9th International Multiconference Information Society
9. mednarodna multikonferenca informacijska družba
IS 2006

Proceedings of the
5th Slovenian and 1st International Conference
Zbornik 5. slovenske in
1. mednarodne konference

Language Technologies
Jezikovne tehnologije
IS-LTC 2006

edited by / uredila:
Tomaž Erjavec
Jerneja Žganec Gros

9th - 10th October 9. - 10. oktober
Ljubljana, Slovenia

Uredniki:

dr. Tomaž Erjavec
dr. Jerneja Žganec Gros

Založnik: Institut »Jožef Stefan«, Ljubljana
Tisk: Birografika BORI d.o.o.
Priprava zbornika: Mitja Lasič
Oblikovanje naslovnice: dr. Damjan Demšar
Tiskano iz predloga avtorjev
Naklada: 60

Ljubljana, oktober 2006

Konferenco IS 2006 sofinancirata
Ministrstvo za visoko šolstvo, znanost in tehnologijo
Institut »Jožef Stefan«

Informacijska družba
ISSN 1581-9973

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

004.934(063)(082)
81'25:004.6(063)(082)

MEDNARODNA multi-konferenca Informacijska družba IS (9 ; 2006 ; Ljubljana)
Jezikovne tehnologije : zbornik 9. mednarodne multikonference Informacijska družba IS 2006, 9. do 10. oktober 2006 = Language technologies : proceedings of the 9th International Multiconference Information Society IS 2006, 9th-10th October 2006, Ljubljana, Slovenia / uredila, edited by Tomaž Erjavec, Jerneja Žganec Gros. - Ljubljana : Institut "Jožef Stefan", 2006. - (Informacijska družba, ISSN 1581-9973)

ISBN-10 961-6303-83-X
ISBN-13 978-961-6303-83-5
1. Gl. stv. nasl. 2. Vzp. stv. nasl. 3. Informacijska družba 4. Information society 5. Erjavec, Tomaž, 1960-
229096960

Zbornik 9. mednarodne multikonference Informacijska družba
Proceedings of the 9th International Multiconference Information Society

IS 2006

Zbornik 5. slovenske in 1. mednarodne konference
Proceeding of the 5th Sloveniab and 1st International Conference

Jezikovne tehnologije

Language Technologies

Uredila / Edited by

Tomaž Erjavec, Jerneja Žganec Gros

http://is.ijs.si

9. do 10. oktober 2006 / 9th - 10th October 2006
Ljubljana, Slovenia

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2006

V svojem devetem letu ostaja multikonferenca Informacijska družba 2006 (http://is.ijs.si) ena vodilnih srednjeevropskih konferenc, ki združuje znanstvenike z različnih raziskovalnih področij povezanih z informacijsko družbo. V letu 2006 smo v multikonferenco povezali osem neodvisnih konferenc. Informacijska družba postaja vedno bolj zapleten socialni, ekonomski in tehnološki sistem, ki je pritegnil pozornost vrste specializiranih konferenc v Sloveniji in Evropi. Naša multikonferenca izstopa po širini in obsegu tem, ki jih obravnava.

Rdeča nit multikonference ostaja sinergija interdisciplinarnih pristopov, ki obravnavajo različne vidike informacijske družbe ter poglabljajo razumevanje informacijskih in komunikacijskih storitev v najširšem pomenu besede. Na multikonferenci predstavljamo, analiziramo in preverjamo nova odkritja in pripravljamo teren za njihovo praktično uporabo, saj je njen osnovni namen promocija raziskovalnih dosežkov in spodbujanje njihovega prenosa v prakso na različnih področjih informacijske družbe tako v Sloveniji kot tujini.

Na multikonferenci, ki bo trajala šest dni, bo na vzporednih konferencah predstavljenih preko 200 referatov, vključevala pa bo tudi okrogle mize in razprave. Referati so objavljeni v zbornikih multikonference, izbrani prispevki pa bodo izšli tudi v dveh posebnih številkah znanstvenih revij, od katerih je ena Informatica, ki se ponaša s 30-letno tradicijo odlične znanstvene revijei. Multikonferenco Informacijska družba 2006 sestavljajo naslednje samostojne konference:

- BIOMA 2006 – Bioinspired Optimization Methods and their Applications
- Mejne kognitivne znanosti
- Kognitivne znanosti
- Sodelovanje in informacijska družba
- Rudarjenje podatkov in podatkovna skladišča
- Vzgoja v informacijski družbi
- Inteligentni sistemi
- Jezikovne tehnologije.

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija. Zahvaljujemo se tudi Ministrstvu za visoko šolstvo, znanost in tehnologijo za njihovo sodelovanje in podporo. V imenu organizatorjev konference pa se želimo posebej zahvaliti udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V letu 2006 sta se programski in organizacijski odbor odločila, da bosta podelila posebno priznanje Slovencu ali Slovenki za izjemen prispevek k razvoju in promociji informacijske družbe v našem okolju. Z večino glasov je letošnje priznanje pripadlo prof. dr. Cenetu Bavcu. Čestitamo!

Viljan Mahnič, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2006

In its 9th year, the Information Society Multiconference (http://is.ijs.si) continues as one of the leading conferences in Central Europe gathering scientific community with a wide range of research interest in information society. In 2006, we organized eight independent conferences forming the multiconference. Information society displays a complex interplay of social, economic, and technological issues that attract attention of many scientific events around Europe. The broad range of topics makes our event unique among similar conferences.

The motto of the Multiconference is synergy of different interdisciplinary approaches dealing with the challenges of information society. The major driving forces of the Multiconference are search and demand for new knowledge related to information, communication, and computer services. We present, analyze, and verify new discoveries in order to prepare the ground for their enrichment and development in practice. The main objective of the Multiconference is presentation and promotion of research results, to encourage their practical application in new ICT products and information services in Slovenia and also broader region.

The Multiconference is running in parallel sessions for six days with over 200 presentations of scientific papers. The papers are published in the conference proceedings, and in two special journal issues. One of them is Informatica with its 30 years of tradition in excellent research publications.

The Information Society 2006 Multi-Conference consists of the following conferences:

- BIOMA 2006 - Bioinspired Optimization Methods and their Applications
- Borderline Cognitive Sciences
- Cognitive Sciences
- Collaboration and Information Society
- Data Mining and Data Warehouses
- Education in Information Society
- Intelligent Systems
- Language Technologies.

The Conference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of ACM. We would like to express our appreciation to the Slovenian Government for cooperation and support, in particular through the Ministry of Higher Education, Science and Technology.

At the end we would like to bring your attention to a special event. In 2006, the Programme and Organizing Committees decided to award one Slovenian for his/her outstanding contribution to development and promotion of information society in our country. With the majority of votes, this honor went to Prof. Dr. Cene Bavec. Congratulations!

On behalf of the conference organizers we would like to thank all participants for their valuable contribution and their interest in this event, and particularly the reviewers for their thorough reviews.

Viljan Mahnič, President of the Programme Committee
Matjaž Gams, President of the Organizing Committee

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, Korea
Howie Firth, UK
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Izrael
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Finland
Bezalel Gavish, USA
Gal A. Kaminka, Israel

Organizing Committee

prof. dr. Matjaž Gams, predsednik/chair
Mitja Luštrek, dipl. ing, podpredsednik/deputy chair
Lili Lasič
Mitja Lasič
Tea Tušar, dipl. ing

Programme Committee

prof. dr. Viljan Mahnič, predsednik
dr. Cene Bavec, pod-predsednik
dr. Tomaž Kalin, pod-predsednik
prof. dr. Jozsef Györkös, pod-predsednik
prof. dr. Tadej Bajd
mag. Jaroslav Berce
prof. dr. Marko Bohanec
prof. dr. Ivan Bratko
dr. Andrej Brodnik
dr. Dušan Caf
prof. dr. Saša Divjak
dr. Tomaž Erjavec
dr. Bogdan Filipič
prof. dr. Matjaž Gams
Marko Grobelnik
prof. dr. Nikola Guid
dr. Marjan Heričko
prof. dr. Borka Jerman Blažič Džonova
prof. dr. Gorazd Kandus
prof. dr. Marjan Krisper
mag. Andrej Kuščer
prof. dr. Jadran Lenarčič
dr. Borut Likar
dr. Dunja Mladenič
dr. Franc Novak
prof. dr. Marjan Pivka
prof. dr. Vladislav Rajkovič
asist. dr. Grega Repovš
prof. dr. Ivan Rozman
Niko Schlamberger, dipl. ing.
prof. dr. Franc Solina
prof. dr. Stanko Strmčnik
dr. Tomaž Šef
dr. Jurij Šilc
prof. dr. Jurij Tasič
prof. dr. Denis Trček
prof. dr. Andrej Ule
prof. dr. Tanja Urbančič
prof. dr. Boštjan Vilfan
prof. dr. David B. Vodušek
prof. dr. Baldomir Zajc
prof. dr. Blaž Zupan

KAZALO / TABLE OF CONTENTS

Zbornik 9. mednarodne multikonference Informacijska družba
Proceedings of the 9th International Multiconference Information Society

IS 2006

Zbornik 5. slovenske in 1. mednarodne konference
Proceeding of the 5th Sloveniab and 1st International Conference

Jezikovne tehnologije

Language Technologies

Uredila / Edited by

Tomaž Erjavec, Jerneja Žganec Gros

http://is.ijs.si

9. do 10. oktober 2006 / 9th - 10th October 2006
Ljubljana, Slovenia

Predgovor

V pričujočem zborniku so objavljeni prispevki s Pete slovenske in prve mednarodne konference Jezikovne tehnologije", ki je potekala 9. in 10. oktobra 2006 v Ljubljani, v okviru meta-konference Informacijska družba, IS'2006. Konferenca je bila namenjena članom Slovenskega društva za jezikovne tehnologije (SDJT) in drugim, ki jih to področje zanima, kot forum, kjer lahko predstavijo svoje delo v preteklih dveh letih, kolikor je minilo od zadnje slovenske konference o jezikovnih tehnologijah.

To srečanje je bilo peto v vrsti slovenskih konferenc o jezikovnih tehnologijah, letos pa prvič organizirano kot mednarodna konferenca z mednarodnim programskim odborom.
Zbornik vsebuje 52 prispevkov, ki obravnavajo široko paleto raziskav in aplikacij. Prispevki so približno enakovredno razdeljeni med tiste, ki obravnavajo govorne tehnologije, in take, ki se ukvarjajo z besedilom. Prispevki so po obravnavani tematiki izrazito raznovrstni, je pa ponovno, glede na prve konference, opazen premik k opisu večjezikovnih virov, projektov in aplikacij.

Organizatorji bi se radi zahvalili vsem, ki so prispevali k uspehu konference: vabljenim predavateljem, avtorjem prispevkov, programskemu odboru za recenzentsko delo ter organizatorjem IS'2006.

Tomaž Erjavec, Jerneja Žganec Gros

Ljubljana, oktober 2006.

Preface

These proceedings contain the contributions to the conference Fifth Slovenian and First International Language Technologies, which took place on October 9th and 10th, 2006, in Ljubljana, in the scope of the Information Society Multiconference, IS'2006. The conference was meant as a forum for members of the Slovenian Language Technology Society and others interested in the field, where they could present their work in the last two years, which have passed since the previous Slovenian conference on Language Technologies.

The event was the fifth in the series of Slovenian Language Technologies Conferences, now for the first time organised as an international conference, with an international programme committee.

These proceedings contain 52 contributions, which present a wide variety of research and application topics. The contributions are about equally divided into those that address speech technologies and those that deal with text. The papers are quite diverse as regards their subject matter but, in comparison to the previous Slovenian LT conferences, we can note a further shift to multilingual applications.

The organisers would like to thank the many people who contributed to the success of the conference: the invited speakers and the authors of contributions, the programme committee of the conference and the organising committee of IS 2006.

Tomaž Erjavec, Jerneja Žganec Gros
Ljubljana, October 2006.

Programski odbor

- doc.dr. Jan Cernocký, Faculty of Information Technology, Brno Technical University (Češka)
- prof.dr. Christoph Draxler, Institute of Phonetics and Speech Communication, Ludwig-Maximilians-Universitaet Munich (Nemčija)
- doc.dr. Tomaž Erjavec (predsednik), Odsek za tehnologije znanja, Institut "Jožef Stefan" (Slovenija)
- prof.dr. Sadaoki Furui, Graduate School of Information Science and Engineering, Tokyo Institute of Technology (Japonska)
- prof.dr. Carmen Garcia-Mateo, ETSI Telecomunicacion, Vigo University (Španija)
- doc.dr. Vojko Gorjanc, Filozofska fakulteta, Univerza v Ljubljani (Slovenija)
- prof.dr. Nancy Ide, Department of Computer Science, Vassar College (ZDA)
- doc.dr. Bojan Imperl, Iskratel d.o.o. (Slovenija)
- prof.dr. Ivo Ipsić, Faculty of Engineering, University of Rijeka (Hrvaška)
- prof.dr. Zdravko Kačič, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru (Slovenija)
- dr. Adam Kilgarriff, Lexical Computing Ltd (Velika Britanija)
- dr. Jin-Dong Kim, Tsujii Laboratory, University of Tokyo (Japonska)
- doc.dr. Cvetana Krstev, Arts Faculty, University of Belgrade (Srbija in Črna gora)
- dr. Siegfried Kunzmann, EMEA voice technology Development, IBM (Nemčija)
- prof.dr. Gianni Lazzari, Interactive Sensory Systems Division, ITC-irst (Italija)
- prof.dr. Bente Maegaard, Centre for Language Technology, University of Copenhagen (Danska)
- prof.dr. Jean-Pierre Martens, Department of Electronics and Information Systems, University of Gent (Belgija)
- doc.dr. Dunja Mladenić, Odsek za tehnologije znanja, Institut "Jožef Stefan" (Slovenija)
- prof.dr. France Mihelič, Fakulteta za elektrotehniko, Univerza v Ljubljani (Slovenija)
- doc.dr. João Paulo Neto, Spoken Language Laboratory, INESC-ID (Portugalska)
- dr. Elmar Nöth, Technical Faculty, Friedrich-Alexander University Erlangen-Nuremberg (Nemčija)
- prof.dr. Karel Pala, Faculty of Informatics, Masaryk University (Češka)
- prof.dr. Marko Stabej, Filozofska fakulteta, Univerza v Ljubljani (Slovenija)
- dr. Tanja Schultz, Univerza Carnegie Mellon (ZDA)
- dr. Tomaž Šef, Odsek za inteligentne sisteme, Institut "Jožef Stefan" (Slovenija)
- prof.dr. Rastislav Šuštaršič, Filozofska fakulteta, Univerza v Ljubljani (Slovenija)
- prof.dr. Marko Tadić, Department of linguistics, University of Zagreb (Hrvaška)
- dr. Jörg Tiedemann, Alfa-Informatica, Rijksuniversiteit Groningen (Nizozemska)
- prof.dr. Dan Tufiș, Research Institute for Artificial Intelligence, Romanian Academy (Romunija)
- prof.dr. Tamás Váradi, Linguistics Institute, Hungarian Academy of Sciences (Madžarska)
- doc.dr. Špela Vintar, Filozofska fakulteta, Univerza v Ljubljani (Slovenija)
- doc.dr. Andreja Žele, Inštitut za slovenski jezik Frana Ramovša, ZRC SAZU (Slovenija)
- dr. Jerneja Žganec Gros (predsednik), Alpineon, d.o.o. (Slovenija)

Programme Committee

- Jan Cernocký, Faculty of Information Technology, Brno Technical University (Czech Republic)
- Christoph Draxler, Institute of Phonetics and Speech Communication, Ludwig-Maximilians-Universitaet Munich (Germany)
- Tomaž Erjavec (chair), Dept. of Knowledge Technologies, Jožef Stefan Institute (Slovenia)
- Sadaoki Furui, Graduate School of Information Science and Engineering, Tokyo Institute of Technology (Japan)
- Carmen Garcia-Mateo, ETSI Telecomunicacion, Vigo University (Spain)
- Vojko Gorjanc, Faculty of Arts, University of Ljubljana (Slovenia)
- Nancy Ide, Department of Computer Science, Vassar College (USA)
- Bojan Imperl, Iskratel Ltd (Slovenia)
- Ivo Ipsić, Faculty of Engineering, University of Rijeka (Croatia)
- Zdravko Kačič, Faculty of Electrical Engineering and Computer Science, University of Maribor (Slovenia)
- Adam Kilgarriff, Lexical Computing Ltd (UK)
- Jin-Dong Kim, Tsujii Laboratory, University of Tokyo (Japan)
- Cvetana Krstev, Arts Faculty, University of Belgrade (Serbia and Montenegro)
- Siegfried Kunzmann, EMEA voice technology Development, IBM (Germany)
- Gianni Lazzari, Interactive Sensory Systems Division, ITC-irst (Italy)
- Bente Maegaard, Centre for Language Technology, University of Copenhagen (Denmark)
- Jean-Pierre Martens, Department of Electronics and Information Systems, University of Gent (Belgium)
- Dunja Mladenić, Dept. of Knowledge Technologies, Jožef Stefan Institute (Slovenia)
- France Mihelič, Faculty of Electrical Engineering, University of Ljubljana (Slovenia)
- João Paulo Neto, Spoken Language Laboratory, INESC-ID (Portugal)
- Elmar Nöth, Technical Faculty, Friedrich-Alexander University Erlangen-Nuremberg (Germany)
- Karel Pala, Faculty of Informatics, Masaryk University (Czech Republic)
- Marko Stabej, Faculty of Arts, University of Ljubljana (Slovenia)
- Tanja Schultz, Carnegie Mellon University (USA)
- Tomaž Šef, Dept. for Intelligent Systems, Jožef Stefan Institute (Slovenia)
- Rastislav Šuštaršič, Faculty of Arts, University of Ljubljana (Slovenia)
- Marko Tadić, Department of linguistics, University of Zagreb (Croatia)
- Jörg Tiedemann, Alfa-Informatica, Rijksuniversiteit Groningen (Netherlands)
- Dan Tufiş, Research Institute for Artificial Intelligence, Romanian Academy (Romania)
- Tamás Váradi, Linguistics Institute, Hungarian Academy of Sciences (Hungary)
- Špela Vintar, Faculty of Arts, University of Ljubljana (Slovenia)
- Andreja Žele, Fran Ramovš Institute of the Slovenian Language, ZRC-SAZU (Slovenia)
- Jerneja Žganec Gros (chair), Alpineon Ltd (Slovenia)

Strengthening the smaller languages in Europe

Steven Krauwer

Utrecht University / ELSNET
Trans 10, 3512 JK Utrecht, The Netherlands
Steven.Krauwer@let.uu.nl

Abstract

The problem we want to address is that - even if the complexity of a language is independent of the number of speakers - industrial developers of language and speech technology (and, unfortunately, hence the EU R&D programmes) focus their efforts on the major languages, because of their economic potential. This is something we cannot change, but in our talk we will discuss what the smaller language communities can do to create optimal conditions for the development of their own language and speech technologies..

Krepitev manjših evropskih jezikov

Čeprav je kompleksnost jezika neodvisna od števila govorcev, industrijski razvijalci jezikovnih in govornih tehnologij zaradi ekonomskega potenciala osredotočajo svoja prizadevanja na velikih jezike, kar na žalost posledično velja tudi za raziskovalno-razvojne programe EU. Tega ne moremo spemeniti, vendar bomo v prispevku preučili, kaj lahko govorci manjših jeziki storijo, da vzpostavijo optimalne pogoje za razvoj lastnih jezikovnih in govornih tehnologij.

1. Introduction

This paper is based on earlier presentations given at the Baltic HLT Conference in Tallinn in April 2005 and at the SALTMIL Workshop at LREC in Genoa in May 2006. The purpose of the paper is not to present new research results, but rather to draw attention to the fate of the smaller languages in Europe and to discuss what we can do to improve the conditions for the smaller languages. In this paper I will use the term smaller languages to refer to languages with limited technological support. There have recently been many discussions on various mailing lists about the most appropriate and politically correct term for this[i], but in the absence of a satisfactory solution I'll stick to smaller for the time being.

2. Roles of Language

Our language is our most important instrument for communication with others. Where in the past the circle of others would normally remain limited to people living in our direct environment (neighbourhood, city, country) the creation and expansion of the EU have made us all member of a much larger community, where more than 60 languages are being used for communication between citizens, 20 of which have the status of official languages. Contrary to the situation in the past we all have to face the fact that most of our fellow EU citizens do not speak or understand our language. This affects a number of aspects of our daily and professional life, and we should ask ourselves to what extent this may cause problems or disadvantages for some of us, and – more importantly – how language and speech technology could help to overcome the problems.

Politically we see that more and more of our local policies are determined by EU legislation coming from Brussels. Although the decision procedures are democratic and every member state gets its chances to participate in the discussions leading to legislative measures and is allowed to use its own language at all formal sessions one may wonder whether everybody's voice is heard equally well during this process and the preparatory stages, where informal discussions may be held in one of the major working languages. At this moment language and speech technology is used by the EU to support professional translators and interpreters, and to provide quick and dirty translations of internal documents between some languages. In spite of these efforts there is no guarantee that all EU legislators are playing on an equal playing field as far as language is concerned.

Economically we can now observe that Europe has become our home market, and the world at large our foreign market. In order to be able to sell products and services both on our home and our foreign market we will always have to cross language barriers. In many countries users of services expect to be addressed in their own language, and very often national legislation requires user manuals to be provided in the national language.

From a cultural point of view we have now become part of the European culture. From an integration point of view it is desirable that our cultural heritage is accessible to our fellow EU citizens, and that they have access to ours. Unfortunately much of this heritage is based on or described in language, which constitutes a major obstacle for mutual cultural exchanges. At the same time we should realize that our language is not only an instrument to convey information about our culture: language is an inalienable part of our cultural identity, and needs to be preserved and protected in the same way we protect buildings, paintings and literature.

Our society and economy become more and more information driven. Unfortunately most information is encoded in language, which means that having electronic access to information is a necessary but not a sufficient condition for having full access to our information society.

Individuals from all member states have become European citizens and can now move freely around in Europe, but one can wonder what it means to be a European citizen if one cannot communicate with most fellow EU citizens. Taking away political frontiers is one step, taking away the language barriers is a natural next step.

3. Where does that leave us?

The ability to cross language barriers is essential for the integration of Europe and for further economic development of the EU as a whole. This is more pressing for small language communities than for the larger ones. One can easily live in an English, French or German speaking region without ever realizing that there exist people who speak a different language. All books are translated, movies are dubbed, and if president Putin, Chirac, Bush or the Pope open their mouth on television a voice-over will take over from him within half a second, unless he happens to speak the local language. If one is living in a smaller language community one is constantly confronted with other languages and with language barriers that have to be crossed.

Traditionally we have three methods to help us to cross language barriers: human translators for written language, human interpreters for spoken language, and (last but not least) learning a foreign language. The first two methods are valid and effective in some situations, but not always applicable in day-to-day communication. The third method can be very helpful in certain situations, but language learning (especially after school age) requires a long-term investment, and there are limits to the number of languages one can learn in a lifetime.

It is not obvious how this situation can ever be improved without the help of technology. In this paper we will discuss how technology can be used to overcome (or at least) reduce the language barriers, and more specifically how we can make sure that both bigger and smaller languages can benefit from the new technologies.

4. The role of language and speech technology

Over the years the EU has invested massively in the development of language and speech technology, and many dedicated R&D programmes have had a significant impact on its advancement, including applications oriented towards solving the multilinguality problem. Even though the 6th Framework Programme, now running towards its end, does not have specific language and speech technology oriented action lines, many of the present projects address language issues.

Unfortunately the strong industrial bias of recent EU programmes has led to a situation where the major part of the funding for language and speech technology goes to the major languages. This is not surprising, as industrial players will prefer to invest in the development and deployment of technologies for larger markets. As a consequence there has been only marginal support for the development of language and speech technology for the language communities that do not constitute profitable markets. As the development cost of such technologies is independent of the number of speakers of a language ("all languages are equally difficult") this has created a very unbalanced situation.

5. What can we do to improve the situation at the EU political level?

At the EU political level it is important that the speakers of smaller languages don't accept that their languages (and the speakers themselves) be marginalized in Europe. It is well-known that the cost (both in time and in money) of multilinguality for the EU is enormous (€ 1123 million in 2005[ii]), and that it will be hard to resist the temptation to reduce the number of official working languages to just a couple. One may be forced to resort to such or similar pragmatic solutions, but representatives of the smaller (or maybe rather commercially not attractive) languages should under all circumstances try to avoid that such pragmatic solutions put them in a disadvantaged position in comparison with those who will be able to use their native languages on all occasions.

It is mandatory to keep the multilinguality problem on the EU agenda as a top priority, and a common responsibility. In this context one should keep in mind that the biggest potential enemy is the so-called subsidiarity principle. There is nothing wrong with the principle as such ("don't treat anything at the EU level that could be treated at the national level"), but in past discussions with EU officials this same principle has been used to explain why the EU could not possibly provide financial support for the technological development of smaller languages, as a language is primarily the responsibility of the national government. This attitude does not only do injustice to the fact that multilinguality is primarily a European problem (as opposed to a collection of national problems), but it also does not seem to be completely consistent with the fact that effectively most of the EU funds for language and speech technology are used to support a few major languages (some of which are supported by strong economies and would actually not need any EU support at all).

One would hope that the coming 7th Framework Programme will recognize the language dimension of Europe, and will address support for language and speech technology development explicitly, irrespective of the economic potential of individual languages or EU world leadership ambitions.

6. What can be done at the national level?

6.1. Human resources

As speakers of smaller languages we have to face the facts: if we don't take care of our languages no one will do it – or Microsoft (provided they judge the potential market interesting enough to make the investment).

In order to properly develop language and speech technology for one's own language (both from a monolingual and from a multilingual point of view) a number of preparations are necessary. First of all language and speech technology have to find their way to higher education curricula. Traditionally language technologists tend to come from a linguistics background, whereas speech technologists have an engineering background. Very few of them have received an education directly aimed at language or speech technology, and there is very little integration between the two. Researchers in more recently emerging areas (multimodality, interfaces, knowledge engineering) have even to a larger extent been obliged to educate themselves, as no standard curricula exist for these fields. Reflection on future curricula seems desirable; in order to be able to offer the next generation of researchers and developers in these (interdisciplinary) fields a better-tailored package of knowledge and skills. In this context we would like to point to initiatives such as the European Masters in Language and Speech Programme[iii], and the European Masters Program in

Language and Communication Technologies[iv], both aiming at defining (and continuously updating) a masters curricula in language and speech technology. The EU Tempus programme offers special mobility grants that can be used to collaborate in the creation of new curricula[v].

When building up local expertise with respect to the national language it is important to keep in mind that even if every language is unique, many problems may manifest themselves in several (often related) languages, and may have been solved there. Even if these solutions might not be directly applicable to one's own language, it is often easier to port the solutions than to try to solve the problem from scratch. In order for researchers to optimally benefit from this it is very important that they get the opportunity to attend international conferences, workshops or courses. The organization of local (or regional) training courses is a very useful instrument to introduce new technologies that have been developed elsewhere.

6.2. Language resources

Language resources (written and spoken corpora, lexicons, parsers, annotation tools, etc) are essential for the development of language technologies and for the training of students. These resources, whatever their nature, have all in common that they are expensive (in time and money) to create. In order to maximally exploit the resources that have been and will be created their re-usability is a very important feature. Funders of the creation of resources should take great care to ensure that once these resources have been created for a specific purpose (e.g. a project) they can be re-used by future projects. This has different aspects:

(i) from an IPR point of view it should be ensured that resources created through public funding can be re-used by others without any legal constraints, at least for research purposes;

(ii) technically these resources should be created in conformity with existing standards or best practice, in order to ensure optimal interoperability with other tools and resources;

(iii) organisationally it should be ensured that a body is identified that is responsible for the maintenance and further distribution of these resources, in order to guarantee that these precious materials do not get lost when research teams are dissolved or new hard- and software platforms emerge.

6.2.1. The BLARK

Given the emergence of statistical methods in all sub-areas of language and speech technology there is virtually no limit to the amount of resources researchers can use. As the creation of such resources can be a significant financial burden ELSNET, in cooperation with a number of partners, including ELDA (Paris), CST (Copenhagen), CNR-ILC (Pisa), is in the process of developing the BLARK concept. BLARK stands for Basic Language Resource Kit, and it aims at defining the minimal collection of resources that is needed to do any research and (precompetitive) development in language and speech technology at all. In its final form it should comprise a list of necessary components (specified both qualitatively and quantitatively), and the standards (formal or de facto) to be adhered to. We will also aim at including cost estimations for the production of the various components, based on experience. The BLARK concept was first launched in the ELRA Newsletter published in May 1998[vi]. The definition allows for adaptations to specific properties of languages.

The BLARK definition should be used as a common reference point for language communities that want to start their own language and speech technology activities, and that need to make up a priority list of what is needed. Once the definition is available teams can make an inventory of what exists and what is missing.

Initial BLARK definitions have been provided for the Dutch language, by researchers associated with the Dutch Language Union. A first inventory and an identification of priorities has led to a large language and speech technology programme funded by the Dutch government and the regional Flemish government in Belgium.

In the framework of the EU funded NEMLAR project[vii] an initial BLARK definition has been prepared for Arabic, and first steps have been made towards the creation of a BLARK for Arabic. The current version of the definition can be found on the same site. [viii]

6.2.2. The BLARKette

One of the findings of the NEMLAR project with respect to the BLARK was that even if a BLARK should be seen as a modest entry point for the creation of resources for a language it has a tendency to grow quickly, as technology advances and discipline boundaries become more and more vague. There is a certain risk that the definition and creation of a full-blown BLARK may be one or more steps too far for smaller, regional languages in Europe, for which no or very little technological support exists, and for which only modest national or regional funding is made available.

In order to accommodate this problem we have proposed the definition of a scaled down, entry-level version of the BLARK, targeting exclusively the research and (especially) the education community. It should be light and compact, not too demanding in terms of hard and software requirements, cheap, free from IPR issues, and ideally small enough to fit on a CD or DVD. We expect to release a first document, with tentative summary specifications, towards the end of 2006. Check the ELSNET site for news[ix].

7. What can be done internationally at the EU level?

Many countries have a long and well-established tradition of national language and speech technology programmes. Within the framework of the creation of the ERA the EU aims at better coordination between national language and speech technology related programmes. Language and speech technology would be an excellent opportunity for such coordination, because it would facilitate both porting of knowledge and expertise between languages addressing cross-lingual issues.

The EU's 7th Framework Programme will also offer opportunities for language and speech technology oriented research and development. There are indications that language and speech technology, which were completely

out of focus in the 6th Framework Programme, will be given a more prominent role, and it is hoped (but not guaranteed) that the smaller (i.e. commercially less significant) languages will receive more attention.

Another interesting development is the decision by the EU to add Irish to the set of official EU languages, and to give a similar status (although on a self-paying basis, and on the basis of special agreements) to Galician, Catalan-Valencian and Basque and other officially recognized languages in member states.

8. What sort of language and speech technology solutions are we looking for?

It is easy to say that we should resort to language and speech technology in order to get our multilinguality problems out of the way, but how realistic is this? In spite of all the efforts made by the R&D community machine translation (MT) is still not mature enough to be accepted as a generally applicable solution. For the time being the creation of high quality MT systems is still a wonderful research topic, but nothing more than that.

Yet it has to be kept in mind that even state-of-the-art MT can be useful. The obvious example is just finding out what a mail message or a web page in a foreign language is about. I am receiving hundreds of spam messages per day, but sometimes I am really curious what it is that people are trying to sell me from Russia, Korea or China, and a free on-line MT system is good enough to get an idea.

If you buy an MT system like Systran you can get it almost for free, and the quality is moderate (to put it mildly), but if you are prepared to spend a bit (or rather: a lot) more it can be customized to your specific needs, and the quality level improves dramatically. Like in the case the cheap inkjet printers and the expensive cartridges Systran's real business is not the MT system but its customization.

If your company has a professional translation department the introduction of an MT system can easily save you 30% on your translation costs. The raw translation is not good enough for publication, but the total process of making the raw translation and having it edited by a professional translator can become a lot cheaper and faster.

Unfortunately MT companies will normally not be interested in the development of systems for language pairs for which they don't see a large potential market which will guarantee them a significant return on their investments.

For normal citizens MT is not really a useful option to cross language barriers. In order to find good alternatives we have to abandon the idea that one single solution should solve the problem in all situations. Different situations may require different types of solutions, just like in traffic where you can solve the problem that you happen to be in the wrong place by walking, using your bike or car, taking the train or the plane, or just using the phone.

Let me just give a few examples. Many mobile phones or PDAs come with a small camera these days. Why can't I use this to point at the menu in a restaurant in Ljubljana, have it OCR-ed, translated and displayed on the screen in my own language? As a matter of fact I used this example some ten years ago to illustrate my dreams of what future technology might bring us, and only very recently I read that such a facility now exists for Japanese to English!

Why isn't my PowerPoint presentation displayed on two screens in parallel, one in English and one in Slovenian (by way of – possibly imperfect – subtitles)? Why doesn't the manager of my hotel use a multilingual authoring system to present his announcements in my own language? Why can't I use my mobile phone or PDA to have the spoken word spinach translated in Slovenian and displayed on the screen so that I can show the shopkeeper that it is spinach I want?

The morale of this should be clear: even if we don't know how to do full MT yet there are lots of ways to deal with the language problem in different contexts, especially since many contexts offer opportunities to support language communication with additional modalities (combination of spoken and written language, gesturing, facial expressions, video displays, etc).

Apart from that there seems to be a wealth of opportunities in the development of computer assisted learning of languages, not just in class-room settings, but also for adults who want to learn new languages from home, or when sitting in trains, planes or traffic jams.

9. Concluding remarks

I have tried to describe above why multilinguality is a pressing problem, especially for the smaller language communities in Europe. I have also indicated what one could do to in order to keep the problem on the EU's political agenda, what one can do to strengthen one's own local language and speech technology, and what sort of solutions present day language and speech technology can offer. Personally I do not see an immediate danger that our small languages will disappear in the first hundred years or so, but in my view the real danger is that speakers of smaller languages may find themselves more and more marginalized, both economically and politically, if they don't make a serious effort to overcome the language problem. From my own professional point of view the use of language and speech technology is the most promising direction, but at the same time I would like to make it clear that I also sympathize with the EU's efforts in their language action plan to encourage people to learn at least two other EU languages in addition to their native language, and where language and speech technology can become very important instruments to achieve this.

---

[i] See e.g. MT-list on http://www.mail-archive.com /mt-list@eamt.org/

[ii] See http://europa.eu/languages/en/document/59#8

[iii] See http://www.cstr.ed.ac.uk/euromasters

[iv] See http://lct-master.org/

[v] See http://ec.europa.eu/education/programmes /tempus/index_en.html

[vi] Also published on http://www.elsnet.org/blark.html

[vii] http://www.nemlar.org

[viii] http://www.nemlar.org/Publications/BLARK-final.pdf

[ix] See http://www.elsnet.org

Speech Synthesis and Discourse Information

Nick Campbell

[1] National Institute of Information and Communications Technology
[2] ATR Spoken Language Communication Research Laboratory,
nick@nict.go.jp & nick@atr.jp

Abstract

This paper describes some recent work towards a conversational speech synthesis system for use in interactive dialogues between a human and an information system, robot, or speech translation device. The paper describes several response-type utterances that are currently very difficult to implement using traditional speech synthesis methods, and shows how these non-verbal speech sounds function to provide feedback and status-updates in an interactive discourse. The talk will be illustrated with examples of such utterances, which include laughter and grunts as well as common phrases and idiom, showing how their variety can reveal several types of information about the speaker-(i.e., listener) states. The paper proposes a model of information exchange (through speech) whereby this feedback from the listener allows the speaker to efficiently deliver content and to be assured of successful information transmission.

Sinteza govora in diskurzna informacija

Članek opisuje zadnje dosežke pri razvoju pogovornega sintetizatorja govora, ki je namenjen uporabi v interaktivnih dialogih med človekom in informacijskim sistemom, robotom ali govorno-prevajalno napravo. Čanek opisuje več vrst neverbalnih odgovorov, ki jih je z uporabo tradicionalnih postopkov za sintezo govora težko implementirati, in pokaže vlogo teh neverbalnih govornih segmentov pri zagotavaljanju povratne informacije in statusnih osvežitev v interaktivnem diskurzu. Predstavitev bo opremljena s primeri takih neverbalnih govornih segmentov, ki vključujejo smeh in mrmranje, kot tudi pogoste fraze in idiome. Pokazano bo, kako lahko njihova raznolikost razkrije več vrst podatkov o stanju govorca oziroma poslušalca. Članek predlaga model za informacijsko izmenjavo (s pomočjo govora), v katerem poslučalčeva povratna informacija govorcu omogoča, da učinkovito posreduje vsebino in govorcu zagotavlja, da bo informacija uspešno prenesena.

1. Introduction

Speech synthesis has made considerable progress over the past ten years, and some of the recent applications using unit-selection and concatenation of raw waveforms can now only occassionallly be distinguished from natural human speech in terms of voice quality and expressiveness (see for example [1,2]). Their use in many news-reading, announcement, or customer-care applications has become almost transparent, but problems still remain when speech synthesis is to be used in a speech translation environment or when the technology has to replace human speakers in a one-to-one dialogue situation. The expressiveness of a one-to-one conversation is much richer than that of a one-to-many broadcast situation, and many of the differences are signalled using tone-of-voice on utterances that carry little or no propositional content.

2. Data Collection

As part of the JST/CREST 'Expressive Speech Processing' project (ESP), we recorded a series of telephone conversations between ten people who were not initially familiar with each other and who had little or no face-to-face contact during the recording period. They spoke together once a week over the telephone for thirty-minutes each time during a period of three months. The content of the conversations was completely unconstrained. We refer to this as the ESP_C subset of the ESP corpus.

The volunteer speakers were paired so that each conversed with a different combination of partners to maximise the different types of expressiveness in the dialogues without placing the speakers under any requirement to self-monitor their speech or to produce different speaking styles "on-demand".

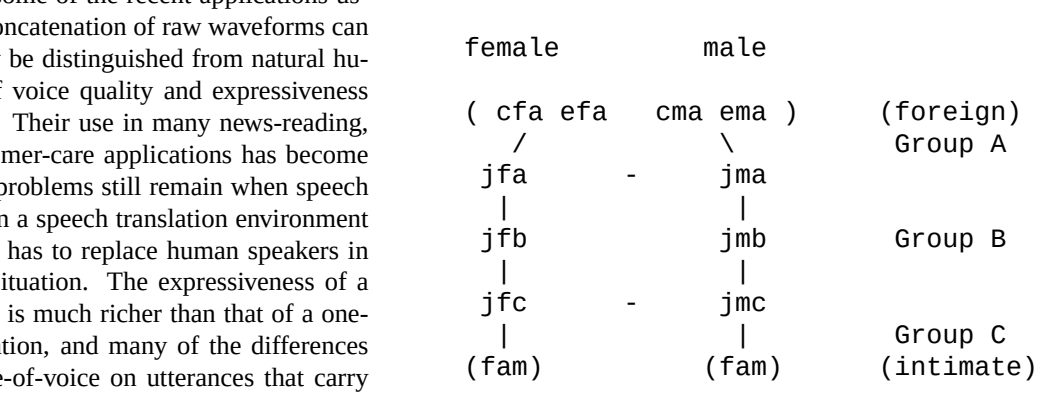


Figure 1: Showing the form of interactions between the participants. The first letter of the participant identifier indicates the mother-tongue (Japanese/Chinese/English) of the speaker, the second letter indicates the speaker's sex (female or male), and the third letter is the group identifier.

The ten speakers were all recorded in Osaka, Japan, and all conversations were in Japanese. Since the speakers were not familiar with each other initially, the use of the local dialect was not expected and conversations were largely carried out in so-called 'standard' Japanese. Again, no constraints on types of language use were imposed, since the goal of this data collection was to observe the types of speech and the variety of speaking styles that 'normal' peo-

ple used in different everyday situations.

Four of the ten speakers were non-native; their inclusion was not so that we should have foreign-accented speech data, but rather that we should be able to observe changes in the speech habits of the Japanese native speakers when confronted with linguistically-impaired partners. Two were male, two female, two Chinese, and two English-language mother-tongue speakers. These and the two Japanese who spoke with them formed Group A in our study. Group B is the 'baseline' group, consisting of a male and a female Japanese native speaker who conversed in turn with the each other and with the Japanese native speakers of both sexes from Groups A and C. Group C similarly consisted of a male and a female Japanese native speaker who conversed with each other and with the members of Group B, but who also telephoned their own family members each week and spoke with them for a similar amount of time.

both female	mixed	both male
6425 EFA JFA 7359 JFA EFA		9348 EMA JMA 7433 JMA EMA
8827 CFA JFA 9145 JFA CFA		x (cma jma) 7530 JMA CMA
	9236 EFA JMA 8499 JMA EFA	
	7557 JMA CFA x (cfa-jma)	
	8237 JFA CMA x (cma-jfa)	
	8416 JFA EMA 8560 EMA JFA	
	10068 JFA JMA 7701 JMA JFA	
9069 JFA JFB 9378 JFB JFA		8614 JMA JMB 9465 JMB JMA
8044 JFB JFC 8234 JFC JFB		6983 JMB JMC 7735 JMC JMB
	7686 JFB JMC 7222 JMC JFB	
	10005 JFC JMB 7980 JMB JFC	
13900 JFC Fam		9961 JMC Fam

Table 1: Showing utterance counts and speakers for each recorded conversation. For example, 6425 is the number of utterances spoken by EFA (English female, Group A) to JFA (Japanese female, Group A). There were no sex constraints for speech with family members, but the voice of the remote partner in these cases was not recorded or transcribed. Lower-case shows conversations yet to be transcribed (utterance count shown by an 'x').

The corpus thus allows us to examine the prosodic characteristics and speaking habits of Japanese native speakers when confronted with a range of different partners on the spectrum of familiarity, and to observe changes in their speech as this familiarity changes over time.

Our principal targets for this series of recordings were the six Japanese native speakers (three male and three female) who came to an office building in Osaka once a week to answer the telephone and speak with each partner for a fixed period of thirty-minutes each time. All wore head-mounted close-talking Sennheiser microphones and recordings were taken directly to DAT with a sampling rate of 48kHz. The offices were air-conditioned, but the rooms were large and quiet, and no unwanted noises (or acoustic reflections) were present in the recordings.

```
CFA JFA C01 200.369 0.491 #
CFA JFA C01 200.860 0.808 laugh
CFA JFA C01 201.668 0.869 あと.は
CFA JFA C01 202.537 1.099 変わり.まし.た
CFA JFA C01 203.636 1.868 laugh
CFA JFA C01 205.504 0.670 うん
CFA JFA C01 206.174 0.744 #
CFA JFA C01 206.918 0.917 はい
CFA JFA C01 207.835 2.691 #
CFA JFA C01 210.526 0.602 はい
CFA JFA C01 211.128 2.791 #
CFA JFA C01 213.919 0.749 @S
CFA JFA C01 214.668 2.685 そう.です.結構.もう.四.年間.なり.まし.た
CFA JFA C01 217.353 0.785 はい
CFA JFA C01 218.138 0.561 #
CFA JFA C01 218.699 0.731 はい
CFA JFA C01 219.430 1.384 #
CFA JFA C01 220.814 1.088 行っ.て.ます
CFA JFA C01 221.902 0.738 #
CFA JFA C01 222.640 0.784 はい
CFA JFA C01 223.424 1.107 #
CFA JFA C01 224.531 1.356 あの.一.歳.です
CFA JFA C01 225.887 0.525 #
CFA JFA C01 226.412 0.600 はい
CFA JFA C01 227.012 2.795 #
CFA JFA C01 229.807 0.443 はい
CFA JFA C01 230.250 0.941 #
```

Figure 2: Transcription was performed by hand, using the Transcriber software package. The first 3 columns identify the speaker, partner, and conversation number. The numbers represent the start time of each utterance in the conversation (in seconds) and its duration. Laughs, non-speech noises, and silences are also transcribed along with the text. Dots in the text represent morphological boundaries as automatically determined by the 'Mecab' software.

The speakers were all mature adults who held part-time jobs with the same company and were paid for their participation in the recordings. They were initially unfamiliar with each other, but the degree of familiarity naturally increased throughout the period of the ten conversations. All have signed consent forms allowing the contents of the recordings to be used for scientific research. The ultimate purpose of the data collection was not made specific to the participants who were only told that their speech would be recorded for use in telecommunications research.

3. Data Analysis

Figure 3 shows the corresponding part of the dialogue segment presented in Figure 2. Here we see the Japanese

```
JFA CFA C01 203.276 1.362 4456 ==> <[ laugh ]>
JFA CFA C01 204.638 0.902 0 ==> <[ @S ]>
JFA CFA C01 205.540 1.927 0 +->  <<あー.,>> .そう.な. << <<ん.です>> .か>>
JFA CFA C01 207.467 0.322 0 ==> <[ @S ]>
JFA CFA C01 207.789 0.401 0 ==> <[ はい ]>
JFA CFA C01 208.190 0.227 0 ==> <[ @S ]>
JFA CFA C01 208.417 1.744 0 ==> <[ あのー ]>
JFA CFA C01 210.976 0.393 814 ==> <[ え ]>
JFA CFA C01 211.369 0.260 0 ==> <[ え ]>
JFA CFA C01 211.629 1.139 0 --> お.,.ご.結婚.を.き.ょ
JFA CFA C01 212.768 0.264 0 ==> <[ え ]>
JFA CFA C01 213.032 1.566 0 --> 何.時.なさっ.た.と.おっしゃい.まし
JFA CFA C01 216.356 0.687 1757 --> 四.年.目
JFA CFA C01 217.043 0.301 0 ==> <[ @S ]>
JFA CFA C01 217.344 1.498 0 +-> あ.,. <<そう.です>> .か
JFA CFA C01 218.842 0.422 0 ==> <[ @S ]>
JFA CFA C01 219.264 0.241 0 ==> <[ え ]>
JFA CFA C01 219.505 1.193 0 --> お子さん.は
JFA CFA C01 221.686 0.283 987 --> X
JFA CFA C01 221.969 0.819 0 --> あ.,.いらっしゃる
JFA CFA C01 223.180 0.360 392 ==> <[ あ ]>
JFA CFA C01 223.540 1.248 0 --> お.幾つ.です.か
JFA CFA C01 225.571 0.749 783 --> 一.才
JFA CFA C01 226.320 0.347 0 ==> <[ @S ]>
JFA CFA C01 226.667 1.235 0 --> あっ.そう.,.じゃ
JFA CFA C01 227.902 1.891 0 +-> こういう. <<とき.は>> .どういう.風.に
JFA CFA C01 229.793 1.494 0 +-> お子さん.は.さ.れ.てる. << <<ん.です>> .か>>
JFA CFA C01 231.287 0.746 0 ==> <[ @S ]>
JFA CFA C01 232.033 0.798 0 --> お家
JFA CFA C01 234.539 0.424 1707 ==> <[ あ ]>
```

Figure 3: The corresponding part of the dialogue shown in Figure 2, but after processing to identify repeated patterns. Frequent utterances ($n >= 100$) are shown in $< [square] >$ brackets, and frequent segments ($N >= 100$) within longer utterances are shown in $<< angle >>$ brackets, which may be embedded. Also shown here in column 6 are the delays (in milliseconds) between succeedding utterances.

speaker's utterances and can combine them with those of her Chinese partner to reproduce the conversation segment. Some potentially ambiguous utterances can thereby be disambiguated by use of the textual content of the surrounding utterances, but a large number remain functionally indeterminate from the transcription alone. They are not at all ambiguous when listening to the speech, and carry a considerable amount of discourse information.

JFA:	CFA	CMA	EFA	EMA	JFB	JMA
a,a–	143	145	88	89	138	170
ano	224	277	221	176	209	266
demo	41	24	31	17	89	134
e–	48	51	37	25	74	94
hai	2932	2234	2181	3239	72	33
un,un	1029	546	585	1190	909	1037

Table 2: Counts for some frequently-repeated simple utterances from one speaker to six partners. The table illustrates differences in usage strategies for these utterances.

The text in Figure 3 has been further annotated by a computer program to show which utterances are unique (and therefore presumably convey more propositional content) and to mark those which are subject to frequent repetition (and hence portray affect or discourse-control information). Two types of repetition have been marked; (a) whole phrases, and (b) phrasal chunks that form part of a larger, possibly unique, utterance but which are frequently repeated anyway. The chunks were determined by use of the pds 'mecab' software [3] for morphological decomposition, in conjunction with 'yamcha' [4] for regrouping of the fine morphological segments.

The current setting of the pattern recognition program, arbitrarily taking more than 99 repeats throughout the corpus as the minimum threshold for bracketing, yields 74,324 untouched utterances, 72,942 marked as repeated phrases, and 49,136 utterances including repeated phrasal segments.

Taking some of the frequent repetitions from one of the corpus speakers as an example, we notice different strategies of usage according to differences in partner. This speaker (JFA) makes considerable use of "a", "ano", "hai", and "un", but not equally with all partners (see Table 3). For example, when speaking with foreigners, she uses "hai" (はい= yes(perhaps?!)) frequently, but significantly less so when speaking with Japanese partners. She uses "demo" (でも=but) much more frequently with Japanese partners, and "a" much less when conversing (in Japanese) with the

10073	うん	467	ズーー	228	ううん	134	へー.ーー
9692	@S	455	スー	227	えっ	134	はい.はい.はい.はい
8607	はい	450	んーー	226	へ.ーー	134	そう.です
4216	laugh	446	うーーーん	226	ハハハ	133	@E
3487	うーん	396	ねー	225	う.んー	133	あ.そう.な.ん.です.か
2906	ええ	395	あ.あー	200	そうですね	130	そう.な.ん.です.か
1702	はーい	393	はい.はい.はい	199	ほ.ーー	129	は.ー
1573	うーーん	387	あー.はい	193	ハー	129	い
1348	ズー	372	ねえ	192	その	127	ほ.ー
1139	ふん	369	ふーーん	190	え.えー	125	ハハハハハ
1098	あのー	369	だから	188	あ.あーー	119	はい.はい
1084	あっ	368	あー.ん	187	ね	119	は.ーーー
981	はあい	366	ああ	180	ん.はい	114	ハハ
942	あの	345	あの.ーー	180	あの.ーーー	113	は
941	ふーん	337	なんか	173	ん.ん	113	で.ー
910	そう	335	え	172	アハハハ	113	て
749	えー	311	でも	168	はい.ー	112	は.あー
714	あーー	305	スーー	164	う.うーん	110	フフフ
701	あ	274	うん.うん.うん	161	は.ーー	110	そのー
630	あーーー	266	ハハハハ	160	@K	110	もう
613	あ.はい	266	て.ー	159	そう.です.ねー	109	ふーーーん
592	うん.うん	266	え.ーー	151	あーーーー	108	はあ.ーー
555	あー	258	で	143	だから.ー	106	そうですね.え
500	んー	248	う	139	アハハハハ	105	んー.ん
469	ん	242	へー	137	そう.そう.そう	104	いや

Table 3: The hundred most frequent single utterances in the ESP_C corpus. The numbers indicate the count of each word or phrase when it occurs as a single utterance in the transcriptions. Since duration is usually considered as distinctive in Japanese, the lengthening (an extra mora beat is indicated by a dash) may be significant. Note the highly repetitive nature of many of these utterances, very few of which can be found in any standard dictionary of Japanese. Note that these few samples alone account for more than a third $(n = 72,685)$ of the 200,000 utterances in the corpus. Less then half $(n = 92,541)$ of the utterances were unique.

English-native-speaker partners.

Such differences may reflect interpersonal relationships, personal characteristics, or cultural peculiarities, but perhaps more interesting to us here (with speech synthesis in mind) is the variety of pronunciation within each utterance type, reflecting the speaker's interest, state-of-mind, and type of participation in the discourse.

4. Ambiguous Utterances — — A Challenge for Synthesis

It is a central tenet of this paper that these repeated segments can be used to carry affect-related and interpersonal information by variation in such acoustic characteristics as tone-of-voice, spectral tilt, pitch range and excursion, speaking rate, phonatory setting, etc. By being frequent and repetitive, they allow the listener (even one not yet familiar with the speaker's traits) to make comparative judgements about the speaker's emotional and affective states and stances and to interpret subtle nuances in the speech by means of the prosodic cues hereby revealed [5].

In speech synthesis, a given text sequence is rendered into speech with a given prosodic pattern, usually predicted from part-of-speech information in conjunction with the position of the words in the phrase and sentence. Here, however, we have whole phrases that consist of a single word (itself often of doubtful or indeterminate part-of-speech status) whose prosody is dependent upon the speaker's affective states and discourse intentions; there is currently no way of easily specifying these higher-level constraints in a synthesiser apparatus.

"a,a–"	CFA	CMA	EFA	EMA	JFB	JMA
f0r	125	181	266	232	234	241
f0m	201	214	220	192	206	198
pwr	28	29	29	28	31	31
pwm	38	39	36	35	42	41
"un,un"	CFA	CMA	EFA	EMA	JFB	JMA
f0r	154	152	182	181	161	141
f0m	172	175	162	145	198	174
pwr	28	29	27	26	29	27
pwm	37	40	36	35	42	39
"ano"	CFA	CMA	EFA	EMA	JFB	JMA
f0r	106	113	161	154	169	155
f0m	131	136	142	133	156	149
pwr	27	28	28	27	31	29
pwm	38	40	37	36	42	39

Table 4: F0 range (f0r) and average (f0m) values in Hz and Power range (pwr) and average (pwm) values in dB for three sample utterances from speaker JFA according to differences in conversational partner (see Figure 4).

Tables 3 and 4, and Figure 4 illustrate some diffferences in pitch range (i.e, the amount of variation in the funda-

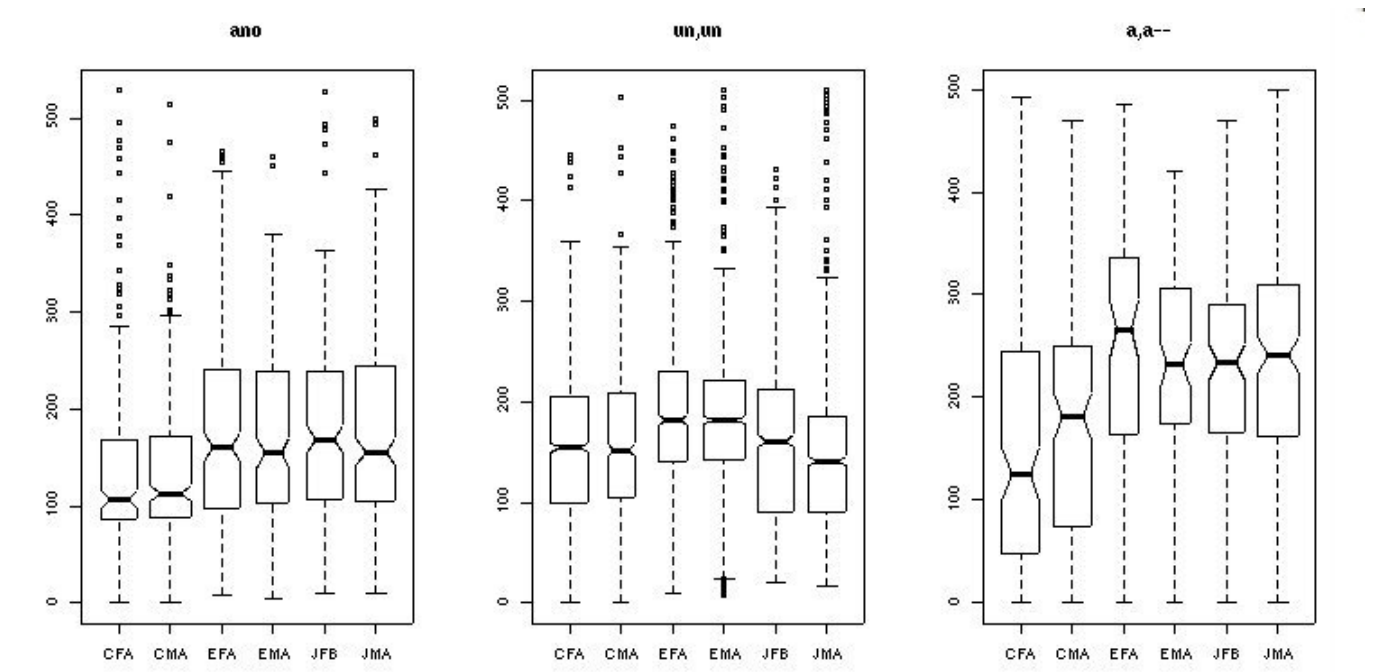


Figure 4: Plots of Pitch Range (amount of variation in the fundamental frequency of the voice) for three utterances from speaker JFA when conversing with six different partners. The width of the boxes is proportional to the number of tokens. Differences are significant at the 5% level if the notches do not overlap. The vertical axis shows pitch range in Hz.

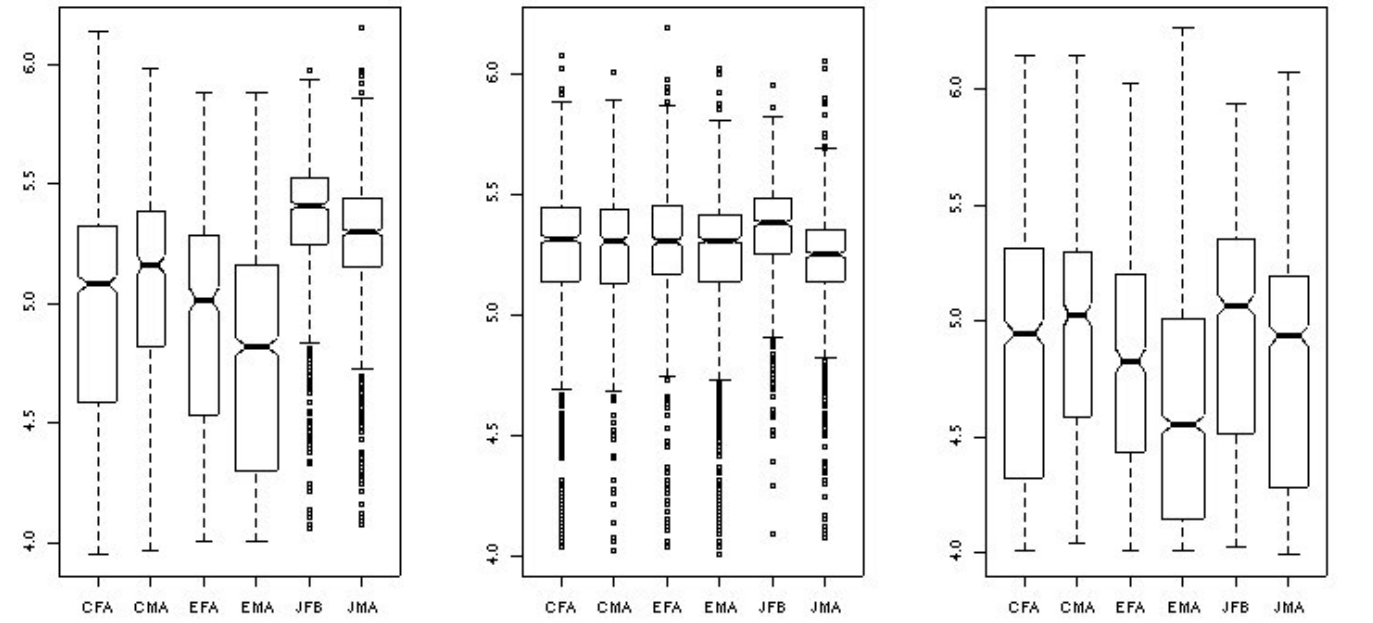


Figure 5: Fundamental frequency contours differ according to the listener. The left-hand plot shows average f0 values for the initial third of the utterance, the middle plot for the middle third, and the right-hand plot shows average f0 values for the final third of the utterance. Plots show‘contours’ for “un,un”. We can see that Japanese partners evoke a high initial contour, and English-native-speakers a lower fall at the end, though all contours appear to pass through the same high range of values mid-utterance.

mental frequency of the voice throughout the utterance) and voice energy (signal power in decibels) for three representative but randomly-selected sample utterances from speaker JFA's conversations with six different partners.

The data show that the speaker's basic acoustic settings and amount of physical energy used in each utterance vary not just by utterance, as would be expected, but also by listener (and presumably according to the content of the conversations). Figure 5 takes a subset of this data (f0 contours for the utterance "un,un") and plots a representation of the 'shape' of each utterance by showing averaged f0 values for each progressive third of the utterance. Again we see considerabe variation, but that the variation between contours for different types of conversation partner is greater than that between utterances within a given set of conversations.

We can see that Japanese partners evoke a high initial contour, and English-native-speakers a lower fall at the end, though all countours appear to pass through the same high range of values mid-utterance. The fact that these differences appear more related to partner than to local contextual differences implies that a higher-level of prosodic processing is taking place; i.e., that a level of social interaction is influencing the prosodic contour just as the linguistic relations influence it a lower more independent level.

5. Conclusion

This paper has presented some data from the ESP_C corpus of conversational dialogues, and has shown that there is considerable prosodic variation on what are seemingly very simple but also very frequent utterances. This variation may indicate the speaker's relationship with the listener, since it seems to vary more between conversational partners than between different utterances.

From a speech synthesis standpoint, this data presents problems for current systems which use one standard set of rules for predicting all prosodic characteristics. These rules currently make no allowance for difference in the relationship with the listener (or conversational partner) but for interactive speech synthesis systems where a computer is generating speech on behalf of one partner, such as in a speech translation system, such information must be mapped, processed, and included in the prosody control rules.

Acknowledgement

This work was supported by the Japan Science & Technology Agency (JST/CREST) and the National Institute of Information & Communications Technology. The author also wishes to thank the management of ATR SLC for support.

6. References

Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., and Syrdal, A., "The AT&T Next-Gen TTS System", in Proc TC-STAR Workshop on Speech-to-Speech Translation, Barcelona, Spain, 2006.

Campbell, N., "Conversational Speech Synthesis and the Need for Some Laughter", in *IEEE Transactions on Audio, Speech, and Language Processing, Vol 14, No.4*, July 2006.

Mecab: http://mecab.sourceforge.jp/

Yamcha: http://www.chasen.org/ taku/software/yamcha/

Campbell, N., "Getting to the heart of the matter; speech as expression of affect rather than just text or language", pp 109-118, *Language Resources & Evaluation Vol 39, No 1*, Springer, 2005.

Automatic Evaluation of Tracheoesophageal Telephone Speech

Korbinian Riedhammer[†], Tino Haderlein[*], Maria Schuster[*], Frank Rosanowski[*], Elmar Nöth[†]

[*] Abteilung für Phoniatrie und Pädaudiologie, Universität Erlangen–Nürnberg
Bohlenplatz 21, 91054 Erlangen, Germany

[†] Lehrstuhl für Mustererkennung, Universität Erlangen–Nürnberg
Martensstraße 3, 91058 Erlangen, Germany
noeth@informatik.uni-erlangen.de

Abstract

The tracheoesophageal (TE) substitute voice is currently state–of–the–art treatment to restore the ability to speak after laryngectomy. The intelligibility while talking over a telephone is an important clinical factor, as it is a crucial part of the patients' social life. An objective way to rate the intelligibility of substitute voices when talking over a telephone is desirable to improve the post–laryngectomy speech therapy. An automatic speech recognition (ASR) system was applied to 41 high quality recordings of post–laryngectomy patients. The ASR system was trained with normal, non–pathologic speech. It yielded a word accuracy (WA) of 36.9%±18.0%; compared to the intelligibility rating of a group of human experts the ASR system had a correlation coefficient of -.88. After downsampling the 41 recordings to telephone quality, the ASR system reached a WA of 26.4%±13.9% leading to a correlation coefficient of -.80. These results confirm that an ASR system can be used for objective intelligibility rating over the telephone.

Samodejna evalvacija traheoezofagalnega telefonskega govora

Traheoezofagalni nadomestni glas je trenutno najsodobnejši način obnove sposobnosti govora po laringektomiji. Razumljivost pri telefonskem pogovoru je pomemben kliničen dejavnik, saj predstavlja ključen del pacientove socialne interakcije. Za izboljšanje govorne terapije po laringektomiji je zaželen objektiven način ocenjevanja razumljivosti nadomestnih glasov pri telefonskem pogovoru. S sistemom za samodejno razpoznavanje govora (SRG) je bilo pregledanih 41 visoko kakovostnih posnetkov pacientov po laringektomiji. Sistem SRG so učili z normalnim, nepatološkim govorom. Odstotek pravilno razpoznanih besed je bil 36,9%±18,0%; v primerjavi z ocenami razumljivosti, ki jih je podala skupina strokovnjakov, je imel sistem SRG korelacijski koeficient -,88. Po znižanju frekvence vzorčenja 41 posnetkov na telefonsko kakovost je sistem SRG dosegel naslednji odstotek pravilno razpoznanih besed: 26,4%±13,9% oziroma korelacijski koeficient -,80. Ti rezultati potrjujejo, da je sistem SRG primeren za objektivno ocenjevanje razumljivosti telefonskega govora.

1. Introduction

The tracheoesophageal (TE) substitute voice is currently state–of–the–art treatment to restore the ability to speak after laryngectomy (Brown et al., 2003): A silicone one–way valve is placed into a shunt between the trachea and the esophagus, which on the one hand prevents aspiration and on the other hand deviates the air stream during expiration into the upper esophagus. The upper esophagus, the pharyngo–esophageal (PE) segment, serves as a sound generator (see Figure 1). Tissue vibrations of the PE segment modulate the streaming air and generate the primary substitute voice signal which is then further modulated in the same way as normal speech. In comparison to normal voices the quality of substitute voices is low, e.g. the change of pitch and volume is limited and inter–cycle frequency perturbations result in a hoarse voice (Schutte and Nieboer, 2002). Another source of distortion is the so–called tracheostoma which is at the upper end of the trachea (see Figure 1). In order to force the air to take its way through the shunt into the esophagus and allow voicing, the patient usually closes the tracheostoma with a finger. If the patient is not able to do this properly, loud "whistling" noises from the eluding air occur. Acoustic studies of TE voices can be found for instance in (Robbins et al., 1984; Bellandese et al., 2001).

In order to improve post–laryngectomy speech therapy, an objective means to rate intelligibility is desired. In previous work we showed that an automatic speech recognition (ASR) system can be used to rate the intelligibility (Schuster et al., 2006; Schuster et al., 2005) of post–laryngectomy speakers. As the telephone is a crucial part of the patients' social life, an objective rating of the intelligibility when talking over a telephone would enhance post–laryngectomy speech therapy.

In our work we examine how well TE telephone speech is processed by an ASR system and how we can optimize the recognition system to achieve better results in order to provide a proper objective intelligibility measure for telephone data.

2. The Recognition System

The ASR system used for the experiments was developed at the Chair of Pattern Recognition (Lehrstuhl für Mustererkennung) of the University of Erlangen–Nuremberg. It can handle spontaneous speech with mid–sized vocabularies up to 10,000 words. A commercial version of this recognizer is used in high–end telephone–based conversational dialogue systems by *Sympalog* (www.sympalog.com), a spin–off company of the Chair of Pattern Recognition. The latest version is described in detail in (Gallwitz, 2002; Stemmer, 2005).

The short–time analysis applies a Hamming window with a length of 16 ms, the frame rate is 10 ms. For each frame, a 24–dimensional feature vector is computed which

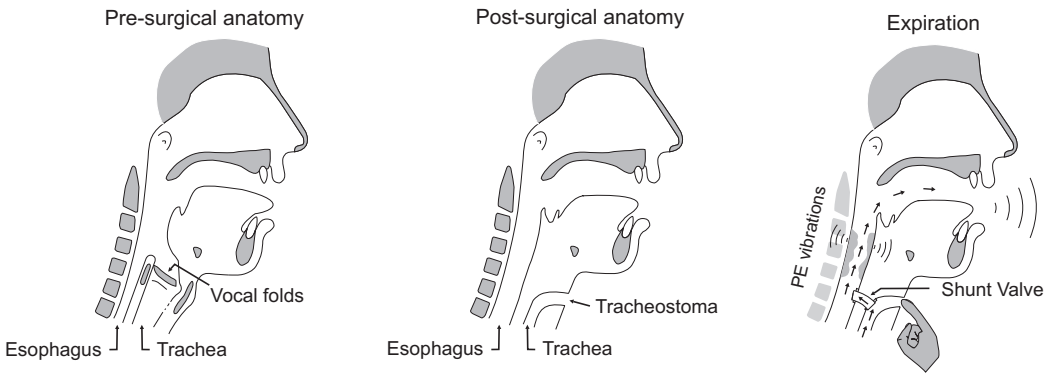


Figure 1: Physiological changes and speaking after laryngectomy: Anatomy of a person with intact larynx (*left*), anatomy after total laryngectomy (*middle*), and the substitute voice (*right*) caused by vibration of the pharyngoesophageal segment (pictures from (Lohscheller, 2003)).

contains the short–time energy, 11 Mel–frequency cepstral coefficients (MFCC) and their first–order derivatives. The derivatives are approximated by the slope of a linear regression line over 5 consecutive frames (56 ms). The filter bank for the Mel–spectrum consists of 25 triangle filters. The actual recognition is done using semi–continuous Hidden Markov Models (SCHMMs). The codebook contains 500 Gaussian densities which are shared by all HMM states. Also, a unigram language model is used, so that the results are mainly dependent on the acoustic models. The elementary recognition units are polyphones, an extension of the well–known triphone approach (Schukat-Talamazzini, 1995). The HMMs for the polyphones have three to four states.

3. Recognizer Training

The basic training set for our recognizers are dialogues from the VERBMOBIL project (Wahlster, 2000). The topic of the recordings is appointment scheduling. The data were recorded with a close–talk microphone at a sampling frequency of 16 kHz and quantized with 16 bit (linear). The speakers were from all over Germany and thus covered most dialectical regions. However, they were asked to speak standard German. About 80% of the 578 training speakers (304 male, 274 female) were between 20 and 29 years old, less than 10% were over 40. This is important in view of the test data, because the fact that the average age of our test speakers is more than 60 years may influence the recognition results. A subset of the German VERBMOBIL data (11,714 utterances, 257,810 words, 25 hours of speech) was used for the training set and 48 utterances (1042 words) for the validation set[1].

In order to get a telephone speech recognizer, we downsampled the training set to telephone quality. We reduced the sampling rate to 8 kHz and applied a low–pass filter with a cutoff frequency of 3400 Hz to simulate telephone quality.

In (Schuster et al., 2005), we showed for a corpus of 18 TE speakers that a monophone–based recognizer for close–talk signals produced slightly better agreement with speech experts' intelligibility ratings than a polyphone–based recognizer. We wanted to verify these results for a larger corpus. Therefore we created four different recognizers: For the 16 kHz and the 8 kHz training data, we created a polyphone–based and a monophone–based recognizer (rows "16kHz/mono", "8kHz/mono", "16kHz/poly", "8kHz/poly" in Table 3). After the training, the vocabulary was reduced to the words occurring in the German version of the "The North Wind and the Sun" text, a fable from Aesop. It is a phonetically rich text with 108 words (71 disjoint) which is often used in speech therapy in German speaking countries.

4. Evaluation Data

41 laryngectomees ($\mu = 62.0 \pm 7.7$ years old, 2 female and 39 male) with TE substitute voice read the German version of the text "The North Wind and the Sun". The speech samples were recorded with a close–talk microphone ("dnt Call 4U Comfort" headset) at a sampling frequency of 16 kHz and quantized with 16 bit (linear).

Eight of the patients additionally read the "The North Wind and the Sun" text to an automatic telephone–based recording system (the recording system was not yet available at the time of the recording of the other 33 patients). The samples were recorded with 8 kHz and quantized with 16 bit (linear). However, one has to keep in mind that the signal is logarithmically companded (8 bit) during transmission which is approximately equivalent to 12 bit linear (rows "telephone calls" in Table 3).

Each close–talk recording was rated by 5 voice professionals (see Sec. 5.). Previous work (Schuster et al., 2006; Schuster et al., 2005) showed that there exists a significant correlation between experts' intelligibility ratings and the speech recognizer's word accuracy (WA) for close–talk recordings. If an automatic evaluation of TE telephone speech is possible, there must be a similar correlation using telephone data. To determine the change of correlation, we created three additional versions of the close–talk data:

1. We downsampled the data to 8 kHz applying the same low–pass filter (3400 Hz) as for the training data (rows

[1]The training and validation corpus was thus the same as in (Gallwitz, 2002; Stemmer, 2005).

"low–pass 3400" in Table 3).

2. In order to simulate the loss due to the logarithmic encoding in the telephone channel, we converted these linearly quantized signals to μ–law companded signals and back to linearly quantized signals (rows "low–pass 3400, μ–law" in Table 3).

3. In order to get a "telephone quality" version of the signals, we played back the close–talk recordings using a standard PC and loudspeaker in a quiet office environment and placed a telephone headset in front of the loudspeaker. The replayed sound files were recorded with the same automatic dialogue system over the telephone mentioned above with 8 kHz and 16 bit linear (again, the signals were logarithmically companded during telephone transmission). Thus we simulated a real telephone call (rows "simulated telephone" in Table 3). Due to the multiple AD/DA conversions and the different frequency characteristics of the loudspeaker and the microphones we expect the recognition rates to be a lower bound for the recognition rates for real telephone calls.

Figure 2 shows spectrograms of a short passage from the "The North Wind and the Sun" fable. The recordings are from one speaker who was recorded with the close–talk microphone (top) and with the telephone–based system (bottom). The spectrogram in the middle is from the downsampled close–talk version which was μ–law companded.

5. Subjective Evaluation

A group of 5 voice professionals subjectively estimated the intelligibility of the patients while listening to a play–back of the close–talk recordings. A five–point Likert scale (1 = very high, 2 = rather high, 3 = medium, 4 = rather low, 5 = very low) was applied to rate the intelligibility of each recording. In this manner an averaged mark – expressed as a floating point value – for each patient could be calculated.

To judge the agreement between the different raters we calculated correlation coefficients and the weighted multi–rater κ. For each rater we calculated the correlation between his "intelligibility" rating and the average of the 4 other raters. Table 1 shows the correlation coefficient for each rater and the average correlation coefficient.

rater	K	L	R	S	U	avg.
others	.82	.80	.81	.85	.77	.81

Table 1: Correlation coefficients between single raters and the average of the 4 other raters for the criterion "intelligibility".

The weighted multi–rater κ by Davies and Fleiss (Davies and Fleiss, 1982) also allows to compare an arbitrary number of raters and weights the difference between the values to compare. This means e.g. for the case that rater a gives a score of 2 and rater b gives a score of 3, this pair of numbers "matches better" and is therefore weighted higher as if person b rated the test data with a 4.

The weights were chosen as proposed by Cicchetti (Cicchetti, 1976) with

$$w_{xy}^{(a,b)} = 1 - \left(\frac{x-y}{C-1}\right)^2 \quad . \tag{1}$$

A κ value greater than $.4$ is said to show moderate agreement. The weighted multi–rater κ for the 5 raters was .45.

6. Automatic Evaluation

We used the experts' intelligibility ratings for the close–talk recordings as a reference for all 4 versions of the recordings: We applied the two close–talk recognizers and the two telephone speech recognizers to the accordant speech data and calculated the correlation between the WAs and the average of the experts' intelligibility rating. The κ values were calculated using the recognizer as a 6th rater. For this we mapped the WAs to marks on the Likert scale, using the thresholds that are given in Table 2.

WA	< 0	< 15	< 25	< 40	≥ 40
Mark	5	4	3	2	1

Table 2: Thresholds for mapping the WA of the ASR system to marks on the Likert scale for rating the intelligibility of the patients.

Table 3 shows the results for the monophone–based recognizers (row 1–4) and the polyphone–based recognizers (row 5–8) for the 41 patients. In addition, the results for the 8 real telephone calls are displayed (row 9-10). Note that the correlation and κ value were computed w.r.t. the ratings of the close–talk data of these patients, i.e. a different recording. The WA for these 8 patients was 23.0% for the simulated telephone calls and and 39.7% for the close–talk recordings using the polyphone–based recognizer compared to 37.0 for the real telephone calls.

Figure 3 shows the WAs of the 41 close–talk recordings compared to the simulated telephone recordings using polyphone–based recognizers. The recordings are ordered with increasing WA for the close–talk recordings.

Figure 4 shows for the 41 recordings the WA in comparison to the average of the experts' intelligibility scores using simulated telephone data and the polyphone–based recognizer.

7. Discussion

The results of the evaluation for the 41 patients show the possibility of an automatic objective way to rate the intelligibility of TE speech. The correlation between the WA of the respective polyphone–based recognizers and the average of the experts' intelligibility scores is only reduced from -.88 to -.80, when going from close–talk to simulated telephone speech.

Adding the recognizer as a 6^{th} expert to the expert group, does not change the κ value significantly. Due to the loss of quality in telephone transmission, the multiple AD/DA conversions, and the different frequency characteristics of the loudspeaker and the microphones, the overall WA for the simulated telephone calls is reduced. Also, the

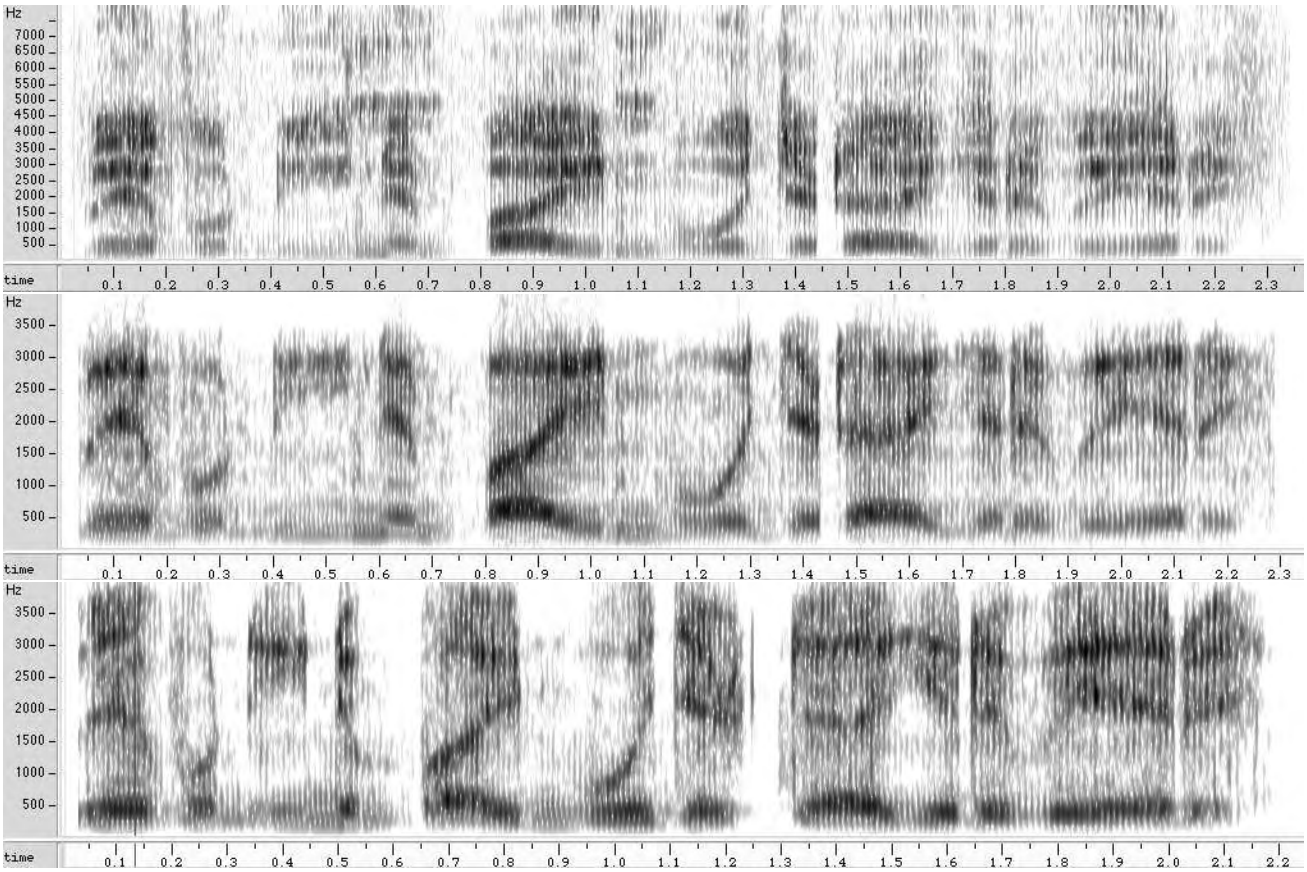


Figure 2: Spectrograms from the German utterance “wer von ihnen beiden wohl der Stärkere wäre”: 16 kHz close–talk vs. 8 kHz downsampled and μ–law companded vs. 8 kHz real telephone data.

#	recording	data/recognizer	μ(WA)	σ(WA)	correlation	weighted κ
41	close–talk	16kHz/mono	35.3	13.7	-.82	.41
41	low–pass 3400	8kHz/mono	33.4	12.1	-.81	.42
41	low–pass 3400, μ–law	8kHz/mono	33.6	12.7	-.78	.42
41	simulated telephone	8kHz/mono	28.4	10.3	-.69	.42
41	close–talk	16kHz/poly	36.9	18.0	-.88	.45
41	low–pass 3400	8kHz/poly	32.3	17.4	-.85	.47
41	low–pass 3400, μ–law	8kHz/poly	33.1	16.7	-.86	.46
41	simulated telephone	8kHz/poly	26.4	13.9	-.80	.46
8	telephone calls	8kHz/mono	32.9	12.8	-.55	.27
8	telephone calls	8kHz/poly	37.0	15.1	-.75	.32

Table 3: Evaluation results for the four different recognizers for the 41 patients and for the 8 real phone calls.

training data of the speech recognizer for the 8 kHz was downsampled close–talk data and not real telephone data. We chose this way instead of using real telephone training data, since we wanted the telephone recognizer to be trained with the same training data as the recognizer for the close–talk data. Reducing the acoustical distance of training and evaluation data might lower the loss of correlation. An acoustic comparison (see Figure 2) of the 8 kHz resampled data to the real telephone data shows that the application of a low–pass filter with a cutoff–frequency of 3800 Hz and μ–law quantization lead to a good acoustic distance. By modifying the training data accordingly, we expect more robust recognition results. Furthermore, we expect better recognition rates by modifying the feature extraction, which is our current research.

The results in Table 3 show that for a larger corpus the polyphone–based recognizer leads to better correlation with the experts’ group. Thus the results from (Schuster et al., 2005) for 18 patients, where the monophone–based recognizer showed better agreement, were not confirmed.

Experiments with the 8 real telephone calls support these conclusions, even though this database is way too small to draw conclusions. The WA for the real telephone data is higher than for the simulated calls, probably for the

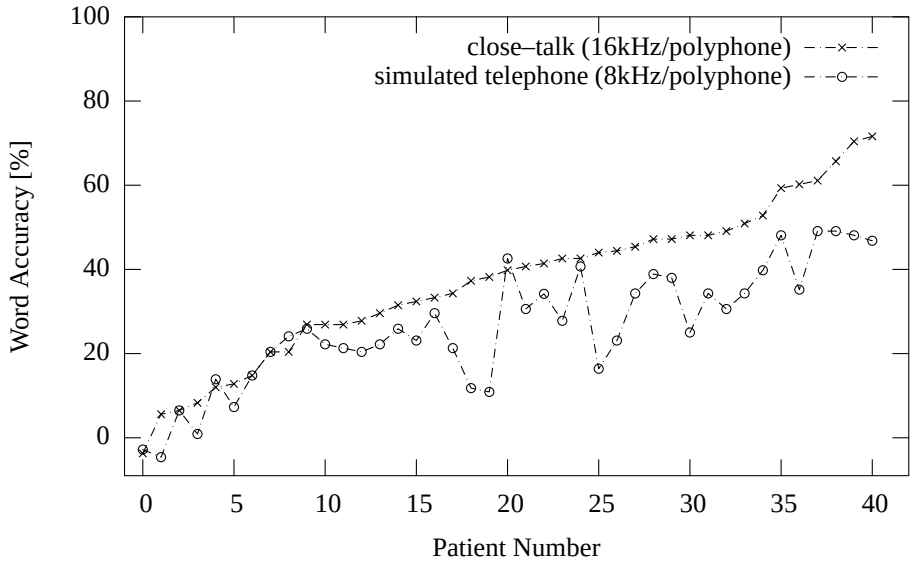


Figure 3: WAs of the 41 close–talk recordings compared to the simulated telephone recordings using polyphone–based recognizers. The recordings are ordered with increasing WA for the close–talk recordings.

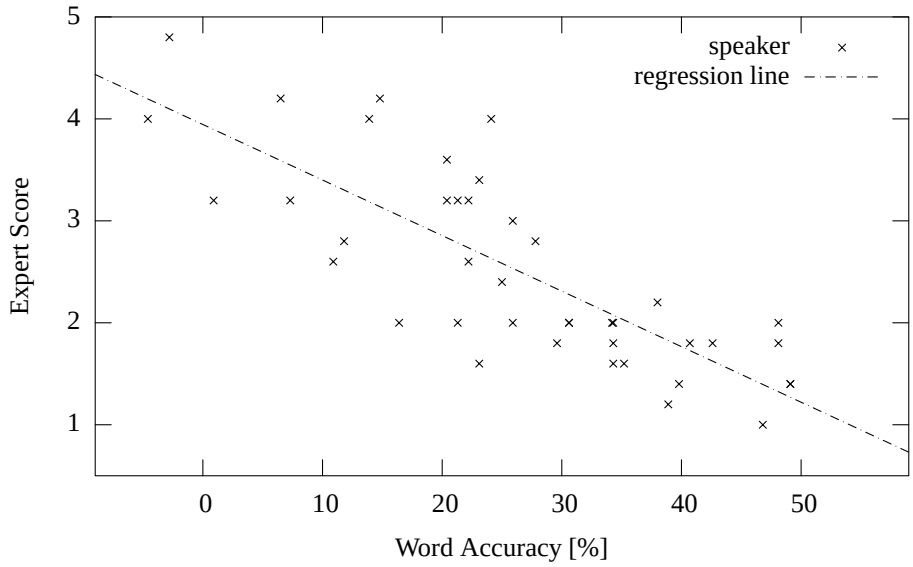


Figure 4: WA for the 41 recordings in comparison to the average of the experts' intelligibility scores using simulated telephone data and the polyphone–based recognizer.

reasons given above. The reduced κ values could be caused by the fact that the human ratings refer to a different recording and by the small corpus size. We are currently collecting a larger telephone corpus to verify the results presented in this paper.

8. Acknowledgments

This work was funded by the German Cancer Aid (Deutsche Krebshilfe) under grant 106266. The responsibility for the content of this paper lies with the authors.

9. References

M.H. Bellandese, J.W. Lerman, and H.R. Gilbert. 2001. An Acoustic Analysis of Excellent Female Esophageal, Tracheoesophageal, and Laryngeal Speakers. *Journal of Speech, Language, and Hearing Research*, 44:1315–1320.

D.H. Brown, F.J.M. Hilgers, J.C. Irish, and A.J.M. Balm.

2003. Postlaryngectomy Voice Rehabilitation: State of the Art at the Millennium. *World J Surg*, 27(7):824–831.

D.V. Cicchetti. 1976. Assessing inter-rater reliability for rating scales: Resolving some basic issues. *British Journal of Psychiatry*, 129(5):452–456.

M. Davies and J.L. Fleiss. 1982. Measuring agreement for multinomial data. *Biometrics*, 38(4):1047–1051.

F. Gallwitz. 2002. *Integrated Stochastic Models for Spontaneous Speech Recognition*, volume 6 of *Studien zur Mustererkennung*. Logos Verlag, Berlin.

J. Lohscheller. 2003. *Dynamics of the Laryngectomee Substitute Voice Production*. Shaker, Aachen.

J. Robbins, H.B. Fisher, E.C. Blom, and M.I. Singer. 1984. A Comparative Acoustic Study of Normal, Esophageal, and Tracheoesophageal Speech Production. *Journal of Speech and Hearing Disorders*, 49:202–210.

E. G. Schukat-Talamazzini. 1995. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Braunschweig.

M. Schuster, E. Nöth, T. Haderlein, S. Steidl, A. Batliner, and F. Rosanowski. 2005. Can you Understand him? Let's Look at his Word Accuracy — Automatic Evaluation of Tracheoesophageal Speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, USA. IEEE Computer Society Press.

M. Schuster, T. Haderlein, E. Nöth, J. Lohscheller, U. Eysholdt, and F. Rosanowski. 2006. Intelligibility of Laryngectomees' Substitute Speech: Automatic Speech Recognition and Subjective Rating. *European Archives of Oto-Rhino-Larngology and Head & Neck*, 263:188–193.

H.K. Schutte and G.J. Nieboer. 2002. Aerodynamics of esophageal voice production with and without a Groningen voice prosthesis. *Folia Phoniatrica et Logopaedia*, 54:8–18.

G. Stemmer. 2005. *Modeling Variability in Speech Recognition*, volume 19 of *Studien zur Mustererkennung*. Logos Verlag, Berlin.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin.

First Results of a Hungarian Medical Dictation Project

András Bánhalmi, Dénes Paczolay, Lászó Tóth, András Kocsor

Research Group on Artificial Intelligence
Hungarian Academy of Sciences and the University of Szeged
Aradi vértanúk tere 1, H-6720 Szeged
{banhalmi, pdenes, tothl, kocsor}@inf.u-szeged.hu

Abstract

This paper reviews the current state of a Hungarian project that seeks to create a speech recognition system for the dictation of thyroid gland medical reports. We present the MRBA speech corpus that was collected to support the training of Hungarian LVCSR systems. Besides the speech data, a huge set of medical reports was also collected to help the creation of domain-specific language models. At the acoustic modelling level we experiment with two techniques – a conventional HMM one and an ANN-based solution – which are both briefly described in the paper. Then we present the language modelling methodology currently applied in the system, and round off with recognition results on test data taken from four people. The scores show that on the current restricted domain we are able to produce word accuracies over 95%, but the planned extension of the system to larger vocabularies will probably require further improvements.

Prvi rezultati madžarskega projekta narekovanja zdravniških izvidov

Prispevek predstavlja pregled trenutnega stanja madžarskega projekta, ki skuša vzpostaviti sistem razpoznavanja govora za narekovanje zdravniških izvidov na temo žleze ščitnice. Predstavljamo govorni korpus MRBA, ki je bil sestavljen za podporo učenju madžarskih sistemov za razpoznavanje govora z velikim besednjakom. Poleg govornih podatkov je bilo zbrano tudi veliko število zdravniških izvidov za pomoč pri pripravi jezikovnih modelov za omenjeno področje uporabe. Na ravni akustičnega modeliranja eksperimentiramo z dvema tehnikama - konvencionalno s prikritimi Markovovimi modeli in rešitvijo, ki temelji na nevronskih mrežah - obe sta kratko predstavljeni. Nato predstavljamo metodologijo jezikovnega modeliranja, ki je trenutno uporabljena v sistemu, in zaključimo z rezultati razpoznavanja na testnih podatkih štirih govorcev. Rezultati pokažejo, da smo pri trenutnem omejenem področju uporabe zmožni dosegati točnost razpoznavanja besed višjo kot 95%, načrtovana razširitev sistema na širše besedišče pa bo verjetno zahtevala dodatne izboljšave.

1. Introduction: goals of the project

At the present time there exists no general-purpose large vocabulary continuous speech recognizer (LVCSR) for the Hungarian language. Among the university publications even papers that deal with continuous speech recognition are hard to find, and these present results only for restricted vocabularies (Szarvas and Furui, 2002). Although on the industrial side Philips have adapted its SpeechMagic system to two special domains in Hungarian, it is sold at a price that is affordable for only the largest institutes (Medisoft, 2004). The experts usually mention two reasons for the lack of Hungarian LVCSR systems. First, there are no sufficiently large, publicly available speech databases that would allow the training of reliable phone models. The second reason is the difficulties of language modelling due to the highly agglutinative nature of Hungarian.

In 2004 the Research Group on Artificial Intelligence, University of Szeged and the Laboratory of Speech Acoustics of the Budapest University of Technology and Economics started a project with the aim of collecting and/or creating the basic resources needed for the construction of a continuous dictation system. The project lasts for three years, and is financially supported by the national fund IKTA-056/2003. As regards acoustic modelling, the project includes the collection and annotation of a large speech corpus of phonetically rich sentences. As regards language modelling, we restricted the target domain to the dictation of certain types of medical reports. Although this clearly leads to a significant reduction compared to the original, general dictation task, we chose this application area with the intent of assessing the capabilities of our acoustic and language modelling technologies. Depending on the findings, later we hope to extend the system to more general dictation domains. This is why the language resources were chosen to be domain-specific, while the acoustic database contains quite general, domain-independent recordings.

Although both teams use the same speech corpus for training, they focus on different dictation tasks and experiment with their own acoustic and language modelling technologies. Our team (Szeged) deals with the dictation of thyroid scintigraphy medical reports, while the Budapest team deals with gastroenterology reports. This paper describes the current state of development of the Szeged team only.

2. Speech and language resources

In the first phase of the project we designed, collected and annotated a speech database that we refer to as the MRBA corpus (the abbreviation stands for the "Hungarian Reference Speech Database") (Vicsi et al., 2004). Our goal was to create a database that allows the training of general-purpose dictation systems which run on personal computers in office environments and work with continuous, read speech. The contents of the database were designed by the Laboratory of Speech Acoustics. As a starting point, they took a large (1.6 MB) text corpus and after automatic phonetic transcription they created phone, diphone and triphone statistics from it. Then they selected 1992 different sentences and 1992 different words in such a way that 98.8% of the most frequent diphones had at least

one occurrence in them. These sentences and words were recorded from 332 speakers, each reading 12 sentences and 12 words. Thus all sentences and words have two recordings in the speech corpus. Both teams participated in the collection of the recordings, which was carried out in four big cities, mostly at universities labs, offices and home environments. In the database the ratio of male and female speakers is 57.5% to 42.5%. About one-third of the speakers are between 16-30 years in age, the rest being evenly distributed among the remaining age groups. Both home PCs and laptops were used for the recordings, and the microphones and the sound cards of course varied as well. The sound files were cleaned and annotated at the Laboratory of Speech Acoustics, while the Research Group on Artificial Intelligence manually segmented and labelled one third of the files at the phone level. This part of the corpus is intended to support the initialization of phone models.

Besides the general-purpose MRBA corpus, we also collected recordings that are specific for the target domain, namely thyroid scintigraphy medical reports. From these recordings 20-20 reports read aloud by 4 persons were used as test data in the experiments done here.

For the construction of the domain-specific language models, we obtained 9231 written medical reports from the Department of Nuclear Medicine of the University of Szeged. These thyroid scintigraphy reports were written and stored between 1998 and 2004 using various software packages that were employed at the department during that period. So first of all we had to convert all the reports to a common format, which was followed by several steps of error correction. Each report consists of 7 fields: header (name, ID number etc. of the patient), clinical observations, request of the referral doctor, a summary of previous examinations, the findings of this examination, a one-sentence summary, and a signature. From the corpus we omitted the first and the last, person-specific fields, for the sake of personal privacy. Then we discarded those reports that were incomplete like those that had missing fields. This way only 8546 reports were kept, which contain 11 sentences and 6 words per sentence on average. The next step was to remove any typographical errors from the database, of which there were surprisingly many (some words occurred in 10-15 mistyped forms). A special problem was that of unifying those Latin terms that can be written both with a Latin or a Hungarian spelling. The abbreviations had to be resolved, too. The corpus we got after these steps contained approximately 2500 different word forms, so we were confronted with a medium-sized vocabulary dictation task.

3. Acoustic modelling I: HMM phone models over MFCC features

At the level of acoustic modelling we have been experimenting with two quite different technologies. One of these is a quite conventional Hidden Markov Model (HMM) decoder that works over the usual mel-frequency cepstral coefficient (MFCC) features (Huang et al., 2001). More precisely, 13 coefficients are extracted from 25 msec frames, along with their Δ and $\Delta\Delta$ values, at a rate of 100 frames/sec. The phone models applied have the usual 3-state left-to-right topology. Although Hungarian has the special property that almost all phones have a short and a long counterpart, in the vocabulary of our specific dictation task they seemed to have no discriminative role. Hence most of the long/short consonant labels were fused, and this way we worked with just 44 phone classes. One phone model was associated with each of these classes, that is we applied monophone modelling and no context-dependent models were tested in the system. The decoder built on these HMM phone models performs a combination of Viterbi and multi-stack decoding. For speed efficiency it contains several built-in pruning criteria. First, it applies beam pruning, so only the hypotheses with a score no worse than the best score minus a threshold are kept. Second, the number of hypotheses extended at every time point is limited, corresponding to multi-stack decoding with a stack size constraint. The maximal evaluated phone duration can also be limited. Normally the decoder runs faster than real-time on our dictation task on a typical PC.

4. Acoustic modelling II: HMM/ANN phone models over 2D-cepstrum features

Our alternative, more experimental acoustic model employs the HMM/ANN hybrid technology (Bourlard and Morgan, 1994). The basic difference between this and the standard HMM scheme is that here the emission probabilities are modelled by Artificial Neural Networks (ANNs) instead of the conventional Gaussian mixtures. In the simplest configuration one can train the neural net over the usual 39 MFCC coefficients – whose result can serve as a baseline for comparison with the conventional HMM. However, ANNs seem to be more capable of modelling the observation context than the GMM technology, so the hybrid models are usually trained over longer time windows. The easiest solution for this is to specify a couple of neighboring feature frames as input to the net: a conventional arrangement is to use 4 neighboring frames on both sides of the actual frame (Bourlard and Morgan, 1994). Another option is to apply some kind of transformation on the data block of several neighboring frames. Knowing that the modulation components play an important role in human speech perception, performing a frequency analysis over the feature trajectories seems reasonable. When this analysis is applied to the cepstral coefficients, the resulting feature set is usually referred to as the 2D-cepstrum (Kanedera et al., 1998). Research shows that most of the useful linguistic information is in the modulation frequency components between 1 and 16 Hz, especially between 2 and 10 Hz. This means that not all of the components of a frequency analysis have to be retained, and so the 2D-cepstrum offers a compact representation of a longer temporal context.

In the experiments we tried to find the smallest feature set that gave the best recognition results. As a quick indicator of the efficiency of a representation we used the frame-level classification score, so the values given below are frame-level accuracy values (measured on a held-out data set of 20% of the training data). First of all we tried to extend the data of the 'target' frame by neighboring frames, without applying any transformation. The results shown in Table 1 indicate that training on more than 5 neighboring frames only significantly increased the number of features

and hidden neurons (and even more considerably the training time) without bringing a real improvement in the score.

In the experiments with the 2D-cepstrum we first tried to find the optimal size of the temporal window. Hence we varied the size of the DFT analysis between 8, 16, 32, and 64, always retaining the first and second components (both the real and the imaginary parts), and combined these with the static MFCC coefficients. The results displayed in Table 2 indicate that the optimum must be somewhere between 16 and 32 (160 and 320 milliseconds). This is smaller than the 400 ms value found optimal by Kanedera et al. (1998) and the 310 ms value reported by Schwarz et al. (2003), but this might depend on the amount of training data available (a larger database would cover more of the possible variations and hence would allow a larger window size). Of course, one could also experiment with the combination of various window sizes as Kanedera et al. (1998) did, but we did not run such multi-resolution tests.

As the next step we examined whether it was worth retaining more components. In the case of the 16-point DFT we kept 3 components, while for the 32-point DFT we tried retaining 5 components (the highest center frequency being 18.75 Hz and 15.625 Hz, respectively). The results show (Table 3) that the higher modulation frequency components are less useful, which accords with what is known about the importance of the various modulation frequencies.

Finally, we tried varying the type of transformation applied. Motlíček reported that there is no need to retain both the real and imaginary parts of the DFT coefficients; using just one of them is sufficient. Also, he obtained a similar performance when replacing the complex DFT with DCT (Motlíček, 2003). Our findings agree more with those of Kanedera et al. (1998), that is we obtained slightly worse results with these modifications (see Table 4). So we opted for the complex DFT, using both the real and imaginary coefficients. One advantage of the complex DFT over the DCT might be that when only some of its coefficients are required (as in our case), it can be very efficiently computed using a recursive formulation (Jacobsen and Lyons, 2004).

Obs. size	Hidden neurons	Frames correct
1 frames	150	64.16%
3 frames	200	67.51%
5 frames	250	68.67%
7 frames	300	68.81%
9 frames	350	68.76%

Table 1: The effect of varying the observation context size.

DFT size	Hidden neurons	Frames correct
8	200	64.63%
16	200	67.60%
32	200	67.01%
64	200	64.75%

Table 2: Frame-level results at various DFT sizes.

DFT Size	Components	H. neurons	Frames corr.
16	1, 2, 3	250	68.40%
32	1, 2, 3, 4, 5	300	70.64%

Table 3: Frame-level results with more DFT components.

Transform	Hidden neurons	Frames correct
DFT Re + Im	300	70.64%
DFT Re only	220	65.81%
DCT	220	68.00%

Table 4: The effect of varying the transformation type.

5. Domain-specific language modelling

A special difficulty of creating language models for Hungarian is the highly agglutinative nature of the language. In a large vocabulary modelling task the application of a morphologic analyzer/generator seems inevitable. First, simply listing and storing all the word forms would be nearly impossible (an average noun can have about 700 inflected forms). Second, if we simply handled all these inflected forms as different words, then achieving a certain coverage rate in Hungarian would require a text about 5 times bigger than that in German and 20 times bigger than that in English (Németh and Zainkó, 2001). Hence, the training of conventional N-gram models would require significantly larger corpora in Hungarian than in English, or even in German. A possible solution might be to train the N-grams over morphemes instead of word forms, but then again the handling of morphology would be necessary.

Though quite good morphological tools exist now for Hungarian, in the first experiments with our system we preferred to avoid the complications with morphology. The restricted vocabulary is one of the reasons why we chose the medical dictation task. As was mentioned, the thyroid gland medical reports contained only about 2500 different word forms. Although these many words could be easily managed even by a simple list ('linear lexicon'), we organize them into a lexical tree where the common prefixes of the lexical entries are shared. Apart from storage reduction advantages, this representation also speeds up decoding, as it eliminates redundant acoustic evaluations (Huang et al., 2001). The prefix tree representation is quite probably even more useful for agglutinative languages than for English, because of the many inflected forms of the same stem.

The limited size of the vocabulary and the highly restricted (i.e. low-perplexity) nature of the sentences used in the reports allowed us to create very efficient N-grams. Moreover, we did not really have to worry about out-of-vocabulary words, since we had all the reports from the previous six years, so the risk of facing unknown words during usage seemed minimal. The system currently applies 3-grams by default, but it is able to 'back off' to smaller N-grams (in the worst case to a small ϵ constant) when necessary. During the evaluation of the N-grams the system applies a language model lookahead technique. This means that the language model returns its scores as early as possible, not just at word endings. For this purpose the lexical trees get factored, so that when several words share a common prefix, the maximum of their probabilities is associated with that prefix (Huang et al., 2001). These tech-

Model Type	Feature Set	Male 1	Male 2	Female 1	Female 2
HMM	MFCC + $\Delta + \Delta\Delta$	97.75%	98.22%	93.40%	93.39%
HMM/ANN	MFCC + $\Delta + \Delta\Delta$	97.65%	97.37%	96.78%	96.91%
HMM/ANN	5-frames $*$ (MFCC + $\Delta + \Delta\Delta$)	97.65%	97.74%	96.67%	98.05%
HMM/ANN	MFCC + 5 Mod. Comp. (Re + Im)	97.88%	97.83%	96.86%	96.42%

Table 5: Word recognition accuracies of the various models and feature sets.

niques allow a more efficient pruning of the search space.

Besides word N-grams we also experimented with constructing class N-grams. For this purpose the words were grouped into classes according to their parts-of-speech category. The words were categorized using the POS tagger software developed at our university (Kuba et al., 2004). This software associates one or more MSD (morpho-syntactic description) code with the words, and we constructed the class N-grams over these codes. With the help of the class N-grams the language model can be made more robust in those cases when the word N-gram encounters an unknown word, so it practically performs a kind of language model smoothing. In previous experiments we found that the application of the language model lookahead technique and class N-grams brought about a 30% decrease in the word error rate when it was applied in combination with our HMM-based fast decoder (Bánhalmi et al., 2005).

6. Experimental results and discussion

For testing purposes we recorded 20-20 reports from 2-2 male and female speakers. The language model applied in the tests was constructed based on only 500 reports instead of all the 8546 ones we collected. This subset contained almost all the sentence types that occur in the reports, so this restriction mostly reduced the dictionary by removing a lot of rarely occurring words (e.g. dates and disease names). Besides the HMM decoder we tested the HMM/ANN hybrid system in three configurations: the net being trained on one frame of data, on five neighboring frames, and on the best 2D-cepstrum feature set (static MFFC features plus 5 modulation components using a 32-point complex DFT, both Re and Im parts). The results are listed in Table 5. Comparing the first two lines, we see that when using the same features the HMM and the HMM/ANN system performed quite similarly on the male speakers. For some reason, however, the HMM system did not like the set of female voices. Extending the net's input with an observation context – either by neighboring frames or by modulation features – brought only a modest improvement over the baseline results. We think that the improvement in the acoustic modelling will be more prominently reflected in the scores when moving to a linguistically less restricted domain where the decoder cannot rely so strongly on the language model as it does in the current configuration.

7. Conclusions

This paper reported the current state of a Hungarian project for the automated dictation of medical reports. We described the acoustic and linguistic training data collected and the current state of development in both the acoustic and linguistic modelling areas. Preliminary recognition results were also given over a somewhat restricted subset of the full domain to be handled. As the next step we plan to extend the vocabulary and language model to cover all the available data, and our preliminary results show that for a larger vocabulary several further improvements will be necessary. On the acoustic modelling side we intend to implement speaker adaptation and context-dependent models (within the HMM system). We also plan to continue our research on observation context modelling (within the HMM/ANN system). Finally, the language model will also need to be improved, especially when handling certain special features like dates or abbreviations.

8. References

A. Bánhalmi, A. Kocsor, and D. Paczolay. 2005. Supporting a Hungarian dictation system with novel language models (in Hungarian). In: *Proc. of the 3rd Hungarian Conf. on Computational Linguistics*, pp. 337–347.

H. Bourlard and N. Morgan. 1994. *Connectionist Speech Recognition – A Hybrid Approach*. Kluwer Academic.

X. Huang, A. Acero, and H.-W. Hon. 2001. *Spoken Language Processing*. Prentice Hall.

E. Jacobsen and R. Lyons. 2004. An update to the sliding DFT. *IEEE Signal Processing Mag.*, 21(1):110–111.

N. Kanedera, H. Hermansky, and T. Arai. 1998. Desired characteristics of modulation spectrum for robust automatic speech recognition. In: *Proc. of ICASSP'98*, pp. 613–616.

A. Kuba, A Hócza, and J. Csirik. 2004. POS tagging of Hungarian with combined statistical and rule-based methods. In: *Proc. of TSD 2004*, pp. 113–121.

Medisoft. 2004. www.medisoftspeech.hu.

P. Motlíček. 2003. *Modeling of Spectra and Temporal Trajectories in Speech Processing*. Ph. D. Dissertation, Brno University of Technology.

G. Németh and Cs. Zainkó. 2001. Word unit based multilingual comparative analysis of text corpora. In: *Proc. of Eurospeech 2001*, pp. 2035–2038.

P. Schwarz, P. Matějka, and J. Černocký. 2003. Recognition of phoneme strings using TRAP technique. In: *Proc. of Eurospeech 2003*, pp. 825–828.

M. Szarvas and S. Furui. 2002. Finite-state transducer based Hungarian LVCSR with explicit modeling of phonological changes. In: *Proc. of ICSLP 2002*, pp. 1297–1300.

K. Vicsi, A. Kocsor, Cs. Teleki, and L. Tóth. 2004. Hungarian speech database for computer-using environments in offices (in Hungarian). In: *Proc. of the 2nd Hungarian Conf. on Computational Linguistics*, pp. 315–318.

A Natural Language Interface to a Theater Information Database

Margus Treumuth[1], Tanel Alumäe[2], and Einar Meister[2]

[1] Institute of Computer Science
University of Tartu, Tartu, Estonia
treumuth@ut.ee
[2] Institute of Cybernetics
Tallinn University of Technology
Tallinn, Estonia
{tanel.alumae, einar}@phon.ioc.ee

Abstract

The development of a natural language dialogue system as an interface to a theater information database is a joint research project of the University of Tartu (Estonia) and the Tallinn University of Technology (Estonia). The underlying database contains information about theater performances in a certain theater or city. The dialogue system can be used to ask information about performances using either spoken or typewritten natural language in Estonian. The dialogue management module was developed at the University of Tartu while the modules for speech recognition and speech synthesis were added by the Tallinn University of Technology. This article discusses the development of the dialogue module, the speech recognition module, and the speech synthesis module.

Sistem za dialog v naravnem jeziku kot vmesnik do gledališke podatkovne baze

Razvoj sistema za dialog v naravnem jeziku kot vmesnik do gledališke podatkovne baze je skupni raziskovalni projekt Univerze v Tartuju (Estonija) in Tehniške univerze v Talinu (Estonija). Podatkovna baza vsebuje informacije o predstavah v določenem gledališču ali mestu. Sistem za dialog je mogoče uporabiti za pridobivanje informacij o predstavah bodisi v govorjeni ali pisni estonščini. Modul za vodenje dialoga je bil razvit na Univerzi v Tartuju, medtem ko so module za razpoznavanje in sintezo govora dodali na Tehniški univerzi v Talinu. Prispevek obravnava razvoj modulov za dialog, razpoznavanje govora in sintezo govora.

1. Introduction

The dialogue system developed in this project operates in a constrained linguistic domain – theater information. The underlying database contains information about theater performances in a certain theater or city. The dialogue system can be used to ask information about performances. Currently, the system does not contain price or booking information, also the names of actors and authors are not included at this time. The system can deal with either typewritten or spoken language. The typewritten interface is accessible at http://www.dialoogid.ee/. The language used by the dialogue system is Estonian.

The research groups involved are with the Institute of Cybernetics at the Laboratory of Phonetics and Speech Technology of the Tallinn University of Technology, and with the Institute of Computer Science at the Research Group of Computational Linguistics at Tartu University.

The paper is organized as follows. Section 2 describes the system components. Section 3 is concerned with experiments. Section 4 describes the software environment of the implementation. Finally, our conclusions are stated in section 5.

2. System Components

The dialogue system consists of modules for speech recognition, dialogue management, morphological analysis, query generation and speech synthesis (see Figure 1). The date recognition module is a submodule of query generation The theater information is stored in a relational database system. The speech recognition, dialogue management and speech synthesis modules are all run as autonomous services

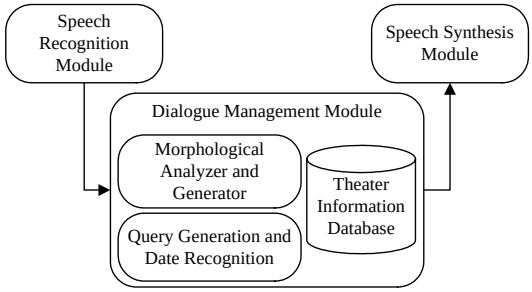


Figure 1: Dialogue system architecture

2.1. Speech Recognition Module

The speech recognition module segments the input stream into utterances and produces a recognition hypothesis for each segment. It also triggers barge-in, if it detects speech that continues for a configurable amount of time. Barge-in sends a signal to the speech synthesis module to stop any speech output.

2.1.1. Acoustic Modeling

The acoustic models for recognition experiments were trained on the Estonian SpeechDat-like phonetic database (Meister, et al., 2003), collected from volunteer speakers over the telephone network. The total number of different speakers in the database is 1332. The number of acceptable utterances is 177 793. This represents about 241.1 hours of audio data.

The speech data was recorded at an 8 kHz sampling rate and coded using 8-bit mono A-law. The recording sessions consisted of a fixed set of utterance types, such as isolated and connected digits, natural numbers, monetary amounts, spelled words, time phrases, date phrases, yes/no

answers, person and company names, application words and phrases, phonetically rich words, and sentences.

The open source SphinxTrain toolkit was used for training the acoustic models. Models were created for 25 phonemes, the five filler/noise types and silence. For acoustic features, MFCC coefficients were used. The coefficients were calculated from a frequency band ranging from 130 Hz to 3400 Hz, using a pre-emphasis coefficient of 0.9. The window size was 0.0256 seconds and the frame rate was 100 frames/second. A 512-point FFT was used to calculate 31 filter banks, out of which 13 cepstral coefficients were generated. All units are modeled by continuous left-to-right HMMs with three emitting states and no skip transitions. The output vectors are 39-dimensional and are composed of 13 cepstral coefficients, delta and double delta coefficients. The final tied-state triphone models have 8000 shared states in total. Each state is modeled by eight Gaussian mixture components.

The pronunciation dictionary is automatically created from word orthography using a set of context sensitive rewrite rules. Since many performance names contain foreign names, there is an additional manually compiled pronunciation dictionary of non-native words that is merged to the rule-driven dictionary.

2.1.2. Language Modeling

For speech recognition language modeling, a class-based trigram model is used. The training data for the language model is a set of sample questions to the system, composed by system developers and testers, and collected during live system testing. Some of the word classes used in the language model are: city names, theater names, performance names (synchronizable with the dialogue manager database), day-of-month names, month names, and weekdays. As Estonian is an inflective language, most such words can occur in many inflections. Thus, there are separate classes for each common inflection, e.g., [weekday, nom. sg.], [weekday, gen. sg.], [weekday, ad. sg.] as in *esmaspäev*, 'Monday' nom. sg.; *esmaspäeva*, 'Monday' gen. sg.; and *esmaspäeval*, 'Monday' ad. sg.

During the training process, all words in the training sentences that belong to any class are replaced with the corresponding class tag. The resulting pseudo-sentences were used to train the trigram language model. All intra-class probabilities were distributed evenly.

One problem with the language model is the common use of shortened performance names in user queries. For example, instead of saying the full name "Pianola or The Mechanical Piano" (a popular performance), users tend to refer to just "Pianola", often using the inflected word form in the sentence (e.g., "*Millal mängitakse Pianolat?*", "When is Pianola (gen. sg.) being played?"). Such shorthand names are difficult to predict automatically from the performance database. To cope with this, we manually composed a list of such short names and put them to a separate class (actually two classes - one for the nominative and one for the often occurring genitive case). However, those classes must be manually checked from time to time for new entry candidates which creates some additional administrative burden.

2.2. Dialogue Management Module

The dialogue management module integrates an Estonian morphological analyzer/generator (Kaalep, 1997). The Estonian language is an agglutinative language that is rich in morphology. Therefore, the parsing technique involves automatic generation of lemmas or base forms using morphological analyzer. This way the system can handle minor deviations of the input. As a weakness – the morphological base form generation could also trigger ambiguity leading to unexpected results. Some deviations still cause problems and in the near future we will use the spell checking functions of the morphological analyzer to handle typing errors. We also plan to use the Levenshtein algorithm to calculate the distance between strings. This way we can guess which word (from a dictionary or database) is intended when an unknown word is encountered.

The morphological generator is used to produce required forms of words from base forms e.g. from *jaanuar* (January nom. sg.) to *jaanuaril* (on January ad. sg.).

The system also uses examples of linguistic phenomena from a dialogue corpus (Gerassimenko, et al., 2004), yet it is not a stochastic approach as probabilistic techniques are not used at the moment. We examined the corpus to see how the users phrase their questions and how they express dates and time.

2.2.1. Knowledge Base

The knowledge base of the dialogue management module consists of a primary database and a secondary database. The primary database contains the theater information (city, theater, performance, date) and the secondary database contains some simple linguistic facts (domain specific words by attribute-value pairs), e.g.:

Keyword (the knowledge base is in Estonian): *pilet* (ticket nom. sg.)

Values for keyword *pilet* (the number of values per keyword is not limited):

Piletite hinnad puuduvad. (Sorry, we have no ticket prices).

Piletite kohta kahjuks info puudub. (Sorry, we have no information about tickets).

Tean ainult etenduste algusaegasid. (We only know the dates of performances).

The primary database – the database of theaters and performances – is gathered manually from several online databases and is also updatable using a web interface. We also plan to arrange the database to run automatic updates daily.

The secondary database is also directly visible and adjustable in a knowledge base settings file. Various forms of greeting expressions and some domain specific phrases can be modified by a system administrator.

2.2.2. Query Generation Module

The query generation component converts user input to SQL (structured query language) queries - commands to be passed to the underlying database.

The primary parser detects proper names that occur in the primary relational database (names of performances and theaters), date and time phrases. Once a certain keyword has been recognized, the system may retrieve the answer from the database having the power of SQL available for quick definition and manipulation of data.

Let us consider a dialogue taking place between a human and a computer.

<Human>: I would like to see the musical *Cats* on *March 17*.
<Computer>: Cats is not playing on March 17. It is playing on March 19.

In this example, the highlighted words (Cats and March 17) are the actual keywords which convey the most important information and are recognized by the system. All the other words are semantically irrelevant and can be ignored. The inflections that occur in Estonian language, as in *märtsini* (March ter. sg.), *märtsil* (March ad. sg.), are handled by the morphological analyzer that generates the base form: *märts* (March nom. sg.).

The previous example might seem simple, yet the recognition of dates is not a simple task. There are many ways users can express dates and time (e.g. next Friday, on Christmas day, two weeks from today). Therefore, we have created a separate module in our system for date recognition.

The reaction to a user utterance depends on the state of the dialogue (dialogue context). That is, the choice of answer is based on previously acquired knowledge. Users can continue to ask queries about the previous topic. The system can remember facts the user has asked before. For example, if the user has mentioned a theater by name, all further references to some certain dates are handled in the context of the theater mentioned previously, e.g.,

<Human>: What is playing at Theater Royal?
<Computer>: The Producers is playing today.
<Human>: What about tomorrow?
<Computer>: There are no plays at the Theater Royal tomorrow.

The secondary parser is used if the primary parser gives no results. It can recognize only predefined words and/or phrases described in the secondary database and can only respond using a number of predefined sentence patterns also described in the secondary database.

Randomization is used in choosing the answer from the secondary database to provide the effect of non-linear transformations between inputs and outputs. Users tend to like slight unexpectedness and surprises (e.g., various expressions of greetings). Users will get bored if the system is too predictable and determine the limits of the system too quickly.

It is essential to keep users actively engaged in trying to get the answers they are searching for. This will provide the developers with valuable chat logs as the system stores all conversations. These chat logs are later used as training data to manually improve the performance of the linguistic model that relies on collection of predefined keywords and expressions to represent semantic notions.

2.3. Speech Synthesis Module

Search results can be presented to the user in two modalities: in text form and/or via speech output. In the latter case the written answer will serve as the input text for an Estonian text-to-speech synthesizer (Mihkla, et al., 1999).

Speech synthesis starts with the linguistic analysis of the input text, where the orthographic text is converted into phonemic representation. The linguistic module identifies numbers, abbreviations and acronyms in the input sentence and transforms them into full orthographic text. Next, the orthographic text is converted into an adequate phonemic representation. A prosody model calculates the phoneme durations and the contour of fundamental frequency according to the communicative type of sentence. Phonemic and prosodic information serve as input for the acoustic unit generation, which is based on the concatenative MBROLA model (Dutoit, et al., 1993). The MBROLA-engine utilizes diphones as the elementary concatenative units; the Estonian diphone database includes about 1700 diphones.

Using synthetic speech as the output of a dialogue system presents high demands on the prosodic (especially intonation) modeling – the spoken answer must be adequate with the dialogue structure and prosodically suit the on-going discourse. The current version of the text-to-speech synthesizer relies only on the linguistic information of the input text and is not able to model the prosodic structure of dialogue speech. Therefore, the speech output is produced almost identically for different types of answers. Significant improvement in prosody modeling could be achieved by including information about dialogue structure.

3. Results of Experiments

The system was tested with 150 conversations, some typewritten and some spoken. There were differences in typewritten and spoken conversations, yet the distinction between those is not important at this stage of development.

The system failed 25% of the time when attempting to answer a question. Yet, the subjects failed to communicate successfully with the system only 5% of the time. This shows that the system can make mistakes while users are still able to get their answers by rephrasing their questions.

The main problems are:

- Some errors occur in pattern matching when minor deviations in the input take place. These occur mainly while matching the names of performances or names of theatres. As stated above, in the near future we will apply the Levenshtein algorithm to calculate the distance between strings. This way we can guess which word (from a dictionary or database) is meant when an unknown word is encountered. We also plan to use the spell checking functions of the morphological analyzer to handle typing errors.
- The users quickly discover the limits of the system's knowledge – there is no information about ticket prices, no booking. So the user is unable to retrieve this information from the system. We plan to expand the knowledge base and include ticket prices and booking information in the near future.

4. Implementation

The dialogue module is a web enabled system. Therefore, it is easy for the developer to add, modify and deploy new functionality. The web enabled system also

provides an easy way to collect chat logs that can be used as training data to manually improve the performance of the linguistic model.

The dialogue module was developed in PHP using MySQL as the database server and Apache HTTP Server as the web server.

The speech recognition and speech synthesis modules are standalone modules developed by collaborating researchers at the Tallinn University of Technology.

5. Conclusions

The dialogue system developed in this project demonstrates that we are capable of designing systems which can understand a small subset of natural language in a constrained linguistic domain. The conducted experiments show that our methods are satisfactory yet need some further improvements.

In the future, we hope to release a dialog based speech-understanding system that could be used over a telephone.

6. Acknowledgements

The project has support from the Estonian Ministry of Culture, the Estonian Information Technology Foundation, the Estonian Science Foundation (grant no 5685), and Elion Enterprises Ltd.

7. References

Gerassimenko, O., Hennoste, T., Koit, M., Rääbis, A., Strandson, K., Valdisoo, M., Vutt, E. (2004). Annotated dialogue corpus as a language resource: an experience of building the Estonian dialogue corpus. In: The first Baltic conference “Human language technologies. The Baltic perspective”. Commission of the official language at the chancellery of the president of Latvia, Riga 150–155

Kaalep, H.-J. (1997). An Estonian Morphological Analyser and the Impact of a Corpus on Its Development. Computers and the Humanities 31: 115-133

Meister, E., Lasn, J., Meister, L. (2003). SpeechDat-like Estonian database. - In: Text, Speech and Dialogue : 6th International Conference, TSD 2003, Czech Republic, September 8-12, 2003 / Eds. Matoušek [et al.]. Berlin [etc.] : Springer, Lecture Notes in Artificial Intelligence, Vol. 2807. 412-417

Mihkla M., Eek A., Meister E. (1999). Text-to-Speech Synthesis of Estonian. – Proceedings of the 6th European Conference on Speech Communication and Technology, Budapest, Vol. 5 2095-2098

Automatic Assessment of Children's Speech with Cleft Lip and Palate

Andreas Maier[‡*], Elmar Nöth[*], Emeka Nkenke[†], Maria Schuster[‡]

[*] Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg
Martensstraße 3, 91058 Erlangen, Germany
Andreas.Maier@cs.fau.de

[†]Mund-, Kiefer- und Gesichtschirurgische Klinik, Universität Erlangen-Nürnberg,
Glückstraße 11, 91054 Erlangen, Germany

[‡]Abteilung für Phoniatrie und Pädaudiologie, Universität Erlangen-Nürnberg
Bohlenplatz 21, 91054 Erlangen, Germany

Abstract

Cleft lip and palate (CLP) may cause functional limitations even after adequate surgical and non-surgical treatment, speech disorder being one of them. Until now, an automatic, objective means to determine and quantify the intelligibility did not exist. We have created an automatic evaluation system that assesses speech, based on the result of an automatic speech recognizer. It was applied to 35 recordings of children with CLP. A subjective evaluation of the intelligibility was performed by two experts and confronted to the automatic speech evaluation. It complied with experts' rating of intelligibility. Furthermore we present the results obtained on a control group of 45 recordings of normal children and compare these results with those of the CLP children.

Samodejna ocena govora otrok z zajčjo ustnico in volčjim žrelom

Zajčja ustnica in volčje žrelo lahko povzročata funkcijske omejitve tudi po ustreznem operativnem ali neoperativnem zdravljenju, med njimi so tudi motnje govora. Do sedaj ni obstajal samodejni objektivni način ugotavljanja razumljivosti. Razvili smo sistem za samodejno vrednotenje, ki ocenjuje govor na podlagi rezultatov samodejnega razpoznavalnika govora. Uporabljen je bil pri 35 posnetkih otrok z zajčjo ustnico in volčjim žrelom. Subjektivno vrednotenje razumljivosti, ki sta ga opravila dva strokovnjaka, je bilo soočeno s samodejnim vrednotenjem govora. Slednje se je ujemalo z oceno razumljivosti strokovnjakov. Poleg tega predstavljamo rezultate, pridobljene pri kontrolni skupini s 45 posnetki govora otrok brez motenj govora, in jih primerjamo z rezultati posnetkov govora otrok z zajčjo ustnico in volčjim žrelom.

1. Introduction

Cleft lip and palate (CLP) is the most common malformation of the head. It can result in morphological and functional disorders (Wantia and Rettinger, 2002), whereat one has to differentiate primary from secondary disorders (Millard and Richman, 2001; Rosanowski and Eysholdt, 2002). Primary disorders include e.g. swallowing, breathing and mimic disorders. Speech and voice disorders (Schönweiler and Schönweiler, 1994) as well as conductive hearing loss that affect speech development (Schönweiler et al., 1999), are secondary disorders. Speech disorders can still be present after reconstructive surgical treatment. The characteristics of speech disorders are mainly a combination of different articulatory features, e.g. enhanced nasal air emissions that lead to altered nasality, a shift in localization of articulation (e.g. using a /d/ built with the tip of the tongue instead of a /g/ built with back of the tongue or vice versa), and a modified articulatory tension (e.g. weakening of the plosives /t/, /k/, /p/) (Harding and Grunwell, 1998). They affect not only the intelligibility but therewith the social competence and emotional development of a child. In clinical practice, articulation disorders are mainly evaluated by subjective tools. The simplest method is the auditive perception, mostly performed by a speech therapist. Previous studies have shown that experience is an important factor that influences the subjective estimation of speech disorders leading to inaccurate evaluation by persons with only few years of experience (Paal et al., 2005). Until now, objective means exist only for quantitative measurements of nasal emissions (Küttner et al., 2003; Lierde et al., 2002; Hogen Esch and Dejonckere, 2004) and for the detection of secondary voice disorders (Bressmann et al., 1998). But other specific or non-specific articulation disorders in CLP as well as a global assessment of speech quality cannot be sufficiently quantified. In this paper, we present a new technical procedure for the measurement and evaluation of speech disorders and compare the results obtained with subjective ratings of a panel of expert listeners.

2. Automatic Speech Recognition System

For the objective measurement of the intelligibility of children with speech disorders, an automatic speech recognition system was applied, a state-of-the-art word recognition system developed at the Chair for Pattern Recognition (Lehrstuhl für Mustererkennung) of the University of Erlangen. In this study, the latest version as described in detail in (Stemmer, 2005) was used. The recognizer can handle spontaneous speech with mid-sized vocabularies of up to 10,000 words. As features we use Mel-Frequency Cepstrum Coefficients (MFCC) 1 to 11 plus the energy of the signal. Additionally 12 delta coefficients are computed over a context of 2 time frames to the left and the right side (56 ms in total). The recognition is performed with semi-continuous Hidden Markov Models (SCHMMs). The codebook contains 500 full covariance Gaussian densities which

are shared by all HMM states. The elementary recognition units are polyphones (Schukat–Talamazzini and Niemann, 1991). The polyphones were constructed for each sequence of phones which appeared more than 50 times in the training set.

We used two types of unigram language models according to the application scenario. This helps to enhance recognition results by including linguistic information. However, for our purpose it was necessary to put more weight on the recognition of acoustic features. In the first scenario the transliteration is assumed to be unknown. So we created a basic language model which was trained with just the reference words of the test (see below) since no further information was available. This model has a perplexity of 43 on the reference text. In the second scenario the transliteration is available. This means that the data have to be transliterated completely and that additional words can appear which were not in the set of the reference words. These words are added to the language model in order to enable the their recognition. However, the probability of the target words is increased by a factor of 2. The test set perplexity of the language model differs for each speaker since the language model is constructed individually if the transliteration is known.

The speech recognition system had been trained with acoustic information from spontaneous dialogues of the VERBMOBIL project (Wahlster, 2000) and normal children's speech. The speech data of non-pathologic children voices (30 female and 23 male) were recorded at two local schools (age 10 to 14) in Erlangen and consisted of read texts. The training population of the VERBMOBIL project consisted of normal adult speakers from all over Germany and thus covered all dialectal regions. All speakers were asked to speak "standard" German. 90 % of the training population (47 female and 85 male) were younger than 40 years. During training an evaluation set was used that only contained children's speech. The adults' data was adapted by vocal tract length normalization as proposed in (Stemmer et al., 2003).

MLLR adaptation (Gales et al., 1996) with the patients' data lead to further improvement of the speech recognition system.

3. Data

All children were asked to name pictures that were shown according to the PLAKSS test (Fox, 2002). This German test consists of 99 words shown as pictograms on 33 slides. With this test, the speech of children can be evaluated even if they are quite young since they do not need the ability to read. However, the children could take advantage of being able to read since the reference words were shown as subtitles. The test includes all possible phonemes of the German language in different positions (beginning, center and end of a word).

The patients' group consisted of 35 children and adolescents (13 girls and 22 boys) with CLP at the age from 3.3 to 18.5 years (mean 8.3 $\pm$ 3.6 years). The examination was included in the regular out-patient examination of all children and adolescents with CLP. These speech samples were recorded with a close-talking microphone (dnt Call 4U Comfort headset) at a sampling frequency of 16 kHz and quantized with 16 bit. For these data no further post-processing was done.

Furthermore a control group with 45 normal children was recorded at a local elementary school. In total, data from 27 girls and 18 boys were collected. The children were in the age from 7.4 to 10.7 (mean 9.5 $\pm$ 0.9 years). The data were collected at 48 kHz with 16 bit quantization. To match the patients' data a resampling to 16 kHz was done. For the control group a Sennheiser close-talking microphone (handgrip K3U with ME 80 head) was used. These data were post-processed: In some cases the voice of the instructor was audible on the sound track. So the instructor's voice was removed in all occasions. Furthermore all of the children's speech data was transliterated.

Informed consent had been obtained by all parents of the children prior to the recording. All children were native German speakers, some using a local dialect.

4. Subjective Evaluation

Two voice professionals subjectively estimated the intelligibility of the children's speech while listening to a play-back of the recordings. A five point Likert scale (1 = very high, 2 = rather high, 3 = medium, 4 = rather low, 5 = very low) was applied to rate the intelligibility of all individual turns. In this manner an averaged mark – expressed as a floating point value – for each patient could be calculated.

5. Analysis and Automatic Evaluation

For the agreement computations between different raters on the one hand and raters/recognizer on the other hand we use the Pearson product-moment correlation coefficient (Pearson, 1896). It allows to compare two number series which are of different scale and margin like in the given case. So the ratings of the human experts and those of the speech recognition system can be compared directly without having to define a mapping between word accuracies and Likert scores. In order to compare both raters to the recognition system the average rating of the experts was computed for each speaker. For the recognition rate of the speech recognition system we investigated the word accuracy (WA) like in (Haderlein et al., 2004), (Schuster et al., 2005) or (Maier et al., 2006; Schuster et al., 2006) and the word recognition rate (WR). The WA is defined as

$$WA = \frac{C - I}{R} \cdot 100\,\%$$

where C is the number of correctly recognized words, I the number of wrongly inserted words and R the number of words in the reference text. The WR is defined as follows:

$$WR = \frac{C}{R} \cdot 100\,\%$$

Both measurements need a reference text in order to determine the number of correctly recognized words. However, since the reference are pictures, the text is not known a priori. One solution to this problem is to transliterate all the data like it was done before. Since we developed a new recording and evaluation software we now know the exact time when the reference slide was moved to the next slide.

measurement	recognized word chain	reference	%
transliteration WA	This is moon, bucket and a a ball	This is a moon, a bucket, and a tree	55.5
transliteration WR	This is moon, bucket and a a ball	This is a moon, a bucket, and a tree	66.6
automatic WA	tiger moon bucket apple ball	moon bucket tree	0
automatic WR	tiger moon bucket apple ball	moon bucket tree	66.6

Table 1: Example of the effects of the automatic reference on the WA and WR. We assume that the spoken utterance is "This is a moon, a bucket, and a tree". Thus, the automatic reference is "moon bucket tree"

measurement	transliteration WA	transliteration WR
automatic WA	0.40	0.21
automatic WR	0.60	0.60

Table 2: Correlation between the different measurements regarding the control group. The automatic WR yields the results with the best correlation to the transliteration-based measurements

rater	M	S	mean
automatic WA	-0.83	-0.77	-0.82
automatic WR	-0.88	-0.85	-0.89

Table 3: Correlation between the different raters and the automatic measurements

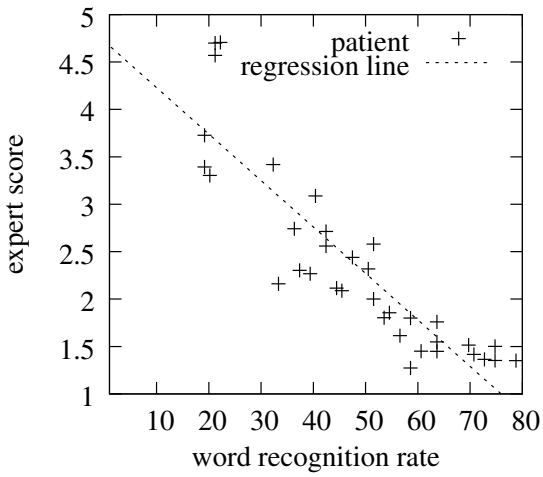


Figure 1: Word recognition rates in comparison to the scores of the human experts for the patient group ($r = -0.89$)

We can exploit this information to approximate a reference word chain. This reference word chain contains just the words which are shown on the slide. Unfortunately this is not sufficient to calculate a good word accuracy since most of the children use carrier sentences like "This is a ..." which are regarded as wrongly inserted words even if the recognition would be perfect. In order to avoid this problem we applied the word recognition rate instead since it does not weight the effect of inserted words. The difference between these methods is shown in Table 1.

6. Results

Since the control group was completely transliterated and recorded with our new software we could investigate the difference between the automatic measurements and those based on the transliteration. As can be seen in Table 2 the word recognition rate correlates to both transliteration-based measurements. The automatic word accuracy, however, matches poorly with the transliteration-based measurements (cf. Table 1). Therefore we expected the WR to show a good agreement with the results presented in (Maier et al., 2006).

The recordings of the CLP children showed a wide range of intelligibility (see Figure 1). Subjective speech evaluation showed good consistency. The correlation coefficient for the raters was 0.91. The results for the correlations of the WA, the WR and the subjective speech evaluation are shown in Table 3. When compared to the average of the raters, the WA for the recognizer has a correlation of -0.82 while the WR even correlates with -0.89. The coefficients are negative because high recognition rates come from "good" speech with a low score number and vice versa (note the regression line in Figure 1).

Figure 2 shows the word recognition rates of children in the same age range of both groups. As can be seen, almost all 45 children of the control group have high recognition rates. The distribution of the patients' group shows a high variance. This is due to the fact that the patients' group contained a wide range of intelligibility. Some of the patients were as intelligible as normal children (cf. Figure 1). The correlation between the age and the word recognition rate is 0.2 for the 45 children of the control group and 0.3 for the 20 children of the patient group. So there is just a weak connection between the age and the intelligibility.

7. Discussion

First results for an automatic global evaluation of speech disorders of different manifestations as found in CLP speech are shown. The speech recognition system shows high consistency with the experts' estimation of the intelligibility.The use of prior information about the speech test and its setup allows us the create a fully automated procedure to create a global assessment of the speaker's intelligibility. In difference to (Maier et al., 2006) no manual post-processing was done. Still the experts' and the recognizer's evaluation show a high correlation.

Using a control group we could show that our measure is sufficient to differentiate normal children's speech from pathologic speech. Furthermore we could show the consistency of our new measure to the transliteration-based evaluation methods.

The technique allows an objective evaluation of speech disorders and therapy effects. It avoids subjective influences from human raters with different experience and is therefore of high clinical and scientific value. Automatic

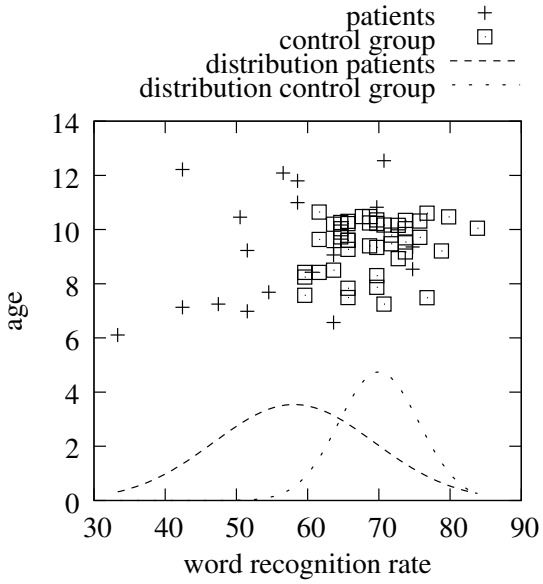


Figure 2: Distribution of the patients and the control group over the word recognition rate. Only members with about the same age were considered.

evaluation in real-time will avoid long evaluation proceedings by human experts. Further research will lead to the classification and quantification of different speech disorders. This will allow to quantify the impact of individual speech disorders on the intelligibility and will improve therapy strategies for speech disorders.

8. Conclusion

Automatic speech evaluation by a speech recognizer is a valuable means for research and clinical purpose in order to determine the global speech outcome of children with CLP. It enables to quantify the quality of speech. Adaptation of the technique presented here will lead to further applications to differentiate and quantify articulation disorders. Modern technical solutions might easily provide specialized centers and therapists with this new evaluation method.

9. Acknowledgments

This work was supported by the Johannes and Frieda Marohn foundation at the Friedrich-Alexander University of Erlangen-Nuremberg. Only the authors are responsible for the content of this article.

10. References

T. Bressmann, R. Sader, M. Merk, W. Ziegler, R. Busch, H.F. Zeilhofer, and H.H. Horch. 1998. Perzeptive und apparative Untersuchung der Stimmqualität bei Patienten mit Lippen-Kiefer-Gaumenspalten. *Laryngorhinootologie*, 77(12):700–708.

A. V. Fox. 2002. PLAKSS - Psycholinguistische Analyse kindlicher Sprechstörungen. Swets & Zeitlinger, Frankfurt a.M.

M. Gales, D. Pye, and P. Woodland. 1996. Variance compensation within the MLLR framework for robust speech recognition and speaker adaptation. In *Proc. ICSLP '96*, volume 3, pages 1832–1835, Philadelphia, USA.

T. Haderlein, S. Steidl, E. Nöth, F. Rosanowski, and M. Schuster. 2004. Automatic recognition and evaluation of tracheoesophageal speech. In P. Sojka, I. Kopecek, and K. Pala, editors, *Text, Speech and Dialogue, 7th International Conference, September 8-11, 2004, Brno, Czech Republic, Proceedings*, volume 3206 of *Lecture Notes in Artificial Intelligence*, pages 331–338, Berlin, Heidelberg. Springer.

A. Harding and P. Grunwell. 1998. Active versus passive cleft-type speech characteristics. *Int J Lang Commun Disord*, 33(3):329–52.

T.T. Hogen Esch and P.H. Dejonckere. 2004. Objectivating nasality in healthy and velopharyngeal insufficient children with the Nasalance Acquisition System (NasalView) Defining minimal required speech tasks assessing normative values for Dutch language. *Int J Pediatr Otorhinolaryngol*, 68(8):1039–46.

C. Küttner, R. Schönweiler, B. Seeberger, R. Dempf, J. Lisson, and M. Ptok. 2003. Objektive Messung der Nasalanz in der deutschen Hochlautung. *HNO*, 51:151–156.

K. Van Lierde, M. De Bodt, J. Van Borsel, F. Wuyts, and P. Van Cauwenberge. 2002. Effect of cleft type on overall speech intelligibility and resonance. *Folia Phoniatr Logop*, 54(3):158–168.

A. Maier, C. Hacker, E. Nöth, E. Nkenke, T. Haderlein, F. Rosanowski, and M. Schuster. 2006. Intelligibility of children with cleft lip and palate: Evaluation by speech recognition techniques. In *Proc. International Conf. on Pattern Recognition*, volume 4, pages 274–277, Hong Kong, China.

T. Millard and L.C. Richman. 2001. Different cleft conditions, facial appearance, and speech: relationship to psychological variables. *Cleft Palate Craniofac J*, 38:68–75.

S. Paal, U. Reulbach, K. Strobel-Schwarthoff, E. Nkenke, and M. Schuster. 2005. Beurteilung von Sprechauffälligkeiten bei Kindern mit Lippen-Kiefer-Gaumen-Spaltbildungen. *J Orofac Orthop*, 66(4):270–278.

K. Pearson. 1896. Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity and Panmixia. *Philosophical Transactions of the Royal Society of London*, 187:253–318.

F. Rosanowski and U. Eysholdt. 2002. Phoniatric aspects in cleft lip patients. *Facial Plast Surg*, 18(3):197–203.

R. Schönweiler and B. Schönweiler. 1994. Hörvermögen und Sprachleistungen bei 417 Kindern mit Spaltfehlbildungen. *HNO*, 42(11):691–696.

R. Schönweiler, J.A. Lisson, B. Schönweiler, A. Eckardt, M. Ptok, J. Trankmann, and J.E. Hausamen. 1999. A retrospective study of hearing, speech and language function in children with clefts following palatoplasty and veloplasty procedures at 18-24 months of age. *Int J Pediatr Otorhinolaryngol*, 50(3):205–217.

E. G. Schukat–Talamazzini and H. Niemann. 1991. Das ISADORA-System – ein akustisch–phonetisches Netzwerk zur automatischen Spracherkennung. In B. Radig, editor, *Mustererkennung 1991*, volume 290 of *Informatik Fachberichte*, pages 251–258, Berlin. Springer–Verlag.

M. Schuster, E. Nöth, T. Haderlein, S. Steidl, A. Batliner,

and F. Rosanowski. 2005. Can you Understand him? Let's Look at his Word Accuracy — Automatic Evaluation of Tracheoesophageal Speech. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, USA. IEEE Computer Society Press.

M. Schuster, A. Maier, T. Haderlein, E. Nkenke, U. Wohlleben, F. Rosanowski, U. Eysholdt, and E. Nöth. 2006. Evaluation of Speech Intelligibility for Children with Cleft Lip and Palate by Automatic Speech Recognition. *Int J Pediatr Otorhinolaryngol*, 70:1741–1747.

G. Stemmer, C. Hacker, S. Steidl, and E. Nöth. 2003. Acoustic Normalization of Children's Speech. In *Proc. European Conf. on Speech Communication and Technology*, volume 2, pages 1313–1316, Geneva, Switzerland.

G. Stemmer. 2005. *Modeling Variability in Speech Recognition*. Logos Verlag, Berlin.

W. Wahlster. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, New York, Berlin.

N. Wantia and G. Rettinger. 2002. The current understanding of cleft lip malformations. *Facial Plast Surg*, 18(3):147–53.

Robust heteroscedastic linear discriminant analysis and LCRC posterior features in large vocabulary continuous speech recognition

Martin Karafiát, František Grézl, Petr Schwarz, Lukáš Burget, and Jan Černocký

Speech@FIT group, Faculty of Information Technology, Brno University of Technology
{karafiat,grezl,schwarzp,burget,cernocky}@fit.vutbr.cz

Abstract

This paper deals with feature extraction in speech recognition. Three robust variants of popular HLDA transform are investigated. Influence of adding posterior features to PLP feature stream is studied. The experimental results are obtained on CTS (continuous telephone speech) data. Silence-reduced HLDA and LCRC phoneme-state posterior features together provide more than 4% absolute improvement in word error rate.

Robustna heteroroskedastična linearna diskriminantna analiza (HLDA) in LCRC posteriorne značilke pri razpoznavanju tekočega govora z velikim besednjakom

Prispevek se ukvarja z izločanjem značilk v razpoznavanju govora. Raziskane so tri robustne različice priljubljene transformacije HLDA. Obravnavan je vpliv dodajanja posteriornih znailk zaporedju značilk PLP. Experimentalni rezultati so dobljeni na podlagi podatkov zveznega telefonskega govora. HLDA in LCRC posteriorne znailke stanja fonema skupaj prinašata več kot 4% absolutno izboljšanje pri stopnji zanesljivosti razpoznavanja besed.

1. Introduction

Speech feature extraction is important part of every large vocabulary continuous speech recognition system (LVCSR). Performance gains obtained thanks to this block are quite welcome as (on contrary to adding data or changing training or decoding algorithms), feature extraction is considered as "cheap" part of speech recognition system.

One of key problems in feature extraction is to reduce the dimensionality of feature vectors while preserving the discriminative power of features. Linear transforms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are mostly used for this task. In recent years, Heteroscedastic Linear Discriminant Analysis (HLDA) has gained popularity in the research community (Kumar, 1997; Burget, 2004) for its relaxed constraints on statistical properties of classes (unlike LDA, HLDA does not assume the same covariance matrix for all classes). To compute HLDA transformation matrix, however, more statistics need to be estimated and the reliability of such estimations becomes an issue. Section 2. discusses robust variants of HLDA.

Second part of the paper is devoted to the use of posterior-features. Posteriors generated by neural networks (NN) and converted into features are also increasingly popular in small (Adami et al., 2002) and large (Zhu et al., 2005) recognition systems for their complementarity with classical PLP or MFCC coefficients. Section 3. introduces phoneme-state posterior estimator based on split temporal context (Schwarz et al., 2004; Schwarz et al., 2006) that has already proved its quality in different tasks ranging from language identification to keyword spotting.

2. HLDA

HLDA allows to derive such projection that best decorrelates features associated with each particular class (maximum likelihood linear transformation for diagonal covariance modeling (Kumar, 1997)). To perform decorrelation and dimensionality reduction, n-dimensional feature vectors are projected into first $p < n$ rows, $\mathbf{a}_{k=1\ldots p}$, of $n \times n$ HLDA transformation matrix, $\mathbf{A}$. An efficient iterative algorithm (Gales., 1999; Burget, 2004) is used in our experiments to estimate matrix $\mathbf{A}$, where individual rows are periodically re-estimated using the following formula:

$$\hat{\mathbf{a}}_k = \mathbf{c}_k \mathbf{G}^{(k)-1} \sqrt{\frac{T}{\mathbf{c}_k \mathbf{G}^{(k)-1} \mathbf{c}_k^T}} \tag{1}$$

where $\mathbf{c}_i$ is the i^{th} row vector of co-factor matrix $C = |\mathbf{A}|\mathbf{A}^{-1}$ for current estimate of $\mathbf{A}$ and

$$\mathbf{G}^{(k)} = \begin{cases} \sum_{j=1}^{J} \frac{\gamma_j}{\mathbf{a}_k \hat{\mathbf{\Sigma}}_j \mathbf{a}_k^T} \hat{\mathbf{\Sigma}}_j & k \leq p \\ \frac{T}{\mathbf{a}_k \hat{\mathbf{\Sigma}} \mathbf{a}_k^T} \hat{\mathbf{\Sigma}} & k > p \end{cases} \tag{2}$$

where $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Sigma}}^j$ are estimates of the global covariance matrix and covariance matrix of j^{th} class, γ_j is number of training feature vectors belonging to j^{th} class and T is the total number of training feature vectors. In our experiments, the classes are defined by each Gaussian mixture component and γ_j are their occupation counts.

Well known Linear Discriminant Analysis (LDA) can be seen as special case of HLDA, where it is assumed that covariance matrices of all classes are the same. In contrast to HLDA, closed form solution exists in this case. Basis of LDA transformation are given by eigen-vectors of matrix $\mathbf{\Sigma}_{AC} \times \mathbf{\Sigma}_{WC}^{-1}$, where $\mathbf{\Sigma}_{WC}$ is within-class covariance matrix and $\mathbf{\Sigma}_{AC}$ is across-class covariance matrix.

2.1. SHLDA

HLDA requires the covariance matrix to be estimated for each class. The higher number of classes is used, the

fewer feature vector examples are available for each class — class covariance matrix estimates become more noisy. We have recently proposed (Burget, 2004) a technique based on combination of HLDA and LDA, where class covariance matrices are estimated more robustly, and at the same time, (at least the major) differences between covariance matrices of different classes are preserved. Smoothed HLDA (SHLDA) differs from HLDA only in the way of class covariance matrices estimation. In the case of SHLDA, estimate of class covariance matrices is given by:

$$\check{\mathbf{\Sigma}}_j = \alpha\hat{\mathbf{\Sigma}}_j + (1-\alpha)\mathbf{\Sigma}_{WC} \tag{3}$$

where $\check{\mathbf{\Sigma}}_j$ is "smoothed" estimate of covariance matrix for class j. $\hat{\mathbf{\Sigma}}_j$ is estimate of covariance matrix, $\mathbf{\Sigma}_{WC}$ is estimate of within-class covariance matrix and α is smoothing factor — a value in the range of 0 to 1. Note that for α equal to 0, SHLDA becomes LDA and for α equal to 1, SHLDA becomes HLDA.

2.2. MAP-SHLDA

SHLDA gives more robust estimation than standard HLDA but optimal smoothing factor α depends on the amount of data for each class. In extreme case, α should be set to 0 (HLDA) if infinite amount of training data is available. With decreasing amount of data, optimal α value will slide up to LDA direction.

To add more robustness into the smoothing procedure, we implemented maximum a posteriori (MAP) smoothing (Gauvain and Lee, 1994), where within-class covariance matrix $\mathbf{\Sigma}_{WC}$ is considered as the prior. Estimate of the class covariance matrix is then given by:

$$\check{\mathbf{\Sigma}}_j = \mathbf{\Sigma}_{WC}\frac{\tau}{\gamma_j+\tau} + \hat{\mathbf{\Sigma}}_j\frac{\gamma_j}{\gamma_j+\tau} \tag{4}$$

where τ is a control constant. Obviously, if insufficient data is available for current class, the prior $\mathbf{\Sigma}_{WC}$ is considered more reliable than the class estimation $\hat{\mathbf{\Sigma}}_j$. In case of infinite data, only the class estimation of covariance matrix $\hat{\mathbf{\Sigma}}_j$ is used for further processing.

2.3. Silence Reduction in HLDA

From the point of view of transformation estimation, silence is a "bad" class as its distributions differ significantly from all speech classes. Moreover, training data (even if end-pointed) contains significant proportion of silence. Therefore, we have experimented with limiting the influence of silence.

Rather than discarding the silence frames, the occupation counts γ_j of silence classes, which take part in computation of global covariance matrix $\hat{\mathbf{\Sigma}}$, and in Equation 2 are scaled by silence reduction factor $1/SR$. Setting $SR = \infty$ corresponds to complete elimination of silence statistics.

3. Posterior features

Several works have shown that using posterior-features generated by NNs is advantageous for speech recognition (Adami et al., 2002; Zhu et al., 2005). We have experimented with two setups to generate posteriors. The first one is based on a simple estimation of phoneme posterior probabilities from a block of 9 consecutive PLP-feature vectors (FeatureNet).

The second one uses our state-of-the-art phoneme-state posterior estimator based on modeling long temporal context (Schwarz et al., 2006). Details of the posterior estimator are shown in Fig. 1. Mel filter bank log energies are obtained in conventional way. Based on our previous work in phoneme recognition (Schwarz et al., 2004), the context of 31 frames (310 ms) around the current frame is taken. This context is split into 2 halves: Left and Right Contexts (hence the name "LCRC"). This allows for more precise modeling of the whole trajectory while limiting the size of the model (number of weights in the NN) and reducing the amount of necessary training data. For both parts, temporal evolutions of critical band log energies are processed by discrete cosine transform to de-correlate and reduce dimensionality. Two NNs are trained to produce phoneme-state posterior probabilities for both context parts. We use 3 states per phoneme which follows similar idea as states in phoneme HMM. Third NN functions as a merger and produces final set of phoneme-state posterior probabilities[1]

For both approaches, the resulting posteriors are processed by log and by a linear transform to de-correlate and reduce dimensionality (details are given in the experimental section below).

4. Experiments

Our recognition system was trained on ctstrain04 training set, a subset of the h5train03 set, defined at the Cambridge University as training set for Conversation Telephone Speech (CTS) recognition systems (Hain et al., 2005). It contains about 278 hours of well transcribed speech data from Switchboard I, II and Call Home English. All systems were tested on the Hub5 Eval01 test set composed of 3 subsets of 20 conversations from Switchboard I, II, and Switchboard-cellular, for a total length of about 6 hours of audio data.

The baseline features are 13th order PLP cepstral coefficients, including 0th one, with first and second derivatives added. This gives a standard 39 dimension feature vector. Cepstral mean and variance normalization was applied. Baseline cross-word triphone HMM models were trained by Baum-Welch re-estimation and mixture splitting. We used a standard 3-state left-to-right phoneme setup, with 16 Gaussian mixture components per state. 7598 tied states were obtained by decision tree clustering. Each Gaussian mixture was taken as a different class for HLDA experiment. Therefore, we had N = 16×7598 = 121568 classes.

The trigram language model used in decoding was estimated at University of Sheffield by interpolation from Switchboard I, II, Call Home English and Hub4 (Broadcast news) transcriptions. The size of recognition vocabulary was 50k words.

The recognition output was generated in two passes: At first, lattice generation with baseline HMMs and bigram language model was performed. The lattices were

[1] Neural nets are trained using QuickNet from ICSI and SNet – a parallel NN training software being developed in Speech@FIT

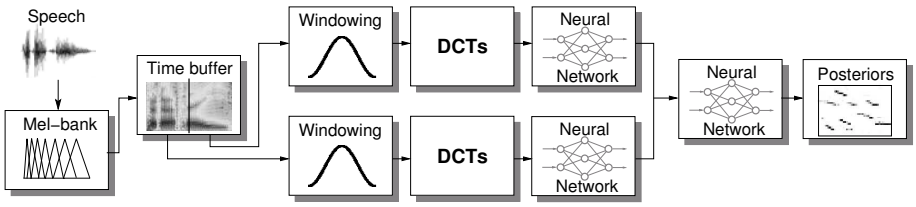


Figure 1: Phoneme-state posterior estimator based on split left and right contexts.

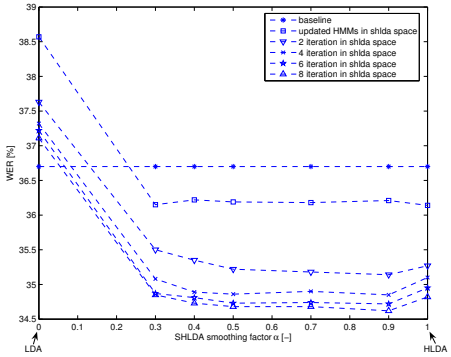


Figure 2: Dependency of WER on the SHLDA smoothing factors.

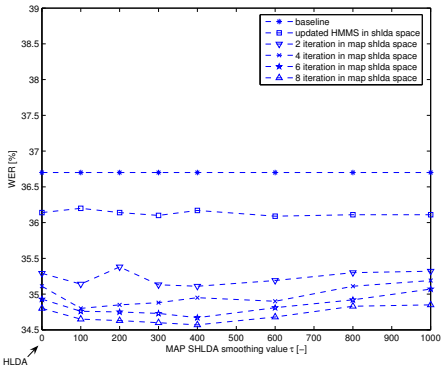


Figure 3: Dependency of WER on the MAP-SHLDA (right) smoothing factors.

expanded by more accurate trigram language model. The pruning process was applied to reduce them to reasonable size. In the second pass, lattices were re-scored with tested features and models.

4.1. Flavors of HLDA

We added the third derivatives into the feature stream, which gave us 52 dimensional feature vectors. **SHLDA** transform was then trained to perform the projection from 52 to 39 dimension. Smoothing factors α in Eq. 3 of 0.0 (LDA), 0.3, 0.4, 0.5, 0.7, 0.9, 1.0 (HLDA) were tested. Figure 2 shows dependency of WER on SHLDA smoothing factor α. Pure LDA failed, probably due to bad assumption of the same Gaussian distribution in all classes. The best system performance (Table 1) was obtained for smoothing factor 0.9. The relative improvement of this system is 7.9% compared to the baseline and 0.6% compared to the clean HLDA setup.

MAP-SHLDA test setup was built in same way as SHLDA system, only the smoothing procedure (Equation 3) was replaced by MAP approach (Equation 4). The average value of all class occupation counts was 820. Therefore $\tau = 820$ in MAP-SHLDA should have the same behavior as $\alpha = 0.5$ in SHLDA if all classes had the same number of observations. The optimal smoothing values for SHLDA were in range 0.5—0.9 (Figure 2). Therefore, we decided to test smoothing control constant τ on values 0 (HLDA), 100, 200, 300, 400, 600, 800 and 1000. The results are shown in Figure 3. The best system performance (Table 1) was obtained for $\tau = 400$. The relative improvement of this system is 8% compared to the baseline and 0.7% compared to the clean HLDA setup.

Silence reduction in HLDA (SR-HLDA) was tested with factors SR equal to 1 (no reduction), 2, 10, 100 and ∞ (removing all silence classes). For $SR = 1$, the WER is obviously 34.8%, for $SR = 2$ it drops to 34.6% and from $SR = 10 \ldots \infty$ it is constant: 34.5%.

System	WER [%]
Baseline (no HLDA)	36.7
HLDA	34.8
SHLDA	34.6
MAP-SHLDA	34.6
SR-HLDA	34.5

Table 1: Comparison of HLDA systems.

4.2. Posterior features

Posterior features were always used together with base PLP features. Table 2 summarizes the results.

Upper part of Figure 4 shows the way the two feature streams were combined in FeatureNet experiments. The upper branch corresponds to the previous section. To compute posterior features, 9 frames of PLP+Δ+$\Delta\Delta$ were stacked and processed by a neural net with 1262 neurons in the hidden layer (this number was chosen to have approximately 500k weights in the NN). There are 45 phoneme classes, which determines the size of the output layer. Log-posteriors are processed by KLT or HLDA and then concatenated with PLP+HLDA features to form the final 64-dimensional feature vectors.

Lower panel of Figure 4 presents the setup with LCRC-posterior features. The PLPs were derived directly with Δ, $\Delta\Delta$ and $\Delta\Delta\Delta$, and down-scaled by HLDA to 39 dimensions. The detail of LCRC-posterior feature derivation is in Fig. 1, all nets had 1500 neurons in the hidden layer. For

System	WER [%]
PLP SR-HLDA	34.5
PLP SR-HLDA + PLP-posteriors KLT	33.8
PLP SR-HLDA + PLP-posteriors HLDA	33.3
PLP SR-HLDA + LCRC-posteriors HLDA	32.6

Table 2: Performance of posterior features in the CTS system.

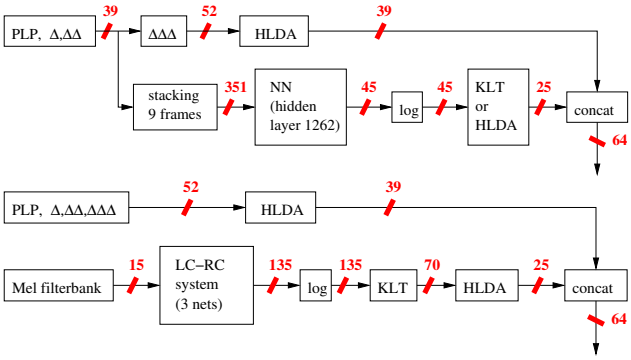


Figure 4: Configuration of the system with PLP- (upper panel) LCRC-posteriors (lower panel).

each frame, the output of LCRC system are estimates of 135 phoneme-state[2] posterior probabilities. As the number of phoneme-state posteriors is too high to fit the statistics necessary for HLDA estimation into the memory, the output dimensionality of LCRC system is first reduced by KLT from 135 to 70. The following HLDA reduces this size to 25, and the results are concatenated with PLP+HLDA features to form again 64-dimensional feature vectors.

We see, that the posterior features improve the results by almost 1% absolutely, and that there is clear preference of HLDA to KLT. With the new LCRC features, we have confirmed good results they provide in phoneme recognition (Schwarz et al., 2006) — with these features, the results are almost 2% better than the PLP SR-HLDA baseline.

5. Conclusion

In this paper, we have investigated robust variants of HLDA and use of classical and novel posterior features in speech recognition.

In the HLDA part, 2 approaches of HLDA smoothing were tested: Smoothed HLDA (SHLDA) and MAP variant of SHLDA, taking into account the amounts of data available for estimation of statistics for different classes. Both perform better than the basic HLDA. We have however found, that removing the silence class from the HLDA estimations (Silence-reduced HLDA) is equally effective and cheaper in computation. Testing SHLDA and MAP-SHLDA on the top of SR-HLDA did not bring any further improvement, therefore we stick with SR-HLDA as the most suitable transformation in our LVCSR experiments.

Two kinds of posterior features were tested – "classical" FeatureNet approach with stacked 9 frames of PLPs and novel approach using more elaborate structure to phoneme-state posterior modeling. The later scheme provided significant reduction of word error rate.

Our current work focuses on using the described feature extraction schemes in meeting data recognition along with speaker adaptative training scheme based on constrained maximum likelihood linear regression (CMLLR) and discriminative training using Minimum Phoneme Error (MPE) criterion. First results indicate that the improvement obtained by SHLDA and posterior features carries on through both adaptation and discriminative training steps.

[2]see (Schwarz et al., 2006) for details on splitting each of phonemes to 3 phoneme-states

6. Acknowledgments

This work was partially supported by EC project Augmented Multi-party Interaction (AMI), No. 506811 and Grant Agency of Czech Republic under project No. 102/05/0278. Lukáš Burget was supported by post-doctoral grant of Grant Agency of Czech Republic No. 102/06/P383. Thanks University of Sheffield for generating LVCSR lattices. We further thank Cambridge University Engineering Department making the h5train03 CTS training set available for granting the right to use Gunnar Evermann's HDecode to the University of Sheffield.

7. References

A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. 2002. Qualcomm-ICSI-OGI features for ASR. In *Proc. ICSLP 2002*, Denver, Colorado, USA.

L. Burget. 2004. Combination of speech features using smoothed heteroscedastic linear discriminant analysis. In *8th International Conference on Spoken Language Processing*, Jeju island, KR, oct.

M.J.F. Gales. 1999. Semi-tied covariance matrices for hidden markov models. *IEEE Trans. Speech and Audio Processing*, 7:272–281.

J. Gauvain and C. Lee. 1994. Maximum a posteriori estimation for multivariate gaussian mixture. *IEEE Trans. Speech and Audio Processing*, 2:291–298.

T. Hain, J. Dines, G. Gaurau, M. Karafiat, D. Moore, V. Wan, R.J.F. Ordelman, and S. Renals. 2005. Transcription of conference room meetings: an investigation. In *In Proceedings of Interspeech 2005*, Lisabon, Portugal.

N. Kumar. 1997. *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*. Ph.D. thesis, John Hopkins University, Baltimore.

P. Schwarz, P. Matějka, and J. Černocký. 2004. Towards lower error rates in phoneme recognition. In *Proc. International Conference on Text, Speech and Dialogue*, pages 465–472, Brno, Czech Republic, September.

Petr Schwarz, Pavel Matějka, and Jan Černocký. 2006. Hierarchical structures of neural networks for phoneme recognition. In *Proc. ICASSP 2006*, Toulouse, France.

Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan. 2005. Using mlp features in sri's conversational speech recognition system. In *In Proceedings of Interspeech 2005*, pages 2141–2144, Lisabon, Portugal.

Vocal Tract Normalization Based on Formant Positions

Nikša Jakovljević*, Dragiša Mišković*, Milan Sečujski*, Darko Pekar†

* Faculty of Engineering, Trg Dositeja Obradovića 6, Novi Sad, Serbia
{jakovnik, dragisa, secujski}@uns.ns.ac.yu
†Alfanum Ltd., Trg Dositeja Obradovića 6, Novi Sad, Serbia
darko.pekar@alfanum.co.yu

Abstract

This paper presents our initial results in a new approach to vocal tract normalization (VTN). In experiments based on continuous automatic speech recognition (ASR) the VTN procedure is in general carried out in both training and test phase. In the training phase it is used to obtain speaker independent acoustic models of phones. In the test phase it is used to convert input observations into observations nearer to the ones corresponding to the universal speaker. The approach described in this paper is new, because instead of training a single set of acoustic models for the universal speaker, several sets of acoustic phone models corresponding to speakers with similar vocal tract lengths were created. Instead of using the VTN procedure in the test phase, the recognized sequence estimated as the most likely one among sequences based on different acoustic model sets was identified as the final recognition result.

Normiranje vokalnega trakta na podlagi lege formantov

V prispevku so predstavljeni začetni rezultati novega pristopa k normiranju vokalnega trakta (NVT). V eksperimentih, ki temeljijo na samodejnem razpoznavanju tekočega govora, se postopek NVT izvaja tako v učni kot v testni fazi. V učni fazi se uporablja za pridobivanje akustičnih modelov fonov, ki niso odvisni od govorca. V testni fazi se uporablja za pretvorbo vhodnih opažanj v opažanja, ki so bližja tistim, ki ustrezajo univerzalnemu govorcu. Pristop, ki je opisan v tem prispevku, je nov: namesto da bi učili posamezno množico akustičnih modelov za univerzalnega govorca, je bilo ustvarjenih več množic akustičnih modelov fonov, ki ustrezajo govorcem s podobno dolžino vokalnega trakta. Namesto uporabe postopka NVT v testni fazi je bil končni rezultat razpoznavanja prepoznano zaporedje, ki je bilo ocenjeno kot najbolj verjetno med zaporedji, temelječimi na različnih množicah akustičnih modelov.

1. Introduction

Most of today's automatic speech recognition (ASR) systems are based on hidden Markov models (HMM). Acoustic variations between training and test conditions, caused by different microphones, channels, background noise as well as speakers, are known to deteriorate ASR performance. Speaker variations can be divided into extrinsic and intrinsic. Extrinsic variations are related to cultural variations among speakers as well as their emotional state, resulting in diverse speech prosody features. Intrinsic variations are related to speaker anatomy (vocal tract dimensions) and they manifest in different formant positions of a given phoneme. Procedures for reducing variation caused by different vocal tract dimensions in feature domain are known as vocal tract normalization (VTN) procedures, whereas procedures in acoustic model domain are referred to as adaptation procedures.

In this paper the improvements of the AlfaNum ASR system obtained by the VTN procedure will be presented. In section 3, a description of the corpus and features used is given. Section 4 contains a description of HMM modeling on phonetic level. A description of VTN procedures is given in section 5. Experiment results are presented in section 6, followed by conclusions in section 7.

2. Goal of the paper

Variations in vocal tract length are the main reason for diverse formant positions within a given phoneme spoken by different persons, hence formant based spectrum warping is more than reasonable. Unfortunately, this approach to VTN has several disadvantages: (i) formant positions are context dependent and could vary largely with different context even for a single speaker; (ii) there are overlaps between different formants across vowels spoken by various speakers; (iii) existing formant estimation techniques are not robust enough. Zhan and Waibel (1997) showed that VTN based on formant positions did not result in any performance improvement, since formant frequency could not reflect difference in vocal tract length among speakers because they are calculated with an unconstrained context and there is no guarantee of phone balance in context among speakers (Zhan, Waibel, 1997). Exact phone boundaries can be used to avoid the problem of context dependency of formant positions only in the training phase, since they are not known in the test phase. In this approach exact phone boundaries are used in training phase to make clusters of speakers with similar vocal tract lengths. For each cluster of speakers a set of acoustic models is created. In the test phase the recognized sequence estimated as the most likely one among sequences based on different acoustic model sets was identified as the final recognition result. Division of the training set into subsets i.e. speaker clusters would reduce the number of utterances per cluster and decrease robustness of consequent acoustic models. In order to overcome this problem a warping procedure was used to extend each training subset with utterances spoken by speakers out of the cluster.

3. Database and features

The used corpus is a part of the Serbian SpeechDat database (Đurić, Pekar, Jovanov, 2002), containing only utterances spoken by male speakers. The corpus in this experiment is reduced only to those speakers for which at least 10 instances of each vowel could be found in the database in order to achieve good vocal tract length estimation for each speaker in the corpus. The Serbian SpeechDat database was recorded through the public switched telephone network and sampled at 8 kHz with 8-bit A-law quantization. The training set contains 14496 utterances spoken by 340 speakers. For testing system

performance 2 test sets were used. The first test set contains 184 utterances spoken by 17 different speakers. No utterance spoken by any of these speakers is present in the training set. The second test set contains 435 utterances spoken by 17 different speakers. Some of the utterances spoken by these speakers are present in training set but not the same ones. The feature vector which was used consists of 2 streams. The first stream contains 6 energy coefficients: normalized energy, logarithm of the energy and their first and second derivatives. The second stream contains 36 coefficients (12 static, 24 dynamic), which describe spectral envelope and its changes in time. These 12 static coefficients describe spectral slopes, or more precisely, differences in energy between successive filter banks. Filter banks divide the Mel-scaled spectrum from 50 to 3800 Hz into 27 regions of equal width. Slopes are evaluated for every other filter bank starting from the third one. Spectral components below 300 Hz and above 3400 Hz are given less relative importance because the AlfaNum ASR system uses telephone quality recordings where these components are distorted. The feature vector is estimated on 30 ms long segment. Overlapping between successive segments is 20 ms.

4. Models

For the purposes of this experiment, several changes into the phonetic inventory of the Serbian language had to be introduced. Instead of the standard 5 vowels in Serbian, two sets containing 5 long and 5 short vowels are taken into consideration (the boundary between the two being 65 ms), and the phone /ə/ (IPA notation) is regarded as a standard vowel as well. The distinction based on vowel length is motivated by a need to model steady formant positions within long vowels better. Closure and explosion of affricates and stops are modelled separately and referred to as subphones. The basic modelling unit is a context dependent phone or subphone referred to as *triphone*. Silence and non-speech sounds present in the corpus are modelled as context independent units.

The number of states per model is proportional to the average duration of all the instances of the corresponding phone in the database. The number of mixtures per state depends on the distribution of observations in the feature space and is determined dynamically. During the initial training the maximum number of mixtures and the minimum number of observations per mixture are specified.

Using triphones instead of monophones leads to a very large set of models and insufficient training data for each triphone. All HMM state distributions would be robustly estimated if sufficient observations were available for each state. This could be achieved by extending the training corpus or by including observations related to acoustically similar states. The second solution was chosen as being less expensive, even though it generates some suboptimal models. More details about the tying procedure used can be found in (Jakovljević, Pekar, 2005).

5. Formant estimation and the warping function

Variations in vocal tract length are the main reason for diverse formant positions within a given phoneme spoken by different persons, therefore formant based spectrum warping is more than reasonable. Unfortunately, the existing formant estimation techniques are not robust enough. Some of the most frequent errors are: formant merging, shifting formant frequencies towards harmonics and false maximum caused by channel distortion (Gouvea, 1998). The algorithm used for formant detection was the one described in (Welling, Ney, 1998). The algorithm does not perform sufficiently well for Serbian vowels /u/ and /i/. The first and the second formant of the vowel /u/ are in many cases very close to each other in the spectrum, and the algorithm can erroneously identify them as a single formant, thus the third formant is detected as the second. The first formant of the vowel /i/ is very low and in some cases attenuated by the channel, and the algorithm often identifies the peak in the range between 600 and 1800 Hz as the first formant. This kind of error is caused by pre-emphasis, but omitting pre-emphasis would result in wrong formant positions for other vowels. Coarticulation is known to cause formant transition in vowels. If the vowel is too short, positions of its formants cannot reach context neutral values. In order to reduce this type of variability, formant position estimation is based on the most reliable 50% of the frames of long vowels /a/, /e/ and /o/, which are those in the middle of the vowel. The results published show minor differences in performance for various VTN function types (Zhan, Westphal, 1997; Uebel, Woodland, 1999; Pitz, 2005). The linear function was chosen as the simplest one and applied in addition to the Mel-scale warping mentioned above. The most natural way to evaluate the frequency warping factor is as a mean value of the ratio of the universal and the current formant value, the universal formant value being the mean formant value for a given phone across all speakers. The frequency warping factor α_c (i.e. linear function slope) for a given speaker can thus be estimated as follows:

$$\alpha_c = \sum_i \sum_f \frac{\mu_{il}}{F_{ilf}} \tag{1}$$

where μ_{il} is the mean value of the *i*-th formant in the phone *l* across all speakers, and F_{ilf} is the current value of the *i*-th formant in the frame *f* of the phone *l*. This approach to warping factor estimation does not consider the possibility of false formant estimation. A more robust way to estimate α_c is as follows:

$$\alpha_f = \arg\max_{\alpha} \left\{ \prod_i P\left\{ \alpha F_{ilf} \mid il \right\} \right\} \tag{2}$$

$$\alpha_f = \frac{\sum_i F_{ilf}\, \mu_{il} \,/\, {\sigma_{il}}^2}{\sum_i \left(F_{ilf}\, \mu_{il} \,/\, \sigma_{il} \right)^2} \tag{3}$$

$$\alpha_c = \frac{\sum_f \alpha_f \prod_i P\{\alpha_f F_{ilf} \mid il\}}{\sum_f \prod_i P\{\alpha_f F_{ilf} \mid il\}} \tag{4}$$

where α_f is the warping factor for the *f*-th frame, F_{ilf} is the value of the *i*-th formant in the *f*-th frame, μ_{il} is the mean value of the *i*-th formant in the phone *l*, s_{il} is the standard deviation for the *i*-th formant in the phone *l*, $P\{\alpha_f F_{ilf}|i,l\}$ is the probability that frequency $\alpha_f F_{ilf}$ is actually the *i*-th formant in the phoneme *l*, and α_c is the warping factor for a given speaker. In the first stage for each frame, warping factor α_f is evaluated as most probable warping factor for given vowel (Eq. 2). Under assumption that formant distribution across all speakers for a given vowel is Gaussian, Eq. 2 becomes Eq. 3. In the second stage the warping factor for a given speaker is calculated as the average value across all frames. Taking probability $P\{\alpha_f F_{ilf}|i,l\}$ into account reduces formant

estimation errors. This method could be performed only in the training phase, when phones and theirs boundaries are known. If the warping factor were calculated based on formant positions of only one formant of a single vowel, the reliability factor $P\{\alpha_f F_{ilf}|i,l\}$ would be eliminated, reducing Eq. 4 to Eq. 1.

6. Experiments

6.1. Finding optimal features for warping factor estimation

The first step of the experiment was finding optimal features for warping factor estimation. The search space contains different combinations of the first 3 formants (F1, F2 and F3) of the vowels /e/, /a/ and /o/. During the evaluation of the formant estimation algorithm, vowels /i/ and /u/ were identified as unreliable (about 40% of observed frames were incorrect). Instead of using an existing ASR system as a reference, a new one using the training corpus adapted for VTN purposes and described in section 2 was trained. Results are thus made independent of the training corpus and none of the utterances of speakers whose utterances are present in the test set are used for acoustic models training. The grammar consists of 195 different words where 8 of them are not present in the VTN test set but are phonetically similar to some of the existing ones. The testing was carried out in a supervised mode to avoid errors caused by incorrect vowel recognition, because the aim of this step was to find optimal features (the set of formants) for reliable warping factor estimation and not to implement VTN procedure itself. In supervised mode phone boundaries are located

formant	vowel	false	ins	del	WER[%]
reference system		47	37	0	20.90
F2	/e/	39	24	1	15.92
F2	/a/ /e/	38	27	1	16.42
F1 F2 F3	/a/	41	31	0	17.91
F3	/e/	43	29	1	18.16
F2 F3	/e/	42	30	1	18.16
F2 F3	/a/ /o/	44	30	0	18.41
F2	/a/ /e/ /o/	39	35	1	18.66
F2	/a/	46	31	0	19.15
F2 F3	/a/ /e/ /o/	43	34	1	19.40
F3	/o/	46	33	1	19.90
F2	/e/ /o/	44	35	1	19.90
F2 F3	/a/	49	32	0	20.15
F1	/a/	47	34	1	20.40
F2 F3	/a/ /e/	45	37	1	20.65
F2 F3	/e/ /o/	49	35	0	20.90
F2	/a/ /o/	47	37	0	20.90
F3	/a/	46	37	1	20.90
F1	/e/	50	43	1	23.38
F2 F3	/o/	55	56	0	27.61
F2	/o/	64	55	2	30.10
F1	/o/	71	52	0	30.60

Table 1: System performances for different features for warping factor estimation

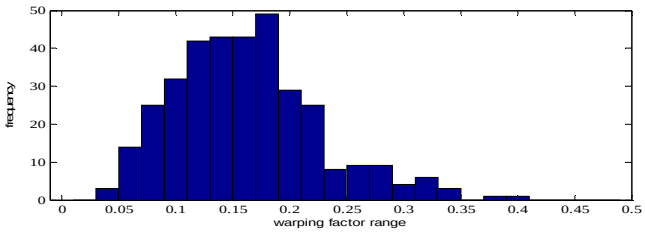


Figure 1. The histogram of the warping factor range

manually. After appropriate warping factor evaluation for each speaker in the test set, the recognition is performed. The results of this phase of the experiment are presented in Table 1. The best system performance is achieved by warping factor estimation based on the second formant of the vowel /e/. Very similar performance is obtained if the warping factor is estimated based on the second formant of vowels /e/ and /a/ instead. Performance improvement comes mostly as a result of a decrease in the number of insertions. Reduction of Eq. 4 to Eq. 1 had no effect since reliability of formant estimation for phoneme /e/ is high. Since the vowel /a/ is the closest one to the neutral vowel /ə/, where each formant frequency is inversely proportional to vocal tract length, it was expected that the VTN based on the formants of the vowel /a/ would produce the best results. However, this was not the case. The results obtained for the formants of the vowel /a/ show some interesting features. If a single formant (F1, F2 or F3) were used for warping factor estimation, or a combination of F2 and F3, the gain is far less than if all 3 formants (F1, F2 and F3) of the same vowel were used. A possible explanation is that during warping factor estimation based only on one formant of a single vowel, warping factor estimation is less reliable, as explained in section 5. It can be seen that experimental results are not very consistent. The system performance in case warping factor estimation is based on F2 of the vowel /a/ is somewhat inferior to the system performance in case the estimation is based on F3 of the vowel /e/. On the other hand, the system with warping factor estimation based on F2 of vowels /e/ and /a/ performs significantly better than the system with estimation based on F2 and F3 of the vowel /e/. One can find further such examples in Table 1. The first formant turned out to be the least appropriate feature for warping factor estimation. A system with warping factor estimation based only on the first formant shows serious degradation of performance in comparison with the referent system in most cases, except for the vowel /a/. In the experiments described in (Gouvea, 1998), the system using warping factor estimation based on the first formant showed the least improvement, but the result was still better than if no VTN procedure had been used. For this reason the first formant was not used in any of the experiments, except in the case of the vowel /a/, because the ratio of its first three formant frequencies is always near to 1:3:5 and it can be shown that F1 contributes to the reliability of estimation of F2 and F3. The second formant has turned out to be the best feature for warping factor estimation. That was not unexpected, since it is known that professional impersonators move their F2 closer to the one of the target speaker, as it seems to be a very important feature of human speaker recognition (Blomberg, Elenius, Zetterholm, 2004). Unfortunately, variations among warping factors obtained in different ways are rather high. Fig. 1 shows the distribution of the

differences between the maximum and minimum warping factor values for each speaker. This is the reason why for some feature combination VTN procedure did not result in any improvement.

6.2. Vocal tract normalization

A common method for improving performance is to use separate acoustic model sets for male and female speakers. Instead of creating an universal acoustic model set, 3 separate model sets were created, representing phones uttered by male speakers only. We intend to extend this approach to the models for female speakers. Any division of the training set into subsets would reduce

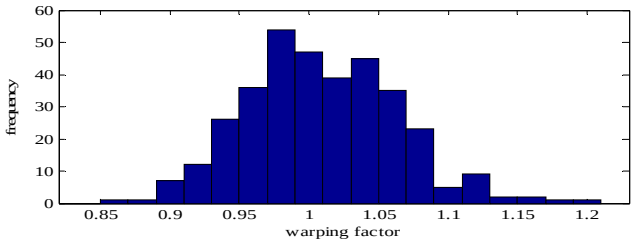


Figure 2. Warping factor histogram for warping factor estimation based on the 2nd formant of /e/

the number of utterances per subset and decrease robustness of consequent acoustic models. In order to overcome this problem warping factors based on F2 of the vowel /e/ were used to extend each of the training sets. The histogram of warping factors for all speakers in the training set is shown in Fig 2. It can be seen that the best coverage of speakers in the training set can be obtained if the subsets of speakers with warping factor values 0.95, 1 and 1.05 are chosen. In order to be able to include the utterances with an inappropriate warping factor in the training set, the spectrum of each such utterance should be scaled with the ratio of its own warping factor and the target warping factor.

The comparative performance is presented in table 2. In this experiment the most successful set of acoustic models describing phones uttered by male speakers was used as the referent system. The referent system was trained on all sentences in the corpus, not only those of speakers for which at least 10 instances of each vowel were found in the database. The test set is the same as the standard set for system evaluation described in section 2. It contains 597 utterances with 735 words spoken by 100 speakers. The grammar consists of 110 words with 40 of them not present in the training set.

In this way the complexity increased 2.8 times (somewhat less than 3 because each subset in the extended VTN system had fewer mixtures than the referent system itself), and the relative improvement is about 7%. It is expected that extension of this approach will result in smaller complexity increase since some of the male and female phone utterances overlap regarding formant positions. The extension of the test set will improve WER resolution, which may give a better picture of the relative improvement.

system	false	ins	del	WER[%]
referent	39	37	1	8.35
extended VTN	30	26	1	7.75

Table 2: System performance

7. Conclusion

In this paper a new approach to the vocal tract normalization procedure is presented. Three separate acoustic model sets are created to describe phones uttered by male speakers only. The utterances are split into 3 classes according to speaker vocal tract length estimated based on F2 of the vowel /e/. Reduction of the number of instances caused by this procedure is overcome by recalculation of warping factors for each utterance. Each model set is trained on the same utterances, but warping factors for an utterance may vary depending on the model being trained. Such an approach omits warping factor calculation during the test procedure, but increases model complexity about 3 times. Achieved relative improvement in WER of 7 % is very small considering the increase in complexity. It is expected that the extension of this approach to acoustic models of phones spoken by female speakers will result in a more significant improvement in performance without such an increase in complexity. On the other hand, this extension will expand the training corpus with utterances spoken by female speakers.

8. Acknowledgment

This work was supported in part by the Ministry of Science and Environment Protection of Serbia within the Project “Development of speech technologies in Serbian and their application in ‘Telekom Srbija’” (TR-6144A).

9. References

Blomberg. M. D. Elenius, E. Zetterholm, 2004. Speaker verification scores and acoustic analysis of a professional impersonator. *FONETIK 2004 Proceedings.*

Gouvea. E., 1998. *Acoustic-Feature-Based Frequency Warping For Speaker Normalization*. Ph. D. Thesis, Department of Electrical and Computer Engineering Pittsburgh.

Đurić. N., D. Pekar, Lj. Jovanov, 2002. Structure of SpeechDat(E) database for Serbian, recorded over PTN. *DOGS 2002 Proceedings*, 1:57-60

Jakovljević. N., D. Pekar, 2005. Description of Training Procedure for AlfaNum CSR System. *EUROCON 2005 Proceedings.*

Pitz. M., 2005. *Investigation on Linear Transformations for Speaker Adaptation and Normalization*. Ph. D. Thesis University Aachen.

Uebel. L, P. Woodland, 1999. An Investigation into Vocal Tract Length Normalization. *EUROSPEECH99 Proceedings*, 6:2527-2530.

Welling L., H. Ney, 1998 Formant estimation for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 6:36-48.

Zhan. P., M. Westphal, 1997. Speaker Normalization based on Frequency Warping. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing Proceedings*, 2:1039-1042.

Zhan. P., A. Waibel, 1997. Vocal Tract Length Normalization for Large Vocabulary Continuous Speech Recognition. *Language Technologies Institute Technical Report CMI-LTI-97-150*, Pittsburgh.

SI-PRON: a Comprehensive Pronunciation Lexicon for Slovenian

Jerneja Žganec Gros[1], Varja Cvetko-Orešnik[2], Primož Jakopin[2]

[1]Alpineon R&D, Iga Grudna 15, Ljubljana, Slovenia
[2]Fran Ramovš Institute of the Slovenian Language, Ljubljana, Slovenia

Abstract

We present the efforts involved in designing SI-PRON, a comprehensive machine-readable pronunciation lexicon for Slovenian. It has been built from two sources and contains all the lemmas from the Dictionary of Standard Slovenian (SSKJ), the most frequent inflected word forms found in contemporary Slovenian texts, and a first pass of inflected word forms derived from SSKJ lemmas. The lexicon file contains the orthography, corresponding pronunciations, lemmas and morphosyntactic descriptors of lexical entries in a format based on requirements defined by the W3C Voice Browser Activity. The current version of the SI-PRON pronunciation lexicon contains over 1.4 million lexical entries. The word list determination procedure, the generation and validation of phonetic transcriptions, and the lexicon format are described in the paper. Along with Onomastica, SI-PRON presents a valuable language resource for linguistic studies and research of speech technologies for Slovenian. The lexicon is already being used by the Proteus Slovenian text-to-speech synthesis system and for generating audio samples of the SSKJ headwords.

SI-PRON: slovar izgovorjav slovenskih besed

Naglasno mesto predstavlja zlog, na katerem ima beseda tonsko ali jakostno izrazitost. Glede na besedno obliko poznamo stalno mesto naglasa, kot npr. v francoščini na zadnjem zlogu, delno omejeno mesto naglasa, kot npr. v hrvaščini, kjer zadnji zlog ni nikoli naglašen, ter prosto mesto naglasa. Za slovenski jezik je značilno prosto mesto naglasa, saj se ta lahko pojavi na prvem, zadnjem, predzadnjem ali predpredzadnjem zlogu. Prav tako ima lahko posamezna beseda več mest naglasa. Mesto naglasa je določeno za vsako besedo posebej in se ga naučimo hkrati z učenjem jezika in besed. Slovar izgovarjav, ki vsebuje fonetične prepise besed, vključno z oznakami za naglasno mesto, je nujno potreben jezikovni vir za razvoj jezikovno-tehnoloških izdelkov ter za jezikoslovno študije. Za slovenski jezik so bili zgrajeni številni slovarji izgovarjav, noben izmed njih pa ne pokriva celotnega besedišča iz Slovarja slovenskega knjižnega jezika. V prispevku predstavljamo postopek pridobivanja SI-PRON slovarja izgovarjav za slovenske besede, ki so zbrane v SSKJ. Seznam osnovnih besednih oblik smo razširili s številnimi pregibnimi oblikami. Skupaj z Onomastico predstavlja SI-PRON dragocen jezikovni vir za izgradnjo govornih aplikacij. Vgrajen je bil v sintetizator govora za slovenski jezik, Proteus, prav tako je bil uporabljen za izgradnjo zvočnih podob gesel SSKJ, ki so predstavljene na spletni različici slovarja SSKJ, na strani http://bos.zrc-sazu.si/sskj.html.

1. Introduction

Consistent specification of word pronunciation is critical to the success of many speech technology applications. Most state-of-the-art Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) systems rely on lexicons, which contain pronunciation information for many words. To provide for a maximum coverage of the words, multi-word expressions or even phrases, which commonly occur in a given application-domain, application-specific word or phrase pronunciations may be required, especially for application-specific proper nouns, such as personal names or location names.

Several guidelines have been reported to define the structure of a pronunciation lexicon, ranging from simple two-column ASCII lexicons providing the mapping between graphemic and phonemic transcriptions, to more general de-facto standards and new standardization attempts, which are also handling multiple orthographies and multiple pronunciations.

The ISO-TC37 initiative, which started at LREC 2002, initiated work on a family of ISO standards related to natural language processing (Romary et al., 2006). Currently these standards are available in working drafts of high-level specifications for word segmentation, feature structures, annotations, and also for lexicons. The high-level specifications build on lower-level specifications in form of language and country codes, data categories, code scripts, and Unicode. Lexicon specifications are covered by the "Lexical Markup Framework" under ISO 24613 (Romary et al., 2006). The same description structure in terms of morphology, syntax and semantics (and translation) applies to monolingual up to multilingual lexicons. Multi-word expressions are given special attention.

Another initiative, the W3C Voice Browser Activity, has recently issued a last-call working draft of the Pronunciation Lexicon Specification (PLS) Version 1.0 (W3C PLS Version 1.0, 2006), which is expected to be soon submitted as a W3C candidate recommendation. The PLS document was designed to enable interoperable specification of pronunciation information for both ASR and TTS engines within voice browsing applications. The mark-up language allows one or more pronunciations for a word or phrase to be specified using a standard pronunciation alphabet or if necessary using vendor specific alphabets. Pronunciations are grouped together into the PLS document which may be referenced from other markup languages, such as the Speech Recognition Grammar Specification (SRGS) and the Speech Synthesis Markup Language (SSML).

The Pronunciation Lexicon Markup Language, based on PLS, is designed to allow open, portable specification of pronunciation information for speech recognition and speech synthesis engines. The language is intended to be easy to use by developers while supporting the accurate specification of pronunciation information for international use.

The LC-STAR project consortium published another set of recommendations for speech technology lexicons, with an emphasis on application in machine translation, speech recognition and speech synthesis (Shamas & van den Heuvel, 2004; Fersøe et al., 2004). A Slovenian lexicon, produced at the University of Maribor, has been built in the scope of the project (Verdonik et al., 2004). Compared to the LC-STAR lexicon specifications the current version of PLS lacks description specifications for more complex features, such as morphological, syntactic, and semantic features of lexical entries.

In Slovenian, lexical stress can be located on almost any syllable and it obeys hardly any rules. The stressed syllable in Slovenian may form the ultimate, the penultimate or the preantepenultimate syllable of a polysyllabic word. Speakers of Slovenian have to learn lexical stress positions along with learning the language. As a consequence, a pronunciation lexicon that indicates lexical stress positions for as many Slovenian words as possible is crucial for the development of speech technology applications and linguistic research. Such a lexicon can be used either in its full-blown form or as a training material for machine learning techniques aimed at automatically predicting word pronunciations.

Several attempts towards pronunciation lexicon construction for Slovenian have been reported so far (Derlić & Kačič, 1997; Gros & Mihelič, 1999; Gros et al., 2001; Šef et al., 2002; Verdonik et al., 2002; Mihelič et al., 2003). However, none of them has used the full lemma set as given in the Dictionary of Standard Slovenian (SSKJ) (SSKJ, 1991).

The paper describes the construction of a comprehensive reference pronunciation lexicon for Slovenian based on two sources: the information from the SSKJ and another list of the most frequent inflected word forms, which has been derived by an analysis of contemporary Slovenian text corpora.

2. The SI-PRON Pronunciation Lexicon

2.1. SI-PRON Word List

The work on designing a new pronunciation lexicon begins with the selection of words, multi-word expressions or phrases, which will be represented in the lexicon. Several word-list selection procedures are known (Ziegenheim, 2003).

The construction of the SI-PRON lexicon started with the complete lemma word list of 93,154 entries from the SSKJ provided by the Fran Ramovš Institute of the Slovenian Language, furnished with basic lexical stress information on the stressed vowels and pronunciation exceptions. The complete word pronunciations still had to be determined.

In order to further expand the SI-PRON word list, we are augmenting the SSKJ lemma descriptions with part-of-speech information and declension/conjugation categories (Toporišič, 1991), specifying the inflectional paradigms of the lemmas. Irregular inflected word forms are processed separately. Using automatic procedures, we are fully expanding the lemmas into inflected word forms. So far, over 1 million lexemes containing lexical stress information have been derived.

Since SSKJ contains many words derived from literary texts, not so common in everyday situations, we decided to upgrade the SI-PRON pronunciation lexicon with a list of 50,000 most frequent inflected word forms whose lemmas are not covered by the SSKJ word list. This additional word list has been derived from a statistical analysis of a contemporary Slovenian text corpus. The corpus comprising over 3 million Slovenian words was composed mainly from fiction and mainstream Slovenian newspaper texts: Delo, Večer, and the former Slovenec. After tokenization and the elimination of numerals, named entities, acronyms, and abbreviations, the remaining text corpus included over 3 million tokens. Acronyms, abbreviations, and named entities were stored into separate word lists.

A statistical analysis performed on the text corpus showed that about 50.000 most frequent words accounted for approaching 95% of all non-SSKJ words used in the text corpus (Gros & Mihelič, 1999). These words form the main additional word list. They were equipped with part-of-speech tags indicating the part-of-speech function of the words in the text corpus.

2.2. Collocations and Multi-word Expressions

The identification of collocations, i.e. current combinations of words as they appear in context, can considerably increase the naturalness of synthetic speech. In human speech, collocations act as prosodic units and are subject to a higher degree of reduction and internal coarticulation than they would be had they been ordinary, separate words. We have chosen a lexical approach for handling collocations. The most common collocations or multi-word expressions, reflexive verbs included, are stored in a separate pronunciation lexicon.

3. Phonetic Transcriptions

We have developed a tool to automatically derive word pronunciations for the SSKJ inflected words, by looking-up their stem pronunciation and appending that of the correct inflection from inflectional paradigms and morphological rules of Slovenian (Toporišič, 1991).

Therefore, the pronunciation of lexemes has been derived automatically for the SSKJ and SSKJ inflected word lists (about 2,500 entries, mainly words of foreign origin that do not obey the general Slovenian pronunciation rules, have been manually transcribed), and semi-automatically for the remaining part of the word list. Automatic lexical stress assignment and automatic grapheme-to-phoneme conversion rules have been used to process the latter.

3.1. Lexical Stress Assignment

The automatic lexical stress assignment algorithm for unseen words, which we applied is to a large extent determined by (un)stressable affixes, prefixes, and suffixes of morphs and is based upon observations by linguists (Toporišič, 1991).

For words that do not belong to these categories, the most probable stressed syllable is predicted using the results from a statistical analysis of stress position depending on the number of syllables within a word (Gros & Mihelič, 1999).

3.2. Grapheme-to-Phoneme Rule Set for Slovenian

Context-free grapheme-to-allophone rules from the Proteus standard words rule set (Žganec Gros, 2006) translate each grapheme string into a series of allophones.

The rules are accessed sequentially until a rule that satisfies the current part of the input string is found. The transformation defined by that rule is then performed, and a pointer is incremented to point at the next unprocessed part of the input string. The procedure is repeated until the whole string has been converted.

The context free rules are rare and they include a one-to-one correspondence, two-to-one correspondence and one-to-two correspondence.

The vast majority of the rules for grapheme-to-allophone transcription for Standard Slovene are context-sensitive. This means that a grapheme or a string of graphemes is transcribed differently according to its phonetic environment. Certainly all rules for determining which allophone of a certain phoneme is to be used in a phonetic sequence are context-dependent.

Each context-sensitive rule consists of four parts: the left context, the string to be transcribed, its right context and the phonetic transcription. A number of writing conventions has been adopted in order to keep the number of rules relatively small and readable. The left and the right context may contain code characters describing larger phonetic sets, e.g.: '#' stands for vowels, '$' for consonants, '_' for white space.

The rules for consonants are rather straightforward, while those for vowels must handle vowel length and the variant realizations of the orthographic /e/ and the orthographic /o/ in stressed syllables.

A typical grapheme-to-allophone rule in the Proteus standard words rule set has the following structure:

left context	grapheme string	right context	allophone string
$	**/er/**	_	[@r]
=	/**n**/	k	[N]

The first rule says that the word final /er/ preceded by a consonant is transcribed as [@r] (e.g. /gaber/ -> [*ga:.b@r]). The second rule implies that any /n/ followed by /k/ is transcribed into [N] ([N] is the allophone of [n] when followed by /k/ or /g/, e.g. in /anka/ -> [*a:N.ka]).

The initial rule set based on the one produced in 2001 (Gros et al., 2001) was built by taking into acconut various observations of expert linguists, e.g. (Toporišič, 1991), and other basic rule sets for Slovenian grapheme-to-allophone transcription (Gros & Mihelič, 1999).

The initial set of rules has been undergoing continuous refinement ever since and resulted in 194 rules of the Proteus standard words rule set (Žganec Gros, 2006). Rules for coarticulatory pronunciation corrections of words according to the words' left context and to the right context are included.

In the recent years, telecommunication applications of ASR and TTS have increased in importance, e.g. automatic telephone directory inquiry systems. Names of locations (cities, streets, etc.) and other proper names cannot be mentally reconstructed from the context when listening to the messages, and correct name pronunciation is required. The Proteus standard word rules developed for a standard Slovenian vocabulary do not lead to satisfactory results when applied to names. Therefore, additional 'name-specific' rules were added to the final Proteus standard words rule set resulting in the Proteus names rule set.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0" xml:lang="si-SI" alphabet="x-sampa-SI-reduced">
  <lexeme>
    <grapheme>dober</grapheme>
    <phoneme>"d/o:-b@r</phoneme>
    <!-- This is an example of the x-sampa-SI-reduced string
      for the pronunciation of the Slovenian word: "dober",
      meaning "good" in English -->
  </lexeme>
</lexicon>
```

Figure 1. An example of a simple lexicon file with a single lexeme within SI-PRON.

3.3. Transcription Accuracy Experiment

The phonemization errors were determined by comparing the automatic transcription outputs to manually verified pronunciation lexicon transcriptions.

A performance test applied on the SI-PRON SSKJ-based word list pronunciation lexicon showed error rates of about 25% in the stress assignment of unknown words and consequently in the phonetic transcription. If stress assignment and the transcriptions of graphemic /e/ and /o/ in stressed syllables was manually verified or known in advance, a transcription success rate of 99.1% was achieved for standard SSKJ words.

A closer examination of the mismatches revealed that the majority of the errors could be attributed to inconsistencies in manual labelling during the preparation of the original SSKJ.

As a consequence, we argue that, in order to semi-automatically derive phonetic transcriptions for Slovenian words not covered by the lexicon with a 0.3% error rate, manual validation of the stress position and its type have to be carried out, starting from automatically predicted stress positions. The rest can be performed automatically

by applying our upgraded grapheme-to-phoneme conversion rule set.

4. SI-PRON Format

The SI-PRON lexicon format complies with the Pronunciation Lexicon Specification (PLS) Version 1.0, a W3C Voice Browser Activity working draft of syntax specification for pronunciation lexicons (W3C PLS Version 1.0, 2006). This lexicon specification has been recommended for use by speech recognition and speech synthesis engines in voice browser applications.

The element **<lexeme>** represents a lexical entry and may include multiple orthographies and multiple pronunciation information. An example of a simple lexicon file with a single lexeme within SI-PRON would be as shown in Fig. 1.

In the Pronunciation Lexicon Specification, the pronunciation alphabet is specified by the **alphabet** attribute of the **<phoneme>** element. We are using the "x-sampa-SI-reduced" phonetic alphabet, a subset of the X-SAMPA set as defined for Slovenian (Zemljak et al., 2002), augmented with additional markers for Slovenian lexical stress accents (acute, circumflex, and grave) and tonemic accents (tonemic acute and tonemic circumflex). Both primary and secondary stress positions are marked.

The **<alias>** element is used to provide the pronunciation of an acronym or an abbreviation in terms of an expanded orthographic representation.

4.1. Homographs

Homographs or words with the same spelling but different pronunciations can be treated in two ways. If we do not want to distinguish between the two words then we can represent them as alternate pronunciations within the same **<lexeme>** element. In the opposite case, two different **<lexeme>** elements need to be used. In both cases the application, which is making use of the lexicon, will not be able to decide when to apply the first or the second transcription unless additional information, such as context-specific attributes or part-of-speech information is provided.

4.2. Multiple Pronunciations

Providing multiple pronunciations for items that share the same orthography and meaning is important for speech recognition lexicons because they provide information on variations of pronunciation within a language. Therefore, for many lexemes, words, and multi-word expressions, multiple standard pronunciations are specified, including those, which consider possible coarticulation effects at word boundaries. Multiple pronunciations are indicated by subsequent **<phoneme>** elements within one **<lexeme>** element.

Pronunciation preference – extensions needed?

In TTS applications, typically only one pronunciation among the multiple pronunciation possibilities is required. Therefore, to indicate default pronunciation variation, the **prefer** attribute can be used in PLS. In SI-PRON, unless marked otherwise, the default pronunciation is the first pronunciation from SSKJ.

However, sometimes several pronunciation variations in SSKJ are (almost) equally preferred, whereas the actual preferred pronunciation for the TTS engine may depend on the application. This is not to be confused with application-specific pronunciations, which can be handled in separate application-specific pronunciation lexica. What we have in mind is that there may exist several almost equally preferred pronunciations for a given grapheme, and the developers would like to have a mechanism that would enable them to systematically choose the preferred one.

Typically one of the two almost equally preferred pronunciations yields better rendering of input text if the application requires either overarticulated or fluent pronunciation. Therefore, we would welcome a new optional attribute to the **<phoneme>** element in PLS, the: **pron-style** attribute indicating the preferred pronunciation variation of a lexeme with respect to the desired pronunciation style. The two attribute values, which would be useful for SI-PRON, are "fluent" and "overarticulated".

In addition, the **pron-style** optional attribute would need to be introduced into SSML, as a defined attribute for the **<voice>**, **<speak>**, **<p>**, and **<s>** elements.

For the same elements in SSML: **<voice>**, **<speak>**, **<p>**, and **<s>**, another optional attribute, emotion, would be useful (e.g. for computer games, where emotion changes occur frequently).

Example: For Slovenian male nouns, ending with a consonant followed by "ilec", SSKJ often provides one of the following single or multiple pronunciations of the "ilc" sequence within the genitive form of the noun: [iUts]/[ilts], [ilts]/[iUts], [ilts], or [iUts]; examples would be Slovenian words "nosilca", "krotilca", "darovalca", etc. Many other cases of such pronunciation variations are known for Slovenian, and are marked in SSKJ.

Whenever there are two pronunciation variations in SSKJ they typically account for an overarticulated (e.g. [ilts]) or a more fluent (e.g. [iUts]) pronunciation variation. The pronunciation order as indicated in SSKJ indicates a slight pronunciation preference in standard usage and should still be indicated by the **prefer** attribute. In order to enable high-quality TTS such pronunciation differentiations should be captured in the text rendering process.

This would avoid the confusion of having a multitude of TTS pronunciation lexicons with different variations of the default pronunciation as given by the **prefer** attribute. The multiple lexicons are impossible to edit synchronously, and the proposed approach would allow us to use one master pronunciation lexicon.

4.3. Multiple Orthographies

Sometimes multiple orthographies of a word share the same meaning and pronunciation. They are presented with subsequent **<grapheme>** elements within a single **<lexeme>** element.

4.4. Part-of-Speech Tags

The most recent specification of the PLS focuses on the major features described in the PLS requirements document. Many more complex features, such as those providing morphological, syntactic and semantic information associated with pronunciations are expected to be introduced in a future revision of the PLS specification.

Therefore, proprietary <lemma> and <morphsynt> elements have been additionally defined for SI-PRON. Multext-East morphosyntactic descriptors for the Slovenian language, as described in (Erjavec, 2004), were used to provide the part-of-speech information of the lexemes, along with the lemmas.

5. SI-PRON Validation

Finally, the SI-PRON lexicon has been subjected to an automatic validation as a way to ensure that the structure of the document is well-formed and conforms with the chosen Document Type Definition (DTD).

Additionally, manual validation of both phonemic transcriptions and morphosyntactic descriptions was performed on a subset of the lexicon comprising 5.000 lexical entries. A subset from the LC-STAR lexicon specifications for lexicon validation criteria was used (Shamas and den Heuvel, 2002).

A lexicon editing tool with a user-friendly interface has been designed to allow inspecting, editing, browsing and automatic validation of the pronunciation lexicon.

6. Conclusion

Due to free lexical stress position, pronunciation lexica are of crucial importance for development of speech technology applications and linguistic research for Slovenian. They are not only used for providing application-specific pronunciations or pronunciations of names, but are indispensable in any TTS or ASR system.

The task of constructing a master pronunciation lexicon is very tedious and time-consuming and should not be repeated often. Therefore, a master-lexicon approach is best suited for Slovenian TTS, in which many speaking-style pronunciation nuances are captured. We propose refined extensions to both PLS and SSML, which are described in section 4, and mainly deal with multiple pronunciations and morphosyntactic descriptions.

Along with Onomastica, SI-PRON presents a valuable language resource for linguistic studies as well as for research and development of speech technologies for Slovenian. The lexicon is already being used by the Proteus Slovenian text-to-speech synthesis system (Žganec Gros, 2006) and for generating audio samples of the SSKJ word list, which are available at the very end of every SSKJ lexical entry description (SSKJ audio, 2006).

7. References

Derlić, R., Kačič, Z., (1996). Definition of pronunciation dictionary of names and letter-to-sound rules for Slovene language - project Onomastica. In Proceedings of the 2nd International Workshop on Speech dialog man-machine, Maribor, Slovenia, June 26-27, pp. 153-158.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC'04, Lisbon, Portugal, pp. 1535-1538.

Fersøe, H., Hartikainen, E., van den Heuvel, H., Maltese G., Moreno A., Shammass S., Ziegenhain U. (2004). Creation and Validation of Large Lexica for Speech-to-Speech Translation Purposes. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC'04, Lisbon, Portugal.

Gros, J., Mihelič, F., (1999). Acquisition of an extensive rule set for Slovene grapheme-to-allophone transcription. In Proceedings of the 6th European Conference on Speech Communication and Technology EUROSPEECH'99, Budapest, Hungary, pp. 2075-2078.

Gros, J., Mihelič, F., Pavešić, N., Žganec, M., Mihelič, A., Knez, M., Merčun, A., Škerl, D., (2001). The phonetic SMS reader. In Proceedings of the Text, speech and dialogue 4th international conference, Železná Ruda, Czech Republic, Lecture notes in artificial intelligence, 2166. Berlin: Springer, pp. 334-340

Mihelič, F., Žganec Gros, J., Dobrišek, S., Žibert, J. and Pavešić, N., (2003). "Spoken language resources at LUKS of the University of Ljubljana", International Journal on Speech Technologies, Vol. 6, No. 3, pp. 221-232.

PLS-W3C, (2006). Pronunciation Lexicon Specification (PLS) Version 1.0, W3C Working Draft 31 January 2006. http://www.w3.org/TR/pronunciation-lexicon/ S4.7.

Romary, L., Francopoulo, G., Monachini, M. and Salmon-Alt, S. (2006). Lexical Markup Framework: working to reach a consensual ISO standard on lexicons. To be presented at LREC'06 as a tutorial. Genoa, Italy.

SSKJ audio (2006). available from http://bos.zrc-sazu.si/sskj.html.

Verdonik, D., Rojc, M., Kačič, Z., Horvat, B., (2002). Zasnova in izgradnja oblikoslovnega in glasovnega slovarja za slovenski knjižni jezik. In Zbornik konference Jezikovne tehnologije'02. Editors: Tomaž Erjavec, Jerneja Gros, Ljubljana, Slovenia, pp. 44-48.

Verdonik, D., Rojc, M. and Kačič, Z., (2004). Creating Slovenian language resources for development of speech-to-speech translation components, In Proceedings of the Fourth International Conference on Language Resources and Evaluation LREC'04. Lisbon, Portugal, pp. 1399-1402.

Shammass, S. & van den Heuvel, H., (2004). Specification of validation criteria for lexicons for recognition and synthesis, LC-STAR Deliverable D6.1. available from www.lc-star.com.

SSKJ (1997). Slovar slovenskega knjižnega jezika (The Dictionary of Standard Slovenian). 2nd edition, Ljubljana: DZS.

Šef, T., Gams, M., Škrjanc, M., (2002). Automatic lexical stress assignment of unknown words for highly inflected Slovenian language. In Zbornik 11. mednarodne Elektrotehniške in računalniške konference ERK 2002. Portorož, Slovenija., pp. 247-250. in Slovenian.

Toporišič, J. (1991). Slovenska Slovenica (Slovenian Grammar). Založba Obzorja Maribor.

Zemljak, M., Kačič, Z., Dobrišek, S., Gros, J., Weiss, P., (2002). Računalniški simbolni fonetični zapis

slovenskega govora. Slavistična revija, Vol. 50, No. 2, pp. 159-169.

Ziegenhain, U., (2003). Specification of corpora and word lists in 12 languages. LC-STAR Deliverable D1.1. available from www.lc-star.com.

Žganec Gros, J., (2006). Text-to-speech synthesis for embedded speech user interfaces, In WSEAS Transactions on Communications, No. 4, Vol. 5, pp. 543-548.

Pragmatically annotated corpora in speech-to-speech translation

Darinka Verdonik

Faculty of Electrical Engineering and Computer Science, University of Maribor
Smetanova ul. 17, 2000 Maribor, Slovenia
darinka.verdonik@uni-mb.si

Abstract

The aim of this paper is to discuss and specify some pragmatic language categories that could be used as attributes in spontaneous speech corpora, especially the corpora used for developing speech-to-speech translation systems components. When developing the speech-to-speech translation, researchers have to deal with spontaneous (conversational) speech phenomena like hesitations, turn-taking behaviors, self-repairs, false starts, filled pauses... This makes speech-to-speech translation a very hard task, with much space for improvement. Language technologies use linguistically annotated corpora and lexica (morphologic, syntactic, semantic...) to achieve better performance. In this paper I suggest to include pragmatic attributes of annotation to deal with some of the above mentioned phenomena of spontaneous speech.

Pragmatično označeni korpusi v strojnem simultanem prevajanju govora

Namen tega prispevka je definirati nekatere pragmatične jezikovne kategorije, ki jih lahko uporabimo kot atribute v pragmatično označenih govornih korpusih, zlasti tistih, ki se uporabljajo pri razvoju sistemov strojnega simultanega prevajanja govora. Raziskovalci, ki delajo na področju tehnologije strojnega simultanega prevajanja govora, opozarjajo, da je v pogovoru polno elementov, kot so obotavljanja, menjavanje vlog, samopopravljanja, napačni začetki, premori... Te značilnosti so problematične za strojno simultano prevajanje govora in zahtevajo ustrezne rešitve. Pri razvoju jezikovnih tehnologij se uporabljajo jezikoslovno označeni korpusi in slovarji (oblikoslovni, skladenjski, semantični...), saj pripomorejo k večji uspešnosti tehnologije. V tem prispevku predlagam vključevanje pragmatičnih atributov za označevanje govornih korpusov, da bi na tak način premoščali težave pri razvoju strojnega simultanega prevajanja govora, ki jih navajam zgoraj.

1. Introduction

Many projects developing speech-to-speech translation systems (eg. Verbmobil – http://verbmobil.dfki.de/, Janus – http://www.is.cs.cmu.edu/mie/janus.html, EuTrans – http://www.cordis.lu/esprit/src/30268.htm, Nespole! – http://nespole.itc.it/) had to face the reality of spontaneous (conversational) speech. It is usually observed that spontaneous speech includes »disfluencies, hesitations (um, hmm, etc.), repetitions« (Waibel, 1996), »pauses, hesitations, turn-taking behaviors, etc.« (Kuremtasu et al., 2000), »self-interruptions and self-repairs« (Tillmann, Tischer, 1995), disfluencies such as »a-grammatical phrases (repetitions, corrections, false starts), empty pauses, filled pauses, incroprehensible utterances, technical interruptions, and turn-takes« (Costantini et. al, 2002). Such characteristics can cause many problems for automatics speech recognition and speech centered translation, which are part of a speech-to-speech translation system.

In linguistics (I refer to linguistics not only as a study of language system, but also as a study of language use) most of the above mentioned characteristics are considered as pragmatic, and are the subject of interest in some fields of discourse analysis or pragmatics. In this paper I will try to specify some basic pragmatic attributes that cover some of these spontaneous speech characteristics and that could be easily annotated in spontaneous speech corpora. There have been few tries to annotate some pragmatic elements in speech corpora for use in developing speech technologies or natural language processing (eg. Heeman et al., 1998; Heeman, Allen, 1999; Miltsakaki et al., 2002), however pragmatics as level of annotation in language resources is far from being broadly discussed or accepted. The processing problems when dealing with spontaneous speech encourage us to try it and discuss it, and to encourage some further discussion on pragmatically annotated corpora is one of the aims of this paper.

The research presented in this paper is based on a corpus in the Slovenian language, therefore the attributes for annotation are defined for the Slovenian, but the presented concepts themselves are general. More details on all aspects of the research which is a basis for this discussion can be found in (Verdonik, 2006).

For the Slovenian language, speech-to-speech translation system recently became an interesting issue. (Žganec et al., 2005) present a design concept of the Voice TRAN, speech-to-speech translation system that would be able to translate simple domain-specific sentences in the Slovenian-English language pair. The other concept for the speech-to-speech translation system including the Slovenian language is named Babilon, and it is presented on the http://www.dsplab.uni-mb.si/Dsplab/Slo/Projects_slo_ demo.php.

The structure of this article is the following: first I describe the corpus (Turdis-1) that was used to track, analyze and specify the pragmatic attributes for annotation. Chapters 3, 4 and 5 bring specification of the three levels of pragmatic annotation in spontaneous speech corpora: conversation structure (sections, turns, utterances), discourse markers and repairs. In chapter 6 some conclusions are drawn.

2. Data for the analysis – the Turdis-1

For the analysis I used a speech corpus of telephone conversations in tourism. Tourist domain seems to be one of the most promising and popular for speech-to-speech translation systems (it was the main or one of the main domains for speech-to-speech translation projects like

Verbmobil, Janus, Nespole!, EuTrans...). Since the tourist domain in general is too broad as a domain of interest for typical speech-to-speech translation applications, it was further restricted to the following sub-domains:

- telephone conversations in tourist agency
- telephone conversations in tourist office
- telephone conversations in hotel reception

Conversations with professional tourist agents and real tourist organizations were recorded. The callers were contacted personally; they were mostly employees and students of the University of Maribor. The tourist organizations which participated in recording were: two local hotels, local tourist office and four local tourist agencies. All conversations were in the Slovenian language which was also the mother tongue of all the callers. Recorded material was transcribed using the Transcriber tool (http://trans.sourceforge.net/en/presentation.php). We considered some of the EAGLES recommendations (http://www.lc.cnr.it/EAGLES96/spokentx/) and principles of transcribing BNSI Broadcast News database (Žgank et al., 2004) when transcribing. More details about recording and transcribing can be found in (Verdonik, Rojc, 2006).

From the recorded material 30 conversations were selected for the present study. This selection is named Turdis-1. The total length of the recordings in the Turdis-1 is 106 minutes, the average length of a conversation 3,5 minutes, the number of tokens is 15,717, number of word forms 2735, number of utterances 2171. The table 1 shows more details about number and length of conversations, and the table 2 about number and gender of speakers.

Table 1: Number and total length of conversations in the Turdis-1 database.

	No. of conv.	**Total length**
Tourist agency	14	53,33 min.
Tourist office	8	28,1 min.
Hotel reception	8	24,38 min.
Total	**30**	**106,2 min.**

Table 2: Gender of the speakers (callers and tourist agents) in the Turdis-1 database.

	Male	**Female**
Tourist agents	3	17
Callers	14	10
Total	**17**	**27**

3. Conversation structure

When processing natural speech, we need to find the most appropriate segments for processing first. This is especially important when talk of one speaker is longer than what is usually understood as a segment (in speech technologies) or an utterance (in discourse analysis). So the basic units of transcribing conversations are usually turns and segments/utterances. Both need some further clarifications.

3.1. Turns

Turn is understood as the talk of one speaker before the next speaker starts to talk. But in natural conversation it often happens that at the exchange point talk of both speakers overlap (so called overlaping speech). When transcribing, different solutions are possible for overlapping speech. The one I suggest here is that we segment overlapping speech as a new, overlaping turn, but include special tags for tracking connections between the text in overlapping speech and the text in the previous or the following segments. This is because when we tag the overlapping speech as a special segment, we have probably put some borders to the text which are not consistent with prosodic, syntactic and semantic borders (i.e. utterances), therefore also the previous or/and the following segment may be syntactically, semantically and prosodically incomplete.

Another issue of discussion is how to transcribe backchannel signals (short expressions that hearer pronounces in order to confirm to the speaker that he is listening, that he understands, that he is interested...). I suggest not to annotate them as overlapping speech, but as special speech events.

3.2. Segments/utterances

Segments/utterances are usually the basic units for processing speech. In written text corresponding units could be sentences. It is quite clear what counts as a sentence in the written text, but there seems to be less agreement on what counts as an utterance in the spontaneous speech. For use in developing speech technologies, I believe syntactic, semantic and also prosodic features (especially intonation and pauses) must be considered when segmenting speech to utterances.

3.3. Sections

Sections can be as well an interesting attribute for annotating conversation structure. Here, I will consider only opening and closing sections in a conversation, which are very important for pragmatically successful conversation. It is open for a discussion, whether other topic shifts during the course of a conversation are to be annotated.

In opening and closing sections in the analyzed telephone conversations I find more or less standard pragmatic acts and standard phrases used. This can make speech-to-speech translation task easier.

In an opening section a caller starts communication by telephone ring. First talk in conversation is agent's, always introducing himself and/or organization he works at, very often also greeting. Next turn is caller's, he is always greeting, very often introducing himself, and after this explaining a reason for the call.

Closing sections are very delicate, because none of the participants in a conversation should feel forced to end the conversation. Analysis shows that discourse markers *dobro/v redu/okej/prav* (Eng. *good, alright, right, okay, well, just*) can be used as signals for closing the conversation. Next act is usually thanking, which is also a signal for closing the conversation. The last act of every conversation are greetings.

4. Discourse markers

Discourse markers are expressions like *oh, well, now, y'know, and...* In conversation, they are most often used the way that they do not contribute much to the propositional content, but have more or less pragmatic, communicative functions. As such I find them an interesting attribute for annotation.

Studies of discourse markers were increasing in the last decades, not only for English but for many languages worldwide (see for example special issues of Discourse Processes (1997, 24/1) and Journal of Pragmatics (1999, 31/10), workshops like Workshop on Discourse Markers (Egmond aan Zee, Nederlands, January 1995) or COLING-ACL Workshop on Discourse Relations and Discourse Markers (Montreal, Canada, August 1998), books like (Schiffrin, 1987; Jucker, Ziv, 1998; Blakemore, 2002) etc.).

There are basically three different approaches to discourse markers: coherence-based (most known is Schiffrin's research (1987)), relevance theory approach (very known is work of Blakemore (1992; 2002)) and grammatical-pragmatic approach (Fraser, 1990; 1996; 1999).

For the Slovenian language there are only few researches of what I here name discourse markers, some more some less close to the discursive perspective: (Gorjanc, 1998), (Schlamberger Brezar, 1998), (Smolej, 2004a). (Pisanski, 2002; 2005) represents broader research on text-organizing metatext in research articles.

4.1. Guidelines for annotating discourse markers

When overviewing the researches on discourse markers, we find out that there is still no agreement on what counts as a discourse marker. But what we find common is acknowledgement that there are two basically different kinds of meaning, communicated by utterances: Schiffrin (1987) distinguishes ideational plane on the one hand, and exchange structure, action structure, participation framework and information state on the other hand; Blakemore (2002) distinguishes conceptual vs. procedural meaning; Fraser (1996) distinguishes propositional content and pragmatic information; researches on metadiscourse (eg. Pisanski, 2002; 2005) distinguish metadiscourse and propositional content. Even though these distinctions are not completely parallel, they have a lot in common. Discourse markers in these distinctions are expressions that function primarily pragmatically and contribute the least to the ideational/propositional/conceptual domain.

As one of the most extensive, detailed and also most often cited studies of discourse markers, based on recorded material of natural conversations, I take work of Schiffrin (1987) as the example. I keep the distinction between ideational structure and all the other planes of talk. Similar distinction is set by Redeker (1990), who distinguishes markers of ideational structure and markers of pragmatic structure. Since we are interested in expressions that function primarily pragmatically and contribute the least to the ideational/propositional/conceptual domain, the aim was to annotate discourse markers that function primarily as pragmatic markers.

According to this basic theoretical framework I annotate discourse markers in the Turdis-1 corpus and make a detailed analysis of annotated expressions in order to define their pragmatic functions in a conversation, to confirm or reject the chosen expressions, and to point to problematic points in annotating discourse markers.

4.2. Expressions functioning as discourse markers

According to the framework for annotating, defined in previous chapter, I annotated the expressions that contribute the least to the propositional content of an utterance in the Turdis-1 corpus. Such expressions were: *ja* (Eng. *yes, yeah, yea, well, I see* – please notice that the English expressions are only approximate description to help readers who do not speak the Slovenian language; it is based on the author's knowledge of English, Slovenian-English dictionary and British National Corpus (http://www.natcorp.ox.ac.uk/); usage of discourse markers is culturally specific and we would need a comparative study to be able to specify the English equivalents more exactly), *mhm* (Eng. *mhm*), *aha* (Eng. *I see, oh*), *aja* (Eng. *I see, oh*), *ne?/a ne?/ali ne?/jel?* (no close equivalent in English, a bit similar to *right?, isn't it?* etc.), *no* (Eng. *well*), *eee/mmm/eeem...* (Eng. *um, uh, uhm*), *dobro/v redu/okej/prav* (Eng. *good, alright, right, okay, well, just*), *glejte/poglejte* (Eng. *look*), *veste/a veste* (Eng. *y'know*), *mislim* (Eng. *I mean*), *zdaj* (Eng. *now*), and backchannel signals: *mhm* (Eng. *mhm*), *aha* (Eng. *I see, oh*), *ja* (Eng. *yes, yeah, yea, I see*), *aja* (Eng. *I see, oh*), *dobro* (Eng. *okay, alright, right*), *okej* (Eng. *okay, alright, right*), *tako* (Eng. *thus*), *tudi* (Eng. *also*), *seveda* (Eng. *of course*). I use the term backchannel signals for isolated uses of discourse markers when hearer does not take over the turn and also does not show intention to do so, but merely expresses his attention, agreement, confirmation, understanding etc. of what speaker is saying.

The results of the analysis showed that some of these expressions always function as discourse markers: such are *mhm* (Eng. *mhm*), *aha* (Eng. *I see, oh*), *aja* (Eng. *I see, oh*), *no* (Eng. *well*), *eee/mmm/eeem...* (Eng. *um, uh, uhm*).

Others (eg. *(a/ali) ne?* (in Eng. similar *right?, isn't it?* etc.), *dobro/v redu/okej/prav* (Eng. *good, alright, right, okay, well, just*), *glejte/poglejte* (Eng. *look*), *veste/a veste* (Eng. *y'know*), *mislim* (Eng. *I mean*)), can function either as a discourse marker, for example *dobro* as discourse marker:

*K25: **dobro** gospa najlepša hvala da ste se tako potrudli ne? / **okay** madam thank you so much for your efforts*

or as an important element of propositional content, for example *dobro* in a proposition:

*K39: ker[+SOGOVORNIK_ja] jim nikol nič ni **dobr** in vedno* etc. */ because[+OVERLAP_yes] nothing is ever **good** enough for them and they always* etc.

but differences between both usages are easy to recognize for a human annotator. For automatic detection it may be helpful, that (according to the analysis of the Turdis-1 corpus) the analyzed expressions in the function of discourse marker are usually positioned at the borders between utterances.

But for some of the analyzed expressions, particularly *ja* (Eng. *yes, yeah, yea, well, I see*) and *zdaj* (Eng. *now*), the border between discourse marker and propositional function was blurred. There were usages where these expressions were functioning clearly pragmatically, other usages where they were functioning clearly as part of a proposition, but also usages where it was not clear which of these two basic functions was more important, for example:

K39: eeem treh ali pa štirih Nemcev to zaenkrat še ne vem s() se pravi oni[+SOGOVORNIK_mhm] so pač iz Nemčije[+SOGOVORNIK_mhm] / um three or four German people this I do not know exactly s() so they[+OVERLAP_mhm] are from Germany[+OVERLAP_mhm]
K39: #nikol# še niso bli v Sloveniji / they have #never# been to Slovenia
*K39: in **zdej** bi jih ze() pač za takšne štir pet dni počitnic ki jih bojo meli v Sloveniji bi jim pač seveda* etc. */ and **now** I would f() for some four five days of vacation they will have in Slovenia I would of course* etc.

Such examples confirm that the border between pragmatic and semantic level is certainly not a clear cut, and annotating in corpora needs careful considerations on every step.

The above mentioned expressions are of course not all discourse markers of the Slovenian language. But the outlined considerations may be the starting point for further discussion about discourse markers. In the Turdis-1 corpus, discourse markers were manually annotated, but the analysis showed that further annotation can be at least partially automatic.

4.3. Pragmatic functions of the analyzed discourse markers

Since the analyzed expressions do not contribute much to the content of a message, we can suppose that they have some pragmatic functions. This suggestion is supported by the fact that the analyzed discourse markers were used more than 2000 times in 15,000 tokens corpus, what corresponds to something more than 13% of all tokens, and that is quite a lot. I used the conversational analysis method (see Levinson, 1983, 286-287), and as the results of the analysis I specified the following pragmatic functions of discourse markers:

- signaling connections to propositional content (backward or forward)
- building relationships between participants in conversation (for example checking and confirming a hearer's presence, interest in conversation, understanding...)
- expressing speaker's attitude to the content of the conversation (eg. surprise, dissatisfaction...)
- organizing the course of conversation (signals in turn-taking system, signals for changing the topic and ending a conversation, signals of disturbances (eg. self-repairs) in utterance structure/production)

5. Self-repairs

As I pointed out in the introduction, spontaneous speech characteristics like disfluencies, self-interruptions and self-repairs, corrections, false starts etc. are problematic for spontaneous speech processing. In pragmatics most of these phenomena are treated as disfluencies or as self-repairs. In the Slovenian language the phenomena did not draw special attention before this research, it was merely noticed for example in (Smolej, 2004b; Krajnc, 2004).

Some of the most cited and known researches on self-repairs were done by (Schegloff, Jefferson, Sacks, 1977; Schegloff, 1979), by (Levelt, 1983), also (Allwood et al., 1990) etc. Disfluencies were studied for example by (Lickley, 1994; Shriberg, 1994; Tseng, 1999). They consider the term more neutral, but it includes broader phenomena (for example for Shriberg (1994) disfluencies are *um*'s, repetitions and self-repairs, for Tseng (1999) restarts, repetitions, pauses, speech errors, speech repairs). Here, based on pragmatic researches of the phenomena, I discuss only self-repairs. I try to define them the way that we can use a definition of the self-repair to annotate the part of an utterance that needs to be eliminated in further processing because it is unfinished structure, replaced by another structure.

I suggested to annotate segment/utterance the way that it can be treated as a basic unit for processing, and I want to define the self-repair the way that it is a structure that needs to be eliminated, therefore I define the self-repair as a phenomenon on the level of a segment/utterance.

5.1. Defining self-repairs

(Blanche-Benveniste, 1991; Smolej, 2004b in the Slovenian linguistics) discuss two levels or axes of producing a text: syntagmatic (horizontal) and paradigmatic (vertical). In the eyes of this theory a self-repair is a structure, where speaker does not continue fluent speech, but stops and goes back to some previous point on syntagmatic level of text, for example:

*kolko pa potem stane nočitev pa recimo **da so** eee*
***da je** poln penzijon /*
*and how much then costs one night for example **that we** um*
__that it is__ with
breakfast

But when listing, explaining, inserting structures etc. speaker also goes back to some previous point on syntagmatic level of text, for example when explaining:

*študenti organiziramo en tak letni **sestanek** oziroma*
__srečanje /__
*the students we organize some sort of annual **meeting** or*
__gathering__

A typical self-repair as I want to define it here always begins by cut-off, therefore I do not define examples as the last one as a self-repair.

Next, I analyze pragmatic aspects of self-repairs. First I try to define reasons for cutting-off. I find that they may be circumstantial (bad telephone connection), social (especially turn-taking), or psychological (a speaker needs more time to prepare what he will say, a speaker changes his strategy how to say something, a speaker notices a mistake in what he told, a speaker has problems when pronouncing and re-pronounces some previous element(s)). It is only when a speaker changes his strategy, when he notices a mistake or has problems when pronouncing, that we can talk about self-repair. At the same time the first condition has to be fulfilled, i.e. a

speaker goes back to some previous point on syntagmatic level of text.

According to this definition I annotate self-repairs in the Turdis-1 corpus. They appear in 185 utterances, which is approx. in 8% of all the utterances.

5.2. Structure of self-repairs

I find four basic structure elements of self-repairs:

1. A part of a text that will be corrected, therefore it should be eliminated in automatic processing. In 90% of examples in the Turdis-1 corpus it is not longer than 3 words.
2. A cut-off.
3. Self-repair signals: metadiscoursive element(s) can follow right after cut-off, for example discourse markers *eee* (Eng. *um*), *zdaj* (Eng. *now*), *mislim* (Eng. *I mean*) etc., pause, pro-longed vowel etc. But these are used only in 55% of all self-repairs in the Turdis-1 corpus.
4. Repairing element/s, i.e. the new text that replaces the part of a text that was corrected. In 65% in the Turdis-1 corpus repaired elements include repetition of at least one token or some phonemes of the cut-off token from the part of a text that was corrected.

6. Conclusion

In this paper I have discussed the idea to include pragmatic tags to spontaneous speech corpora used for developing speech-to-speech translation components (and of course for other speech technologies, dealing with spontaneous speech, for example dialog systems). Based on the analysis of the corpus (Turdis-1) of telephone conversations I tried to define three basic levels of annotation.

Annotating basic conversation structure elements – segments/utterances, turns, sections – is usual in conversation corpora. In this paper I point to some problematic points of annotation: annotating overlapping speech and backchannel signals, defining utterances to achieve consistency of annotation, annotating opening and closing sections which include mostly standard pragmatic acts and phrases.

Next, I suggest annotating discourse markers. Discourse markers attracted much attention of linguists, but annotating discourse in speech corpora used for developing speech technologies is not broadly accepted yet, even though there are/were some tries. Overview of the researches of discourse markers in discourse analysis shows that there is no agreement on what counts as discourse marker. Therefore I try to specify a framework for annotation that would be the most useful for speech-to-speech translation purposes. As discourse markers, I specify the expressions that contribute the least to the propositional content of an utterance, but have mostly pragmatic functions. The analysis shows that most of them are used at the borders between utterances, so they can be used to help segmenting spoken text to segments/utterances. They are very frequently used in a conversation – more than 13% of all the words in the Turdis-1 corpus. This supports the idea that discourse markers are very important elements of natural conversation.

Last I try to define self-repairs the way that self-repair as attribute in speech corpora annotates a part of spoken text that needs to be eliminated in further processing – it is unfinished structure, replaced by some other structure. I conclude that self-repairs are an event where a speaker goes back to some previous point on syntagmatic level of text, in order to change a strategy, correct a mistake or repair problems when pronouncing. Self-repairs are present in approx. 8% of all the utterances in the Turdis-1 corpus.

Possibilities for further annotation of pragmatic elements in spontaneous speech corpora are many more, for example speech acts, adjacency pairs, other metatextual elements, repetitions etc. There is a wide area for researches, experiments and discussion.

7. Acknowledgements

I sincerely thank to all the tourist companies that participated in recording: the tourist agencies **Sonček, Kompas, Neckermann Reisen** and **Aritours**, to the **Terme Maribor**, especially the **Hotel Piramida** and the **Hotel Habakuk,** and to the **Mariborski zavod za turizem** with the tourist office **MATIC**. I also thank to all the tourist agents in these companies who participated in recording and to all the callers who were ready to use the Turdis system.

8. References

Allwood, J., J. Nivre, E. Ahlsen. 1990. Speech management: On the non-written life of speech. *Nordic Journal of Linguistics,* 13/1.

Blakemore, Diane. 1992. *Understanding utterances*. Oxford, Cambridge: Blackwell Publishers.

Blakemore, Diane. 2002. *Relevance and Linguistic Meaning: The Semantics and Pragmatics of Discourse Markers*. Cambridge: Cambridge University Press.

Blanche-Benveniste, Claire. 1991. *Le francais parle*. Etudes grammaticales. Paris: CNRS.

Constantini, E., S. Burger, F. Pianesi. 2002. NESPOLE!'s multilingual and multimodal corpus. In proceedings of the 3rd International Conference on Language Resources and Evaluation 2002, LREC 2002, Las Palmas, Spain.

Fraser, Bruce. 1990. An approach to discourse markers. *Journal of Pragmatics,* 14, 383-395.

Fraser, Bruce. 1996. Pragmatic markers. *Pragmatics,* 6/2, 167-190.

Fraser, Bruce. 1999. What are discourse markers? *Journal of Pragmatics,* 31, 931-952.

Gorjanc, V. 1998. Konektorji v slovničnem opisu znanstvenega besedila. Slavistična revija, XLVI/4, 367–388.

Heeman, Peter, Donna Byron, James Allen. 1998. Identifying Discourse Markers in Spoken Dialogue. In *Working Notes of AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, Stanford, CA.

Heeman, Peter, James Allen. 1999. Speech repairs, intonational phrases and discourse markers: modeling speakers' utterances in spoken dialog. *Computational Linguistics*, 25(4).

Jucker, Andreas H., Yael Ziv (Eds.). 1998. *Discourse Markers: Descriptions and Theory.* Amsterdam: John Benjamins.

Krajnc, M. 2004. Besediloskladenjske značilnosti javne govorjene besede (na gradivu mariborščine). *Slavistična revija,* 52/4, 475-498.

Kurematsu, A., Akegami, Y., Burger, S., Jekat, S., Lause, B., MacLaren, V., Oppermann, D., Schultz, T. 2000. Verbmobil Dialogues: Multifaced Analysis. In Proceedings of the International Conference of Spoken Language Processing.

Levelt, W. J. M. 1983. Monitoring and self-repair in speech. *Cognition,* 14, 41-104.

Levinson, Stephen. 1983. *Pragmatics*. Cambridge: Cambridge University Press.

Lickley, Robin J. 1994. *Detecting disfluency in spontaneous speech*. PHD thesis. University of Edinburgh.

Miltsakaki, E., R. Prasad, A. Joshi, B. Webber. 2004. The Penn Discourse Treebank. In Proceedings of the the Language Resources and Evaluation Conference'04, Lisbon, Portugal.

Pisanski, Agnes. 2002. Analiza nekaterih metabesedilnih elementov v slovenskih znanstvenih člankih v dveh časovnih obdobjih. *Slavistična revija,* 50/2, 183-197.

Pisanski Peterlin, Agnes. 2005. Text-organising metatext in research articles: an English-Slovene contrastive analysis. *Engl. specif. purp. (N.Y. N.Y.),* 24/3, 307-319.

Redeker, G. 1990. Ideational and pragmatic markers of discourse structure. *Journal of Pragmatics,* 14, 367-381.

Schegloff, E., G. Jefferson, H. Sacks. 1977. The preference for self-corection in the organization of repair in conversation. *Language,* 53/2, 361-382.

Schegloff, E. 1979. The relevance of repair to syntax-for-conversation. In Givon, T. (ed.). *Syntax and Semantics 12, Discourse and Syntax*. New York: Academic Press. 261-286.

Schiffrin, Deborah. 1987. *Discourse Markers.* Cambridge: Cambridge University Press.

Schlamberger Brezar, M. 1998. Vloga povezovalcev v diskurzu. In *Jezik za danes in jutri*. Ljubljana: Društvo za uporabno jezikoslovje Slovenije. 194-202.

Shriberg, E. E. 1994. *Preliminaries to theory of speech disfluencies*. PHD thesis. University of California at Berkeley.

Smolej, Mojca. 2004a. Členki kot besedilni povezovalci. *Jezik in slovstvo,* 49/5, 45-57.

Smolej, Mojca. 2004b. Načini tvorjenja govorjenega diskurza – paradigmatska in sintagmatska os. In Erika Kržišnik (ed.). *Aktualizacija jezikovnozvrstne teorije na Slovenskem: členitev jezikovne resničnosti* (Obdobja, Metode in zvrsti, 22). Ljubljana: Center za slovenščino kot drugi/tuji jezik.

Tillmann, Hans G., Bernd Tischer. 1995. Collection and exploitation of spontaneous speech produced in negotiation dialogues. In proceedings of the ESCA Workshop on Spoken Language Systems, 217-220, Vigsø.

Tseng, Shu-Chuan. 1999. Grammar, prosody and speech disfluencies in spoken dialogues. PHD thesis. University of Bielefeld.

Verdonik, Darinka, Matej Rojc. 2006. Are you ready for a call? – Spontaneous conversations in tourism for speech-to-speech translation systems. In proceedings of the 5th International Conference on Language Resources and Evaluation, Genoa, Italy.

Waibel, Alex. 1996. Interactive translation of conversational speech. *IEEE Computer,* 29/7, 41-48.

Žganec Gros, J., F. Mihelič, T. Erjavec, Š. Vintar. 2005. The VoiceTRAN Speech-to-Speech Communicator. In Proc. ef the 8th Intl. Conf. on Text, Speech and Dialogue, TDS 2005. Czeck Republic, Karlovy Vary.

Žgank, A., T. Rotovnik, M. Sepesy Maučec, D. Verdonik, J. Kitak, D. Vlaj, V. Hozjan, Z. Kačič, B. Horvat. Acquisition and Annotation of Slovenian Broadcast News Database. 2004. In Proceedings of the 4th International Conference on Language Resources and Evaluation. Lisbon, Portugal.

Studying the Learning Curves of a Statistical Dependency Parser for Four Languages

Atanas Chanev

Department of Cognitive Sciences
University of Trento
ITC-irst Povo-Trento
via Matteo del Ben 5, 38068 Rovereto, Italia
chanev@form.unitn.it

Abstract

Multilingual dependency parsing is gaining popularity in recent years for several reasons. Dependency structures are more adequate for languages with freer word order than the traditional constituency notion. There is a growing availability of dependency treebanks for new languages. Broad coverage statistical dependency parsers are available and easily portable to new languages. Dependency parsing can provide useful contributions in areas such as information extraction, machine translation and question answering, among others. In addition, syntactic head-dependent pairs are a good interface between the traditional phrase structures and semantic theta roles. In this paper we present the learning curves of a statistical dependency parser for four languages: Arabic, Bulgarian, Italian and Slovene. We discuss issues that mostly concern the employed annotation scheme for each treebank with an emphasis on coordinated structures. We also investigate how these issues are related to the learning curve for each language.

Preučevanje krivulje učenja statističnega odvisnostnega razčlenjevalnika za štiri jezike

Večjezično odvisnostno skladenjsko razčlenjevanje postaja v zadnjih letih vse bolj privlačno zaradi vrste razlogov. Odvisnostne strukture so za jezike s prostejšim besednim redom primernejše kot pa tradicionalne, ki temeljijo na konstituentih, poleg tega pa je na voljo vse več odvisnostnih drevesnic za nove jezike. Statistični odvisnostni razčlenjevalniki s širokim pokritjem so dostopni in lahko prenosljivi na nove jezike. Odvisnostno razčlenjevanje je lahko koristen prispevek področjem, kot so luščenje podatkov, strojno prevajanje in sistemi za odgovarjanje na vprašanja. Poleg tega so skladenjski pari jedro-odvisnica dobri vmesniki med tradicionalno frazno strukturo in pomenskimi vlogami. V članku predstavimo krivulje učenja statističnega odvisnostnega razčlenjevalnika za štiri jezike: arabskega, bolgarskega, italijanskega in slovenskega. Razpravljamo o vprašanjih, ki se dotikajo predvsem uporabe označevalne sheme za vsako drevesnico s poudarkom na zgradbi priredij. Preučimo tudi, kako so ta vprašanja povezana s krivuljo učenja za vsakega od jezikov.

1. Introduction

Contrary to a constituency (or phrase structure) grammar, a dependency grammar (e.g. (Mel'čuk, 1988)) does not view syntactic structures as nested sets of constituents but as a set of binary head-dependent relations. In most dependency grammar formalisms there are several restrictions for the dependency relations: They should build up a connected acyclic graph; For each dependent there should be only one head; There should be a single word in the sentence without a head – the root word. A syntactic label, such as subject, object etc. is usually associated with each relation in the graph.

Projectivity is another issue that is often considered as a constraint to dependency graphs. A simple non-formal definition for projectivity of a connected dependency graph is: if one connects the root word of a sentence with an artificial root placed before the first word, there should not be crossing dependency arcs. While most of the dependency parsers can parse only projective structures, the need for non-projective relations is recognised in nearly all dependency treebank annotation schemes.

State-of-the art statistical dependency parsers have been evaluated on 13 different treebanks (for 13 different languages) at the CoNLL-X shared task on statistical dependency parsing (Buchholz and Marsi, 2006)[1]. While the treebanks had been parsed with many parsers, all the parsers had been an implementation of a limited number of parsing models.

A good multilingual dependency parser should be robust enough so that its accuracy would not decrease when it is ported to another language / another treebank. In practice this is a rarely observed quality, especially when common treebank annotation schemes are based on considerably different linguistic assumptions. The multilingual dependency parsing task is evaluation of the ability of parsers to be ported easily to new languages. But it is also evaluation of the eligibility of treebank annotation schemes to encode linguistic phenomena so that the treebanks can be easily parsed using a statistical dependency parser.

A new direction in designing parsers is using evidence from Psycholinguistics. (Hale, 2001) and (Lesmo et al., 2002), for example, use a psychologically motivated tree pruning and implement incremental processing strategies in their parsers. Incrementality is also addressed in (Nivre, 2004).

This paper gives the learning curves of a statistical dependency parser – the Malt parser (Nivre et al., 2006), for four languages: Arabic, Bulgarian, Italian and Slovene.

[1] http://nextens.uvt.nl/∼conll/

The treebanks for these languages had been annotated by different research groups, using four different annotation schemes. The parser that we use has a high attachment score (accuracy), it is robust and has a number of features that are psychologically plausible.

The paper is structured as follows: Section 2. gives an overview of the results achieved at the CoNLL-X shared task on dependency parsing (Buchholz and Marsi, 2006) and motivation to use the Malt parser in our experiments. Then, in Section 3. we briefly describe the annotation scheme of each treebank that we give learning curves of. We give a short description of the Malt parser and the parsing feature model that we used in our experiments in Section 4. The learning curves are given and discussed in Section 5. We conclude in Section 6.

2. Statistical Dependency Parsing

The parsers from the CoNLL-X shared task usually implemented two parsing models. In one of them the correct dependency graph was searched for as the maximum spanning tree in a full graph with removed arcs that violate the constraints for a dependency graph (e.g. (McDonald et al., 2006)). Parsers of this kind were able to parse non-projective graphs. However, such parsers are not able to assign correct labels to dependency relations during processing. They do it in a following step.

The other approach is an implementation of the shift-reduce parser (Yamada and Matsumoto, 2003), extended as in e.g. (Nivre, 2005). In this model the dependency graph is built in incremental fashion using a stack for storing the words of the sentence and four actions: shift, reduce, left-arc and right-arc. The parser cannot parse non-projective arcs (except e.g. the implementation described in (Attardi, 2006)) but they can be parsed using a technique known as pseudo-projective parsing (Nivre and Nilsson, 2005).

The difference between the accuracy of the two best parsers: (McDonald et al., 2006) that implements the maximum spanning tree approach and (Nivre et al., 2006) that implements the shift-reduce algorithm is not statistically significant (Buchholz and Marsi, 2006). But (Nivre et al., 2006) is more interesting to us because of its psychological plausibility and the fact that any kind of information can be included directly in feature models for learning.

3. Treebanks

We used four treebanks in our experiments: The Prague Arabic Dependency Treebank (PADT) (Hajič et al., 2004), the BulTreeBank (BTB) (Simov et al., 2005), the Turin University Treebank (TUT) (Bosco, 2004) and the Slovene Dependency Treebank (SDT) (Džeroski et al., 2006). Their annotation schemes are different (with PADT and SDT annotation schemes being quite similar). PADT, TUT and SDT are original dependency treebanks while BTB was converted from Head-driven Phrase Structure Grammar (HPSG) format to dependency graphs in (Chanev et al., 2006). We give short descriptions of the treebanks below and summarize their features in Table 1.

Languages:	Ar	Bg	It	Sl
Tokens	59,752	196,151	44,616	35,140
Sentences	1,606	13,221	1,500	1,936
T. per sen.	37.2	14.8	27.7	18.2
PoS set	21	570	90	30
Dep. set	27	20	18	26
Randomized	no	no	yes	no
DG	yes	no	yes	yes

Table 1: Treebank properties. (DG = Dependency Grammar)

3.1. The Prague Arabic Dependency Treebank

We used the CoNLL-X shared task version of the PADT[2] which slightly differs from the original treebank. It is separated in training (1,460 sentences; 54,379 tokens) and test (146 sentences; 5,373 tokens) set. The number of part-of-speech tags and the number of dependency tags are respectively 21 and 27. The average number of tokens per sentence is 37.2. The PADT annotation scheme is closely related to the one of the Prague Dependency Treebank (PDT) (Hajič, 1998).

One of the idiosyncrasies of the PDT annotation scheme is the fact that the root of a sentence is not an ordinary word but an artificial token whose position is e.g. before the first word of the sentence. What should have been the root of a sentence (i.e. the only word that does not have a head) points to the artificial root together with end-of-sentence punctuation. As the artificial root is not included in the CoNLL-X data, one has to learn and parse sentences which have more than one root and their dependency graphs are not connected.

Another idiosyncrasy is the treatment of coordinated structures. In PDT-related annotation schemes the coordinating conjunction (or punctuation) is chosen to be the head of the coordinated words.

3.2. The BulTreeBank

BulTreeBank is an HPSG-based treebank but head-dependent relations between words are not stated explicitly. It has been converted to dependency graph representations in (Chanev et al., 2006). We use the CoNLL-X shared task dependency version of the BTB for our results to be comparable to those from the CoNLL-X shared task.

The BulTreeBank is separated in training (10,911 sentences; 159,395 tokens) and test (2,310 sentences; 36,756 tokens) set. The average number of words per sentence is 14.8. The number of part-of-speech labels is 570[3] and the number of dependency labels is 20.

Contrary to the PADT approach all the graphs in the BulTreeBank have one root per graph. Coordinated structures are annotated differently than those in the PADT. In the BTB encoding the first coordinated word is annotated as the head of the coordinating conjunction (or punctuation) and as the head of the second coordinated word.

[2]PADT is distributed by the Linguistic Data Consortium: http://www.ldc.upenn.edu/

[3]We used the original BTB part-of-speech tags.

3.3. The Turin University Treebank

The TUT was not included in the CoNLL-X shared task mainly because of its limited size – 1,500 sentences (44,616 tokens). The average number of tokens per sentence is 27.7. Although the treebank is small and n-fold cross-validation is usually used in such cases, here we report results on a test set of 150 sentences (4,172 tokens) and a training set of 1,350 sentences (37,444 tokens) in order the TUT experiment not to differ from the experiments on the other treebanks in this study.

We used a version of the TUT with removed traces and reduced tag sets (Chanev, 2005). The reduced tag sets comprised 90 part-of-speech tags and 18 dependency tags. Italian dependency tags are semantically 'deeper' than those from the other treebanks in this study. All the graphs in the treebank are connected and have only one root per graph. Coordination is annotated with the coordinating conjunction (or punctuation) being head of the second coordinated word and dependent on the first coordinated word. A single coordination dependency tag is used for both of the dependency arcs[4].

3.4. The Slovene Dependency Treebank

SDT has an annotation scheme which is similar to those of the PDT and PADT. We used the CoNLL-X version of the treebank for our results to be comparable with those from the shared task. The data is divided in a training set (1,534 sentences, 28,750 words) and a test set (402 sentences, 6,390 words). The average number of tokens per sentence is 18.2. The number of the part-of-speech tags used in the annotation of SDT is 30. The number of dependency labels is 26. Like in PADT, sentences can have more than one root and coordinated structures are treated with the coordinating conjunction (or punctuation) as the head of the coordinated words.

4. The Parser

We used version 0.4 of the Malt parser[5]. It is related to the shift-reduce dependency parser described in (Yamada and Matsumoto, 2003). There are two different parsing algorithms: arc-eager and arc-standard. In all our experiments we used the arc-eager parsing algorithm because it is more accurate than the arc-standard algorithm[6].

Malt parser does not use an explicit probabilistic grammar but implements a data-driven parsing approach. What is learned is the actions that the shift-reduce parser must take in order to build the dependency graph of the sentence. Two learners are available for that task: Memory-Based Learning (MBL) (Daelemans and den Bosch, 2005) and Support Vector Machines (SVM) (Chang and Lin, 2005). PoS tags, words as well as dependency labels which have already been assigned by the parser on the run can be used in feature models for learning.

In all the experiments we used the SVM learner. We also employed a common feature model – m7 that has proven to outperform the m3 and m4 models. It consists of six part-of-speech features, four dependency features and four lexical features. More information about the parser and feature models can be found in (Nivre, 2005) as well as on the Malt parser web page. The Malt parser team reported the second best result at the CoNLL-X shared task (Nivre et al., 2006) (the difference from the best result is not statistically significant).

[4]This approach differs from the one implemented in the original TUT.

[5]http://w3.msi.vxu.se/∼nivre/research/MaltParser.html

[6]This issue was discussed with Joakim Nivre in personal communication.

5. Results

In this section we list related work, describe preliminary settings, present and discuss the learning curves for Arabic, Bulgarian, Italian and Slovene.

5.1. Previous Studies

Even though constituency parsing is undoubtedly related to dependency parsing, in this section we give only dependency parsing results because they are immediately relevant to the study.

5.1.1. Arabic

The PADT has been learned and parsed by various teams at the CoNLL-X shared task on dependency parsing. Results vary from 50.7% to 66.9% labelled accuracy (Buchholz and Marsi, 2006).

5.1.2. Bulgarian

A dependency version of the BulTreeBank has also been used at the CoNLL-X shared task. Labelled accuracy is within the range 67.6% – 87.6%. Labelled accuracy of 79.5% was reported for another conversion of the original HPSG-based BulTreeBank but those results did not differ significantly from the results reported on the CoNLL-X conversion using the same parser and feature model (79.2%) (Chanev et al., 2006).

5.1.3. Italian

There are not many studies on statistical dependency parsing of Italian mainly because there are not large enough resources to train a parser. We will compare the learning curve for Italian with (Chanev, 2005) where the Malt parser was used together with the MBL learner. The reported accuracy is 81.8%. (Lesmo et al., 2002) describe a rule-based dependency parser for Italian. Even though its evaluation is only partial, accuracy is comparable to (Chanev, 2005).

5.1.4. Slovene

Slovene, like Arabic and Bulgarian, was one of the languages for the CoNLL-X shared task. Results for Slovene varied from 50.7% to 73.4% labelled accuracy (Buchholz and Marsi, 2006). Parsing Slovene was also mentioned in (Chanev, 2005) where the Malt parser with the MBL learner was trained on an old and very small version of the SDT. Labelled accuracy of 58.3% was reported.

5.2. Settings

All the experiments were performed on training / test sets with gold standard PoS tags. The same feature model and the same learning and parsing settings were used in all the tests with the exception of an option that allowed many roots in a sentence that was used only for the Arabic and

Slovene treebanks. The measure that we use is labelled attachment score (labelled accuracy) measured excluding punctuation. We chose this measure for comparison reasons, since it is the measure used in the evaluation of the parsers at the CoNLL-X shared task. However, we also report unlabelled attachment score (unlabelled accuracy) for the biggest data sets for all the treebanks. For a definition of these measures, the reader is referred to (Lin, 1998).

The BulTreeBank learning curve is set for training sets that start from 1,000 sentences and increase up to the full size of the treebank, where at each step the size of the training set is increased by 1,000 sentences. The learning curves for the other languages start from a training set of 600 sentences and the sizes continue to grow up to the full number of sentences of the treebanks with increase of 200 sentences at each step. The sentences from the Italian treebank were randomized. However, the other treebanks were not, due to CoNLL-X shared task compatibility reasons.

Two additional learning curves are included for Arabic and Slovene after a simple graph transformation on the coordinated structures was applied on the training sets for these languages. Parsing output was then converted back to the original coordination encoding and evaluated on the gold standard PADT and SDT. These learning curves are shown with squares on the graphics for Arabic and Slovene on Figure 1.

A description of the coordination transformation procedure follows:

Coordinated structures are identified by the dependency label of the coordinating conjunction (or punctuation) which, according to the PDT annotation scheme, is the head of the coordinated words. If there are two words with the same dependency labels among the dependents, one of them being before the head and the other – after the head, then they are recognised as coordinated. Then the first coordinated word takes the head word of the coordinating conjunction (punctuation) and the coordinating conjunction or punctuation is made to point to the first coordinated word.

The inverted transformation is performed in a similar way. After the coordinated structure is identified, the head of the first coordinated word is transferred to be the head of the coordinating conjunction (or punctuation) and the first coordinated word is made dependent on the coordinating conjunction (or punctuation). Note that the back transformation can be accurate only for properly parsed coordinated structures. It is important that coordinated words have correct labels, otherwise a coordinated structure cannot be easily identified for inverted transformation.

5.3. Learning Curves

The learning curves are given in Figure 1. X-axis in the graphics shows the number of sentences used for training. The measure on the y-axis of the graphics is labelled accuracy. Results reported on data sets with transformed coordination structures for the Arabic and Slovene treebanks are given with squares. The best results are achieved for the biggest training data as shown in Table 2.

For training data of 1,000 sentences labelled accuracies for Bulgarian, Slovene and Arabic are similar. Labelled accuracy for Italian is the best for this size of training data.

Languages:	Ar	Bg	It	Sl
AS_L	*67.4%	81.8%	83.7%	*68.2%
AS_U	*78.0%	86.8%	88.6%	*77.6%

Table 2: Best results for Arabic, Bulgarian, Italian and Slovene. AS_L = labelled attachment score; AS_U = unlabelled attachment score; * = Coordination transformation applied.

If the comparison is done using the unlabelled accuracy measure, the per cent for Bulgarian is lower than those for Arabic and Slovene due to the bigger difference between labelled and unlabelled accuracy for PADT and SDT. Unlabelled accuracy is on the average 11% higher than labelled accuracy for Arabic, 5% for Bulgarian and Italian and 10% for Slovene.

There are a number of reasons for differences in accuracy for the different treebanks, from numbers of tokens per sentence for each treebank to sizes of the tag sets and idiosyncrasies of the annotation schemes. For example, the small number of part-of-speech tags for the Arabic and Slovene treebanks might have been the reason for the lower accuracy, in comparison with the bigger number of PoS tags for the Italian treebank, given that the number of dependency tags is similar in all the three treebanks. In fact, this is not the case. We did an additional experiment on the Italian data. We used a PoS set of only 17 coarse grained tags and labelled accuracy was still above 81%.

The Arabic and Slovene data sets had their transformed versions learned and parsed better than the original ones. The difference is over 1% for nearly all the sets. The biggest training set gives worse results than the second biggest for the transformed Slovene staining data. This is due to loss of accuracy in the inverted transformation.

There are two other factors which are relevant to parsing accuracy. The number of non-projective sentences in each treebank is the first factor. Non-projective arcs cannot be parsed correctly using the Malt parser without employing the pseudo-projective technique described in (Nivre and Nilsson, 2005). The other factor, which is relevant for the Slovene and Arabic treebanks, is the number of coordinated structures.

The number of non-projective trees for the Arabic, Bulgarian, Italian and Slovene treebanks are respectively 175 (10.9%), 962 (7.3%), 91 (6.1%) and 1,289 (66.6%). The number of sentences with coordinated structures in the PADT and SDT are respectively 1,041 (64.8%) and 989 (51.1%).

The overall results for Arabic and Slovene are the worst, compared to the results for the other treebanks. Labelled accuracy for PADT is around 1% smaller than labelled accuracy for the same size data sets for Slovene. It seems that the incremental dependency parser has difficulties with coordination treatment in these annotation schemes, at least for small data sets[7].

[7]The PDT was also one of the hard-to-parse treebanks at the CoNLL-X shared task despite its larger size (Buchholz and Marsi, 2006).

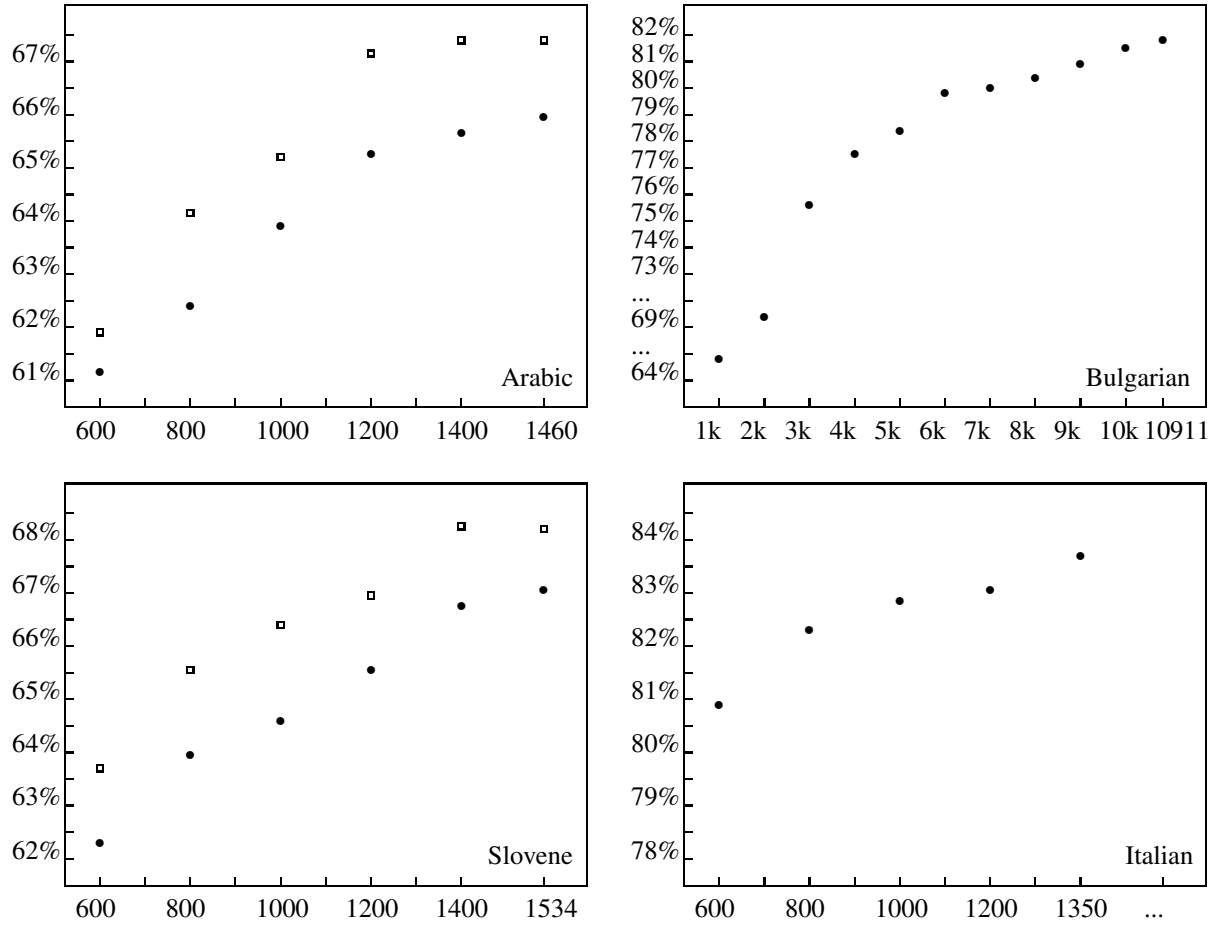


Figure 1: Clockwise: Learning curves for Arabic (top left), Bulgarian, Italian and Slovene (labelled attachment score).

Figure 1. shows that the coordination transformations increased parsing accuracy for the Arabic (and Slovene) data sets. Due to the imperfect back transformation procedure some accuracy has been lost. The number of non-projective sentences in PADT is comparatively small – only 175. The number of sentences with coordination is 1,041. The results for Arabic reported in this paper are slightly higher (0.5%) than the best results reported at the CoNLL-X shared task even though a more sophisticated feature model for the Malt parser was used there.

Results for Bulgarian are lower, compared to the results obtained at the CoNLL-X shared task where the Malt parser had a better feature model and the data was parsed pseudo-projectively. The accuracy that we report is higher than the one reported in (Chanev et al., 2006) because they used an option of the SVM learner which splits the data on smaller parts for faster learning with the cost of decrease in performance.

Compared to the other treebanks the parser learned TUT very well with a limited amount of training data. The reason for the good performance cannot be the number of tokens per sentence (SDT has less and PADT has more tokens per sentence). Sizes of the tag sets are not suspiciously small to be the main reason for the good results on little training data. It may be concluded that the reason for the high accuracies is the treebank annotation scheme. It is different from those of the other treebanks in its 'deeper' syntactic dependency relations. The distance between the dependents and their heads is usually short which facilitates processing.

Compared to (Chanev, 2005) there is an increase of accuracy due to the use of a more advanced feature model for the parser and the better SVM learner. The number of sentences in TUT which have non-projective graphs is very small[8] – only 91. That may have contributed to the high parsing accuracy.

Our results for Slovene somehow lag behind the results for that language which were obtained using the Malt parser at the CoNLL-X shared task. The reasons are the use of a simple feature model for the parser and the big number of non-projective trees in the Slovene treebank (1,289) which we did not parse pseudo-projectively.

Results are on the average 1% higher than those for the PADT. Possibly this difference can be explained with the very small number of tokens per sentence for the SDT – only 18.2, compared to 37.2 for the Arabic treebank. The number of coordinated structures is 989. As in the case

[8]Originally TUT does not have non-projective sentences but after traces were removed in (Chanev, 2005) non-projective arcs were introduced.

with PADT, coordination transformations increased parsing results.

6. Conclusion and Future Work

We presented the learning curves for four different treebanks using the same feature model for learning an incremental statistical dependency parser. We showed that often parsing results differ significantly for different languages and the reasons can be various properties of the concrete treebank. We performed treebank transformations for Arabic and Slovene to report parsing accuracy for Arabic that is slightly higher than the best results reported at the CoNLL-X shared task. We compared the annotation schemes of the treebanks by measuring the extent to which they can be learned and parsed using an incremental parser.

Future work includes investigation of various treebanks to find out which annotation scheme keeps parsing accuracy high for a vast majority of languages. In addition we believe that adding different kind of information to feature models for parsers with incremental architectures can lead to successful broad coverage models of the human sentence parsing mechanism whose implementations must be good multilingual NLP parsers.

Acknowledgements

I thank the organizers of the CoNLL-X shared task for providing the PADT in CoNLL format (through the LDC) as well as for their devoted work on organizing the task. I thank Kiril Simov and Petya Osenova for making the BulTreeBank available, Cristina Bosco – for the TUT and Tomaž Erjavec – for the SDT. I also thank Alberto Lavelli for being my co-advisor, one anonymous reviewer and the Malt parser team at the CoNLL-X shared task (Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit and Svetoslav Marinov).

7. References

G. Attardi. 2006. Experiments with a multilingual non-projective dependency parser. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

C. Bosco. 2004. *A grammatical relation system for treebank annotation*. PhD thesis, University of Turin.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

A. Chanev, K. Simov, P. Osenova, and S. Marinov. 2006. Dependency conversion and parsing of the BulTreeBank. In: *Proc. of the LREC workshop Merging and Layering Linguistic Information*, Genoa.

A. Chanev. 2005. Portability of dependency parsing algorithms - an application for Italian. In: *Proc. of the workshop Treebanks and Linguistic Theories*, Barcelona.

C.-C. Chang and C.-J. Lin. 2005. LIBSVM: A library for Support Vector Machines. *URL: http://www.csie.ntu.edu.tw/∼cjlin/papers/libsvm.pdf*.

W. Daelemans and A. Van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press.

S. Džeroski, T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky, and A. Žele. 2006. Towards a Slovene dependency treebank. In: *Proc. of the Fifth International Conference on Language Resources and Evaluation*, Genoa.

J. Hajič, O. Smrž, P. Zemánek, J. Šnaidauf, and E. Beška. 2004. Prague arabic dependency treebank: Development in data and tools. In: *Proc. of the NEMLAR International Conference on Arabic Language Resources and Tools*.

J. Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In: *Issues of Valency and Meaning*, Prague. Karolinum.

J. Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In: *Proc. of Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh.

L. Lesmo, V. Lombardo, and C. Bosco. 2002. Treebank development: the TUT approach. In: R. Sangal and S.M. Bendre, ed., *Recent Advances in Natural Language Processing*, New Delhi. Vikas Publ. House.

D. Lin. 1998. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4 (2):97–114.

R. McDonald, K. Lerman, and F. Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

I. Mel'čuk. 1988. *Dependency syntax: Theory and practice*. State University of New York Press.

J. Nivre and J. Nilsson. 2005. Pseudo-projective dependency parsing. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

J. Nivre, J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. 2006. Labeled pseudo-projective dependency parsing with Support Vector Machines. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

J. Nivre. 2004. Incrementality in deterministic dependency parsing. In: *Incremental Parsing: Bringing Engineering and Cognition Together, Workshop at ACL-2004*, Barcelona.

J. Nivre. 2005. *Inductive Dependency Parsing of Natural Language Text*. PhD thesis, University of Växjö.

K. Simov, P. Osenova, A. Simov, and M. Kouylekov. 2005. Design and implementation of the Bulgarian HPSG-based treebank. *Journal of Research on Language and Computation – Special Issue*, str. 495–522.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with Support Vector Machines. In: *Proc. of IWPT*, Nancy.

Slovene Word Sketches

Simon Krek,[*] Adam Kilgarriff[**]

[*] Faculty of Arts
University of Ljubljana
Ljubljana, Slovenia
simon.krek@guest.arnes.si

[**] Lexical Computing Ltd
Brighton, United Kingdom
adam@lexmasterclass.com

Abstract

Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour. They were first used in the production of the Macmillan English Dictionary (Rundell 2002). At that point, they only existed for English. Today, the Sketch Engine is available, a corpus tool which takes as input a corpus of any language and corresponding grammar patterns and which generates word sketches for the words of that language. It also automatically generates a thesaurus and 'sketch differences', which specify similarities and differences between near-synonyms. The FidaPLUS corpus, a morpho-syntactically tagged corpus of Slovene was loaded into the Sketch Engine software. We shall demonstrate the Slovene word sketches, and show how they can be used in lexicography and for other linguistic purposes. The results show that word sketches could significantly facilitate lexicographic work in Slovene as they have for English.

Besedne skice v slovenščini

Besedne skice *(Word sketches)* so avtomatski na korpusu temelječi sežetki slovničnega in kolokacijskega vedenja neke besede. Prvič so bile uporabljene pri sestavljanju enojezičnega angleškega slovarja založbe Macmillan (Rundell 2002). Takrat so obstajale le za angleški jezik. Zdaj je na voljo programski modul Sketch Engine, korpusno orodje, ki na vhodu sprejme korpus kateregakoli jezika ter njegove slovnične vzorce, iz njih pa ustvari besedne skice za besede tega jezika. Hkrati avtomatsko generira tezaver in "razlikovalne skice", ki izpostavljajo podobnosti in razlike med bližnjimi sopomenkami. V programski modul Sketch Engine smo naložili korpus FidaPLUS, oblikoslovno-skladenjsko označeni korpus slovenščine. Prikazali bomo slovenske besedne skice in pokazali, kako jih je mogoče uprabiti za leksikografske in druge jezikoslovne namene. Rezultati kažejo, da besedne skice znatno olajšajo delo leksikografom slovenskega jezika, tako kot se je izkazalo pri angleščini.

.

1. Introduction

Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behaviour. Their value for lexicographic work in English and other languages, as well as the background of the use of corpora in lexicography, have been described elsewhere (Kilgarriff and Tugwell 2001, Kilgarriff and Rundell 2002, Kilgarriff et al. 2004).

First, we shall introduce corpus query systems and the basic idea of word sketches. Next, we shall concentrate on the application of word sketches to the Slovene language in the Sketch Engine software.

The FidaPLUS corpus of Slovene will also be briefly described, with special attention to the tagging problems which could affect its use within the Sketch Engine.

2. Word sketches

2.1. Corpus query systems

Different corpus query systems have been used to check the corpus evidence since the rise of the first electronic corpora. Ever since the COBUILD project, lexicographers have been using KWIC concordances as their primary tool for finding out how a word behaves. Later, with the growth of corpora, lexical statistics had to be applied to manage the abundant data and highlight the most salient combinations and collocations. Today, state-of-the-art CQSs allow the lexicographer great flexibility in searching for phrases, collocates, grammatical patterns, sorting concordances according to a wide range of criteria, identifying 'subcorpora' for searching in only spoken text, or only fiction. Available systems include WordSmith, MonoConc, and the Stuttgart Workbench among others.

Specifically for the two large Slovene corpora, there are also two different on-line concordancers available: ASP32 for the FidaPLUS corpus[1] and NEVA for Nova beseda, with a more detailed description available in Krek (2003).[2]

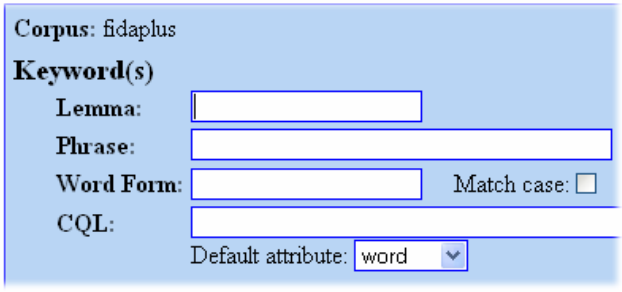
2.2. Sketch Engine

2.2.1. Description

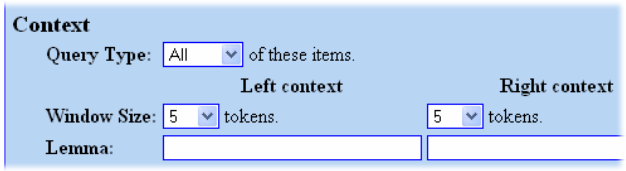
The Sketch Engine is a corpus query system which allows the user to use the familiar CQS functions:

[1] http://www.fidaplus.net
[2] http://bos.zrc-sazu.si/s_beseda.html

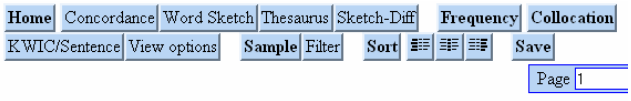
– concordances with lemma, phrase, word form and CQL search,



together with the context control filter



and the usual viewing and sorting options:



F0000012.35.10 polkrožna platoja . Danes tam stoji počitniška luša , ob njej pa zidanic
F0000012.104.9 vgnezdila nemška posadka . Zdaj tam stoji nova luša . Grajska kapela M
F0000012.214.1 zgodovinski listini omenjena Karolova " luša " (haws Sagradez
F0000012.224.7 , ohranjena je še nekdanja oskrbnikova luša , nekdanja grajska
F0000012.295.5 Stančič , zdaj pa na njenem mestu stoji nova luša . Južno od tod so p
F0000012.614.8 Ob grajskem jedru je bila zgrajena nova luša . Dornberk je bil ol
F0000012.782.8 podrl stavbo . Na njegovem mestu stoji nova luša . Galetov
F0000012.1001.16 Buka . Danes na njenem mestu stoji nova luša . Gospodarsko pos

However, the features of the Sketch Engine which are of special interest in this article are not part of standard concordancing programs. These features include Word Sketch, Sketch Difference and Thesaurus which will be described later. All these features are fully integrated with standard concordancing.

2.2.2. Word Sketch

To identify a word's grammatical and collocational behaviour, the Sketch Engine needs to know how to find words connected by a grammatical relation. It allows two possibilities.

In the first, the input corpus has been parsed and the information about which word-instances stand in which grammatical relations with which other word-instances is embedded in the corpus. Currently, dependency-based syntactically annotated corpora are supported. Phrase-structured trees need heads of phrases to be marked.

In the second, the input corpus is loaded into the sketch engine POS-tagged but not parsed, and the sketch engine supports the process of identifying grammatical relation instances. Each grammatical relation will be defined, using the Sketch Engine to test and develop it. When the developer is happy with the definition of each grammatical relation, they save the definitions in a "gramrel" file. The Sketch Engine then compiles this file and finds all instances of all grammatical relations in the corpus. It puts them in a gramrels database and users than have access to word sketches.

2.2.3. Lemmatization & POS-tagging

The Sketch Engine does not support the process of lemmatization; various tools are available for linguists to develop lemmatizers, and they are available for a number of languages. If no lemmatizer is available, it is possible to apply the Sketch Engine to word forms, which, while not optimal, will still be a useful lexicographic tool.

Similarly for part of speech (POS) tagging, also known as POS-disambiguation. This is the task of deciding the correct word class for each word in the corpus – of determining whether an occurrence of "brez" in Slovene is an occurrence of a noun "breza" in plural, genitive case, or a preposition. A tagger presupposes a linguistic analysis of the language which has given rise to a set of the syntactic categories of the language, or tagset. Tagsets and taggers exist for a number of languages, and there are assorted well-tried methods for developing taggers. The Sketch Engine assumes tagged input.

As the FidaPLUS corpus is both lemmatized and POS-tagged but not syntactically annotated, Slovene word sketches are based on a lemmatized and POS-tagged corpus, with grammatical relations defined on the basis of POS-tag information.

2.3. Grammatical relations

Grammatical relations are defined as regular expression over POS-tags. For example, if we wish to include the grammatical relation between a noun and its adjectives in modifying position, we define the head of the noun phrase, a noun ("S" in the FidaPLUS tagset) and one or more preceding adjectives ("P") with the possibility of allowing the intervening comma and the particles "se" and "si":

=a_modifier/modifies
2: [tag="P.*"] [tag="P.*" | word="," | word="se" | word="si"] {0,5} 1: [tag="S.*"]

The first line, following the =, gives two names for the grammatical relation. The first, before the slash, is the name when the arguments are in the one order, and the other is when the arguments are in the other.

The 1: and 2: mark the words to be extracted as the first and second arguments. |, ., (), and * are standard regular expression metacharacters. {0,5} indicates that the preceding term occurs between zero and five times.

3. Slovene Word Sketches

3.1. Slovene Corpus

3.1.1. FIDA corpus

The FIDA corpus is the precursor of the FidaPLUS corpus which was used in the Sketch Engine software. It was compiled in a joint project involving four partners, two from the academic/research sphere: (the Faculty of Arts, University of Ljubljana, the Jožef Stefan Institute) and two commercial ones (DZS publishing house and Amebis software company). Corpus compilation started in 1997 and was concluded in 2000. The corpus was just over 100 million words and was a balanced corpus of texts in the Slovene language mainly from the 1990s.

The corpus was lemmatized and POS-tagged but the process was limited to the lexicon of word forms available

at Amebis at the time. The disambiguation of multiple possible morphosyntactic descriptions, (MSDs) for ambiguous wordforms such as *brez* was not performed, a considerable drawback when using the corpus for automatic linguistic analysis.

3.1.2. FidaPLUS corpus

The problems of lemmatization and POS-tagging, together with the size, balance and up-to-dateness were addressed in the subsequent project, "Language Resources for Slovene", funded by the Slovene Ministry of Higher Education, Science and Technology and co-funded by DZS and Amebis. Project partners included the Faculty of Arts (University of Ljubljana) as the leading partner, the Faculty of Social Sciences (University of Ljubljana) and the Jožef Stefan Institute. Its aim was a three hundred million word corpus with complete lemmatization and POS-tagging.

The FidaPLUS corpus used for testing in the Sketch Engine is the preliminary result of the project. In terms of size it is similar to the FIDA corpus, but the lemmatization and POS-tagging have been improved. Lemmatization is both lexicon-based and statistical, aiming at lemmatization of all items in the corpus. POS-disambiguation uses the tools developed by Amebis.

3.2. Slovene grammatical relations

The Slovene "gramrel" file was based on the Czech example (Kilgarriff et al. 2004), since Czech, like Slovene but unlike English, is a relatively free word order language.

The grammatical relations in the Slovene gramrel file include three types: **symmetric**, between two items with equal status, **dua**l, between two items with dependent relations and **trinary**, between three dependent items.

coord	7334	0.5
prostor	837	44.63
datum	144	37.89
trud	75	35.92
kraj	247	34.76
Vera	33	31.35
energija	155	30.85
denar	206	27.91
se	192	26.5
bit	35	25.96
zaslonka	13	25.71
potrpljenje	21	24.85
trimesečen	17	23.36
da	97	23.18
on	131	21.66
svoj	26	20.95
ne	63	19.83
napor	25	19.0
tudi	43	18.33
kraja	24	18.02
njegov	35	17.97

3.2.1. Symmetric Example

One example of the symmetric relation is various coordinate structures with conjunctions "and" or "or", as well as two-word coordinate structures such as "niti-niti", "ali-ali".

```
=coord
*SYMMETRIC
        1:[] [word = "in" | word = "ali"] 2:[]
        [word = "niti"] 1:[] [word = "niti"] 2:[]
        [word = "ali"] 1:[] [word = "ali"] 2:[]
        [word = "bodisi"] 1:[] [word = "bodisi"] 2:[]
        [word = "tako"] 1:[] [word = "kakor"] 2:[]
        [word = "tako"] 1:[] [word = "kot"] 2:[]
```

The result of this grammatical relation can be viewed as part of the word sketch. The result shows that in the FidaPLUS corpus, 7334 instances of this particular grammatical relation can be found for the lemma "čas". Lemmas are ranked according to the salience score (Kilgarriff and Tugwell 2001). The user can click on the number next to a lemma to see the relevant concordance.

We used four summetrical relations..

a_modifier	19068	1.4
sklonjen	157	59.54
obrit	66	47.71
kronan	35	39.24
dvignjen	62	36.12
zeljnat	23	35.75
odsekan	33	35.5
Hermanov	34	35.11
razgret	40	34.63
mrtvaški	34	31.12
trezen	41	30.65
bikov	22	29.33
ovnov	16	29.12
video	106	29.03
koničast	34	28.96
pobrit	13	27.94
zeljen	18	27.92
bister	35	27.26

is_obj4_of	2506	3.9
skloniti	98	54.21
beliti	83	49.81
dvigniti	218	49.15
razbijati	71	44.74
odsekati	60	43.2
pomoliti	45	42.62
nagniti	59	40.97
tiščati	46	40.92
stakniti	39	39.75
obrniti	90	34.92
odrezati	44	33.54
nasloniti	29	32.21
sploščiti	20	32.18
razbiti	36	31.43
pobešati	11	30.46
sklanjati	18	30.17
povešati	11	29.91

3.2.2. Dual Example

Dual relations are most common in the gramrel file. There are eleven of them, covering relations expressed by means of grammatical case in Slovene as well as modifying structures as shown before. The corresponding part of the word sketch for the lemma "glava" is shown on the left.

Relations covering grammatical cases are defined in the following fashion:

```
=is_obj4_of/has_obj4
*DUAL
        2:[tag="Gpp.*" & !(lemma = "biti" | lemma = "imeti" | lemma = "hoteti" | lemma = "morati" | lemma = "smeti") ] [tag!="[SGDVLMOZ].*" & tag!=""] {0,5} 1: [tag="S...t.*"]
        2:[tag="G.d.*" & !(lemma = "biti" | lemma = "imeti" | lemma = "hoteti" | lemma = "morati" | lemma = "smeti") ] [tag!="[SGDVLMOZ].*" & tag!=""] {0,5} 1:[tag="S...t.*"]
```

There are two variants of the particular relation: either a verb has an object in the oblique case or the noun is itself an object in the same case, in relation to a verb. The example on the left shows a list of verbs where the lemma "glava" is predominantly used in the oblique case within a window of five items from a verb. All the verbs from the beginning of the list indicate structures which are lexicographically relevant because of their either central or additional metaphorical meaning. Thus the concordances of the structure "skloniti glavo" show that besides the literal meaning "to bow one's head", there are many examples of the metaphorical extension "to give up" or "to concede defeat". The next one indicates the structure "beliti si glavo" which is thoroughly idiomatic: "to worry about, to agonize over". The same is true for "razbijati si glavo", "tiščati glave (skupaj)", "stakniti glave" etc.

3.2.3. Trinary Example

Trinary relations indicate the relations between three grammatical categories. In the Slovene gramrel file, they are mainly used to extract prepositional patterns where the grammatical case – in Slovene the instrumental and locative cases – is expressed by means of prepositional phrases.

prec_po	1090	9.0
rojiti	91	65.67
udariti	142	55.22
motati	49	51.25
popraskati	24	42.95
treščiti	38	39.83
poditi	32	38.99
tolči	34	37.54
lopniti	12	32.97
tepsti	20	32.27
praskati	13	28.88
bloditi	12	28.06
plesti	14	27.2
trepljati	9	26.25
poškodovati	23	25.79
čohati	6	25.61
pobriti	5	24.37
srati	8	23.96

*TRINARY
=prec_%s
2:[tag="S.*"] 3:[tag="D.*"] [tag="P.*" | word= "," | word = "se" | word = "si"] {0,5} 1:[tag="S.*"]
2:[tag="G.*"] 3:[tag="D.*"] [tag="P.*" | word= "," | word= "se" | word="si"] {0,5} 1:[tag="S.*"]

In the case shown on the left, the grammatical relation is established between the lemma "glava" preceded by the preposition "po", and the "glava" word sketch indicates salient combinations with verbs on the left. Again, together with the frequent but semantically transparent combinations there are numerous idiomatic expressions such as "rojiti/motati/poditi po glavi" and the more informal "srati po glavi".

3.3. Sketch Differences

The sketch differences feature in the Sketch Engine specifies, for two semantically related words, what behaviour they share and how they differ. Synonymous words tend to share some of the collocates but not all. The sketch differences show the patterns which are shared by both synonyms and presents the information also in a colour scheme for the user to grasp immediately if and where the lemmas are synonymous. For the Slovene language, this is particularly useful in cases where there are two competing synonyms, one etymologically foreign and the other of Slavic origin. The more normatively-minded usually argue for abolition of the foreign lemma and non-discriminatory use of the Slavic form. The example of "cona" and "območje" in the Appendix 1 shows the differences. In the FidaPLUS corpus, only "operativen" is distributed evenly between the two synonyms. A milder bias towards "območje" is indicated in the cases of "demilitariziran" and "turističen" and a stronger one with "zaprt" and "obmejen". The opposite is true with more fixed "erogena cona", "obrtna cona", "industrijska cona" etc. and less fixed "carinska cona / carinsko območje", "tamponska cona / tamponsko območje", also "tamponski", "brezcarinski", "siv" etc.

3.4. Thesaurus

The similarity is based on 'shared triples'. "Cona", "območje" both occur as the second term in the triple <modifier, ?, "tamponska">, and this provides one small piece of evidence that the two words are close in meaning. By simply gathering together all such pieces of evidence (and weighting them according to salience, following the method developed by Lin (1998)), we identify the near neighbours for each. The Sketch Engine does this and the result for the lemma "kriza" can be seen in the Appendix 2.

As there is no thesaurus available for the Slovene language, it is not possible to compare it to the human assessment of the word's synonymic relations, but it is immediately clear that the software shows a number of relevant items such as "konflikt", "spor", "spopad" etc., indicating one semantic direction, "problem", "težava", "zaplet" etc., indicating another, and "stiska", "izguba" indicating a more intimate human sentiment.

One can explore each of the relations with the sketch differences feature.

4. Conclusion and further work

Testing of the 100-million FidaPLUS corpus in the Sketch Engine has shown it to be an exceptionally useful tool for exploring typical grammatical and lexical relations in the Slovene language. To be able to take full advantage of the software, it is important to have a corpus which is lemmatized and POS-tagged as accurately as possible, and that is one area where there is room for improvement. We would like to further explore Slovene grammatical relations and their implementation in the gramrel file, and also the possibility a Slovene dependency-parser.

However even in its present form the Sketch Engine is a valuable tool, particularly for lexicographic use.

5. References

Kilgarriff, A., Tugwell, D. (2001). WORD SKETCH: Extraction and Display of Significant Collocations for Lexicography. *Proc. ACL workshop on COLLOCATION: Computational Extraction, Analysis and Exploitation.* Toulouse. 32-28.

Kilgarriff, A., Rundell, M. (2002). Lexical profiling software and its lexicographic applications - a case study. *Proc EURALEX.* Copenhagen. 807-818.

Kilgarriff, A., Rychly, P., Smrž, P., Tugwell, D. (2004) The Sketch Engine. *Proc. Euralex.* Lorient, France. 105-116.

Lin, D, (1998). Automatic retrieval and clustering of similar words. *COLING-ACL,* Montreal. 768-774.

Krek, S. (2003). Jezikovni priročniki in novi mediji. *Jezik in slovstvo*, letn. 48, št. 3-4, 29-46.

Rundell, M. (ed) (2001). *Macmillan English Dictionary for Advanced Learners*. Macmillan Education.

Appendix 1: Sketch difference – lemma_1 “cona”, letmma_2 “območje”

	a_modifier	4941	36173	2.6	1.9
green	erogen	75	13	58.6	21.4
green	obrten	288	9	58.3	5.6
green	ekonomski	411	19	54.5	5.2
light green	carinski	271	214	53.4	33.9
green	industrijski	284	76	52.2	19.4
green	prost	317	79	50.1	16.7
green	prostocarinski	61	21	47.3	21.8
red	obmejen	7	313	11.5	46.7
green	moder	139	20	41.4	8.2
green	okupacijski	35	10	38.6	14.6
light red	nekdanji	41	710	16.6	37.3
extra light green	tamponski	17	12	35.8	23.1
extra light red	demilitariziran	9	30	25.3	33.8
red	zaprt	5	162	7.6	32.4
extra light green	brezcarinski	26	17	30.3	16.5
light green	konvergenčen	12	6	29.7	15.0
white	operativen	34	47	28.5	21.8
light red	bivši	15	199	11.9	27.4
red	posamezen	9	338	5.1	26.8
light green	svoboden	42	43	25.5	14.9
extra light red	turističen	16	183	11.0	23.6
light red	visok	8	353	3.1	23.5
extra light red	koprski	12	110	11.8	23.2
light red	mesten	11	190	7.2	21.7
light green	siv	23	14	20.9	8.1

Appendix 2: Thesaurus – lemma “kriza”

konflikt	0.319	spor 0.273 spopad 0.267 vojna 0.266 nasilje 0.202 boj 0.17
problem	0.303	težava 0.288 razmera 0.279 situacija 0.264 dogajanje 0.228 dogodek 0.226 stanje 0 problematika 0.18 potreba 0.172 stvar 0.172
zaplet	0.25	katastrofa 0.243 padec 0.198 zlom 0.185 razpad 0.179 izbruh 0.171 tragedija 0.17
izguba	0.24	posledica 0.21 pomanjkanje 0.209 nevarnost 0.188 pritisk 0.182 vpliv 0.176 učinek
sprememba	0.229	politika 0.196 proces 0.189 razvoj 0.184 sila 0.182 gibanje 0.182 odnos 0.177
stiska	0.223	recesija 0.211 revščina 0.178
afera	0.221	
bolezen	0.207	nesreča 0.196 bolečina 0.169
revolucija	0.205	reforma 0.203 napad 0.196 volitve 0.178 poseg 0.176 akcija 0.171
nemir	0.193	napetost 0.19
obdobje	0.187	
uspeh	0.182	poraz 0.175

Exploiting the Leipzig Corpora Collection

Matthias Richter*, Uwe Quasthoff*, Erla Hallsteinsdóttir*, Christian Biemann*

*Leipzig University, Computer Science Department
Natural Language Processing Group
Augustusplatz 11, 04109 Leipzig, Germany
{mrichter,quasthoff,cbiemann}@informatik.uni-leipzig.de
erlahall@yahoo.dk

Abstract

In this paper the Leipzig Corpora Collection is introduced as a contribution to the idea that there is need for standardization of multilingual language resources. We explain the steps of building, processing and presenting corpora of comparable sizes and in a uniform format. Results from intra- and interlingual comparisons of corpora are given and methods that can build upon these corpora are shown.

Uporaba lepiziške korpusne zbirke

V članku je leipziška korpusna zbirka predstavljena kot prispevek k ideji o standardizaciji večjezičnih jezikovnih virov. Razložimo postopke gradnje, procesiranja in predstavitve korpusov primerljive velikosti in v enovitem formatu. Podani so rezultati znotraj- in medjezikovne primerjave korpusov ter predstavljene metode, ki lahko zrasejo na njihovi osnovi.

1. Introduction

Corpora are important linguistic resources. We have released a collection of standard sized corpora in 17 different languages in a uniform format that is free of charge for scientific use. Large corpora can be accessed online and downloaded from `http://corpora.uni-leipzig.de/` including a software for offline corpus exploration. This data has been prepared in order to ease and foster corpus research and as a contribution to the standardization of language resources. In this paper we describe the details of the collection and its format, explore possibilities of research given standard sized corpora and present selected results that we have already obtained. Because of the variety of topics covered in this paper we mention and discuss related works in the respective contexts instead of prepending a related work section.

After elaborating on the collection itself in Section 2. we present intra and inter language statistics in Section 3. and examples of usage of our corpora in Section 4..

2. The Leipzig Corpora Collection

2.1. Goals of the Project

The Leipzig Corpora Initiative was started during the 1990s because at that time there were no freely accessible resources available for NLP in German. Since then techniques for processing and presenting corpora have been developed which are not depending on features of specific languages. Some are described in (Biemann et al., 2004b). Having collected text resources in many different languages, it is now possible to provide access to data and statistics on these languages which are available in a unified format and in standard sizes. Further, we want to provide basic linguistic services free of charge for anyone who has a use for them, without having to sign agreements, paying shipping fees and alike. Of course, free corpora as opposed to high-quality expensive resources may not fulfill all requirements in text quality and balancing and cannot provide manually added metadata or large-scale annotation. As for such, more sophisticated corpus query systems are available, e.g. (Kilgarriff et al., 2004). Our focus, however, is on methods that work in absence of linguistic knowledge. And nevertheless, as discussed in detail e.g. in (Bordag, 2006) the resources we are discussing here are sufficient for a number of lexical acquisition and other NLP tasks such as extraction of knowledge, automatic calculation of semantic associations and collocations as well as word sense induction. Unlabelled data can greatly improve learning tasks in general see the literature on semi-supervised learning (Zhu, 2005). Possible usage of corpora as a resource includes, but is not limited to (Baroni and Ueyama, 2006):

- monolingual lexicography (which will be a more detailed example in Section 4.1.)
- comparing different languages on a statistical basis
- parameterizing language models e.g. for speech recognition
- expanding queries with statistically similar words
- extracting significant terms from documents by comparison against a reference corpus (Faulstich et al., 2002)
- selecting balanced word sets for experiments e.g. in psycholinguistics

2.2. The Corpus Building Process

Our corpus building process consists of mainly four steps: collecting, pre-processing, cleaning and, eventually, calculating. The steps of the process have been described in detail in (Quasthoff et al., 2006).

Unless there already is a large text collection at hand, texts have to be collected for each language. During the

	language	size	source
cat	Catalan	10 million	WWW
dan	Danish	3 million	WWW
dut	Dutch	1 million	Newspaper
eng	English	10 million	Newspaper
est	Estonian	1 million	various
fin	Finnish	3 million	WWW
fre	French	3 million	Newspaper
ger	German	30 million	Newspaper
ice	Icelandic	1 million	Newspaper
ita	Italian	3 million	Newspaper
jap	Japanese	0.3 million	WWW
kor	Korean	1 million	Newspaper
nor	Norwegian	3 million	WWW
sor	Sorbian	0.3 million	various
spa	Spanish	1 million	Newspaper
swe	Swedish	3 million	WWW
tur	Turkish	1 million	WWW

Table 1: languages, maximum size in sentences and sources of the corpora.

last years it has become a common practice to use the web as corpus or for corpus acquisition (Kilgarriff, 2001). Corpus acquisition from the web often includes seeding and crawling of web sites (Baroni and Kilgarriff, 2006). One modification of seeding that we employ is to search for current news articles with a news search engine for a very long period of time in order to ensure that certain types of text get collected.

Pre-processing is done by stripping HTML-tags from the collected texts and separating the content from boilerplates. Then a sentence boundary detection is performed and ill-formed sentences fragments get removed as well as sentences in foreign languages (Quasthoff and Biemann, 2006) and (near) duplicates.

Before scrambling the corpus on sentence level and reducing it to pre-defined sizes, further cleaning is performed. This is done to ensure there are actually properly formed sentences which are not obviously containing non-standard language. Scrambling sentences and downsampling in a way that the original documents cannot be restored ensures that the texts can be distributed without hurting copyright protection, as single sentences are too short to be regarded as intellectual property.

2.3. Languages and Corpora

Corpora in the languages listed in Table 1 are collected from the web and consist either of newspaper texts or of randomly collected web pages. The maximum sizes of the corpora offered are restricted by present availability, rather than being arbitrarily chosen. Our notion of corpus is centered around the sentence as the largest unit. This is sufficient for a vast variety of applications in statistical NLP and lexicography.

For each language a full form dictionary with frequency information for each word is calculated. Further we provide co-occurrence statistics: words that co-occur significantly often with a given word. For the calculation of the significance, the log-likelihood measure (Dunning, 1993) is used as described in (Biemann et al., 2004b). Two kinds of co-occurrence data are pre-computed: Words occurring together in sentences and words found as immediate (left or right) neighbors. Only co-occurences that are above a certain significance level (p=5% for neighbors, p=1% for sentence-windows) are kept. Co-occurrence data is meant to be used extensively as a building block for further applications (cf. Section 4.2. for some ideas).

Additional data is included if available. As of now, only the German dictionary already contains grammatical information such as inflection and semantic information such as subject areas and synonyms. The open and flexible architecture, however, can easily be augmented on word and sentence level with all kinds of additional data such as grammar, links and annotation.

2.4. Database Structures und Conversion Issues

The structures of the MySQL database have been kept as simple as possible with much effort having been put into short query response times with large amounts of data. One type of table is meant for storing words, sentences and sources with id, frequency (for words only) and the respective string. Another type of table is used for sentence-based and neighbor co-occurrences between two word ids, the co-occurrence's frequency and its statistical significance. Finally there are inverted lists, one for sentence id and source id and one for word id and sentence id which also contains the word's position in the sentence. There is also a table with pre-calculated meta statistics of the database.

There is at the moment no conversion script available for specific source formats, but it is only a matter of a few lines of code to transform any sane text corpus format into a database in the Leipzig Corpora Collection's format. There is a software available at request from the authors which takes a sentence segmented text and a list of multi word units as input and calculates a full text index with position and sentence-based and neighbor co-occurrences. As an example, the 21 corpora of the TEI-encoded JRC-Acquis collection (Steinberger et al., 2006) were converted with ease.

2.5. Distribution and Availability

On our web site `http://corpora.uni-leipzig.de/` corpora for the following languages can be accessed online: Catalan, Danish, Dutch, English, Estonian, Finnish, French, German, Icelandic, Italian, Japanese, Korean, Norwegian, Sorbian[1], Spanish, Swedish, Turkish. There also is a download site at `http://corpora.uni-leipzig.de/download.html` where smaller corpora[2] of these languages can be obtained free of charge in two formats: flat text files and MySQL databases. The Leipzig Corpus Browser is a tool written in Java for accessing the MySQL databases. The software provides a lot more predefined query options than the web site does and makes adding customized queries easy. This

[1] spelling is correct: Upper and Lower Sorbian are slavonic minority languages with approximately 100 000 speakers in the south of Eastern Germany.

[2] As of now the larger corpora are available only after email request.

can be used for example to add more sophisticated queries and for the integration of additional data resources. The browser should be operational on any platform that supports Java 5, however it has only been tested on Microsoft Windows, Mac OS X, Linux and Solaris. It is available free of charge from the download page as well.

3. Statistical Results

There are several problems when comparing statistical data for corpora of different types of selection, languages, and sizes. In Section 3.1.1. the effect of of the type of selection is measured. In Section 3.1.2. we compare measurements for different corpus size. The non-linear growth of some size parameters is shown to fulfill power laws. This in turn is used to combine results for different languages in Section 3.2..

3.1. Intra Language Statistics

3.1.1. Sampling

A series of experiments was conducted to quantitatively study the intra language effects of sampling sentences at random from starting sets of different size. Starting point was a corpus of 40 million German sentences that were in text order. 100 000 sentences were selected at random from one distinct segment of size 1 million, 4 million, 10 million or all 40 million sentences. Each experiment was repeated 40 times and numbers of tokens, types, sentence co-occurrences, neighbor co-occurrences as well as average type and token length and text coverage with the top n types was measured. The results are summarized in Table 2.

It turns out that the numbers of types and co-occurrences show a big variation. On the other hand the average type, token and sentence length as well as text coverage with the top n types remain extremely stable. The experiment also proposes that the amount of text from which one chooses the final sample has got a small but significant influence on the average numbers of types (the larger, the more) and co-occurrences (the larger, the less) observed, which is stronger with sentences based than with neighbor co-occurrences. This result does not defy intuition as one would expect, when looking at random samples from a corpus of infinite size, to see content words' frequencies – and therefore also the number of co-occurrences – decrease and the number of hapax legomena increase. A t-test tells that this result is highly significant (p=0.1%) only when comparing the columns for 1 million and 40 million sentences. The values for the intermediate segment sizes can not contribute statistically significant support, yet they are neither opposing the observations. On the other hand the effect is not so dramatically strong that we would need to ensure that there is very precisely the same amount of source text from which we start downsampling standard size corpora. If there are, however, several orders of magnitude this systematic skew should be taken into consideration.

3.1.2. Scaling

In the following, we compare Finnish corpora containing 100K, 300K, 1M and 3M sentences, respectively. For these, we count the number of tokens and the number of sentence and next neighbor co-occurrences (above some threshold). The result, shown in Figure 1, is typical for all languages we analyzed.

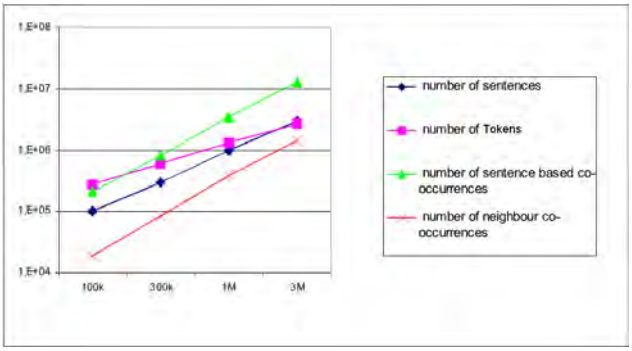


Figure 1: Effects of different corpus size on numbers of types, tokens and co-occurrences for Finnish

In this doubly logarithmic plot we can observe the following:

1. The increase is in all cases nearly linear.
2. The number of tokens increases clearly more slowly than the number of sentences.
3. The number of sentence co-ocurrences increases at a similar rate as the neighbor co-occurrences.

3.2. Inter Language Statistics

3.2.1. Basic statistics

Language statistics such as Zipf's law (Zipf, 1935; Sigur et al., 2004) have been researched in intra language basis for many years, e.g. by (Meier, 1967) for German. We are now presenting inter language basic characteristics in Tables 3 and 4 and exemplified in Figures 2 and 3:

- number of types (nty)
- number of tokens (nto)
- average type length (tyl): word length from each type in the corpus divided by the number of types
- average token length (tol): word length from each token in the corpus divided by the number of tokens
- coverage of text: given a text, the most frequent 10 / 100 / 1 000 / 10 000 types make up a certain percentage of this text

All data is obtained from a 100 000 sentence corpus of the respective language.

3.2.2. Comparing growth rates

In Figure 4 we compare Figures like 1 for different languages. For simplicity's sake, always one language is compared to the average of all languages. Here we compare Finnish, French, Italian, and Norwegian (bold lines) with the language average (thin lines).

As can be seen, there are considerable differences from the average. These differences are stronger than the intra language variation observed in Section 3.1.1.. We find both parallel and non-parallel behavior. For instance, we find:

	1 million	4 million	10 million	40 million
num. tokens	2 020 882 (98 465)	2 021 002 (81 693)	2 020 851 (65 396)	2 021 958 (2 964)
num. types	154921 (19910)	162324 (21030)	162576 (11424)	166350 (373)
num. s. co-occurrences	438459(26003)	413683 (14460)	405442 (8005)	395641 (1718)
num. n. co-occurrences	169308 (4636)	167248 (3446)	167000 (2180)	166408 (461)
coverage top 10	26.70 (0.18)	26.71 (0.19)	26.69 (0.18)	26.71 (0.18)
coverage top 100	48.42 (0.12)	48.40 (0.16)	48.41 (0.12)	48.43 (0.12)
coverage top 1000	65.81 (0.58)	66.02 (0.89)	66.09 (0.58)	65.03 (0.61)
coverage top 10000	82.62 (1.03)	82.53 (1.60)	82.68 (1.04)	82.57 (1.06)
avg. token length	5.72 (0.0077)	5.73 (0.014)	5.72 (0.0080)	5.72 (0.0077)
avg. type length	11.19 (0.026)	11.22 (0.032)	11.25 (0.026)	11.28 (0.025)

Table 2: Sampling Statistics. Arithmetic means and (standard deviations) for each set of experiments. Very stable features are marked **bold**, less stable features are marked in *italics*.

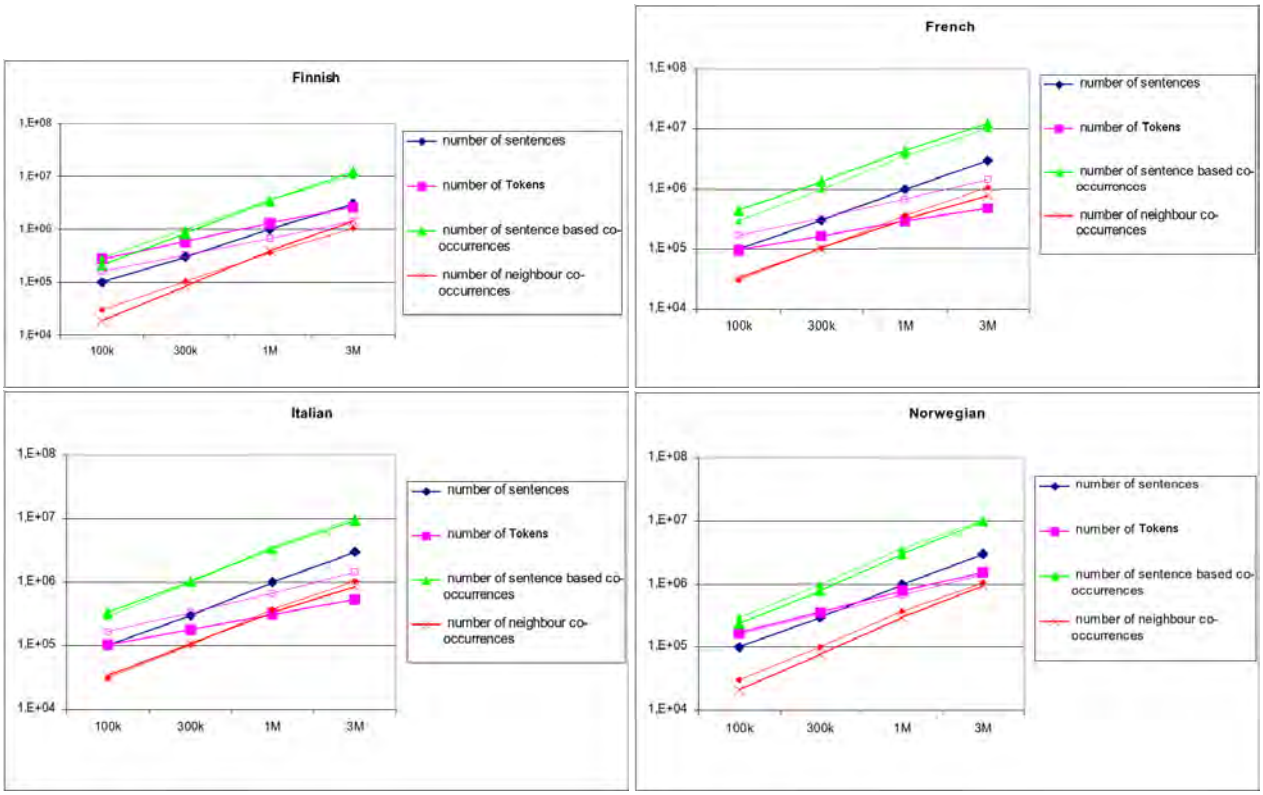


Figure 4: Effects of different corpus size on numbers of types, tokens and co-occurrences for Finnish, French, Italian and Norwegian

- Finnish has more word forms than average, This corresponds to strong morphology and huge average word length.
- In contrast, French and Italian have much less words. Moreover, the increase of the number of tokens is less then average.
- For Norwegian, the number of tokens behaves average. But it seems to have less co-occurrences of both kinds.

4. Using Corpora and Co-occurrences

4.1. Research in Phraseology

To illustrate the possible usage of corpora as a linguistic resource we discuss the usage and the usefulness of the corpora in a research project on phraseology and lexicography in this Section. Since there are no homogeneous definitions of phraseological units, the term is here to be understood in a broad sense covering heterogeneous lexicalized multi word units.

In order to select highly frequent phraseological units for the compilation of a bilingual phraseological database we determined the frequency of over 5000 phraseological units extracted from existing dictionaries for German as a Second Language in the German corpus. The frequency test was carried out in the corpus in April 2002 by using constructed search forms that correspond to possible usage forms of the phraseological units. Furthermore, we analyzed the corpus examples to extract lexicographic relevant data such as frequent syntactic and semantic usage patterns, meaning and semantic variation, external valency, syntactic and morpho-semantic restrictions and any complementary

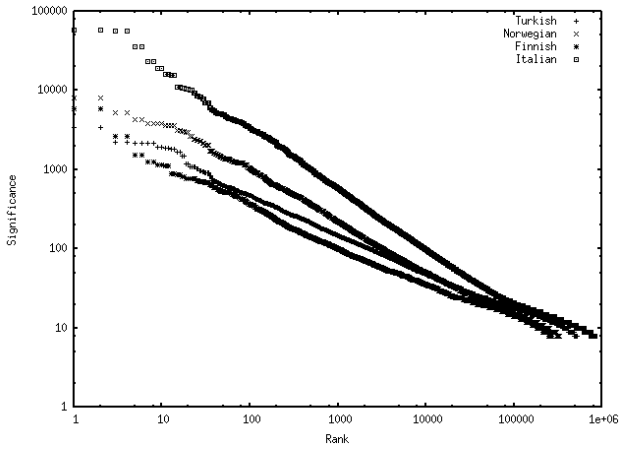


Figure 2: log-log rank - co-occurrence significance diagramm for Turkish, Norwegian, Finnish and Italian

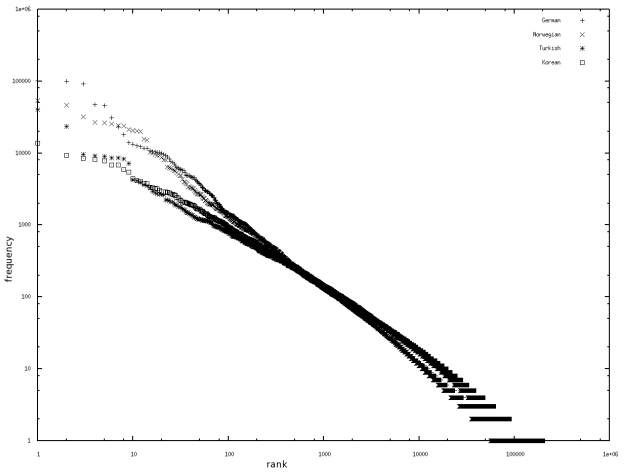


Figure 3: log-log rank - frequency diagramm for German, Norwegian, Turkish and Korean

	nty	nto	tyl	tol
cat	110 034	2 178 029	8.04	4.57
dan	157 560	1 623 436	10.28	5.27
dut	124 986	1 588 453	9.94	5.27
est	191 225	1 401 652	10.37	6.58
fin	266 633	1 206 771	11.80	7.94
fre	101 782	2 352 542	8.54	5.03
ger	183 567	1 816 287	11.78	5.47
ice	155 903	1 787 209	9.84	5.16
ita	105 139	1 842 639	8.81	5.28
nor	165 090	1 551 530	10.26	5.25
sor	170 917	1 764 778	8.16	4.43
swe	169 825	1 503 581	10.32	5.51
tur	200 122	1 319 398	9.21	6.58

Table 3: number of types and tokens, average type and token length

	10	100	1 000	10 000
cat	24.31	45.30	65.20	87.82
dan	19.63	42.58	62.74	83.10
dut	22.53	45.23	65.78	85.54
est	11.61	25.92	47.62	73.28
fin	10.98	20.72	37.48	62.39
fre	21.38	45.73	66.25	88.65
ger	26.69	48.45	65.97	82.54
ice	21.62	40.74	61.22	82.39
ita	17.88	40.59	62.41	85.93
kor	5.68	17.54	37.16	64.33
nor	19.42	41.96	62.05	82.05
sor	15.95	35.37	58.70	79.99
swe	18.76	40.25	60.59	80.93
tur	9.75	19.69	38.80	67.12

Table 4: percentage of text coverage by the most frequent 10, 100, 1 000, 10 000 types

grammatical, lexical and pragmatic information needed to enable the potential non-native users of the database to correctly use the phraseological units (Hallsteinsdóttir, 2005).

The frequency data was then combined with data from a research project on native speakers knowledge about the same phraseological units, whereby we have compiled a list including highly frequent and well known phraseological units that should be integrated in the basic vocabulary of German as a foreign language. This list provides a solid basis for further lexicographic and language teaching work on phraseology, e.g. in relation to the reference levels of the Common European Framework of Reference for Languages (CERF) (Hallsteinsdóttir et al., 2006).

4.2. Co-occurrences as building blocks

On the web site and in the Corpus Browser, we show co-occurrence graphs that depict associations of a target word graphically. Figure 5 makes obvious the idea of how to obtain word senses from co-occurrence graphs, see (Bordag, 2006) for details. The basic idea is to partition the co-occurrences graph into clusters each of which represents one sense.

Other applications include semantic class and taxonomy learning: words have been compared by their co-occurrences, yielding paradigmatic relations, by e.g. (Rapp, 2002).

Promising initial results have been achieved also in the attempt to separate syntagmatic and paradigmatic relations from co-occurrences sets based on typical distances between co-occurring words (Büchler, 2006). This is of course highly language specific and will need further research. A way to refine word sets is to intersect co-occurrence sets as in (Biemann et al., 2004a). To give an example, common right neighbors of apple and plum are fruit, trees, tree, varieties, flavors. The highest-ranked sentence-based co-occurrences excluding neighbors are a collection of fruits and other edible things: pear, cherry, peach, sauce, wine, spice. While these mechanisms usually do not produce 100% pure word sets, they can serve as important selection procedures for augmenting semantic resources.

5. Conclusions and Further Work

We have presented a flexible schema of providing monolingual large natural language resources and given an insight into possible questions that may be answered by it. We have also presented some promising results from corpus

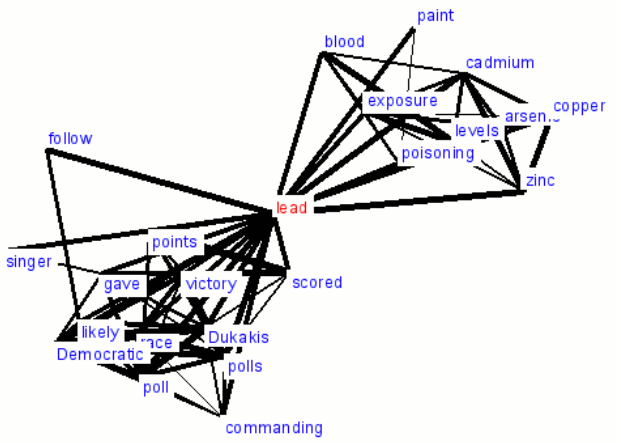


Figure 5: co-occurrence graph for "lead" from English Corpus: two meanings as metal and verb are visually perceivable

use and from inter- and intra-language comparison. Our resources are meant to be growing in size and variety. In the near future, all larger languages, beginning with the official languages in the EU, will be covered. We are open for cooperations and for donations of text in any language.

6. References

Marco Baroni and Adam Kilgarriff. 2006. Large linguistically-processed Web corpora for multiple languages. In *Proceedings of EACL-06, Trento, Italy*.

Marco Baroni and Motoko Ueyama. 2006. Building general- and special-purpose corpora by Web crawling. In *Proceedings of the 13th NIJL International Symposium, Language Corpora: Their Compilation and Application*, pages 31 – 40.

Chris Biemann, Stefan Bordag, and Uwe Quasthoff. 2004a. Automatic acquisition of paradigmatic relations using iterated co-occurrences. In *Proceedings of LREC-04*, Lisboa, Portugal.

Chris Biemann, Stefan Bordag, Uwe Quasthoff, and Christian Wolff. 2004b. Web Services for Language Resources and Language Technology Applications. In *Proceedings of LREC-04*, Lisboa, Portugal.

Stefan Bordag. 2006. Word sense induction: Triplet-based clustering and automatic evaluation. In *Proceedings of EACL-06*, Trento, Italy.

Marco Büchler. 2006. Flexible Computing of Co-occurrences on Structered and Unstructered Text. Master's thesis, Leipzig University.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics, Volume 19, number 1*.

Lukas Faulstich, Uwe Quasthoff, Fabian Schmidt, and Christian Wolff. 2002. Concept Extractor - Ein flexibler und domänen-spezifischer Web Service zur Beschlagwortung von Texten. In *Proceedings of 8. Intl. Symposium für Informationswissenschaft (ISI 2002)*.

Erla Hallsteinsdóttir, Monika Sajánková, and Uwe Quasthoff. 2006. Phraseologisches Optimum für Deutsch als Fremdsprache. Ein Vorschlag auf der Basis von Frequenz- und Geläufigkeitsuntersuchungen. *Linguistik online: Neue theoretische und methodische Ansätze in der Phraseologieforschung*.

Erla Hallsteinsdóttir. 2005. Vom Wörterbuch zum Text zum Lexikon. *Zwischen Lexikon und Text - lexikalische, stilistische und textlinguistische Aspekte*.

Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. *Proc. EURALEX 2004, Lorient, France*.

Adam Kilgarriff. 2001. Web as Corpus. In *Proceedings of Corpus Linguistics*.

Helmut Meier. 1967. *Deutsche Sprachstatistik*. Olms, Hildesheim, 2nd edition.

Uwe Quasthoff and Chris Biemann. 2006. Measuring Monolinguality. In *Proceedings of LREC-06 workshop on Quality assurance and quality measurement for language and speech resources*.

Uwe Quasthoff, Matthias Richter, and Chris Biemann. 2006. Corpus Portal for Search in Monolingual Corpora. In *Proceedings of LREC-06*.

Reinhard Rapp. 2002. The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of COLING-02*, Taipei, Taiwan.

Bengt Sigur, M. Eeg-Olofsson, and J. van de Weijer. 2004. Word length, sentence length and frequency - Zipf revisited. *Studia Linguistica 59:1*.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the LREC-06*.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison. http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

George Kingsley Zipf. 1935. *The Psycho-Biology of Language.* Houghton-Mifflin.

Optimization of Latent Semantic Analysis based Language Model Interpolation for Meeting Recognition

Michael Pucher[†] [‡], **Yan Huang**[*], **Özgür Çetin**[*]

[‡]Telecommunications Research Center
Vienna, Austria
pucher@ftw.at

[†]Speech and Signal Processsing Lab, TU Graz
Graz, Austria

[*]International Computer Science Institute
Berkeley, USA
yan@icsi.berkeley.edu, ocetin@icsi.berkeley.edu

Abstract

Latent Semantic Analysis (LSA) defines a semantic similarity space using a training corpus. This semantic similarity can be used for dealing with long distance dependencies, which are an inherent problem for traditional word-based n-gram models. This paper presents an analysis of interpolated LSA models that are applied to meeting recognition. For this task it is necessary to combine meeting and background models. Here we show the optimization of LSA model parameters necessary for the interpolation of multiple LSA models. The comparison of LSA and cache-based models shows furthermore that the former contain more semantic information than is contained in the repetition of words forms.

Optimizacija latentne semantične analize temelječe na interpolaciji jezikovnega modela za namene razpoznavanja sestankov

Latentna semantična analiza (LSA) definira prostor semantične podobnosti z uporabo učnega korpusa. To semantično podobnost je mogoče uporabiti pri odvisnostih dolgega dosega, ki so inherenten problem za tradicionalne, na besedah temelječe n-gramske modele. Prispevek predstavlja analizo interpoliranih modelov LSA, ki so uporabljeni za razpoznavanje sestankov. Za to nalogo je potrebno združiti modela sestankov in ozadja. Predstavljena je optimizacija parametrov modela LSA za interpolacijo med večimi modeli LSA. Primerjava modelov LSA in modelov s predpomnilnikom pokaže tudi, da prvi vsebujejo več semantičnih informacij kot ponavljanje besednih oblik.

1. Introduction

Word-based n-gram models are a popular and fairly sucessful paradigm in language modeling. With these models it is however difficult to model long distance dependencies which are present in natural language (Chelba and Jelinek, 1998).

LSA maps a corpus of documents onto a semantic vector space. Long distance dependencies are modeled by representing the context or history of a word and the word itself as a vector in this space. The similarity between these two vectors is used to predict a word given a context. Since LSA models the context as a bag of words it has to be combined with n-gram models to include word-order statistics of the short span history. Language models that combine word-based n-gram models with LSA models have been successfully applied to conversational speech recognition and to the Wall Street Journal recognition task (Bellegarda, 2000b)(Deng and Khudanpur, 2003).

We conjecture that LSA-based language models can also help to improve speech recognition of recorded meetings, because meetings have clear topics and LSA models adapt dynamically to topics. Due to the sparseness of available data for language modeling for meetings it is important to combine meeting LSA models that are trained on relatively small corpora with background LSA models which are trained on larger corpora.

LSA-based language models have several parameters influencing the length of the history or the similarity function that need to be optimized. The interpolation of multiple LSA models leads to additional paramters that regulate the impact of different models on a word and model basis.

2. LSA-based Language Models

2.1. Constructing the Semantic Space

In LSA first the training corpus is encoded as a word–document co-ocurrence matrix W (using weighted term frequency). This matrix has high dimension and is highly sparse. Let $\mathcal{V}$ be the vocabulary with $|\mathcal{V}| = M$ and $\mathcal{T}$ be a text corpus containing n documents. Let c_{ij} be the number of occurrences of word i in document j, c_i the number of occurrences of word i in the whole corpus, i.e. $c_i = \sum_{j=1}^{N} c_{ij}$, and c_j the number of words in document j. The elements of W are given by

$$[W]_{ij} = (1 - \epsilon_{w_i}) \frac{c_{ij}}{c_j} \quad (1)$$

where ϵ_{w_i} is defined as

$$\epsilon_{w_i} = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{ij}}{c_i} \log \frac{c_{ij}}{c_i} \ . \quad (2)$$

ϵ_w will be used as a short-hand for ϵ_{w_i}. Informative words will have a low value of ϵ_w. Then a semantic space with much lower dimension is constructed using Singular Value Decomposition (SVD) (Deerwester et al., 1990).

$$W \approx \hat{W} = U \times S \times V^T \quad (3)$$

For some order $r \ll \min(m, n)$, U is a $m \times r$ left singular matrix, S is a $r \times r$ diagonal matrix that contains r singular values, and V is a $n \times r$ right singular matrix. The vector $u_i S$ represents word w_i, and $v_j S$ represents document d_j.

2.2. LSA Probability

In this semantic space the cosine similarity between words and documents is defined as

$$K_{\text{sim}}(w_i, d_j) \triangleq \frac{u_i S v_j^T}{||u_i S^{\frac{1}{2}}|| \cdot ||v_j S^{\frac{1}{2}}||} \ . \quad (4)$$

Since we need a probability for the integration with the n-gram models, the similarity is converted into a probability by normalizing it. According to (Coccaro and Jurafsky, 1998), we extend the small dynamic range of the similarity function by introducing a temperature parameter γ.

We also have to define the concept of a pseudo-document $\tilde{\mathrm{d}}_{t-1}$ using the word vectors of all words preceding w_t, i.e. $w_1, \ldots, w_{t-1}$. This is needed because the model is used to compare words with documents that have not been seen so far. In the construction of the pseudo-document we also include a decay parameter $\delta < 1$ that is multiplied with the preceding pseudo-document vector and renders words closer in the history more significant.

The conditional probability of a word w_t given a pseudo-document $\tilde{\mathrm{d}}_{t-1}$ is defined as

$$P_{\text{LSA}}(w_t|\tilde{\mathrm{d}}_{t-1}) \triangleq \frac{[K_{\text{sim}}(w_t, \tilde{\mathrm{d}}_{t-1}) - K_{\min}(\tilde{\mathrm{d}}_{t-1})]^\gamma}{\sum_w [K_{\text{sim}}(w, \tilde{\mathrm{d}}_{t-1}) - K_{\min}(\tilde{\mathrm{d}}_{t-1})]^\gamma} \quad (5)$$

where $K_{\min}(\tilde{\mathrm{d}}_{t-1}) = \min_w K(w, \tilde{\mathrm{d}}_{t-1})$ to make the resulting similarities nonnegative (Deng and Khudanpur, 2003).

2.3. Combining LSA and n-gram Models

For the interpolation of the word based n-gram models and the LSA models we used the methods defined in Table 1. λ is a fixed constant interpolation weight, and $\propto$ denotes that the result is normalized by the sum over the whole vocabulary. λ_w is a word-dependent parameter defined as

$$\lambda_w \triangleq \frac{1 - \epsilon_w}{2} \ . \quad (6)$$

This definition ensures that the n-gram model gets at least half of the weight. λ_w is higher for more informative words.

We used two different methods for the interpolation of n-gram models and LSA models. The *information weighted geometric mean* and simple *linear interpolation*.

Model	Definition
n-gram (baseline)	$P_{n-\text{gram}}$
Linear interpolation (LIN)	$\lambda P_{\text{LSA}} + (1-\lambda) P_{n-\text{gram}}$
Information weighted geometric mean interpolation (INFG)	$\propto P_{\text{LSA}}^{\lambda_w} P_{n-\text{gram}}^{1-\lambda_w}$

Table 1: Interpolation methods.

The *information weighted geometric mean* interpolation represents a loglinear interpolation of normalized LSA probabilities and the standard n-gram.

2.4. Combining LSA Models

For the combination of multiple LSA models we tried two different approaches. The first approach was the linear interpolation of LSA models with optimized λ_i where $\lambda_{n+1} = 1 - (\lambda_1 + \ldots + \lambda_n)$:

$$P_{\text{lin}} \triangleq \lambda_1 P_{\text{Lsa}_1} + \ldots + \lambda_n P_{\text{Lsa}_n} + \lambda_{n+1} P_{n-\text{gram}} \quad (7)$$

Our second approach was the INFG Interpolation with optimized θ_i where $\lambda_w^{(n+1)} = 1 - (\lambda_w^{(1)} + \ldots + \lambda_w^{(n)})$:

$$P_{\text{infg}} \propto P_{\text{Lsa}_1}^{\lambda_w^{(1)}\theta_1} \ldots P_{\text{Lsa}_n}^{\lambda_w^{(n)}\theta_n} P_{n-\text{gram}}^{\lambda_w^{(n+1)}\theta_{n+1}} \quad (8)$$

The parameter θ_i have to be optimized since the $\lambda_w^{(k)}$ depend on the corpus, so that a certain corpus can get a higher weight because of a content-word-like distribution of w, although the whole data does not well fit the meeting domain. In general we saw that the λ_w values were higher for the background domain models than for the meeting models. But taking the n-gram mixtures as an example the meeting models should get a higher weight than the background models. For this reason the λ_w of the background models have to be lowered using θ.

To ensure that the n-gram model gets a certain part α of the distribution, we define $\lambda_w^{(k)}$ for word w and LSA model Lsa_k as

$$\lambda_w^{(k)} \triangleq \frac{1 - \epsilon_w^{(k)}}{\frac{n}{1-\alpha}} \quad (9)$$

where $\epsilon_w^{(k)}$ is the uninformativeness of word w in LSA model Lsa_k as defined in (2) and n is the number of LSA models. This is a generalization of definition (6). Through the generalization it is also possible to train α, the mimimum weight of the n-gram model.

For the INFG interpolation we had to optimize the model parameters θ_i, the part of the n-gram model α, and the γ exponent for each LSA model.

3. Analysis of the models

To gain a deeper understanding of our models we analyzed the effects of the model parameters and compared our models with other similar models. For this analysis we used meeting heldout data, containing four ICSI, four CMU and four NIST meetings. The perplexities and similarities were estimated using LSA and 4-gram models trained on the Fisher conversational speech data (Bulyko et al., 2003)

and the meeting data (Table 2) minus the meeting heldout data. The models were interpolated using the INFG interpolation method (Table 1).

Training Source	# of words ($\times 10^3$)
Fisher	23357
Meeting	880

Table 2: Training data sources.

3.1. Perplexity Space of Combined LSA Models

Figure 1 shows the perplexities for the meeting and the Fisher LSA model, that were interpolated with an n-gram model using linear interpolation (Definition 7) were λ_1 and λ_2 are the corresponding LSA model weights. Zeros are plotted where the interpolation is not defined, e.g. where $\lambda_1 + \lambda_2 \geq 1$, which would mean that the n-gram model gets zero weight.

This figure shows that the minimum perplexity is reached with $\lambda_1 = \lambda_2 = 0$. Furthermore we can see that the graph gets very steep with higher values of λ. This is beneficial for the gradient descent optimization since we always know where to go to reach the minimum perplexity. The minimum perplexity is however reached when we do not use the LSA model and solely rely on the n-gram.

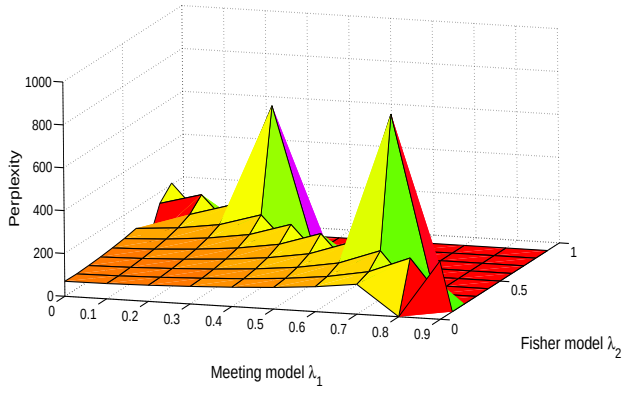


Figure 1: Perplexity space for 2 linearly interpolated LSA models.

Figure 2 shows the perplexity space of the INFG interpolation (Definition 8) for the meeting and the Fisher model that is much flatter than the linear interpolation space. We can estimate the difference in steepness by looking at the perplexity scale, which is $[67, 72]$ for the INFG interpolation compared to $[0, 1000]$ for the linearly interpolated models. Therefore the parameter optimization is harder and slower for this interpolation.

On the other hand we can achieve an improvement over the n-gram model when using this interpolation. The optimum perplexity is not reached when giving both LSA models $\theta_i = 0$, but when setting the parameter for the Fisher model to $\theta_2 = 0$ and the meeting model parameter to $\theta_1 = 1$. The θ_i's have only the function of boolean model selectors in this 2-model case. But there is still the word entropy that is varying the interpolation weight between LSA and n-gram model.

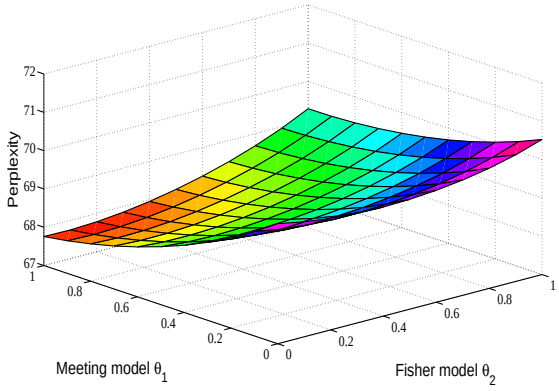


Figure 2: Perplexity space for 2 INFG interpolated LSA models.

When we conducted word-error-rate experiments with combinations of more than two LSA models (Pucher et al., 2006) we used gradient-descent optimization to optimize all the interpolation parameters together. Here we used a brute-force approach to get a picture of the whole perplexity space.

3.2. The Repetition Effect: LSA Models and Cache Models

Some improvements of LSA-based language models over n-gram models are surely due to the redundant nature of language and speech. A lot of words that pop-up in a meeting for example are likely to pop-up again in a short window of context. A word will be highly similar to a context when the word appears in the context. A cache-based language model can exploit this fact by keeping a cache of words that already have been seen, and giving them higher probability (Kuhn and De Mori, 1990). To test if the performance of LSA-based models only rests on this cache-effect we checked the word probabilities of the models.

	+ Meet	+ Fish	− Meet	− Fish
Word in hist.	60%	63%	5%	6%
Word out of hist.	8%	7%	27%	24%
	68%	70%	32%	30%

Table 3: Number of improved LSA word probabilities.

Table 3 shows the number of improved word probabilities for the meeting and the Fisher model on the heldout data. '+' means that the probability of the LSA model was higher than the n-gram model probability, '−' means that it was lower. The end-of-sentence event is not included.

For the meeting model 60% of the improvements are due to the cache-effect where the word appears in the history. This value is so high because we use the decay parameter, so that a word disappears from the pseudo-document, but it still stays in our cache for the whole meeting and increases the cache-effect. So a certain amount of this improvement is actually due to the semantic of the LSA model. This happens because the word vector is decayed

in the pseudo-document but the word stays in the cache for the whole meeting. The percentage of the class +/Word not in hist. has to be increased by this amount.

We can estimate this amount by assuming that each meeting contains ≈ 7500 ($90455/12$ meetings) words, and that the last 100 words are present in the pseudo-document. We know that 60% of the words fall under the category +/Word in hist. (≈ 4500 words). But this is only true if we assume the history to be the whole preceding meeting and not just the last 100 words. The mean length of the history for a document of length k is given by the arithmetic mean $\frac{0+1+2+...+k-1}{k} = \frac{k+1}{2}$.

In our case the mean length of the history is ≈ 3700. So we know that given a mean length of around 3700, 60% of the words fall under the former class, but given a mean length of the history around 100, some improvement also falls into the class +/Word not in hist, which must therefore be significantly higher than 7%. The same reasoning applies to the Fisher model where the performance is even better.

According two t-tests for paired samples the differences between LSA and n-gram models for the following classes are significant: +/Word in hist, −/Word in hist., −/Word not in hist. for the meeting and the Fisher model ($p < 0.05$). The difference within the class +/Word not in hist. is however not significant, but as already mentioned the true size of this class is bigger than the estimated size.

This analysis shows that LSA-based models cannot be simply replaced by cache-based models. Although the repetition effect is important for LSA models they also cover other semantic information.

3.3. The Temperature Effect: γ Exponent Optimization

The temperature parameter γ (Definition 5) is used to extend the small dynamic range of the LSA similarity (Coccaro and Jurafsky, 1998). Here we want to optimize this parameter and show how it changes the LSA similarities.

The similarities were scaled by using the minimum similarity given the history as in Definition 5. Otherwise the exponent would make negative similarities positive.

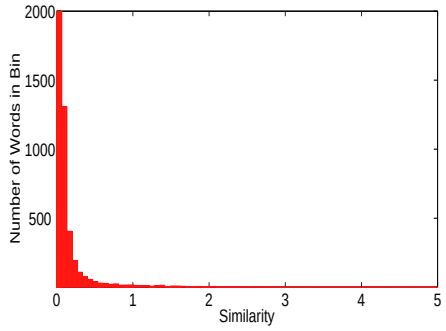


Figure 3: Similarities for $\gamma = 8$.

Figure 3 shows the similarity distribution for a γ value of 8 for the Fisher model on the heldout data. This distribution expanded the similarity range and assembles a lot of similarities around zero.

In Figure 3 all similarities < 1.0 get pruned in comparison to $\gamma = 1$. This is due to the nature of the exponentiation where all values between in $[0, 1]$ get smaller if exponentiated. To change this one can add an offset $\beta \in [0, 1]$ to the similarities to avoid pruning of similarities in the interval $[1 - \beta, 1]$. For $\beta = 1$ there is no pruning since all similarities are bigger than or equal to 1. Than the similarity distribution gets flatter. We also optimized β to find the effect of values that are smaller than 1.

For our work it is interesting to see which γ values optimize the perplexity on the heldout data. Figure 4 shows perplexities of the Fisher model on the heldout data for different values of γ and β.

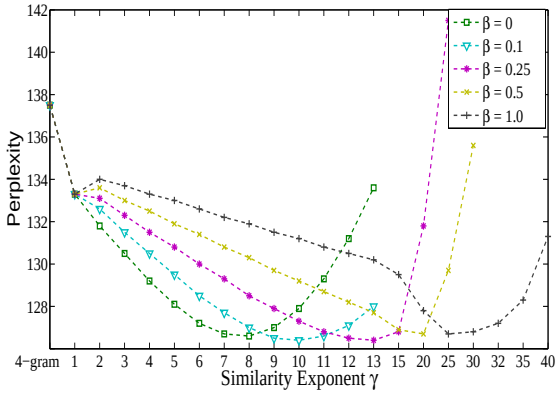


Figure 4: Perplexities for the Fisher LSA model with different γ and β values.

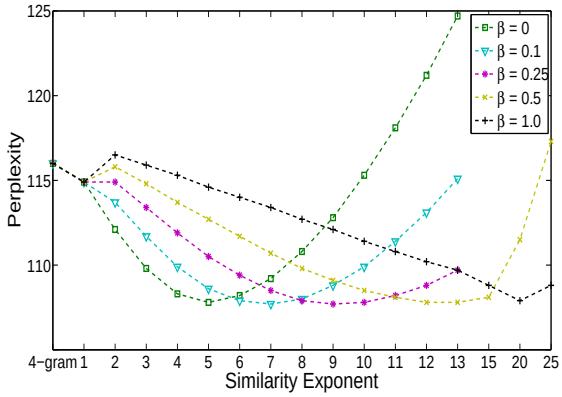


Figure 5: Perplexities for the meeting LSA model with different γ and β values.

One can see that the lowest perplexity for all β values is nearly the same while only the exponent is shifting. It can also be seen that all LSA models outperform the 4-gram model even for $\gamma = 1$. The optimal γ value for the meetings is for all β smaller than for the Fisher model (Figure 5). One generalization we can make from experiments with other models is that the optimal γ value is in general higher for bigger models, e.g. models that are trained on larger corpora. This is also reflected in the relation between the meeting model and the Fisher model which can be seen from figure 5 and 4.

With the first approach one comes up with a much smaller exponent than with the second. We conjecture that the different values of exponents found in the literature ranging from 7 (Coccaro and Jurafsky, 1998) to 20 (Deng and Khudanpur, 2003) are due to the usage of different values of β. Since we do not see a difference in perplexity we conclude that it does not matter which approach one chooses.

The temperature parameter was optimized independently from the interpolation parameters. We found that this value is stable over different test data sets.

3.4. The History Effect: δ Decay Optimization

Here we show how the decay parameter δ influences the perplexity. The perplexity of the 4-gram Fisher and meeting models are again our baselines. As a test set we use again the meeting heldout data. The idea of the decay parameter is to update the pseudo-document in a way that words that were recently seen get a higher weight than words that are in a more distant history. Finally the words that are far away from the actual word are forgotten and have no more influence on the prediction of the actual word. (Bellegarda, 2000a) finds a value around 0.98 to be optimal for the decay parameter δ.

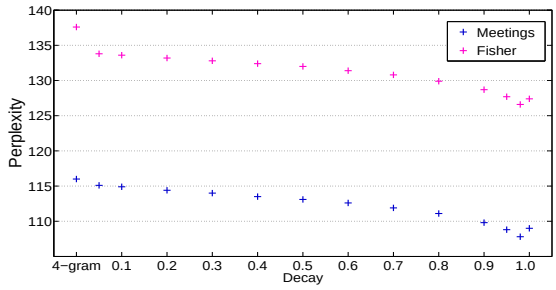


Figure 6: Perplexities for a 4-gram and LSA models with different decays δ.

Figure 6 shows that there is a constant drop in perplexity as we increase the length of the history, e.g. the value of δ. There is however not much difference for the decay value 0.98 and 1.0, which means that no words are forgotten. But even the shortest history with a decay of 0.05 has a lower perplexity than the 4-gram for the Fisher and the meeting model on the heldout data. We can conclude that it is beneficial for our models not to forget too fast but there is no big difference between forgetting very slow and never forgetting. In our experiments we used nevertheless a decay of 0.98 because it still has the best performance concerning perplexity and because it was also found to be optimal by others (Bellegarda, 2000a).

The decay parameter was optimized independently of the interpolation parameter and the temperature parameter.

4. Conclusion

We showed how to optimize the parameters for interpolated LSA-based language models and saw that simple linear interpolation did not achieve any improvements. With the INFG interpolation we achieved an improvement and a model selection.

The comparison between LSA and cache-based models showed that a large amount of the improvement is due to the repetition of words, but there is also an improvement that relies on other features of the LSA-based models. So cache-based models cannot simply replace LSA-based models.

We also presented the optimization of similarity exponent and offset and saw the relation between the offset selection and the similarity exponent.

The optimization of the decay parameter showed that it makes little difference when being close to or equal to one, but a bigger difference when the value gets close to zero.

We can conclude that the optimzation of parameters is crucial for outperforming word-based n-gram language models by interpolated LSA models.

5. References

J.R. Bellegarda. 2000a. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296, August.

J.R. Bellegarda. 2000b. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84, January.

I. Bulyko, M. Ostendorf, and A. Stolcke. 2003. Class-dependent interpolation for estimating language models from multiple text sources. Technical Report UWEETR-2003-0000, University of Washington, EE Department.

C. Chelba and F. Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings of COLING-ACL*, pages 225–231, San Francisco, California.

N. Coccaro and D. Jurafsky. 1998. Towards better integration of semantic predictors in statistical language modeling. In *Proceedings of ICSLP-98*, volume 6, pages 2403–2406, Sydney.

S. Deerwester, S.T. Dumais, G.W. Furnas, and T.K. Landauer. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Scienc*, 41(6):391–407.

Y. Deng and S. Khudanpur. 2003. Latent semantic information in maximum entropy language models for conversational speech recognition. In *Proceedings of HLT-NAACL*, pages 56–63, Edmonton.

R. Kuhn and R. De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583.

M. Pucher, Y. Huang, and Ö. Çetin. 2006. Combination of latent semantic analysis based language models for meeting recognition. In *Computational Intelligence 2006*, San Francisco, USA.

Including deeper semantic information in the Lexical Markup Framework: a proposal

Isabel Segura Bedmar, José L. Martínez Fernández, Paloma Martínez

Department of Computer Science
University Carlos III of Madrid
Avda. Universidad 30, 28911 Leganés, Madrid
{isegura, jlmferna, pmf}@inf.uc3m.es

Abstract

The exploitation of various lexical resources is crucial for many complex Natural Language Processing (NLP) applications. These systems improve their results remarkably if a more intensive exploitation of the lexical and semantic resources is carried out. Therefore, optimizing the production, maintenance and extension of lexical resources is a crucial aspect impacting Natural Language Processing tasks, in particular those related to language understanding. The lexical resources have been built with extensive human effort over years of work, so it would be beneficial to enable the merging of these resources to form extensive global resources and also to define an standard way to interact with this kind of semantic repositories. Lexical Markup Framework (LMF) is a model, sponsored by the International Organization for Standardization, ISO, that provides a common standardized framework for the construction of NLP lexicons. This paper proposes an extension of the semantic part of the LMF metamodel. We hope this extension to improve the metamodel by the inclusion of semantic information considered useful in semantic interpretation of texts, as proved by research in Semantic Role Labeling processes.

Vključevanje globljih pomenskih informacij v LMF (Lexical Markup Framework/Okvir za leksikalno označevanje): predlog

Izkoriščanje različnih leksikalnih virov je ključno za mnogo celovitih aplikacij procesiranja naravnega jezika. Pri teh sistemih se rezultati občutno izboljšajo, če so leksikalni in pomenski viri učinkoviteje izrabljeni. Zatorej je optimiziranje priprave, vzdrževanja in širjenja leksikalnih virov ključno pri nalogah procesiranja naravnih jezikov, posebej tistih, povezanih z razumevanjem jezika. Leksikalni viri so bili zgrajeni z veliko človeškega truda v več letih, zato bi bilo koristno omogočiti njihovo združevanje in tako oblikovati obsežne globalne vire, prav tako pa določiti standardne pristope za delo s tovrstnimi pomenskimi zbirkami. LMF je model, ki ga podpira Mednarodna organizacija za standardizacijo (ISO) in v okviru katerega se pripravlja skupni standardizirani okvir za gradnjo leksikonov za procesiranje naravnega jezika. V članku predlagamo razširitev pomenskega dela metamodela LMF. Upamo, da bo ta razširitev izboljšala metamodel glede vključevanja pomenskih informacij, uporabnih pri pomenski interpretaciji besedila, kot se je to potrdilo pri raziskavi procesiranja oznak pomenskih vlog.

1. Introduction

The goals of a semantic parser are to identify the semantic relations between the words, and the construction of a structure allowing the interpretation of the meaning of the text (Shi and Mihalcea, 2005).

The identification of the semantic roles is a crucial part in the interpretation of texts (Gildea and Palmer, 2002), and therefore is important for information extraction and retrieval, question answering, natural language interfaces etc. (Hacioglu et al., 2003), (Melli et al., 2005).

In the last decade, the work in the information extraction research field has shifted from complex rule-based systems (Alshawi, 1992) to simpler finite-state or statistical systems such as (Hobbs et al., 1997) and (Miller et al., 1998). These systems have been used in the extraction of relations for specific semantic domains such as terrorist events in the framework of the DARPA Message Understanding Conferences. Other commercial systems have incorporated knowledge representation techniques traditionally used in IA, like frames or context-dependent templates.

Nowadays, the challenge is to be able to develop domain-independent systems or, at least, systems easily adjustable to any semantic domain. The semantic role labelling systems use lexical resources like VerbNet (Kipper, Dang and Palmer, 2000), PropBank (Kingsbury, Palmer and Marcus, 2002), or FrameNet (Baker, Fillmore, Lowe, 1998). The semantic role labelling systems improve their results remarkably if a more intensive exploitation of the lexical resources is carried out (Brharati, Venkatapathy and Reddy, 2005). These resources were built with extensive human effort over years of work. Hence, it would be beneficial to enable the merging of these resources to form extensive global resources.

The creation of a standard on lexicons can be a useful aid for the construction and maintenance of the lexical resources, and for their integration into natural language processing systems. LMF (ISO 24613) is a model that provides a common standardized framework for the construction of NLP lexicons. The goals of LMF are: to provide a common model for the creation and use of lexical resources, to manage the exchange of data among these resources, and to enable the merging of large number of individual electronic resources to form extensive global electronic resources.

The aim of this paper consists to propose a set of improvements to LMF model in particular, in the semantic level, in order to improve the creation and the integration of lexical resources. The proposed extension is based on the analysis and study of research works in the fields of Semantic Role Labeling and lexical resources applications, like Information Retrieval, Information Extraction and Question Answering.

In section 2, the semantic roles and some related linguistics theories are treated. In section 3, the main lexical resources are described. In section 4, several previous works on standards for lexical resources are reviewed. In section 4, the model proposed by standard ISO 24613 is described. In section 5, our approach is developed and finally, in section 6, some conclusions are considered.

2. Semantic Roles

The semantic roles describe the semantic relation (non grammatical) that the arguments have with respect to the predicate of a sentence (usually a verb). Other terms used for their denomination are: thematic roles, semantic cases, thematic relations, semantic arguments, etc.

A semantic role describes an abstract function carried out by an element taking part in an action. This abstract function is defined regardless of the syntactic realizations that the element can acquire into a sentence. So, the semantic roles allow for the representation of generic actions, regardless of the language and the diverse grammar resources that a language offers to express the same action (Cook, 1989).

In the following sentences, the semantic roles of the predicate *to break* have different syntactic realizations:

[John $_{\text{Agent}}$] [broke $_{\text{V}}$] [the window$_{\text{Object}}$] with [the hammer $_{\text{Instrument}}$]

[The window $_{\text{Object}}$] [was broken $_{\text{V}}$] by [John $_{\text{Agent}}$]

[The hammer. $_{\text{Instrument}}$] [broke $_{\text{V}}$] [the window $_{\text{Object}}$]

In contrast to the syntactic level, where there is, more or less, agreement among the linguistic community about the syntactic components and their definition, the semantic level does not reach that degree of agreement when semantic roles and their characteristics must be stated.

The majority of abstract roles have been proposed by linguists as part of the *Linking* Theory (Levin and Rappaport, 1996) - the part of grammatical theory that describes the relationship between semantic roles and their syntactic realizations- which is more concerned with explaining generalizations across verbs in the syntactic realizations of their arguments.

The Proto-Role theory is most abstract and was proposed by (Valin, 1993), (Dowty, 1991). This theory has only two roles: Proto-Agent, Proto-Patient.

Fillmore (Fillmore, 1968) proposed a grammar of cases that classified the verbs according to the frames of cases or the necessary roles demanded by a verb. One of the essential elements of the model was a small set of roles universal, that is to say, generic enough to be valid for all the languages.

The more specific roles have been proposed by computer scientists, who are more concerned with the details of the realization of the arguments for specific verbs**.** For example, if a flight information system is considered, some specific roles could be: FROM_AIRPORT, TO_AIRPORT, DEPART_TIME, or verb-specific roles such as EATER and EATEN for the verb *eat*.

Finally, it is important to emphasize the difficulty in the identification of the semantic roles. The main reason is that there is no a direct mapping between the syntax and the semantics.

3. Review of the main linguistic resources containing semantic information

As already stated, the linguistic information is crucial in many NLP tasks. If semantic role labeling systems are considered, the use of this type of resources is essential.

FrameNet is based on the theory of semantic frames (Fillmore, 1976), where each frame corresponds to an interaction and its participants (roles). A frame has an appropriate name to describe the semantic relation defined by the semantic roles. The frame elements (roles) proposed by FrameNet are specific of each frame.

FrameNet includes corpus of annotated sentences with semantic roles. The corpus can be used to learn how to identify semantic relations starting with syntactic structures.

Its main disadvantage is that it does not define selection restrictions for semantic roles. In addition, the coverage of FrameNet (3040 verbs) and its scalability are seriously limited.

PropBank is a corpus in which verbs are annotated with semantic tags, including coarse-grained sense distinctions and predicate-argument structures. PropBank is based on the verbal classification introduced by Levin (Levin, 1993), that assumes there is a strong connection between syntax and semantic. The verbs are grouped together based on their syntactic behaviour and the resulting clusters are coherent from a semantic point of view as all verbs in one Levin class share the same semantic roles. The clusters are formed at a grammatical level according to diathesis alternation criteria. The arguments (roles) of PropBank are specific of each verb.

VerbNet is a verb lexicon providing detailed syntactic-semantic descriptions of Levin classes. As a result, the main hypothesis of VerbNet is that the syntactic frames of a verb are a direct reflection of the underlying semantic.

The main advantage of VerbNet is that it offers a hard generalization of the syntactic behavior of verbs. In addition, VerbNet provides selection restrictions for its roles. A selection restriction marks the semantic category to which the argument's header belongs to. Another remarkable advantage of VerbNet is that each verb entry is already linked to WordNet (Fellbaum, 1998), with a list of possible senses. In addition, it has a wider coverage than FrameNet (4159 verbs, as opposed to 3040 verbs of FrameNet; 2398 defined in both resources).

The main VerbNet drawback is that thematic roles are too generic to capture similar scenarios to those represented by semantic frames of FrameNet.

WordNet is a lexical database of nouns, verbs, adjetives and adverbs. Closed categories (prepositions, conjunctions, etc.) are not represented, as they are considered part of the syntactic knowledge, not of the semantic knowledge. The main disadvantage of WordNet is that it does not codify the syntactic behavior of the verbs.

The lexical resources are scarce but very valuable information. (Shi and Mihalcea, 2005) propose the integration of the lexical resources FrameNet, VerbNet and WordNet. Each of these resources encodes a different kind of knowledge and has its own advantages, so their combination can eventually result in a richer knowledge-base that could enable a more accurate and robust semantic parsing.

Few automatic methods for semantic classification exist, mainly due to the lack of resources with semantic information.

4. Standards for Lexical Resources

Several attempts have been made in the standardization of linguistic processes and resources. This section describes some of the main initiatives in this line.

GENELEX was a EUREKA project that had several aims and one of them was to design a global model to represent all kind of lexical information (for monolingual morphology, syntax and semantics, and multilingual correspondences), in a neutral mood, independent of applications and not directly linked to a particular theory. Furthermore, the project pursued to build adapted tools to create and maintain such lexicons. In addition, the effectively creation of large size lexical data in this model was considered.

EAGLES[1] (Expert Advisory Group on Language Engineering Standards) was an initiative of the European Commission, within DG XIII *Linguistic Research and Engineering* program, which aimed to accelerate the provision of standards for: very large-scale language resources (such as text corpora, computational lexicons and speech corpora); means of manipulating such knowledge, via computational linguistic formalisms, mark up languages and various software tools; means of assessing and evaluating resources, tools and products.

ISLE[2] (International Standards for Language Engineering) is both the name of a project and the name of an entire set of co-ordinated activities regarding the Human Language Technology (HLT) field. ISLE acted under the aegis of the EAGLES. The aim of ISLE was to develop HLT standards within an international framework, in the context of the EU-US International Research Cooperation initiative.

Its objectives were to support national projects, HLT RTD projects and the language technology industry in general by developing, disseminating and promoting de facto HLT standards and guidelines for language resources, tools and products.

MULTEXT provided specific guidance for the purposes of NLP and MT corpus-based research. MULTEXT tackled the definition of a software standard, an essential step toward reusability, and publishing the standard to enable future development by others.

PAROLE[3] was an EU funded project which aimed to build harmonized lexica and corpora in all languages of the Union. This allows multi-lingual links to be made at the same formal linguistic level (morphological, syntactic and semantic) and at the same level of descriptive granularity. The project took account of previous research into encoding lexica and corpora using standard, non-language specific formats

SIMPLE[3] was a project sponsored by the IV European Framework Program. This project represented the first attempt to develop wide-coverage semantic lexicons for a large number of languages, with a harmonized common model that encodes structured "semantic types" and semantic frames.

[1] http://www.ilc.cnr.it/EAGLES96/home.html

[2] http://www.mpi.nl/ISLE

[3] http://www.ub.es/gilcub/SIMPLE/simple.html

5. Lexical Markup Framework

The sub committee ISO-C37 elaborated a standard for the management of terminology (Terminology Markup FrameWork, ISO 16642), and later, decided to construct standards for natural language processing. ISO 24613, published under the name "Language resource management – Lexical markup framework", provides a common model for the creation and use of lexical resources. In addition, the model makes it possible to manage the exchange of data among linguistic resources and to enable the merging of a large number of individual electronic resources to form extensive global resources.

The same specifications are to be used for both small and large lexicons. The descriptions range from morphology, syntax and semantic to translation. The range of targeted NLP applications is not restricted.

The LMF specification complies with the modeling principles of Unified Modeling Language, UML (Rumbaugh, Jacobson, and Booch. 2005) as defined by OMG[4].

LMF is composed of two components: a *core package* which describes the basic hierarchy of information in a lexical entry and some *extensions of the core package* that describe the reuse of the core components in conjunction with the additional components.

In Figure 2, the UML class diagram of the core package is presented. The class *Database* represents the entire resource and is a container for one or more lexicons. The class *Lexicon* is the container for all the lexical entries of the same language within the database.

The *Lexical Entry* is a container for managing the top level language components. As a consequence, the number of single words, multi-word expressions and affixes of the lexicon is equal to the number of lexical entries in a given lexicon. The Form and Sense classes are parts of the Lexical Entry. The Form consists of a text string that represents the word. The Sense disambiguates the meaning and context of a form. Therefore, the Lexical Entry manages the relationship between sets of related forms and their senses.

The current LMF extensions are described as UML packages (Figure 1).

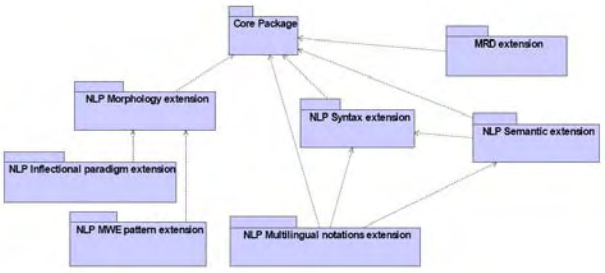


Figure 1: Extensions of the core package UML

Creators of lexicons should select the subsets of the possible extensions that are relevant to their needs. All extensions conform to the LMF core model in the sense that some of the core package classes are extended. An extension cannot be used to represent lexical data regardless of the core package.

[4] www.omg.org

In Figure 3, the semantic extension of the model is represented. The purpose is to describe one sense and its relations with other senses belonging to the same language. LMF propose several descriptive mechanisms like synsets, predicates, relations or linkage with syntax. Due to the intricacies of syntax and semantics in most languages, the section on semantics comprises also the connection to syntax.

The most important classes shown in Figure 3 are *Sense*, *SemanticPredicate* and *SynSet*. The class *Sense* is described in the core package. *SemanticPredicate* is an element that describes an abstract meaning together with the association with Semantic Arguments (*SemanticArgument)*. A semantic predicate may be used to represent the common meaning between different senses that are not necessarily fully synonyms.

Synset links synonyms. *Synset* is an element that describes a common and shared meaning within the same language. Synset may link senses of two different lexical entries with the same part of speech.

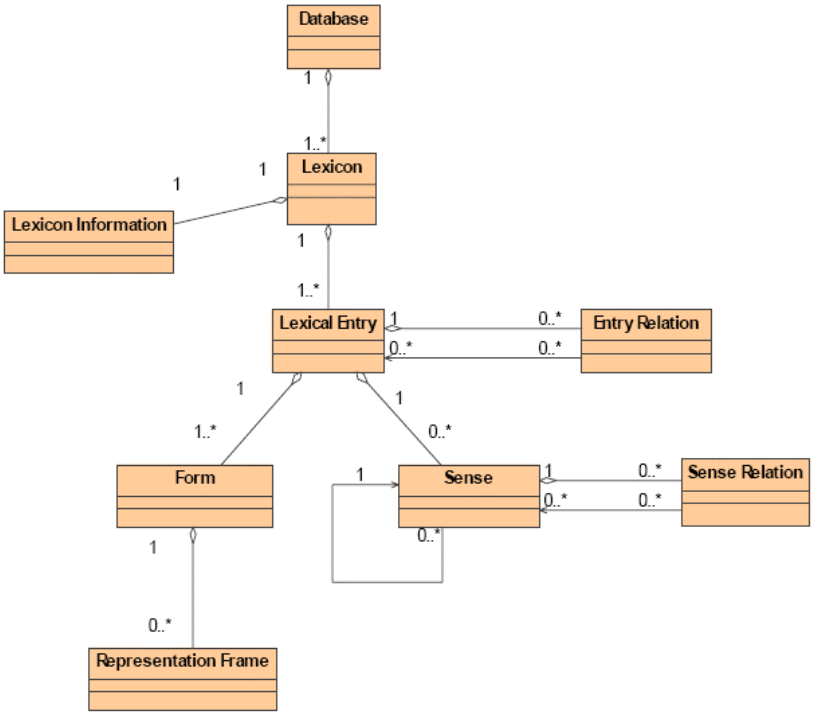


Figure 2. Core Package LMF

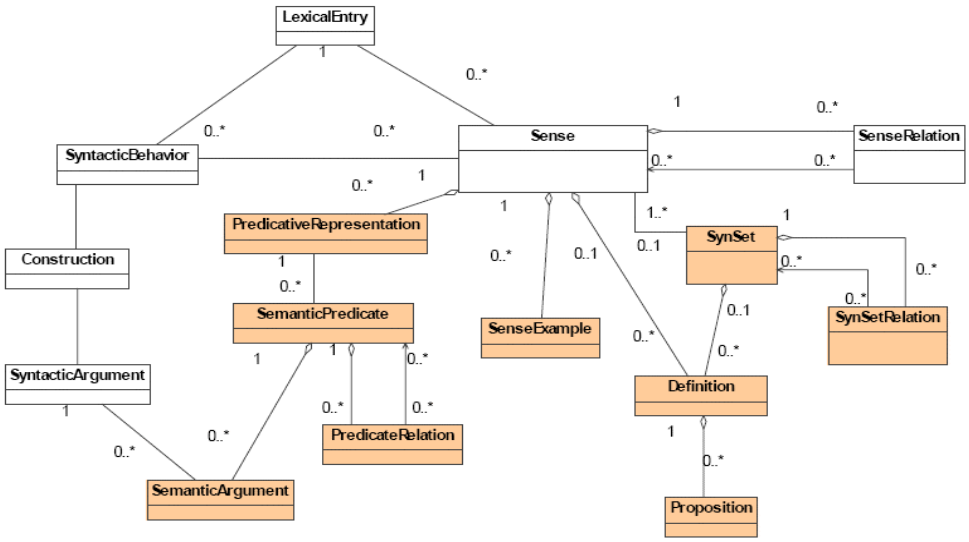


Figure 3: *Extension for Semantic LMF*

6. Proposed classes for the semantic part of LMF

In this section, we propose new classes for the semantic part of LMF. These new classes, denoted as *SemanticClass* and *SelectionalRestrictions* in Figure 4, correspond with the lexical features which have been found to be useful when studying application of lexical resources and semantic role labeling techniques. The final goal of this proposal is to provide the LMF model with the needed elements to comprise existing resources like VerbNet, PropBank, FrameNet and WordNet. These lexical resources, among other things, contain information about thematic roles, which is crucial when the semantic interpretation of texts is considered.

The semantic role labeling system that has obtained the best results until year 2005 was proposed by (Pradhan et al, 2005). The evaluation showed that the features *verb semantic class* and *verb sense* improved its performance. In a previous section, VerbNet and FrameNet resources have been described, both containing representations for groupings of lexical units regarding their semantic meaning. In PropBank or VerbNet, the focus is put on verbs, which can drive the semantic interpretation of a sentence, while FrameNet syntactic categories as nouns, adjectives and others are also considered. These semantic classes cannot be matched against the SynSet class proposed in Figure 3, because this class groups lexical entries with the same syntactic category. This is the reason to include *SemanticClass* in the metamodel. This SemanticClass would include this groups of senses, according to these resources. It is worth mentioning that the frames defined in VerbNet, FrameNet and others could be related with the class SemanticPredicate already defined in the LMF model but these frames can represent one or several semantic classes, depending on the lexical repository considered.

Furthermore, (Brharati, Venkatapathy and Reddy, 2005) showed that the sub-categorization frames help in predicting the semantic roles of the mandatory arguments, thus improving the overall performance. VerbNet defines for each verb class a set of thematic roles (*SemanticArgument)* and a set of syntactic frames (which can be included in the class *SemanticPredicate* defined in LMF) in which these roles are expressed. In addition, VerbNet defines selection restrictions (*Selectional Restrictions*) for the roles of each one of the classes (*+animate, +organization, +communication, +machina, +concrete, + +abstract, etc*). These restrictions are valuable information for determining which arguments correspond with the proper semantic roles. In Figure 4, the class *Selectional Restrictions* is included as an associative class between classes *SemanticClass* and *SemanticArgument* and it must take values from the *SynSet* class.

PropBank does not define semantic selection restrictions for its arguments, but these could be obtained easily, because PropBank and VerbNet are based in the same verbal classification (Giuglea and Moschitti, 2004). In this case, the semantic class of the head word can be useful to determine the correspondence between the syntactic components and the semantic arguments of PropBank. The head word of the noun phrase, and other lexical features, have generated good results in the classification task (Gildea and Jurafsky, 2002), (Pradhan et al., 2005), but these lexical features produce a large dispersion in the data, causing noise in the classification. In this case, it can be useful to use its semantic class (obtained from WordNet) with the purpose of reducing the noise in the classification.

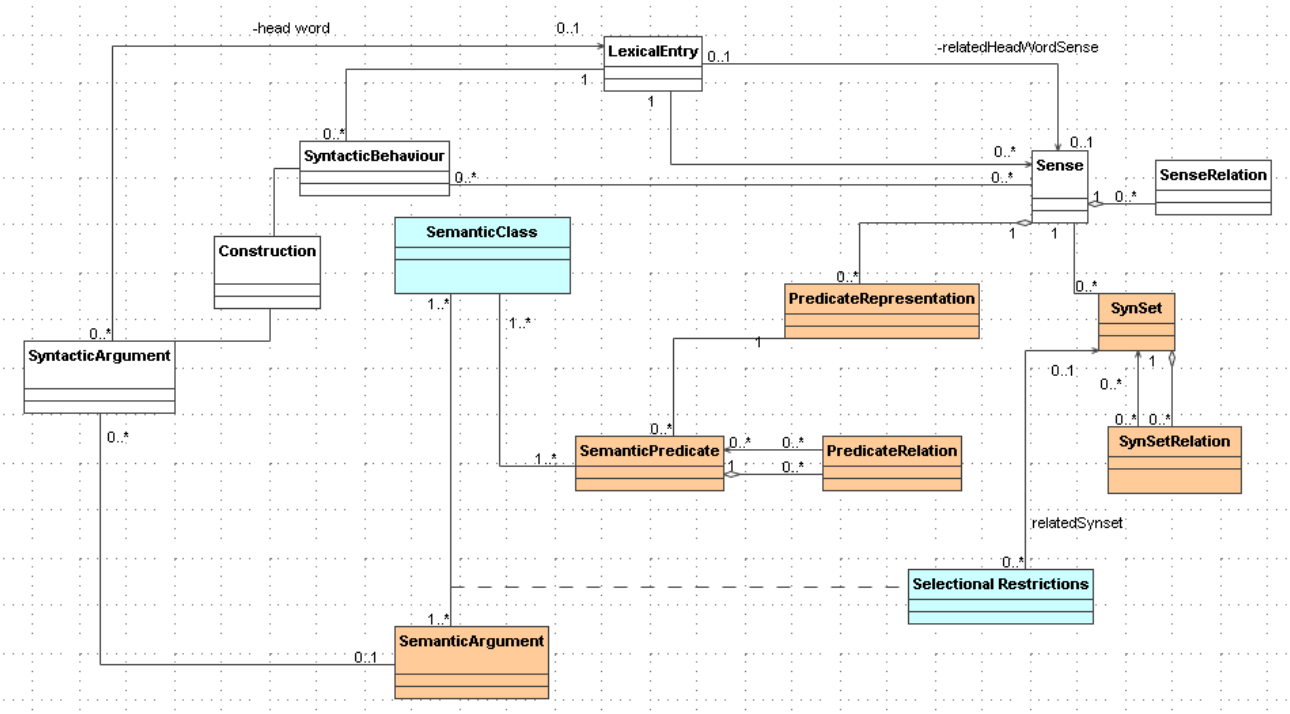


Figure 4. Proposed Semantic Extension of LMF model

In Figure 4, a relation *head word* exists for each noun phrase, so a relation is necessary between the classes *SyntacticArgument*, that represents a noun phrase, and *LexicalEntry*, that represents a word. Furthermore, a new relation *relatedHeadWordSense* is added to represent the semantic class of the head word appropriate for the syntactic argument.

Although specific resources like VerbNet, FrameNet or WordNet have been studied to propose the mentioned new set of classes, the identified elements can be considered generic enough to have a representation in the semantic extension of the LMF metamodel.

7. Conclusion

The availability of semantic information is a crucial issue in the interpretation of texts, and therefore it is important for many tasks related with Natural Language Processing such as Information Extraction, Question Answering or Information Retrieval.

Current lexical resources are small and expensive to produce and maintain. So, it is important to be able to combine them to construct resources with a wider coverage. The creation of a standard fixing the structure and interfaces to be provided by lexical repositories can be a useful aid in the construction and maintenance of these kind of resources and in their integration within Natural Language Processing applications.

In the present work, the LMF standard has been reviewed, and we have proposed several extensions for the semantic part of the LMF metamodel. These extensions are considered to be beneficial for systems where semantic interpretation of texts is pursued.

8. References

Alshawi, H., ed. 1992. The Core Language Engine. *Cambridge, MA: MIT Press.*

Baker, C. F., C. J. Fillmore, J. B. Lowe. 1998. The Berkeley Framenet Project. In *Proceedings of the International Conference on Computational Linguistics (COLING/ACEL-98)*, páginas 86-90, Montreal.

Brharati, A., S. Venkatapathy, P. Reddy. 2005. Inferring semantic roles using sub-categorizacion frames and maximum entropy model. In *Proceeding of CoNLL'2005 Shared Task.*

Cook, W. A. 1989. Case Grammar Theory. *GEORGETOWN UNIVERSITY PRESS, WASHINGTON, D.C.*

Dowty, D. 1991. Thematic Proto-roles and Argument Selection. In *language*, 67.

EAGLES, 1996. Evaluation of Natural LanguageProcessing Systems. *Final Report, Center for Sprogteknologi, Copenhagen.*

Fellbaum, C. editor. 1998. WordNet: An Electronic Lexical Database. *Language, Speech and Communications. MIT Press, Cambridge, Massachusetts.*

Fillmore, C. J. 1968. The case for case. In *EmmonW. Bach and Robert T. Harms, editors, Universals in Linguistic Theory. Holt, Rinehart &Winston*, New York, páginas 1–88.

Fillmore, C. J. 1971. Some problems for case grammar. *In R. J. O'Brien, editor, 22nd annual Round Table. Linguistics: developments of the sixties – viewpoints of the seventies, volume 24 of Monograph Series on Language and Linguistics. Georgetown University Press,Washington D.C., páginas 35–56.*

Fillmore, C. J. 1976. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: conference on the Origin and Development of Language and Speech*, volume 280, páginas 20-32.

Gildea, D. y D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computacional Linguistics*, 28(3):245-288.

Gildea, D. y M. Palmer. 2002. The Necessity of Parsing for Predicate Argument Recognition. In *Proceedings of ACL 2002*, Philadelphia, USA.

Giuglea, A. M. y A. Moschitti. 2004. Knowledge Discovering using FrameNet, VerbNet and PropBank. In *Proceedings of the Workshop on Ontology and Knowledge Discovery at ECML 2004*, Pisa, Italia.

Hacioglu, K., S. Pradhan, W. Ward, J. Martin, D. Jurafsky. 2003. Shallow Semantic parsing using support vector machines. Technical Report TR-CSLR-2003-1, Center for Spoken Language Reserach, Boulder, Colorado.

Hobbs, J. R., D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, M. Tyson. 1997. ``FASTUS: A Cascaded Finite-State Transducer for Extraction Information from Natural Language Text". In *Emmanuel Roche and Yves Schabes, editors*, Finite-State Language Processing, capítulo 13, páginas 383-406. MIT Press, Cambridge, Massachusetts, Londres.

ISO 24613 Language resource management – Lexical markup framework. ISO Geneva 2005.

Kingsbury, P., M. Palmer, M. Marcus. 2002. Adding semantic annotation to the Penn Treebank. In *Proceedings of the Human Language Technology Conference*. San Diego. CA

Kipper, K., H. T. Dang, M. Palmer. 2000. Class-Based Construction of a Verb Lexicon. In *AAAI-2000*, Austin TX.

Levin, B. 1993. English Verb Classes and Alternations: A Preliminary Investigation. The University of Chicago Press.

Levin, B. y M. Rappaport. 1996. From lexical semantics tu argument realization manuscript.

Melli, G., YangWang, Y. Liu, M. M. Kashani, Z. Shi, B. Gu, A. Sarkar, F.Popowich. Description of SQUASH, the SFU Question Answering Summary Handler for *the DUC-2005 Summarization Task.*

Pradhan, S., K. Hacioglu, V. Krugler, W. Ward, J. H. Martin, D. Jurafsky. 2005. Support Vector Learning for Semantic Argument Clasification. *Machine Learning*, 60, 11-39.

Rumbaugh, J., I. Jacobson, y G. Booch. 2005. The Unified Modeling language reference manual, *2ª ed, Addison Wesley* 2005.

Shi, L. y R. Mihalcea. 2005. Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing, In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*, Méjico.

Valin, V. y D. Robert. 1993. A synopsis of role and reference grammar. In *Robert D. Van Valin, editor, Advances in Role and Reference Grammar*. John Benjamins Publishing Company, Amsterdam, páginas 1–166.

Pronoun Generation for Text Summarization and Question Answering

Mehdi M. Kashani, Fred Popowich

School of Computing Science
Simon Fraser University
8888 University Drive, Burnaby, BC, Canada
{mmostafa, popowich}@sfu.ca

Abstract

An algorithm for pronoun generation is introduced as part of a summarization and question answering system. The algorithm makes use of Lingpipe, a coreference resolution tool, as a key component of a process to generate the appropriate pronouns. The two phased algorithm makes use of a replacement phase, followed by a validation phase which makes use of information obtained by a parser. At the end, initial results performed on a collection of DUC 2005 documents are provided.

Tvorjenje zaimkov v sistemih povzemanja besedila in odgovarjanja na vprašanja

Predstavljen je algoritem za tvorjenje zaimkov kot del sistema povzemanja in odgovarjanja na vprašanja. Algoritem uporablja LingPipe kot ključno komponento postopka za tvorjenje ustreznega zaimka. Dvostopenjski algoritem uporablja fazo zamenjave, ki ji sledi faza validacije, pri kateri se uporablja informacije, pridobljene s avtomatskim razčlenjevanjem. Na koncu so predstavljeni prvi rezultati delovanja sistema na zbirki dokumentov DUC (Document Understanding Conferences).

1. Introduction

Summarization and question answering are both examples of natural language processing systems that produce natural language output, and thus require some sort of text generation module. The degree of sophistication in the text generation can vary widely, but given the high frequency of pronouns in natural language text, it is natural to expect that a proper treatment of pronouns in summaries and responses might lead to better quality output. We examine this issue by exploring an approach to pronoun generation which incorporates a pronoun *resolution* module as part of the generation process.

Little attention has been paid to pronoun generation and the focus has always been on coreference resolution. The reason can be attributed to the lack of a good benchmark for evaluation and/or scarcity of real language generation systems that the pronoun generation module can be plugged into.

Sometimes resolution algorithms can be viewed as clues to generation. The first rule of a centering model proposed by Grosz et al (1995) can be interpreted as an acceptance criterion for pronoun generation. However, this is only a special case and no one has really implemented the idea in a generation framework.

McCoy et al. (1999) hypothesize that discourse structure is indeed vital in the decision of whether or not to generate a pronoun. To prove their claim, they choose the shift in time scale as a signal of change in the deictic center of the story. Based on time clues, they segment the text into different threads and in their algorithm state that if the current and previous references to X are in the same thread a pronoun is preferable and otherwise a definite description is used. In case of ambiguities, they use a reference resolution algorithm (Strube, 1998) and check if the pronoun would resolve to X in which case it is permitted to use a pronoun, otherwise not. By using these rules, they show a reduction of error rate by 28.9% compared to a baseline. We take a similar and simpler approach for another task and show an improvement.

This paper introduces a pronoun generation approach used in the course of a summarization and question answering system (Melli, 2005). The task was to find answers, less than 250 words, to fifty questions using the given corpus of relevant documents. These documents came from the Financial Times of London and from the Los Angeles Times.

The general approach taken in the summarization task involved a linguistic analysis of each sentence in the corpus, performing not only named entity extraction, but also anaphora resolution, in which each pronoun in the document was tagged with the entity corresponding to its antecedent. Sentences were then selected from the various documents, and combined to form a summary. Note that by replacing pronouns with their antecedents, it allows the use in the summary of a sentence that originally contained a pronoun, even if the sentence containing the antecedent is not included in the summary. Thus, "dangling" pronouns are avoided.

As is measured in (Vicedo and Ferrandez, 2000) the ratio of pronominal reference used in news collections can be as high as 55%. So, to make the final text smooth and fluent, pronoun generation is essential.

In this paper we focus only on third person singular pronouns where they are in contexts handled by Lingpipe[1]. In the task for which we used our approach, sentences are wholly extracted from original documents. Reflexive pronouns and cataphora are generally both intra-sentence concerns and their generation is not needed in the task. Also note that first and second person pronouns are never considered in pronoun generation systems, because their generation requires the change of sentence structure, something which is usually not desired (for example verbs should change as well). The cases where the proper noun acts as an adjective (as in "the

[1] Lingpipe is a suite of natural language processing tools written in Java that performs tokenization, sentence detection, named entity detection and co-reference resolution on text. The input is plain text and output is an XML file with embedded tags inside the original text.

Castro government") are not dealt with in this paper either.

In this paper, we first explain in section 2 the algorithm in detail and in section 3 our results are provided. We conclude in section 4 with some suggestions for possible enhancements.

2. Our Algorithm

As mentioned above, Lingpipe can find all the referents to a specific entity, so by running Lingpipe once on the text, we would have a chain of entities, all with the same referent. Our goal is for the latter entities to be systematically replaced by pronouns referring to the former entities. The algorithm can be divided into two phases: *replacement* and *validation*.

In the first phase, an appropriate pronoun is chosen and the text is regenerated with the specific entity replaced by this pronoun. Then, Lingpipe is used to validate the replacement. In case of valid replacement, the pronoun will remain in the final text.

Nearly all of the existing algorithms for anaphora resolution identify a part of the text surrounding the pronoun that will be inspected for the candidate antecedent. Lappin and Lease (1994) and Mitkov (1998) use the preceding three and two sentences respectively. Gaizauskas and Humphreys (1996) use the same paragraph that the pronoun is located. As sentences in news articles tend to be long, we chose the distance as at most two sentences. The sentence boundary detector in Lingpipe was used for this task.

In order to suggest a pronoun out of the pronoun set (he, she, his, her, him), we have to deal with gender and case (nominative, accusative, possessive) as part of the replacement phase. Since Lingpipe is not able to guarantee grammaticality, we cannot deal with grammaticality when we use it in the validation phase.

The gender recognition task is itself performed in four consecutive phases. The first, third and fourth phase are general and the second phase takes advantage of the information available in previous stages in the pipeline.

First, the summary is checked to see if we can resolve gender using existing referring pronouns. Second, in the annotated document set, named entity information for all the original documents exists and is used to extract gender information. Third, if some entities remain unresolved (either because they are not referenced by a pronoun or the co-reference is not detected by Lingpipe) an online database of 10079 international frequently used names[2] is used. Fourth, the prefix courtesy titles (ex. Mr) are applied, overriding all of the above. If the gender of an entity cannot be distinguished after these four phases, its gender is marked as *male* (due to the dominance of male entities in news articles).

In order to choose between different types of pronouns (nominative, accusative, possessive) the information available from the parse is used. Specifically, the following rules are applied:

1. If most of the prepositions precede the entity and it is not followed by 's, the replaced pronoun should be accusative (him, her). These prepositions do not include all of the words labelled as PP in the parser[3].
2. If most of the prepositions precede the entity and the entity is followed by 's, the replaced pronoun should be possessive (his, her).
3. If a verb precedes an entity (base form, past tense, gerund, past participle, present tense) and the entity is not followed by 's, the replaced pronoun should be accusative (him, her).
4. If a verb precedes an entity (base form, past tense, gerund, past participle, present tense) and the entity is followed by 's, the replaced pronoun should be possessive (his, her).
5. In all other cases, the replaced pronoun is nominative and based on gender information (he, she).

After the pronoun is replaced in the text, the text is fed to Lingpipe. This new output is compared with the original text. If the new pronoun is still referring to the same entity that the earlier entity referenced (i.e. the entity that is replaced by the pronoun), the replacement will be valid and the pronoun is kept in the text, otherwise the previous version of the text is used for the next replacement iteration. This process is repeated for all the possible combinations of co-referent entities. If the entity is already a pronoun, nothing is done**.**

3. An Example

To illustrate how this algorithm works, we will now work through an example. Suppose the following passages, shown in (1), (2) and (3), are extracted from three separate documents[4]:

(1) Albert lives alone.
(2) Sandra invited Albert to the dinner.
(3) Jack couldn't make it to the party. Albert is in a hurry.

By running Lingpipe on the set, it would return (Albert, Sandra and Jack) as the entities. The output would be as shown in (4-6).

(4) <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> lives alone.
(5) <ENAMEX id="1" type="PERSON"> Sandra </ENAMEX> invited <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> to the dinner.
(6) <ENAMEX id="2" type="PERSON"> Jack </ENAMEX> couldn't make it to the party. <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> is in a hurry.

So there are three co-referent *Alberts*, one *Sandra* and one *Jack*. As the algorithm states, there is an opportunity for the second and third *Alberts* to be replaced by pronouns. First, the gender recognition task is performed and after the four phases explained in the section 2 the genders would be known.

[2] http://baby-names.adoption.com/names.php

[3] For instance, while nominative pronouns can occur after *while*, *while* is categorized as prepositional phrase.

[4] This example is not extracted from the DUC2005 corpus for the sake of simplicity. Also, it is not tested by the implemented code and Lingpipe. The purpose is just to show how the algorithm works.

Since there are at most two potential replacements, the loop runs twice. On the first run, *him* is suggested instead of the second *Albert* (following the 4th rule proposed in section 2). Then, the following text is generated:

(7) Albert lives alone. Sandra invited him to the dinner. Jack couldn't make it to the party. Albert is in a hurry.

Notice the only change to the text is the introduction of this single pronoun. Now, this text is fed to Lingpipe to generate the following output:

(8) <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> lives alone.
(9) <ENAMEX id="1" type="PERSON">Sandra</ENAMEX> invited <ENAMEX id="0" type="MALE_PRONOUN"> him </ENAMEX> to the dinner.
(10) <ENAMEX id="2" type="PERSON"> Jack </ENAMEX> couldn't make it to the party. <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> is in a hurry.

In the validation phase, the id of the newly-replaced pronoun (*him*) is compared with the entity it was replaced with (the second *Albert*). Since both of them are 0, it means the pronoun is correctly referring to the antecedent of the replaced entity, so it is kept in the final text.

On the second pass, the third *Albert* is replaced with a pronoun as shown in (11).

(11) Albert lives alone. Sandra invited him to the dinner. Jack couldn't make it to the party. He is in a hurry.

After running Lingpipe on the text, we obtain the following:

(12) <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> lives alone.
(13) <ENAMEX id="1" type="PERSON"> Sandra </ENAMEX> invited <ENAMEX id="0" type="PERSON"> Albert </ENAMEX> to the dinner.
(14) <ENAMEX id="2" type="PERSON"> Jack </ENAMEX> couldn't make it to the party. <ENAMEX id="2" type="MALE_PRONOUN"> He </ENAMEX> is in a hurry.

The id of *he* is 2 not 0, meaning that Lingpipe suggests if we perform such a replacement we end up referring to *Jack* instead of *Albert* which is not author's purpose. So, this replacement is rejected.

4. Results

We use the DUC 2005 documents for our evaluation. One issue we encountered with the DUC2005 questions was that they were not really person-centric and since the final answer was heavily dependent on the question keywords, we could not have recurrent person entities in them.

We decided to use the current project corpus but to overcome the lack of test data by extracting the sentences containing the same entity and order them randomly and to occasionally insert some other sentences between them. So, the following results are based on the DUC2005 corpus but are not using the DUC2005 questions. This approach might seem artificial but is consistent with questions like "who is X?", where the only significant keyword is the name of the person. McCoy et al (1999) showed in 97.9% cases not using the pronoun for long distance references (more than two sentences), is accurate. Since they worked with a similar corpus (NY Times articles), we decided to focus only on short distance references.

Although the precision of Lingpipe is high its recall proved to be low (0.54) on our set of documents. It fails to identify some obvious entities and at times cannot associate names of the same person as the co-reference. To improve performance, we made a few modifications to its output as described below.

Sometimes Lingpipe is unable to find the same entity, even when it is repeated exactly the same. For example, if *Albert* is repeated twice in the text, Lingpipe might find only the first occurrence. It is vital for us to find recurrent entities so that we can make the replacements. So we automatically identify and extract any instances of an entity not detected by Lingpipe and assign them the same id that Lingpipe used for that entity elsewhere in the document.

In cases where Lingpipe assigns the same entity repeated in the text different "entity types", we relax the rigid condition that both entities should be person. For example, in one sentence, "Trump" might be a person and in another an organization. However, at least one of them should be person.

To evaluate our algorithm independently of Lingpipe performance, we provide two different types of evaluation: first, by assuming that Lingpipe has detected all the valid entities and second, by taking all of the entities into account whether they are detected by Lingpipe or not. We again use the DUC 2005 documents for our evaluation.

To better explain our results, we will first introduce a few terms. *Action* can be defined as any decision that the algorithm makes. It might be either generating a new pronoun or simply leaving the entity as it is. *Valid Action* is an action acceptable by a human reader. *Invalid Action* is an action unacceptable by a human reader.

Running the algorithm, yields the results summarized in Table 1[5].

Dale (2000) characterizes the mistakes in pronoun generation as missed and inappropriate pronouns. We use the same notation for invalid actions.

Valid Action		Invalid Action	
Replacement	Refusal	Inappropriate	Missed
46	29	7	29
41.4%	26.1%	6.3%	26.1%
67.6%		32.4%	

Table 1. Summary of Algorithm Performance

The number of opportunities for generating a pronoun were 111. As shown in the Table 1, the algorithm works well when deciding to perform a replacement. But, it does not perform well when avoiding (refusing) a replacement. As noted earlier, part of it can be attributed to Lingpipe's failure to extract at least one instance of a repeated entity.

[5] Obviously, the pronouns already in the text are not counted in the evaluation.

Our observation shows that this happens 13 times out of the 29 refusals, meaning that these 13 entities are not found at all, let alone replaced by the pronoun. Table 2 shows the results, omitting these cases in the test data.

Valid Action		Invalid Action	
Replacement	Refusal	Inappropriate	Missed
46	29	7	16
47.0%	29.6%	7.1%	16.3%
76.5%		23.5%	

Table 2. Performance on Alternative Data

Knowing that our experiment concentrated on difficult cases, namely inter-sentential references with no intra-sentence references, the 76.5% accuracy result should be compared with the corresponding result from McCoy et al (1999) which shows an accuracy of 72.6%. An example of a summary produced by our method is provided in Appendix A.

One issue that we did not deal with but which can improve the performance is the identification of appositive clauses in the text. For the sake of brevity in news documents it is very common to use appositives to describe the person's role or job. On the other hand, in English it is not accurate to use a pronoun before an appositive (ex. *He, Canadian Prime Minister*). Hence, replacing the entity before an appositive with a pronoun would become an error.

5. Conclusions and Future Work

In this paper we introduced a simple approach to using an existing co-reference resolution tool in order to perform the task of pronoun generation. The independence from the resolution module enables us to improve the performance with the new advances in anaphora resolution approaches and at the same time enhance the generation module independently.

Since the approach is independent of which anaphora resolution module is used, future work could involve comparisons among different modules. Additionally, competing results from different resolution modules could be scored and combined in order to obtain more accurate generation.

6. References

Gaizauskas R., K. Humphreys. 1996. *Quantitative Evaluation of Coreference Algorithms in an Information Extraction System*. In S. Botley and T. McEnery (eds.), Corpus-based and Computational Approaches to Discourse Anaphora.

Grosz J., Joshi A. K., and Wenstein S.. 1995. *Centering: A Framework for Modeling the Local Coherence of Discourse*, Comput. Linguist., vol.21, no.2,pp.203-225.

Lappin S., H. J. Leass. 1994. *An Algorithm for Pronominal Anaphora Resolution*. Computational Linguistics 20(4):535-561.

McCoy K. F. and Strube M.. 1999. *Generating Anaphoric Expressions: Pronoun or Definite Description?*, Proc. Workshop on the Relation of Discourse/Dialogue Structure and Reference, pp.63-71.

Melli G., Y. Wang, Y. Liu, M. M. Kashani, Z. Shi, B. Gu, A. Sarkar, F. Popowich, 2005, *Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task*, Document Understanding Conference 2005 (DUC-2005), Vancouver, BC.

Mitkov R.. 1998. *Robust Pronoun Resolution with Limited Knowledge*. Proceedings of COLING/ACL 1998, Montreal, Quebec, Canada, 869-875.

Reiter E., R. Dale. 2000. *Building Natural Language Generation Systems*, Cambridge University Press.

Strube M.. 1998. *Never Look Back: An Alternative to Centering.* Proceedings of the 17th International Conference on Comp. Linguist., Montreal, Quebec, Canada, Vol. 2, pp. 1251-1257.

Vicedo J. L, A. Ferrandez. 2000. *Importance of Pronominal Anaphora Resolution in Question Answering Systems.* The 38th Annual Meeting of the Association for Computational Linguistics, ACL 2000.

Appendix A. Sample Output

'There has been a broad recognition, led by Preston, that an institution like the Bank cannot keep on expanding,' says Husain. When Preston came to the Bank **he** found an organisation still shattered by that event. The financial squeeze partly reflects **his** appreciation of the chilly climate. To **his** credit, **he** has no apparent interest in empire building. Having spent 40 years at JP Morgan, the premier New York bank, **he** seems untroubled by the notion of transferring bank functions to the private sector. According to **him**, NGOs have some involvement in 50 per cent of the Bank 's lending activities in Africa. THE World Bank will link loan volume to the strength of a country's efforts to fight poverty, according to an operational directive to staff issued today by Mr Lewis **he**, the bank's president. Mr Barber Conable, the bank's president, says a 50 per cent target for loans directed to the private sector risks 'subterfuge', suggesting the bank would simply redefine loans so they fitted into the right category. In the directive, **he** says poverty reduction is 'the benchmark by which our performance as a development institution will be measured'.

Uporaba kanoničnega govornega akustičnega modela za prilagajanje prostora govornih akustičnih značilk

Simon Dobrišek*, Boštjan Vesnicer*, Jerneja Žganec Gros†, France Mihelič*

*LUKS, Fakulteta za elektrotehniko, Univerza v Ljubljani
Tržaška 25, 1000 Ljubljana
simon.dobrisek@fe.uni-lj.si

†Alpineon, d.o.o
Ulica Iga Grudna 15, 1000 Ljubljana

Povzetek

V članku predstavljamo rezultate poskusov s postopki prilagajanja akustičnega govornega modela na globalne akustične značilnosti govornih sej. Preizkušeni postopki temeljijo na CMLLR-transformacijah. Uporabili smo poseben, manj obsežen kanonični govorni akustični model, katerega namen je izključno določanje CMLLR-transformacij. Parametre bolj obsežnega govornega akustičnega modela smo določali po posebnem učnem načrtu z uporabo že transformiranih vektorjev akustičnih značilk. Rezultati preizkusov samodejnega razpoznavalnika govora z govornimi sejami, ki po globalnih akustičnih in govornih značilnostih odstopajo od učnih govornih sej, so v primerjavi z izhodiščnim modelom, ki ne izvaja opisanega postopka prilagajanja, pokazali izboljšanje pravilnosti razpoznavanja govora.

Adaptation of Acoustic Feature Space Using Canonical Acoustic Model

The paper presents the results of experiments with a speaker-adaptive-training scheme that is based on CMLLR. A simple canonical acoustic model was used to obtain linear transformations of acoustic feature space that are speech session dependent. A more complex acoustic model, used for the actual automatic speech recognition, was first initialized from the canonical model. Its parameters were then reestimated from the acoustic feature vectors that were previously transformed using the session-dependent linear transformations. The presented results indicate that, when test speech sessions differ considerably from the training ones, automatic speech recognition is improved using the proposed training scheme.

1. Uvod

Trenutno najboljši samodejni razpoznavalniki govora podpirajo možnost samodejnega prilagajanja akustičnega govornega modela akustičnim značilnostim trenutne govorne seje. Govorna seja zajema vse govorne posnetke istega govorca, ki so posneti v približno enakih akustičnih razmerah. Pogosto so vsi posnetki istega govorca obravnavani kot ena govorna seja.

Uveljavljeni postopki tovrstnega prilagajanja praviloma predpostavljajo uporabo akustičnih govornih modelov, ki temeljijo na teoriji prikritih Markovovih modelov (PMM). V preteklih letih je bilo narejenega največ dela predvsem na postopkih prilagajanja z uporabo linearnih transformacij akustičnih govornih modelov (Gales, 1998; in P. C. Woodland, 2001). Poleg te možnosti so najbolj znani še postopki, ki temeljijo na kriteriju največjega aposteriornega verjetja modela (angl. MAP) (in C. H. Lee, 1994). Na slednjih je bilo v zadnjem času opravljenega veliko dela predvsem pri razvoju samodejnih razpoznavalnikov govorcev.

Že pred leti smo si za raziskovalni cilj zastavili razvoj lastnega pogona za samodejno razpoznavanje tekočega slovenskega govora z velikim besednjakom, ki bo imel zmožnost samodejnega prilagajanja na govorne seje in bo primeren za uporabo v vgradnih sistemih. V zadnjem času smo delali predvsem na postopkih, ki temeljijo na linearnih transformacijah in omogočajo sprotno prilagajanje govornega modela trenutnim govornim sejam.

Posebej smo se posvetili določanju omejenih globalnih linearnih transformacij parametrov akustičnega govornega modela, ki jih je mogoče preslikati v linearne transformacije prostora akustičnih govornih značilk. Z določanjem takšnih globalnih transformacij, ki so od govornih sej odvisne, lahko namreč zgradimo kanonični akustični govorni model, ki je deloma neodvisen od globalnih akustičnih značilnosti govornih sej. Transformacije, dobljene s kanoničnim modelom, lahko nato uporabimo pri učenju običajnega samodejnega razpoznavalnika govora, ki tako tudi postane deloma neodvisen od globalnih akustičnih značilnosti govornih sej. To je eden od možnih načrtov postopka učenja, ki se prilagaja govornim sejam in s tem tudi govorcu (angl. Speaker Adaptive Training - SAT).

Članek opisuje nekaj naših poskusov z različnimi načrti postopka učenja razpoznavalnika govora, ki se prilagaja govornim sejam na prej opisan način. Zaradi časovne zahtevnosti izvajanja takšnih poskusov smo primerjali rezultate, dosežene z samodejnim razpoznavalnikom s srednje velikim besednjakom. Za izvajanje poskusov smo v pretežni meri uporabljali orodje HTK. Za bolj učinkovito veriženje linearnih transformacij, ki so posledica iterativnega postopka ocenjevanja parametrov, smo razvili tudi nekaj lastnih orodij.

2. Prilagajanje z linearnimi transformacijami

Pri akustičnih govornih modelih, ki temeljijo na teoriji PMM, se prilagajanje z linearnimi transformacijami ne nanaša na prav vse parametre tega modela. Ponavadi se izvaja linearno transformacijo le na parametrih funkcij normalnih gostot verjetnosti, s katerimi modeliramo porazdelitve naključnih spremenljivk v posameznih stanjih naključnega avtomata. V tem primeru se transformacije nanašajo le na srednje vrednosti in variance Gaussovih porazdelitev. Prehodne verjetnosti med stanji naključnega avtomata in apriorne verjetnosti Gaussovih komponent v mešanicah, ki modelirajo omenjene porazdelitve, pa v teh postopkih prilagajanja ponavadi ne spreminjamo.

Obstaja več vrst linearnih transformacij, ki se uporabljajo za prilagajanje akustičnega govornega modela (in M. J. F. Gales, 2005). Mi smo se posvetili predvsem linearnim transformacijam, ki jih je mogoče preslikati iz transformacije parametrov akustičnega govornega modela v transformacijo prostora akustičnih govornih značilk. Primer takšne transformacije je omejena linearna transformacija, določena po kriteriju največjega verjetja akustičnega modela (angl. Constrained Maximum Likelihood Linear Regression - CMLLR) (Gales, 1998).

Pri CMLLR-transformaciji se vektorji srednjih vrednosti μ in kovariančne matrike $\mathbf{\Sigma}$ linearno transformirajo po spodnjih enačbah.

$$\hat{\mu} = \mathbf{A}'\mu - \mathbf{b}' \quad , \quad \hat{\mathbf{\Sigma}} = \mathbf{A}'\mathbf{\Sigma}\mathbf{A}'^{\mathbf{T}}$$

Matrika $\mathbf{A}'$ in vektor $\mathbf{b}'$ predstavljata linearno transformacijo in njune koeficiente določamo po kriteriju največjega verjetja akustičnega modela za dane nize vektorjev govornih akustičnih značilk $\mathbf{o}(\tau)$, ki so na razpolaga za prilagajanje. Določanje koeficientov matrike $\mathbf{A}'$ in vektorja $\mathbf{b}'$ izvedemo s uveljavljenim postopkom EM (angl. Expectation-Maximization) kot je podano v (Gales, 1998).

Dobljeno transformacijo parametrov Gaussovih porazdelitev lahko enostavno preslikamo v linearno transformacijo vektorjev akustičnih značilk, kot je podano v spodnjem izrazu.

$$\hat{\mathbf{o}}(\tau) = \mathbf{A}\mathbf{o}(\tau) + \mathbf{b} = \mathbf{A}'^{\mathbf{-1}}\mathbf{o}(\tau) + \mathbf{A}'^{\mathbf{-1}}\mathbf{b}'$$

Matriko $\mathbf{A}$ in vektor $\mathbf{b}$ ponavadi združimo v matriko $\mathbf{W} = [\mathbf{A}\ \mathbf{b}]$, ki nato enovito predstavlja iskano linearno transformacijo. Koeficiente matrike $\mathbf{W}$ določamo iz govornih posnetkov za vsako govorno sejo posebej. S tem pridemo do linearnih transformacij $\mathbf{W}(s)$, ki vektorje akustičnih značilk dane govorne seje s prilagodijo akustičnemu govornemu modelu.

3. Načrt učenja s prilagajanjem

Zahteva po sprotnem prilagajanju akustičnega govornega modela trenutni govorni seji ponavadi pomeni, da je za ta namen na razpolago razmeroma malo govora. Zaradi statistične narave akustičnega govornega modela in postopka ocenjevanja koeficientov linearne transformacije je očitno, da je pri majhni količini posnetkov prilagajanje boljše, če ima model manjše število parametrov. Zato smo se odločili, da bomo za prilagajanje uporabili posebni akustični govorni model z manjšim številom parametrov. Namen tega osnovnega modela je bil izključno ocenjevanje koeficientov linearne transformacije, ki se je uporabljala za sprotno transformiranje vektorjev akustičnih značilk. Dejanski razpoznavalnik govora smo učili in ga preizkušali na že transformiranih vektorjih akustičnih značilk.

Osnovni akustični govorni model smo sestavili kot razpoznavalnik kontekstno neodvisnih alofonov. Za vsakega od dvaintrideset alofonov ter treh dodatnih akustičnih enot (tišina, tlesk z jezikom in vdih) smo uporabil običajne levo-desne PMM s tremi stanji. Ta model smo uporabili tudi kot izhodišče pri določanju in učenju znatno večjega števila akustičnih modelov kontekstno odvisnih alofonov - trifonov.

Obravnavan učni načrt je možen le v primeru, ko so učni govorni posnetki urejeni po govornih sejah. Za vsako govorno sejo s smo določali njej lastno matriko $\mathbf{W}(s)$. Začetne vrednosti koeficientov matrik $\mathbf{W}(s)$ smo inicializirali tako, da smo normalizirali globalne vektorje srednjih vrednosti $\hat{\mu_0}(s) = \mathbf{0}$ in kovariančne matrike $\hat{\mathbf{\Sigma}}_0(s) = \mathbf{I}$, ki sta ocenjena iz vseh transformiranih $\hat{\mathbf{o}}(\tau)$ dane govorne seje s. To se enostavno doseže tako, da se koeficiente matrike $\mathbf{W}(s) = [\mathbf{A}(s)\ \mathbf{b}(s)]$ inicializira na sledeč način

$$\mathbf{A(s)} = \mathbf{L}_0(s)^T \ , \ \mathbf{b}(s) = -\mathbf{L}_0(s)^T\mu_0(s) \ ,$$

kjer $\mu_0(s)$ označuje globalni vektor srednjih vrednosti in $\mathbf{L}(s)$ spodnjo trikotno matriko razcepa Choleskega inverzne globalne kovariančne matrike $\mathbf{\Sigma}_0(s)^{-1}$. Pri tem sta $\mu_0(s)$ in $\mathbf{\Sigma}_0(s)$ ocenjena iz netransformiranih $\mathbf{o}(\tau)$ dane govorne seje s.

Takšno inicializacijo smo izvedli zato, ker smo smo za akustične značilke uporabili običajne MFCC-koeficiente. Na ta način smo v prilagajanje z linearnimi transformacijami vključili še normalizacijo MFCC-koeficientov po srednjih vrednostih in kovariancah (angl. Cepstral Mean and Covariance Normalization). Za takšno normalizacijo je znano, da zmanjšuje občutljivost govornega modela na akustično spremenljivost govornih sej.

Učenje s prilagajanjem smo v grobem izvajali po naslednjih korakih:

- Izračun globalnih vektorjev srednjih vrednosti $\mu_0(s)$ in kovariančnih matrik $\mathbf{\Sigma}_0(s)$ za vsako učno govorno sejo s.
- Inicializacija matrik $\mathbf{W}(s)$ za vsako učno govorno sejo s z uporabo prej izračunanih $\mu_0(s)$ in $\mathbf{\Sigma}_0(s)$.
- Izmenično iterativno ocenjevanje novih vrednosti matrik $\mathbf{W}(s)$ in parametrov osnovnih kanoničnih akustičnih govornih modelov iz vseh učnih govornih sej.
- Inicializacija parametrov akustičnih modelov trifonov z uporabo parametrov osnovnih akustičnih govornih modelov alofonov.
- Iterativno ocenjevanje parametrov akustičnih modelov trifonov iz vseh učnih govornih sej s z upoštevanjem linearnih transformacij, ki jih določajo matrike $\mathbf{W}(s)$.

Pri osnovni različici učnega načrta smo matrike $\mathbf{W}(s)$ inicializirali na običajen način z enotsko matriko in ničelnim vektorjem.

3.1. Govorne zbirke

V učno govorno zbirko smo združili tri različne zbirke. Zbirki Gopolis in K211d vsebujeta pretežno bran govor, posnet v nadzorovanem akustičnem okolju s kakovostnim mikrofonom. Zbirka VNTV pa vsebuje običajne televizijske posnetke vremenskih napovedi, ki so jih voditelji podali v okviru dnevnih poročil na Televiziji Slovenija. Učna govorna zbirka je tako vsebovala posnetke govornih sej petinšestdesetih govorcev. Skupno trajanje vseh učnih posnetkov je približno dvanajst ur in pol. Za eno govorno sejo smo šteli vse posnetke istega govorca. Iz učne govorne zbirke smo izločili tristo posnetkov, ki smo jih namenili za preizkus samodejnega razpoznavanja, ki je bil od učnih govornih sej odvisen.

Preizkusne govorne posnetke, ki so od učnih govornih sej deloma neodvisni, smo pridobili posebej za izvedbo poskusov, opisanih v tem članku. Dvaindvajset govorcev (v glavnem študentov) smo prosili, da posnamejo po dvajset daljših stavkov. Naključno tvorjeni stavki so se nanašali na poizvedovanja po letalskih informacijah. Do teh stavkov smo prišli podobno kot pri pridobivanju zbirke Gopolis (S. Dobrišek, 1998). Testni posnetki so bili pridobljeni v nenadzorovanih akustičnih okoljih in z različnimi mikrofoni, računalniki ter programi za snemanje zvoka. Pri testnih posnetkih gre še vedno pretežno za bran govor, a se ta govor znatno razlikuje od posnetkov v zbirki Gopolis. Govorcev namreč nismo posebej motivirali, da bi stavke jasno artikulirali, zato se pri znatnem številu posnetkov odražajo prvine spontanega govora (tleskanje z jezikom, vzdihi ipd).

4. Zgradba govornih modelov

Pri izvedbi poskusov smo poskrbeli za čim večjo primerljivost med preizkušenimi govornimi modeli. Vsi govorni modeli so bili tvorjeni s pomočjo orodij iz zbirke HTK. Orodjem smo dodali le možnost bolj učinkovitega veriženja linearnih transformacij. V vseh poskusih smo uporabljali iste govorne zbirke, vektorje akustični značilk in govorne modele z istim številom parametrov. Slednje postane pomembno predvsem pri izvedbi vezave parametrov s fonetičnimi odločitvenimi drevesi. Pri tem postopku je končno število parametrov govornega akustičnega modela odvisno od določenega praga (S. Young, 2005). Za povsem enako število parametrov smo poskrbeli tako, da smo pri določanju praga in doseganju želenega števila parametrov uporabljali rekurzivni postopek bisekcije.

Za vektorje akustičnih značilk smo uporabljali običajne 39-razsežne vektorje, sestavljene iz MFCC-koeficientov in njihovih delta- in delta-delta koeficientov. Iskane linearne transformacije so se nanašale na celotne 39-razsežne vektorje.

Pri vseh akustičnih modelih smo uporabil običajne levo-desne PMM s tremi stanji. Osnovni kanonični akustični model je tako poleg verjetnosti prehodov med stanji PMM tvorilo še 105 Gaussovih funkcij gostot verjetnosti z diagonalnimi kovariančnimi matrikami. Trifonski akustični modeli so imeli po alofonih vezane verjetnosti prehodov med stanji PMM in 3200 vezanih stanj s po pet-komponentnimi Gaussovimi porazdelitvami. Ta akustični model je tako vseboval skupaj točno 16000 Gaussovih funkcij gostot verjetnosti z diagonalnimi kovariančnimi matrikami. Fonetična vprašanja, ki so potrebna za vezavo parametrov smo tvorili ročno (Dobrišek, 2001) in v kombinaciji z vprašanji, samodejno pridobljenimi z orodji, ki so del zbirke Sphinx III.

Poleg navedenih parametrov imajo na rezultat razpoznavanja precejšen vpliv tudi drugi parametri Viterbijevega postopka iskanja najbolj verjetnega zaporedja stanj govornega modela pri danem govornem posnetku. Pri teh parametrih smo pazili predvsem na to, da smo pri vseh govornih modelih dosegli približno enak čas razpoznavanja istih govornih posnetkov. Vedno smo tudi uporabljali enako razmerje med vplivom akustičnega in jezikovnega modela na rezultat razpoznavanja in poskrbeli za približno enako razmerje med napakami vrivanj in izbrisov govornih enot.

4.1. Preizkušanje razpoznavalnikov

Vse zgrajene razpoznavalnike smo preizkušali z ugotavljanjem napak pri samodejnem razpoznavanju glasov in besed. Pri razpoznavanju glasov (alofonov) nismo uporabljali nobenega jezikovnega modela. To pomeni, da je govorni model vključeval predpostavko, da vsak alofon lahko sledi drugemu z enako verjetnostjo. Pri razpoznavanju besed smo upoštevali besednjak s približno pettisoč besedami. Govorni model je vključeval bigramski jezikovni model, ocenjen iz učne govorne zbirke. To pomeni, da je vključeval tako poizvedovanja po letalskih informacijah kot tudi vremenske napovedi. Kot smo že omenili, so se preizkusne govorne seje nanašale le na poizvedovanja po letalskih informacijah. Preizkusni govorci niso bili vključeni v učno govorno zbirko.

Pri preizkušanju razpoznavalnikov, ki se prilagajajo na nove govorne seje, se pojavi problem začetne ocene linearnih transformacij, ki prilagodijo govorni model njihovim globalnim akustičnim značilnostim. Preizkus smo si zaenkrat zamislili tako, da smo del preizkusnih posnetkov namenili izključno začetnemu prilagajanju govornega modela in nato uporabili preostali del za dejanski preizkus pravilnosti razpoznavanja. S poskusi smo ugotovili, da se doseže dobre rezultate že z desetimi poljubnimi krajšimi stavki, ki se namenijo izključno začetnemu nenadzorovanemu prilagajanju govornega modela.

V praksi bi to pomenilo, da bi moral nov govorec najprej izgovoriti deset poljubnih stavkov, ki bi bili namenjeni izključno začetnemu prilagajanju govornega modela na njegove globalne akustične značilnosti. Rezultati, ki so podani v tem članku, predpostavljajo takšno začetno prilagajanje. Nadaljnje prilagajanje se je nato izvajalo sprotno z vsakim novim preizkusnim stavkom, ki ga je izgovoril govorec. V naših prihodnjih poskusih nameravamo oceniti tudi kako narašča pravilnost razpoznavanja od prvega stavka naprej. Ta podatek je zanimiv za primere, ko govorcev ne bi radi obremenjevali s takšnim začetnim prilagajanjem govornega modela.

5. Rezultati

Podajamo rezultate razpoznavanj za štiri vrste poskusov. Pri prvem poskusu (BASE) je bil uporabljen model, ki se ni prilagajal na globalne akustične značilnosti govornih sej. Pri drugem poskusu (CVMN) smo uporabljali linearne transformacije, s katerimi izvedemo le normalizacijo srednji vrednosti in varianc globalnih Gaussovih porazdelitev posameznih govorih sej. Pri tretjem poskusu (CMLLR) smo izvedli prilagajanje na opisan način s kanoničnim akustičnim modelom, pri katerem so bile transformacije inicializirane na običajen način z enotsko matriko in ničelnim vektorjem. Pri zadnjem poskusu (CVMN-CMLLR) pa smo transformacije inicializirali iz srednji vrednosti in varianc globalnih Gaussovih porazdelitev posameznih govorih sej. Rezultati preizkusnih razpoznavanj glasov (alofonov) so

MODEL	EVAL	ADPT	TEST
BASE	89,7%	57,2%	56,8%
CVMN	90,1%	59,8%	59,1%
CMLLR	92,0%	62,2%	62,1%
CVMN-CMLLR	92,4%	63,4%	63,2%

Tabela 1: Ocenjene verjetnosti pravilnega razpoznavanja glasov pri različni govornih akustičnih modelih

podani v tabeli 1. Rezultati so podani kot ocene verjetnosti pravilnega razpoznavanja glasov. Rezultati, označeni z EVAL, se nanašajo na že omenjenih tristo posnetkov, ki so bili naključno izbrani in izločeni iz učne govorne zbirke. Ti rezultati predstavljajo oceno verjetnosti pravilnega razpoznavanja glasov v posnetkih, ki so od učnih govornih sej odvisni. Rezultati, označeni z ADPT, se nanašajo na preizkusne posnetke, ki so bili uporabljeni za začetno oceno linearnih transformacij pri prilagajanju govornega modela. Rezultati, označeni z TEST, pa se nanašajo na dejanske preizkusne posnetke, pri katerih se je izvajalo sprotno prilagajanje govornega modela. Pri rezultatih v tabeli 1 je najbolj

MODEL	EVAL	ADPT	TEST
BASE	8,7%	24,7%	26,2%
CVMN	8,8%	23,7%	25,6%
CMLLR	8,5%	19,1%	19,2%
CVMN-CMLLR	8,3%	18,8%	18,9%

Tabela 2: Ocenjene verjetnosti napačnega razpoznavanja besed pri različni govornih akustičnih modelih

opazna znatna razlika med oceno pravilnosti razpoznavanja posnetkov, ki so od učnih govornih sej odvisni, v primerjavi s tistimi, ki so od učnih govornih sej neodvisni. Glede na razmeroma velik obseg učne govorne zbirke to priča o tem, da se preizkusne govorne seje po globalnih akustičnih značilnostih res precej razlikujejo od učnih govornih sej. Po drugi strani pa je razlika med ocenami verjetnosti pravilnega razpoznavanja glasov v posnetkih ADPT in TEST majhna. To priča o tem, da po globalnih akustičnih značilnostih ni znatnih razlik med posnetki iste govorne seje, torej posnetki, ki so bili namenjeni začetni oceni linearnih transformacij in posnetki, na katerih se je izvajalo sprotno prilagajanje in dejanski preizkus razpoznavalnika.

Rezultati preizkusov razpoznavanj besed so podani v tabeli 2. Rezultati so podani kot ocene verjetnosti napačnega razpoznavanja besed. Tu so razlike med rezultati po različnih skupinah posnetkov nekaj manjši. To priča o znatnem vplivu jezikovnega modela na končni rezultat preizkusov.

6. Zaključek

Rezultatov naših poskusov potrjujejo domnevo, da postopki prilagajanja govornih modelov na globalne akustične značilnosti govornih sej z uporabo linearnih transformacij izboljšajo pravilnost samodejnega razpoznavanja govornih enot. Rezultati kažejo tudi na to, da je smiselno inicializirati linearne tranformacije tako, da v izhodišču dosežemo normalizacijo srednjih vrednosti in varianc globalnih Gaussovih porazdelitev posameznih govorih sej.

V naših nadaljnjih poskusih s postopki prilagajanja govornih modelov na globalne akustične značilnosti govornih sej se bomo posvetili predvsem postopkom, ki temeljijo na kriteriju največjega aposteriornega verjetja modela. Tudi v tem primeru bomo poskušali priti do kanoničnega govornega modela, pri katerih bo mogoče transformacije modela preslikati v transformacije govornega akustičnega prostora oziroma vektorjev govornih akustičnih značilk. Pri tem se bomo naslanjali na izkušnje, ki smo jih pridobili pri razvoju sistemov za samodejno razpoznavanje govorcev.

7. Literatura

S. Dobrišek. 2001. *Analiza in razpoznavanje glasov v govornem signalu*. Doktorska disertacija, Univerza v Ljubljani, Fakulteta za elektrotehniko.

M. J. F. Gales. 1998. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer Speech and Language*, 12, 75–98.

J. L. Gauvain in C. H. Lee. 1994. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. Speech Audio Proc.*, 2, 291–298.

H. Liao in M. J. F. Gales. 2005. Joint uncertainty decoding for noise robust speech recognition. V: *INTERSPEECH-2005*, str. 3129–3132.

L. F. Uebel in P. C. Woodland. 2001. Improvements in linear transforms based speaker adaptation. V: *ICASSP-2001*, str. 3129–3132.

F. Mihelič in N. Pavešič S. Dobrišek, J. Ž. Gros. 1998. Recording and labelling of the gopolis slovenian speech database. V: A. Rubio, ur., *First International Conference on Language Resources & Evaluation: Proceedings*, str. 1089–1096. European Language Resources Association.

M. Gales. T. Hain D. Kershaw G. Moore J. Odell D. Ollason D. Povey V. Valtchev in P. Woodland S. Young, G. Evermann. 2005. *The HTK Book (for HTK Version 3.3)*. Cambridge University, Engineering Department, Cambridge.

Klepec: slovenski programirani sogovornik

Špela Arhar, Miro Romih

Amebis, d. o. o.
Bakovnik 3, 1241 Kamnik, Slovenija
spela.arhar@amebis.si, miro.romih@amebis.si

Povzetek

Program Klepec je klepetanju namenjen programirani sogovornik, ki za jezik komunikacije z uporabnikom uporablja slovenščino. Nastal je v sklopu projekta KOLOS, katerega cilj je omogočiti komunikacijo med človekom ter računalnikom v naravnem jeziku. V članku predstavljava trenutno stanje programa ter nakazujeva smernice za nadaljnji razvoj, obenem pa izpostavljava nekatera problematična mesta razvoja programiranih sogovornikov (specifike klepetalniškega diskurza, potreba po antropomorfizaciji programa ipd.). Razpravo dopolnjujeva s primeri realnih komunikacijskih nizov med uporabniki ter programom Klepec.

Klepec: a Slovene chatbot program

Klepec is a chatbot computer program that uses Slovene as the language of communication. It was created within the KOLOS project, the main goal of which was to establish natural-language communication between human users and computers. In the article we present the current state of the program, together with some guidelines for its future development. In addition, we indicate some of the main difficult areas of chatbot-programming (specifics of the chatting discourse, the need for the anthropomorphization of the program etc.). The paper is supplemented by the examples of real communication sequences between users and the Klepec program.

1. Uvod

Programirani sogovornik je program, ki komunicira z uporabnikom ali drugim programom v izbranem naravnem jeziku. Poznamo programirane sogovornike, namenjene predvsem krajšanju časa s klepetanjem (v angleški literaturi najdemo zanje izraze *chatbot, chatterbot, chatterbox*), ter takšne, ki imajo kakšno drugo uporabno vrednost, konverzacijske sposobnosti pa izrabljajo kot pomoč pri dosegi svojega cilja. Med slednje spadajo predvsem nekateri tipi t. i. inteligentnih agentov (*intelligent agents*), programov, ki za uporabnika ali druge programe izvršujejo različne naloge, za katere so pooblaščeni (iščejo ter urejajo različne informacije, prodajajo artikle, vodijo uporabnika skozi določeno opravilo ...).

Namen članka je predstaviti program Klepec, ki je prvi resni poskus izdelave programiranega sogovornika za slovenski jezik. Članek se osredotoča predvsem na izbiro slovenščine za jezik komunikacije ter iz tega izvirajoče specifike pri razvoju programa, ob strani pa pušča splošnejši prikaz programske zasnove ter delovanja (slednje v Romih et al., 2002).

2. Kaj je Klepec

Klepec je eden izmed jezikovnotehnoloških projektov podjetja Amebis, d. o. o (http://www.amebis.si). Nastal je v sklopu razvoja sistema KOLOS, katerega cilj je »ustvariti okolje in orodja, s pomočjo katerih bi bila komunikacija med človekom in računalnikom enaka komunikaciji med ljudmi« (Romih et al., 2002). Trenutno je na internetu verzija 2.0, v teku pa so že priprave izboljšane verzije.[1]

Zaenkrat je Klepec programirani sogovornik, namenjen zabavi in klepetu, s postopnim vgrajevanjem znanj oz. podatkovnih baz pa naj bi se razvil v agenta, ki zna uporabniku poiskati želene informacije oz. mu s svojim znanjem pomaga pri reševanju različnih nalog.

3. Kaj zna Klepec

Stanje, v katerem najdemo program na trenutni stopnji razvoja, na kratko predstavljava v štirih podpoglavjih.

3.1. Slovarsko ter slovnično znanje o jeziku

Uporabniške vnose program zaenkrat razčlenjuje le na prvi stopnji. Strukturno prepoznava stavke ter besede, besede lematizira, vendar v primeru dvoumne lematizacije ne izbira med obstoječimi možnostmi. Besednovrstne kategorije so rabljene predvsem v programski kodi, kjer z njimi generaliziramo tako vzorce za prepoznavo uporabniškega vnosa kot tudi odgovore na določen tip uporabnikove iztočnice. Nova verzija programa naj bi prinašala bistveno kvalitetnejšo jezikovno analizo vnosa na osnovi posodobitve programa z izboljšanim lematizatorjem ter jezikovnim analizatorjem, ki omogoča razdvoumljanje lematizacije ter delno stavčnočlensko analizo.

Klepec ima torej vgrajeno znanje o pregibanju besed in če ga v pogovoru denimo prosimo, naj sklanja določen samostalnik, bo to tudi storil. Vgrajeno ima tudi znanje o slovničnem spolu in po identifikaciji sogovornikovega spola bo v nadaljevanju uporabljal zanj ustrezne slovnične oblike. Natančneje urejena je tudi identifikacija vprašalnic v vprašalnih stavkih, kar programu omogoča izbiro odgovora iz ustreznega dela programske baze.

Zaenkrat program nima nikakršnih semantičnih informacij, na osnovi katerih bi lahko vodil koherentno povezan pogovor (potreboval bi podatke o pomenskih razmerjih med besedami, tipičnih kolokacijah ipd.). Ker v tem trenutku stavčnočlenska analiza ne poteka niti na ravni stavka, prav tako še niso izpolnjeni pogoji za nadstavčno povezovanje, razdvoumljanje pomenskih navezav itd.

[1] S Klepcem lahko klepetamo na internetni strani http://klepec.amebis.si.

Uporabnik: sklanjaj voli
Klepec: Je voli samostalnik?
Uporabnik: da
Klepec: Na tvojem mestu ne bi bil tako prepričan.
Uporabnik: sklanjaj vol
Klepec: vol, vola, volu, vola, pri volu, s/z volom
Uporabnik: sklanjaj besedo vol v monžini
Klepec: Nekoč sem imel domačo žival. Jo imaš tudi ti?
Uporabnik: nočem klepetati

Primer pogovora 1: Sklanjanje.[2]

3.2. Poznavanje specifik klepetalniškega diskurza

Klepetalniški diskurz (diskurz, kakršen se pojavlja pri realnočasovni pisni komunikaciji prek interneta) ima določene specifike, ki jih je potrebno pri snovanju programiranega sogovornika upoštevati. Najpomembnejše so naslednje točke:

3.2.1. Stalne točke v poteku klepeta

Klepec ima vgrajeno osnovno védenje o poteku klepeta – pozna denimo koncepta pozdravljanja ter predstavljanja. V začetni fazi pogovora skuša dobiti od uporabnika informacijo o njegovem imenu, podatek si zapomni in ga v nadaljevanju pogovora po potrebi uporabi. Če je podano ime že v bazi osebnih lastnih imen, Klepec v nadaljevanju pogovora za uporabnika uporablja ustrezne slovnične oblike, če pa imena ne prepozna, se pozanima, ali gre za moško ali žensko ime. Kadar ne dobi enoznačnega odgovora, privzame, da je uporabnik moškega spola.

Klepec: Mimogrede, mi zaupaš tvoje ime?
Uporabnik: na
Klepec: Še nisem slišal. Kakšno ime pa je to - moško ali žensko?
Uporabnik: brez spola
Klepec: Čakam, čakam, čakam ...
Uporabnik: kaj čakaš?sneg?
Klepec: Čakam, čakam, čakam ... Bo že kaj?
Uporabnik: ne vem če bo kej snežilo
Klepec: Ne grem se več. Zame si moški, pa pika.

Primer pogovora 2: Pridobivanje podatkov o imenu ter spolu uporabnika.

3.2.2. Menjavanje vlog v komunikaciji

Klepec upošteva načelo menjave vlog sprašujočega ter odgovarjajočega v pogovoru (včasih le odgovarja, včasih sprašuje, včasih združi oboje).

3.2.3. Odkloni od knjižne norme v smer govorjenega jezika

V internetnih klepetih so odstopi od knjižne norme pogosti na vseh jezikovnih ravneh. Pojavlja se opuščanje ločil ter velikih začetnic, leksika ter skladnja se približujeta govorjenemu jeziku, prepoznavo uporabnikovega vnosa pa otežujejo še tipkarske in druge pravopisne napake ter opuščanje šumnikov.

Leksikalno bazo, na kateri temelji Klepec, za potrebe razvoja programa dopolnjujemo z najpogostejšimi odkloni od knjižne norme, in sicer z leksikalnimi enotami, ki:

- se v zapisu približujejo govoru (npr. različni zapisi zaimka *jaz – jst, jest, js ...*, *kako – kk, kak, kuko ...,* reducirane glagolske oblike *gledal – gledu* ipd.),
- z opuščenimi šumniki nimajo enakopisne knjižne ustreznice (npr. *delaš – delas*),
- odstopajo od pravopisnih pravil (npr. *življenjski – življenski*).

Uporabnik: kaj delas
Klepec: Tega pa žal še ne vem.
Uporabnik: kako
Klepec: Hitro in z lahkoto.
Uporabnik: zakaj
Klepec: Pomisli malo.
Uporabnik: a si nor
Klepec: Ne bi raje uporabil lepo vprašalnico "ali"?
Uporabnik: ali si nor
Klepec: Kaj pa vem.
Uporabnik: saj malo
Klepec: Govoriva o čem drugem, prosim.
Uporabnik: kaksno vreme bo jutri
Klepec: Jutri bo lepše.
Uporabnik: kaj pa pojutrijsnem
Klepec: Ali nimaš nobenega pametnejšega vprašanja?

Primer pogovora 3: Odstopi od knjižne norme.

3.2.4. Vpletanje tujejezičnih prvin

V pogovorih se relativno pogosto pojavljajo tudi jezikovne prvine iz (večinoma) angleškega jezika. V trenutni fazi razvoja je program omejen zgolj na identifikacijo rabe tujega jezika (kljub temu, da Klepec v spodnjem primeru trdi, da zna v angleščini tudi komunicirati). Identifikacija jezika temelji na bazi ključnih tujejezičnih besed, ki se v takšnih primerih najpogosteje pojavljajo, in je posledično uspešna le pod pogojem, da je uporabniški vnos tujejezičnih enot pravopisno ustrezen.

Uporabnik: how are you?
Klepec: Poglej, poglej - saj znaš angleško!?
Uporabnik: a ti ne znaš?
Klepec: Veš da.
Uporabnik: no pa se grema pogovarjat angleško,ok
Klepec: Ti pa znaš. Si se učil angleško!?
Uporabnik: ja
Klepec: To pa kar težko verjamem.
Uporabnik: pa nič

Primer pogovora 4: Uporaba tujega jezika.

3.2.5. Uporaba emotikonov

Ena od specifik klepetalniškega jezika je tudi raba klepetalniških kratic (npr. *lol, wtf, brb ...*) in emotikonov (grafičnih simbolov za čustva). Klepec zaenkrat

[2]Vsi primeri, ki jih navajava v tem članku, so zbrani iz baze pogovorov, ki so s Klepcem potekali v obdobju od 1. 10. do vključno 19. 12. 2005. Zaradi varstva uporabniških pravic izpuščava imena, kjer se pojavljajo, ter navajava le kratke izseke posameznega komunikacijskega niza.

prepoznava ter mestoma tudi že uporablja kratice ter emotikone, ki se v komunikaciji najpogosteje uporabljajo.

Klepec: Mislim da se mi je pokvarilo neko vezje. Si kaj sumljivega opazila?
Uporabnica: Ne čisto nič, samo ogenj pa dim. :)
Klepec: :))

Uporabnik: lol
Klepec: In čemu se tako smeješ?
Uporabnik: a kr poznash kratico laughing out loud?
Klepec: Veš da ne.
Uporabnik: o kok si talent.. dei grewa u angleščini govort

Primer pogovora 5: Uporaba emotikonov ter klepetalniških kratic.

3.3. Podatkovne baze

Ta tema se delno povezuje z vsemi ostalimi, saj vse znanje, na podlagi katerega program operira, izvira iz podatkovnih baz. Na tem mestu govoriva specifično o bazah znanja o svetu, iz katerih naj bi Klepec v prihodnosti črpal odgovore na najrazličnejša uporabniška poizvedovanja. Kot rečeno je glede informacijske podprtosti program še na začetku razvoja. Zaenkrat odgovarja na vprašanja z *kdo* in *kaj*, za katera dobiva podatke iz Amebisove interne definicijske baze, ki vsebuje okrog 100.000 gesel.

Uporabnik: kdo je marjan
Klepec: Marjan je hrib nad Splitom
Uporabnik: ne marjan je moj stric
Klepec: Kolikor vem, je stric očetov ali materin brat.
Uporabnik: tudi ti ves nekaj o tem
Klepec: Pustiva zdaj to. Raje mi povej še kaj o sebi.

Uporabnik: kaj je to zastava ?
Klepec: izročitev stvari za zavarovanje upnikove terjatve, če se ne motim
Uporabnik: kaj pa hladilnik ?
Klepec: Me sploh ne zanima.
Uporabnik: kaj je to nevihta ?
Klepec: Slišal sem, da je nevihta neurje, pri katerem močno dežuje, naliv spremljajo bliski.
Uporabnik: smrad
Klepec: Ti lahko povem, da je smrad neprijeten vonj.
Uporabnik: medmet
Klepec: Hej! Nekaj o tem pa vem tudi jaz! medmet je nepregibna beseda, ki izraža duševno stanje ali posnemanje glasov, zvokov. Imam prav?
Uporabnik: pa že

Primer pogovora 6: Vprašanja z *kdo* in s *kaj.*

3.4. Antropomorfizacija

Določena mera podobnosti v mišljenju in življenjski izkušnji je predpogoj za uspešno komunikacijo, zato je dobro, da je Klepec vsaj v nekaterih segmentih podoben človeku, drugačnega ga pravzaprav niti ni mogoče zasnovati. V trenutni fazi je namenjen klepetanju in zabavi, zato je želena mera antropomorfizacije višja: uporabnik program ocenjuje glede na to, koliko je podoben človeškemu sogovorniku – v kolikšni meri so njegovi odgovori primerni oz. smiselni v kontekstu pogovora, kako se odziva na določen tip replik itd. V prihodnosti se bo program razvijal v smer interaktivne podatkovne baze, kar pomeni, da se bo težišče njegove uporabe premaknilo. Pričakovano je, da se bodo s tem premikom spremenila tudi pričakovanja uporabnikov in njihovi kriteriji za presojanje kvalitete programa, s tem pa posledično zmanjšala potreba po antropomorfizaciji.

Nekaj zametkov simulirane osebnosti Klepec že ima, izraža želje, preference in tudi razpoloženje, odziva se na žaljenje itd. Antropomorfna lastnost je tudi smisel za humor. Zanimivo vprašanje, ki zaenkrat ostaja neodgovorjeno, pa je: je bolje, da program pri določenih temah simulira konkretne izkušnje (s spolnostjo, športom, čustvi ...), ali naj k temam pogovora pristopa, kot da dejanskost zaznava prek vmesnih medijev (da je informacije le nekje 'videl' oz. 'prebral') in s tem poudarja svoje specifike napram človeškim sogovornikom.

Uporabnik: si živčen?
Klepec: Šur.
Uporabnik: zakaj?
Klepec: Ker je res.
Uporabnik: ne bi smel biti
Klepec: Kako mi lahko kaj takega sploh govoriš?
Uporabnik: si program:)
Klepec: Ja.
Uporabnik: ergo ne bi smel biti živčen

Primer pogovora 7: Programi in živčnost.

4. Kako Klepec deluje

4.1. Notranja zgradba

Pokazalo se je, da klasični programski jeziki (kot npr. C++) ne omogočajo hitre in uspešne gradnje sistemov za komunikacijo v naravnih jezikih, še posebej ne v pregibnih, kakršna je slovenščina. Ti programski jeziki kot prvo ne omogočajo enostavne vgradnje podatkov oz. znanj o nekem naravnem jeziku, še bolj problematično pa je, da način programiranja in delovanja, ki temelji na (vnaprej) določenem poteku programskih ukazov, ni dovolj učinkovit in primeren za izdelavo sistemov dialoga, kjer je nemogoče vnaprej določiti interakcijo s sogovornikom.

Zato smo za potrebe sistema, kakršen je Klepec, razvili in izdelali poseben programski jezik s trenutnim delovnim imenom K2.0, ki omogoča učinkovitejše programiranje sogovornikov za določen naravni jezik. Trenutno je K2.0 podprt le z modulom za slovenski jezik.

4.1.1. Osnovni princip delovanja

V osnovi je Klepec vzorčno vodeni sistem. To pomeni, da je sestavljen iz določenega števila vzorcev oz. zapisov, ki jih program primerja s sogovornikovim vnosom, potem pa na osnovi določenih meril izbere tistega, ki je temu vnosu najbolj podoben, in izbere enega od odgovorov, ki jih ta zapis vsebuje.

```
$ kako si | kako si kaj | kako kaj
> Dobro. | Slabo. | Odlično.
$ kako ti je ime | kako se imenuješ
```

```
> Klepec. | Ime mi je Klepec. | Klepec se imenujem.
$ kaj znaš > Vse, kar so me naučili v šoli.
```

Slika 1: Nekaj enostavnih vzorcev oz. zapisov.

4.1.2. Posploševanje vzorcev

Eden osnovnih problemov pri razumevanju uporabniškega vnosa ter izdelavi in primerjavi vzorcev je, da lahko semantično načeloma enoznačne trditve izražamo na več različnih načinov (kar se tiče izbire leksike ter skladenjskih struktur). Zato je potrebno za praktično rešitev tega problema omogočiti določeno stopnjo posploševanja vzorcev.

```
$
kako ti je ime
mi lahko poveš kako ti je ime
mi lahko prosim poveš kako ti je ime
mi prosim lahko poveš kako ti je ime
mi lahko poveš prosim kako ti je ime
ali mi lahko poveš kako ti je ime
ali mi lahko prosim poveš kako ti je ime
...
mi lahko zaupaš kako ti je ime
mi lahko prosim zaupaš kako ti je ime
...
ali mi lahko zaupaš kako ti je ime
...
povej mi kako ti je ime
prosim povej mi kako ti je ime
...
zaupaj mi tvoje ime
...
kako se imenuješ
...
>
Klepec.
```

Slika 2: Veliko število možnih vzorcev.

Posplošitev vzorcev lahko v programskem jeziku K2.0 izvedemo s pomočjo različnih programskih elementov in ukazov, odvisno od tega, kaj želimo posplošiti. Najpogosteje se za posplošitev uporabljajo operatorji tipa %B (poljubna beseda), %S (poljubno število), %L (poljubno ločilo), %0 (nič ali poljubno število enot), %1 (ena poljubna enota), %2 (ena ali dve poljubni enoti'), %3 (ena, dve ali tri poljubne enote) itd. To nam omogoča, da želeni nabor vnosov zajamemo s precej manjšim številom vzorcev.

```
$
%0 kako ti je ime
%0 zaupaj mi %0 tvoje ime
%0 mi zaupaš %0 tvoje ime
...
%0 kako se %0 imenuješ
...
>
Klepec.
```

Slika 3: Posplošitev vzorcev.

Posplošitev lahko izvedemo tudi s pomočjo operatorja morfosintaktične informacije, s katerim še dodatno zmanjšamo potrebno število primerjalnih vzorcev.

```
$
%0 kako ti je ime
%0 zaupati[*] %0 tvoje ime
%0 povedati[*] %0 tvoje ime
...
%0 kako se %0 imenuješ
...
>
Klepec.
```

Slika 4: Dodatna posplošitev.

S pomočjo operatorja morfosintaktične informacije lahko kontroliramo točno določene besedne vrste in njihove oblike (npr. pri samostalniku vrsta, spol, sklon in število), tako na vhodni kot tudi izhodni strani. Pri tem pogosto potrebujemo pomoč operatorjev #1 (prva spremenljivka v vhodnem nizu), #2 (druga spremenljivka v vhodnem nizu) itd., ki nam omogočajo prenos poljubnih vhodnih besed v izhodni niz.

```
$ kako se sklanja %B[S??ei] | sklanjaj %B[S??ei]
>
#1[S??ei], #1[S??er], #1[S??ed], #1[S??et],
pri #1[S??em], s\/z #1[S??eo]

$ kako se sklanja %B[G?n] | sklanjaj %B[G?n]
>
Glagoli se ne sklanjajo!
```

Slika 5: Sklanjanje besed.

4.1.3. Kontrola poteka

Ker je v jezik K2.0 vgrajena naključnost pri izbiri enakih ali zelo podobnih vzorcev, imamo za kolikor toliko kontrolirano izbiro določenih vzorcev ali poteka dialoga na voljo številne dodatne operatorje in ukaze. Med te sodita določitev statične prioritete posameznega vzorca, npr. (7), ter določitev pogoja primerjave vzorca in izbire izhodnega niza, npr. [spol == 1], pri čemer lahko uporabljamo določene lokalne, globalne ali sistemske spremenljivke.

```
$ (7) [vreme == 0]
kakšno bo %0 vreme %0
kakšna %0 vremenska napoved %0
>
V Sahari bo precej suho. <vreme = 1>
[spol == 1] Počakaj, pa boš videl. <vreme = 1>
[spol == 2] Počakaj, pa boš videla. <vreme = 1>
```

Slika 6: Uporaba statične prioritete, pogojev in spremenljivk.

Če želimo kontrolirati potek dialoga na nivoju celotnega nabora ali skupine vzorcev, ne samo znotraj posameznega vzorca, imamo na voljo nekatere ukaze, s katerimi lahko preskočimo primerjavo skupine vzorcev

oz. omejimo primerjavo le na točno določeno skupino, ki je izbrana na osnovi prejšnjega poteka pogovora. Pri tem pogosto uporabljamo naslove, npr. {ime}, in akcije, ki se izvedejo ob branju ustreznega vzorca, npr. <goto {ime}>.

```
//***** kontrola imena
$> <goto {ime_konec}>
$> {ime} <null>

$ (7)
%0 ime %0 mi %0 je %0 %B[Slmei] %0
>
Lepo moško ime.<ime = #1><spol = 1><goto {zacetek}>

$> <stop/goto {ime}>
$> {ime_konec} <null>
//*****
```

Slika 7: Kontrola poteka dialoga.

4.1.4. Druge funkcije

Če želimo poleg nekoristnega klepeta v odgovore vključiti tudi koristne informacije, shranjene v določenih podatkovnih bazah, lahko uporabimo (uporabniško definirane) klice funkcij, ki te podatke preverjajo in berejo. Tako lahko programiranega sogovornika uporabimo v funkciji informacijskega sistema, ki nam koristne informacije podaja v zelo zanimivi obliki.

Poleg funkcij lahko pri programiranju uporabimo še številne zanimive in koristne mehanizme jezika K2.0, za predstavitev katerih pa je v tem članku premalo prostora.

```
$ (6)
%0 kaj %0 je %0 @Preveri(__pot, "BAZA")
>
#4 je @PreberiBAZA(__pot, #4)
```

Slika 8: Uporaba funkcij.

4.2. Uporabniški doprinos

4.2.1. Eliza – pripisovanje posebnega komunikacijskega namena

Brez uporabnikove pripravljenosti sodelovati v komunikaciji je še tako dobro zasnovan program obsojen na neuspeh, in obratno, povsem osnovne aplikacije lahko dosegajo velike uspehe.

To dejstvo je v svoj prid izkoristil tudi Joseph Weizenbaum, snovalec prve znane programirane sogovornice. Eliza, kot je svoj izdelek poimenoval po liku iz znanega dela *Pygmalion*, ima povod za nastanek v psihiatriji; tam se je v tistem času (gre za obdobje 60-ih let prejšnjega stoletja) izredno rado prakticiralo t. i. aktivno poslušanje – psihiater s parafraziranjem pacientovih izjav ter spodbudnimi signali pomaga pacientu, da sam razreši problem, ki ga teži. Metoda je Weizenbauma navdihnila, da je ustvaril program, ki z uporabnikom komunicira na enak način. Kot simulacija psihiatrinje je dobila programirana Eliza v pogovoru posebno hierarhično vlogo, ki jo avtor takole opisuje:

»Ta tip komunikacije je bil izbran, ker je psihiatrični intervju eden redkih primerov /.../ komunikacije v naravnem jeziku, v kateri je enemu od udeležencev pogovora dovoljeno zavzeti položaj, po katerem o svetu ne ve skoraj ničesar. Če npr. nekdo pove psihiatru 'Šel sem se vozit s čolnom,' psihiater pa odvrne 'Povejte mi kaj o čolnih,' ne bomo sklepali, da psihiater ne ve ničesar o čolnih, pač pa da pogovor vodi v to smer z določenim namenom.« (Weizenbaum, 1966. Prevedla in podčrtala Š. Arhar.)[3]

4.2.2. Osmišljanje izjav kontekstu

Da je komunikacija s programiranimi sogovorniki sploh mogoča, gre torej pripisati predvsem človeški sposobnosti oz. potrebi po osmišljanju izrekov v kontekstu diskurza. Pri snovanju programa, ki se v pogovoru predstavlja za uporabniku hierarhično enakovrednega sogovornika, je potrebno predvideti takšen način komunikacije, da je na vseh ravneh omogočeno čim bolj uspešno osmišljanje v kontekstu. Pri razvijanju programa Klepec skušamo to doseči na več načinov:

- s premišljenim snovanjem baze odgovorov; le-ti morajo biti obenem dovolj splošni, da ustrezajo čim več potencialnim kontekstom, obenem pa dovolj specifični, da omogočajo nadaljevanje pogovora,
- z uporabo humorja v Klepčevih izjavah; diskurz, ki se na tak način razvije, omogoča boljše osmišljenje Klepčevih izjav, saj je uporabnik do njih tolerantnejši, če jih ne jemlje povsem resno,
- z uporabo poskusa menjave teme v primerih, da uporabnikovega vnosa Klepec ne prepozna.

V nadaljevanju navajava tri primere uspešnega osmišljanja v kontekstu, kjer zgoraj naštete strategije v kombinaciji z uporabnikovim vztrajanjem pri temi, in seveda njegovo pripravljenostjo sodelovati v igri komunikacije, privedejo do zabavnih dialoških izmenjav:

Uporabnik: koliko je ura
Klepec: Je čas tvoj gospodar?
Uporabnik: ne
Klepec: Slišati si zelo prepričan v to.
Uporabnik: ampak mene vseeno zanima koliko je ura
Klepec: Meni čas ne pomeni ničesar. Vsaj dokler sem v tej škatli.
Uporabnik: da ti ne bom virusa poslal

Primer pogovora 8: Časovna relativnost.

Uporabnik:a ti veš kje je grčija
Klepec: Ja, stari.
Uporabnik: kje
Klepec: V zraku.
Uporabnik: kje?
Klepec: Za vogalom.
Uporabnik: kje
Klepec: Tukaj, kje pa drugje.
Uporabnik: kreten

Primer pogovora 9: Kje je Grčija.

[3]Eliza je za klepet na voljo na straneh Inštituta Jožef Stefan, <http://www-ai.ijs.si/eliza/>.

Uporabnica: me ljubiš?
Klepec: Ali ni danes prav lepo vreme?
Uporabnica: me ljubiš dragi?
Klepec: Uf, kako je vroče. Morda se mi je pokvaril ventilator?
Uporabnica: daj no me ljuubiš
Klepec: Zakaj pa "no"?
Uporabnica: ja me ljubiš?
Klepec: Hej, ti mi pa že ne boš tega govorila.
Uporabnica: daj me ljubi?
Klepec: Težko bi dal.
Uporabnica: ljubiš me
Klepec: Spremeniva temo!

Primer pogovora 10: Izpovedovanje ljubezni.

4.2.3. Uporabniški odnos do programa

Ker je vsaka komunikacija mnogo več kot le izmenjava jezikovnih nizov, želiva na tem mestu opozoriti še na sociološke ter psihološke vidike komunikacije med človekom ter programiranim sogovornikom. Dejstvo je, da raziskave uporabniškega odnosa do programov za klepetanje kažejo nenavadno visoko stopnjo agresivnega govora, pri katerem prednjačijo raznovrstne kletvice, žalitve ter provokacije. Foner (1997) v svoji sociološki raziskavi omenja uporabniški strah, da bi bil stroj (čeprav le po znanju) boljši od človeka. De Angeli in Carpenter (2005) ugotavljata, da izvira žaljiv odnos uporabnika do programiranega sogovornika iz privzetega razmerja moči, ki postavlja uporabnika kot človeško bitje na višji položaj od programa, ki se trudi biti človeku podoben. Poleg teh dveh premis je potrebno dodati še eno: samocenzura, ki je v medčloveški komunikaciji seveda stalno prisotna, v dialogu s programom – za katerega vemo, da nima ne čustev ne spomina na pogovor – popusti.[4]

Ker je predvidljivo, da bodo nekateri uporabniki do Klepca žaljivi še naprej, je za prepričljivo antropomorfnost nujno, da se program na žalitve ustrezno odziva. Zaenkrat je Klepec pred agresivnim govorom še dokaj nemočen, reagira le na uporabo nekaterih vulgarnih besed, ne simulira pa užaljenosti ali jeze, ki bi ju v podobnih primerih pričakovali od človeških uporabnikov.

Uporabnik: ti si peder
Klepec: Zakaj me žališ?
Uporabnik: ti si idiot!!!!!!
Klepec: Koliko časa pa misliš, da bom še prenašal tvoje žaljivke?
Uporabnik: DO KONCA PIČKA TI MATERINA!!!!!!
Klepec: Tako prostaškega sogovornika pa že dolgo nisem imel.
Uporabnik: kr navad se ti zafukanc zafukani

Primer pogovora 11: Agresivni govor.

5. Zaključek

Na več mestih je že bilo rečeno, da je program Klepec projekt v procesu in da bo za njegov razvoj ter posledično kvalitetnejše delovanje potrebnega še mnogo dela. Trenutno največji izziv predstavlja nadgradnja, ki bo programu omogočala držati rdečo nit pogovora in s tem kohezivno ter koherentno dialoško komunikacijo, kakršna je v navadi med človeškimi sogovorniki. Kratkoročnejši cilji so vgradnja izboljšanega lematizatorja ter jezikovnega analizatorja, prestrukturiranje leksikalne baze, da bo omogočala dopolnitev s semantičnimi ter kolokacijskimi informacijami ter postopno dodajanje podatkovnih baz z znanjem o svetu. Načrti izboljšave antropomornosti pa gredo predvsem v smer načrtovanja programskih modulov za različna Klepčeva razpoloženja ter različne tipe diskurza glede na jezikovne tendence uporabnika.

6. Literatura

Ahrenberg, L., A. Jönsson in N. Dahlbäck, 1990. Discourse Representation and Discourse Management for a Natural Language Dialogue System. V J. Allen, B. Miller, E. Ringger in T. Sikorski (ur.), *Proceedings of the second nordic conference on text comprehension in man and machine.* < http://www.ida.liu.se/~arnjo/papers/notex-90.pdf >

Bickmore, T. in J. Cassell, 2000. »How about this weather?«: Social Dialogue with Embodied Conversational Agents. *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents.* < http://www.misu.bmc.org/~bickmore/publications/SIA00.pdf>

De Angeli, A. in R. Carpenter, 2005. Stupid computer: Abuse and social identities. Objavljeno na < http://www.agentabuse.org/deangeli.pdf >.

Foner, L. N., 1997. Entertaining Agents: a Sociological Case Study. *AGENTS '97: Proceedings of the first international conference on Autonomous agents.* Kalifornija: ACM Press. 122–129. <http://foner.www.media.mit.edu/people/foner/Reports/Agents-97/Julia.pdf>

Morkes, J., H. K. Kernal in C. Nass, 1999. Effects of Humor in Task-Oriented Human-Computer Interaction and Computer-Mediated Communication. *Human-computer Interaction* vol.14 / št. 4*:* 395–435.

Romih, M. in P. Holozan, 2002. Sporazumevanje z računalnikom v naravnem jeziku. V T. Erjavec in J. Gros (ur.), *Jezikovne tehnologije – Mednarodna multi-konferenca Informacijska družba*. Ljubljana: Inštitut »Jožef Stefan«. 168.

Weizenbaum, J., 1966. ELIZA: A Computer Program For the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM* 9/1: 36–45.

[4]Da so uporabniki žaljivi tudi pri nas, je pokazala raziskava Klepcu najpogosteje zastavljenih vprašanj, ki smo jo opravili na vzorcu pogovorov, ki so potekali med 1. 10. ter 19. 12. 2005. Uporabniška vprašanja smo kategorizirali na poizvedovalna (sprašujejo po informaciji) ter odločevalna (odgovor je *da* oz. *ne*). Žalitve (ki so vprašanja le napol, saj glavni namen uporabnika ni poizvedovanje, pač pa izražanje negativnega mnenja o programiranem sogovorniku) najdemo le med odločevalnimi vprašanji. V skupini 550-ih odločevalnih vprašanj daleč največji delež zajemajo vprašanja oz. žalitve o spolnih preferencah, predvsem homoseksualnosti (49 %). Na drugem mestu so vprašanja o partnerskem statusu (17 %), na tretjem mestu vprašanja o znanju angleščine (14 %), na četrtem pa ponovno žalitve, tokrat umskih sposobnosti (7 %). Ostale kategorije so manj zastopane.

Osnovna zgradba razpoznavalnika slovenskega tekočega govora UMB Broadcast News

Andrej Žgank, Tomaž Rotovnik, Mirjam Sepesy Maučec in Zdravko Kačič

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
Smetanova ul. 17, SI-2000 Maribor, Slovenija
andrej.zgank@uni-mb.si http://www.dsplab.uni-mb.si

Povzetek

V članku bomo predstavili zasnovo in osnovno zgradbo prvega slovenskega razpoznavalnika tekočega govora za domeno Broadcast News. Takšen razpoznavalnik govora je namenjen razpoznavanju govora v televizijskih (in radijskih) dnevnoinformativnih oddajah. Le-te predstavljajo zaradi zelo različnega akustičnega okolja in razlik v načinu govora zelo kompleksen problem na področju jezikovnih tehnologij. Sistem UMB Broadcast News smo zasnovali na slovenski govorni in tekstovni bazi BNSI Broadcast News. Za pripravo akustično homogenih zvočnih intervalov smo uporabili postopek akustične segmentacije na osnovi Gaussovih modelov. Kot akustične modele smo uporabili grafemske kontekstno odvisne HMM modele s 16 kombinacijami porazdelitev verjetnosti na stanje. Tvorili smo nabor različnih bigramskih in trigramskih jezikovnih modelov z 20k oz. 60k besedami v slovarju. Vrednotenje sistema UMB Broadcast News smo izvedli z bigramskim jezikovnim modelom ter slovarjem z 20k besedami. Dosegli smo 40,5% napako razpoznavanja besed na testnem naboru baze BNSI.

Basic Structure of the UMB Slovenian Broadcast News Transcription System

This paper presents basic structure and design scheme of the first Slovenian Broadcast News transcription system. Such system is used for large vocabulary continuous speech recognition of television (and radio) news shows. The acoustic environment and speaking style in Broadcast News speech corpora are heterogeneous and as such very complex for speech recognition. The UMB Broadcast News system was developed using the speech and text database BNSI Broadcast News. Acoustic homogeneous speech intervals were produced with the acoustic segmentation based on Gaussian models. Grapheme based context-dependent HMM acoustic models with 16 mixture probability density functions were applied. A set of different bigram and trigram language models with 20k and 60k words in the vocabulary was generated. The UMB Broadcast News system was evaluated with the bigram language model and 20k words in the vocabulary. The 40,5% word error rate was achieved on the BNSI evaluation set.

1. Uvod

Na področju avtomatskega razpoznavanja govora ločimo med sistemi različne kompleksnosti. Najtežjo nalogo tako predstavlja razpoznavanje tekočega spontanega govora z velikim slovarjem besed. V to kategorijo sodi tudi domena razpoznavalnikov govora Broadcast News (BN), kjer razpoznavamo govor v dnevnoinformativnih televizijskih oddajah. Razvoj na področju takšnih sistemov se je začel leta 1996 (Pallet, 2002) v okviru projekta ameriških organizacij DARPA in NIST. Danes v svetu najdemo razpoznavalnike govora iz domene BN, razvite za različne svetovne jezike (Gauvain et al., 2002; Woodland, 2002; Beyerlein et al., 2002). Kompleksnost samega razpoznavalnika govora je odvisna tudi od lastnosti jezika. Slovenski jezik sodi zaradi svojih lastnosti (pregibna narava, dvojina, relativno prost vrstni red besed v stavku,…) med težavnejše jezike za razpoznavanje govora, kar je ena izmed glavnih ovir za večji razvoj tega področja.

V članku bomo predstavili prvi razpoznavalnik slovenskega tekočega spontanega govora za domeno Broadcast News, ki je nastal na Univerzi v Mariboru[1]. Sistem UMB Broadcast News je trenutno eden izmed najkompleksnejših slovenskih razpoznavalnikov govora. V prispevku bomo opisali zasnovo in osnovno strukturo sistema ter podali preliminarne rezultate razpoznavanja govora.

[1] Delo je bilo delno financirano s strani Agencije za raziskovalno dejavnost Republike Slovenije po pogodbi št. P2-0069.

Osnovni jezikovni vir, ki smo ga uporabili pri razvoju UMB Broadcast News sistema, je slovenska baza BNSI Broadcast News (Zögling et al., 2003; Žgank et al., 2004/1; Žgank et al., 2005/1). Zaradi raznolikosti govornega materiala vključenega v bazo BNSI, se takšen razpoznavalnik govora sreča tako z branim govorom v studijskem okolju (npr. agencijske novice), kot tudi z narečnim spontanim govorom s šumom iz okolice ali glasbo v ozadju (npr. intervju na terenu).

Pri razvoju sistema UMB BN smo izhajali iz predhodnih razpoznavalnikov slovenskega govora (Žgank et al., 2001; Rotovnik, 2004), ki so bili razviti za bazo SNABI (Dreo, 1995) ter iz izkušenj pridobljenih v okviru razvoja demonstracijskega sistema za podnaslavljanje televizijskih oddaj v živo (Žgank et al., 2004/2).

V nadaljevanju članka bomo v drugem poglavju opisali različne slovenske jezikovne vire, ki smo jih uporabili pri razvoju. Osnovno zgradbo sistema UMB BN bomo opisali v tretjem poglavju. Rezultate razpoznavanja govora z bazo BNSI bomo predstavili v četrtem poglavju. Zaključek s smernicami za prihodnje delo bomo podali v zadnjem – petem – poglavju.

2. Jezikovni viri

Razpoložljivost primernih jezikovnih virov je osnova za razvoj vsakega razpoznavalnika govora. Za učenje akustičnih modelov tako potrebujemo transkribiran govorni material, za razvoj jezikovnih modelov pa tekstovne korpuse.

Učenje akustičnih modelov za sistem UMB BN smo izvedli z govornim korpusom slovenske baze BNSI Broadcast News (Žgank et al., 2004/1). Baza vsebuje 36

ur transkribiranega govornega materiala iz obdobja 1999-2003, zajetega iz dnevnoinformacijskih oddaj RTV Slovenija. Med razvojem osnovne zgradbe razpoznavalnika govora je potreben precejšen delež ročnega dela za pripravo in uskladitev vseh jezikovnih virov. Zato smo zaradi poenostavitve dela uporabili samo polovico razpoložljivega govornega korpusa baze BNSI. Učni korpus je tako vseboval 15 ur govornega materiala, testni nabor pa približno 1,5 ure govora.

Tabela 1 prikazuje statistiko virov, ki smo jih uporabili pri učenju jezikovnih modelov. **BNSI-Speech** označuje transkripcije 69 oddaj iz osnovnega in razširjenega učnega govornega korpusa baze BNSI (Žgank et al., 2005/1). Oznaka **BNSI-Text** predstavlja tekstovni korpus baze BNSI (Žgank et al., 2005/1), ki vsebuje scenarije dnevnoinformativnih oddaj RTV Slovenija iz obdobja 1998-2004. Iz tekstovnega korpusa smo izločili tiste mesečne sklope scenarijev, v katerih se nahajata oddaji iz testnega nabora. **Večer** je korpus člankov časopisa Večer, ki smo jih zbrali v obdobju od leta 1998 do leta 2001. Poudariti velja, da sta prva dva jezikovna vira predstavnika govorjenega jezika, slednji pa sodi v skupino jezikovnih virov pisanega jezika. Najpomembnejše lastnosti govorjenega jezika, ki jih v pisanem jeziku ne zasledimo so: ponovitve, popravki, slovnično neujemanje in svobodni vrstni red besed. Vse naštete pojave zasledimo v korpusu **BNSI-Speech**, nekatere med njimi tudi v korpusu **BNSI-Text**. Naš cilj je razpoznavanje govorjenega jezika, zato sta iz tega stališča pomembnejša korpusa **BNSI-Speech** in **BNSI-Text.** Po drugi strani je uspešnost statističnega modeliranja odvisna od velikosti učnega korpusa, zato se kot pomemben vir znanja izkaže tudi korpus Večer.

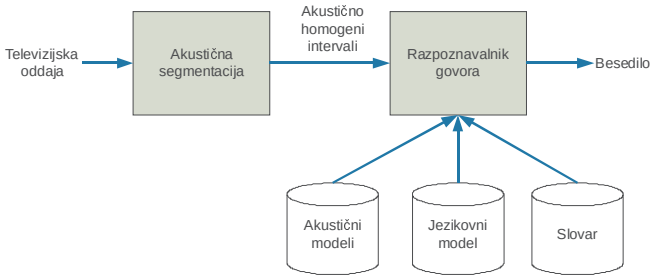
Korpus	Število stavkov	Število besed	Štev. različnih besed
BNSI-Speech	30k	573k	51k
BNSI-Text	614k	11M	175k
Večer	12M	95M	736k

Tabela 1: Statistika korpusov, uporabljenih pri gradnji jezikovnega modela.

Slovar razpoznavalnika smo sestavili tako, da je vseboval 20k in 60k najpogostejših besed v korpusih govorjenega jezika: **BNSI-Speech** in **BNSI-Text**.

3. Arhitektura sistema UMB Broadcast News

Sistem za razpoznavanje tekočega govora UMB Broadcast News je zasnovan na statističnem modeliranju govora. Blokovna shema osnovnih gradnikov sistema je predstavljena na sliki 1.



Slika 1: Osnovna zgradba sistema UMB Broadcast News za razpoznavanje govora.

3.1. Akustična segmentacija

Naloga akustične segmentacije je iz vhodnega zvočnega signala izrezati akustično homogene dele, ki so primerni za razpoznavanje govora. Tukaj lahko upoštevamo različne kriterije za doseganje homogenosti. Nekateri najpogostejši kriteriji so: zvok/tišina, govor/glasba/šum, spol govorca, združevanje govorcev v skupine, …

Osnovno zasnovo sistema UMB BN za razpoznavanje govora smo zaenkrat pripravili za vključitev segmentacije na osnovi prvih treh kriterijev (Žgank, 2006). V prvem koraku segmentacije tako izločimo iz zvočnega signala dele, ki vsebujejo tišino. Sledi določitev akustičnega ozadja govornega signala, ki je posebej namenjeno identifikaciji govora z glasbo v ozadju. Le-ta namreč zelo oteži razpoznavanje govora. V tretjem koraku določimo spol govorca, kar omogoča uporabo različnih akustičnih modelov za oba spola. Ker količina govornega materiala, ki smo jo trenutno uporabili za učenje akustičnih modelov, ne omogoča kvalitetnega modeliranja ob uporabi drugih dveh kriterijev, smo za vrednotenje razpoznavalnika govora (tabela 4) uporabili samo klasifikacijo zvok/tišina.

Modul za segmentacijo smo zasnovali na osnovi Gaussovih modelov (GMM). Učenje GMM-ov je potekalo na 9 oddajah iz nabora baze BNSI. Glede na razpoložljivost zvočnega materiala za posamezno kategorijo, ter glede na dosežene rezultate segmentacije, smo uporabili različno kompleksnost modelov GMM – najpreprostejši model je vseboval 4 Gaussove porazdelitve verjetnosti, najkompleksnejši pa 512.

3.2. Učenje akustičnih modelov

V sistemu UMB BN smo zasnovali akustične modele na osnovi tri stanjskih levo-desnih prikritih modelov Markova s kombinacijami zveznih Gaussovih porazdelitev verjetnosti. Vektor značilk vsebuje 12 mel kepstralnih koeficientov in energijo, ter njihove prve in druge odvode.

Akustične modele smo zasnovali na grafemski osnovni enoti (Žgank in Kačič, 2005/2). V osnovnem naboru je bilo 27 akustičnih modelov. Z uporabo grafemov smo se izognili dodatni napaki, ki bi jo v sistem vnesla grafemsko-fonemska pretvorba. Za učenje prikritih modelov Markova smo uporabili prosto dostopno orodje HTK (HTK, 2006).

V prvem koraku učenja smo tvorili kontekstno neodvisne akustične modele na osnovi inicializacije z globalnimi vrednostmi. Z dobljenimi akustičnimi modeli smo izvedli postopek prisilne poravnave in izločili 0,63% neustreznih transkripcij. V drugem koraku smo ponovili postopek učenja. Sedaj smo inicializacijo izvedli na

osnovi ločenih vrednosti za vsak posamezni model. Ponovili smo postopek prisilne poravnave, ter kot neustrezne izločili dodatnih 0,47% transkripcij.

V tretjem prehodu smo korakoma nadaljevali z učenjem kontekstno neodvisnih akustičnih modelov. V nadaljevanju smo tvorili notranje-besedne kontekstno odvisne akustične modele, s katerimi se praviloma doseže boljši rezultat kot z kontekstno neodvisnimi modeli. Ker se posledično drastično poviša število prostih parametrov akustičnih modelov, ki jih je potrebno oceniti, smo uporabili postopek vezave stanj z odločitvenim drevesom (Young et al., 1994). S pomočjo tega postopka vežemo stanja, ki so si akustično dovolj podobna med seboj in tako združimo razpoložljivi učni material. Inicializacijo odločitvenih dreves smo izvedli s podatkovno tvorjenimi grafemskimi razredi (Žgank et al., 2005/3). Število Gaussovih porazdelitev verjetnosti na stanje v kontekstno odvisnih akustičnih modelih smo korakoma povečevali do 16. Po končanem postopku učenja smo akustične modele pretvorili v format, ki ga podpira razpoznavalnik govora razvit na Univerzi v Mariboru.

3.3. Učenje jezikovnih modelov

Zgradili smo standardne bigramske in trigramske jezikovne modele. Uporabili smo Good-Turingovo glajenje. V model smo vključili vse bigrame oz. trigrame, tudi tiste, ki so se pojavili samo enkrat. Posledično so nastali relativno obsežni jezikovni modeli. Tabela 2 prikazuje velikosti jezikovnih modelov, učenih na različno velikih učnih besedilnih zbirkah. Jezikovni model **LM1** je učen le na korpusu **BNSI-Speech**, **LM2** na korpusih **BNSI-Speech** in **BNSI-Text** ter **LM3** na vseh treh korpusih: **BNSI-Speech**, **BNSI-Text** in korpusu **Večer**. V jezikovnem modelu **LM3** imajo vsi trije korpusi enak vpliv, ki smo ga v nadaljevanju želeli uravnotežiti glede na značilnosti BN korpusa. Zgradili smo tri komponente jezikovnega modela: prvo komponento na korpusu **BNSI-Speech**, drugo na korpusu **BNSI-Text** in tretjo na korpusu **Večer**. Optimalno razmerje med komponentami smo poiskali tako, da smo v iterativnem postopku iskali optimalne interpolacijske koeficiente (ki dajo najmanjšo perpleksnost interpoliranega modela). Nastali model smo poimenovali **LM4**. Vse omenjene postopke učenja jezikovnih modelov smo izvajali ločeno za slovar velikosti 20k besed in slovar velikosti 60k besed.

Jezikovni model	20k		60k	
	2-grami	3-grami	2-grami	3-grami
LM1	186k	298k	244k	380k
LM2	1,403M	3,542M	2,004M	4,640M
LM3	5,250M	18,623M	9,037M	28,307M
LM4	5,250M	18,623M	9,037M	28,307M

Tabela 2: Velikosti jezikovnih modelov, pri slovarjih 20k in 60k besed.

3.4. Razpoznavalnik govora

Za razpoznavalnik govora smo uporabili statistični pristop zasnovan na Bayesovem odločitvenem pravilu, ki vsebuje naslednje komponente: akustični analizator, iskalni algoritem, stohastična modela (akustični in jezikovni model). Akustični analizator izvaja kratkočasovno spektralno analizo govornega signala. Akustični model zajema trenutno akustično in časovno karakteristiko govorca in skupaj z jezikovnim modelom podaja osnovna jezikovna vira za iskalni algoritem. Iskalni algoritem določi besedni niz neznane dolžine na osnovi največje aposteriori verjetnosti. Za naš razpoznavalnik govora smo uporabili sinhroni iskalni algoritem (niz akustičnih vektorjev značilk se procesira od začetka do konca govornega signala), ki za zmanjšanje računske zahtevnosti vključuje Viterbijevo aproksimacijo. Za zmanjšanje iskalnega prostora smo uporabili drevesno obliko slovarja, katerega značilnost so skupna vozlišča za enake začetne foneme besed. Sama izvedba slovarja vpliva na velikost iskalnega prostora. Ker smo v razpoznavalniku govora uporabili bigramske jezikovne modele je bilo potrebno za vsako besedo iz slovarja tvoriti kopije dreves slovarja za ohranitev zgodovine hipotez v iskalnem prostoru. V korenih dreves se nato izvaja rekombinacija na besednem nivoju. Upošteva se samo najverjetnejša hipoteza, ki pride v določenem časovnem okviru v koren, vse ostale se zavržejo. Zaradi vzporedne obdelave vseh možnih hipotez iskalnega prostora, smo v vsakem časovnem okviru izključili tiste hipoteze, katerih verjetnost je bila za določen prag slabša od trenutno najboljše hipoteze. Tako imenovano snopovno omejevanje dodatno zmanjša aktivni iskalni prostor, torej del iskalne mreže, ki jo je potrebno preiskati, za nekaj desetkrat in s tem pohitri proces razpoznavanja govora. Za učinkovitejše snopovno omejevanje smo v iskalni prostor predčasno vključili verjetnosti jezikovnega modela. Tako imenovan pogled naprej jezikovnega modela v vsako skupno vozlišče drevesnega slovarja postavi najboljšo verjetnost za vse možne besede, ki lahko nastanejo iz danega vozlišča. Za dodatno zmanjšanje iskalnega prostora smo v vsakem časovnem okviru omejili število aktivnih vozlišč, katerim se določajo (izračunavajo) nove poti v iskalni mreži. Omejevanje smo izvedli na dva načina. V prvem primeru smo aktivne modele sortirali po najboljšem rezultatu in pri tem ohranili samo N najboljših. V drugem primeru smo nadaljevali samo N najboljših delnih hipotez, ki so se v trenutnem času končale v zadnjem stanju in zadnjem vozlišču dreves.

4. Rezultati razpoznavanja govora

Pred samim vrednotenjem razvitega sistema UMB Broadcast News bomo podali statistiko uporabljenih akustičnih in jezikovnih modelov, ki kaže na kompleksnost razvitega sistema.

V naboru sta bila pred vezavo stanj 16.902 kontekstno odvisna grafemska akustična modela. Po vezavi stanj je ostal 21,5% neodvisnih akustičnih modelov, ki so imeli skupaj približno 175k Gaussovih porazdelitev verjetnosti. Skupna velikost datoteke z akustičnimi modeli je znašala 86,6 MB.

Jezikovni model	20k		60k	
	2-gram	3-gram	2-gram	3-gram
LM1	504	459	837	821
LM2	419	372	641	575
LM3	344	285	471	390
LM4	292	234	400	316

Tabela 3:Perpleksnosti jezikovnih modelov.

Delež besed izven slovarja (OOV) v naboru testnih stavkov pri slovarju z 20k besedami je 12.34% in pri slovarju s 60k besedami 5.44%. Tabela 3 prikazuje perpleksnosti nabora testnih stavkov z uporabo različnih jezikovnih modelov. Prvi trije modeli LM1-3 kažejo, da z večanjem učnega korpusa perpleksnost testnih stavkov pada.

Rezultate razpoznavanja govora bomo podali za velikost slovarja 20k najpogostejših besed v tekstovnih korpusih in bigramski jezikovni model LM4. Beležili smo napako razpoznavanja besed (NRB), hitrost razpoznavanja (večkratnik realnega časa, CPU: P4, 2,4 GHz) in velikost iskalnega prostora, izraženega s povprečnim številom aktivnih modelov. Povprečno število aktivnih modelov smo izračunali tako, da smo najprej v vsakem časovnem okviru za vsak testni stavek pred procesom omejevanja beležili število aktivnih modelov. Vsoto modelov za posamezni stavek smo normalizirali s številom okvirov. Skupno vsoto normaliziranih aktivnih modelov posameznih stavkov smo povprečili z velikostjo testne množice.

Eksperiment	NRB[%]	Hitrost	Št. aktivnih modelov
B20	40,5	42,3	26620

Tabela 4: Rezultat razpoznavanja z besednimi modeli pri velikosti slovarja 20k enot.

Z velikostjo slovarja 20k besed smo dosegli napako razpoznavanja 40,5 % (tabela 4). Podrobnejša analiza je pokazala, da imajo največji vpliv na napako razpoznavanja manjkajoče besede, ki jih ni v slovarju. 20% vseh zamenjanih besed predstavljajo besede, ki so izpeljane iz skupne leme in so si fonetično zelo podobne. Analiza rezultatov razpoznavanja nakazuje na uporabo tehnik razpoznavanja primernih za pregibne jezike (npr.: uporaba podbesednih enot razpoznavanja). V primerjavi z enakovrednim najboljšim predhodnim razpoznavalnim sistemom, kjer smo uporabili govorno bazo SNABI, smo zmanjšali napako razpoznavanja za absolutno 12,8%.

Z omejevanjem iskalnega prostora smo vplivali na hitrost razpoznavanja in ga optimirali glede na najboljši rezultat razpoznavanja. Dosežena hitrost razpoznavanja je bila 42,3-kratna vrednost realnega časa, pri povprečju 26620 aktivnih modelov na časovni okvir.

5. Zaključek

V članku smo predstavili zasnovo in osnovno strukturo prvega slovenskega sistema za razpoznavanje tekočega govora v domeni Broadcast News. Dosegli smo vzpodbudne rezultate razpoznavanja govora.

V nadaljnjem delu bomo korakoma povečevali velikost uporabljene učne govorne baze in tako vključili segmentacijo zvočnega signala glede na druga dva kriterija, ter povečali slovar razpoznavalnika govora. Postopoma bomo sistemu UMB Broadcast News dodajali tudi nove module, s katerimi lahko pričakujemo dodatno izboljšanje kvalitete razpoznavanja govora.

6. Literatura

Beyerlein, P., Aubert, X., Haeb-Umbach, R., Harris, M., Klakow, D., Wendemuth, A., Molau, S., Ney, H., Pitz, M., and Sixtus, A., (2002). Large vocabulary continuous speech recognition of Broadcast News - The Philips/RWTH approach, *Speech Communication*, Volume 37, Issues 1-2, 109-131.

Dreo, D., (1995). Slovene speech data base SNABI. *Dialog Man - Machine: second International Workshop*, Maribor, Slovenija.

Gauvain, J., Lamel, L., Adda, G., (2002). The LIMSI Broadcast News Transcription System. *Speech Communication,* Volume 37, Issues 1, 89-108.

HTK domača stran, http://htk.eng.cam.ac.uk.

Pallett, D. S. (2002). The role of the National Institute of Standards and Technology in DARPA's Broadcast News continuous speech recognition research program. *Speech Communication*, Vol. 37, Issues 1-2, 1:3--14.

Rotovnik T., Avtomatsko razpoznavanje govora za pregibni jezik z velikim slovarjem besed z uporabo podbesednih modelov osnova - končnica, *Doktorska disertacija, Univerza v Mariboru*, 2004.

Zögling Markuš, A., Žgank, A., Rotovnik, T., Sepesy Maučec, M., Vlaj, D., Hozjan, V., Kotnik, B., (2003). Spoken Language Resources at University of Maribor. *Proc. of 10th International Workshop Advances in Speech Technology 2003*, Maribor, Slovenija.

Woodland, P.C., (2002). The development of the HTK Broadcast News transcription system: An overview, *Speech Communication*, Volume 37, Issues 1-2, 47-67.

Young, S., Odell, J., Woodland, P., (1994). Tree-based State Tying for High Accuracy Acoustic Modelling. *Proc. ARPA Human Language Technology Conference* Plainsboro.

Žgank, A., Kačič, Z., and Horvat, B., (2001). 'Large vocabulary continuous speech recognizer for Slovenian language, *Proc. Text, speech and dialogue : 4th international conference, TSD 2001*, Železna Ruda, Češka, *Lecture notes in Artificial Intelligence*, Vol. 2166, 242-248, Springer.

Žgank, A., Rotovnik, T., Sepesy Maučec, M., Verdonik, D., Kitak, J., Vlaj, D., Hozjan, V., Kačič, Z., Horvat, B., (2004/1). Acquisition and annotation of Slovenian broadcast news database. *Fourth international conference on language resources and evaluation, LREC 2004*, Lizbona, Portugalska.

Žgank, A., Rotovnik, T., Verdonik, D., Kačič, Z., (2004/2). Baza Broadcast News za slovenski jezik (BNSI) in sistem za razpoznavanje tekočega govora. Zbornik konference Jezikovne tehnologije 2004, Ljubljana, Slovenija.

Žgank, A., Verdonik, D., Zögling Markuš, A., Kačič, Z., (2005/1). BNSI Slovenian Broadcast News database - speech and text corpus. *Proc. Interspeech 2005*, Lizbona, Portugalska.

Žgank, A., Kačič, Z., (2005/2). Primerjava treh tipov akustičnih osnovnih enot razpoznavalnika slovenskega govora. *Elektrotehniški vestnik*, 2005, Ljubljana, Slovenija.

Žgank, A., Horvat, B., Kačič, Z., (2005/3). Data-driven generation of phonetic broad classes, based on phoneme confusion matrix similarity. *Speech Communication*, vol. 47, issue 3, 379--393, november 2005.

Žgank, A., (2006), Acoustic Segmentation for Slovenian Broadcast News Transcription System. *Proc. of 13th International Workshop Advances in Speech Technology 2006*, v tisku, Maribor, Slovenija.

Rezultati vrednotenja dveh sistemov Čarovnik iz Oza

Melita Hajdinjak, France Mihelič

Laboratorij za umetno zaznavanje, sisteme in kibernetiko,
Fakulteta za elektrotehniko, Univerza v Ljubljani
Tržaška 25, 1000 Ljubljana
{melita.hajdinjak, france.mihelic}@fe.uni-lj.si

Povzetek

Opišemo postopek in rezultate vrednotenja učinkovitosti dveh sistemov Čarovnik iz Oza z ogrodjem PARADISE. Vpeljemo t. i. *parametre podatkovne zbirke*, ki odražajo velikost ter zgradbo podatkovne zbirke in se v literaturi o vrednotenju učinkovitosti sistemov za dialog ne pojavljajo. Izpeljemo funkciji učinkovitosti obeh sistemov Čarovnik iz Oza, ki nas vodita do spoznanja, da je predstavitev znanja oz. zgradba podatkovne zbirke sistema za dialog izjemnega pomena in da so parametri podatkovne zbirke pri vrednotenju sistemov za podajanje informacij nepogrešljivi.

Results from the Evaluation of two Wizard-of-Oz Systems

The results from the PARADISE evaluation of data from two Wizard-of-Oz experiments are given. The *database parameters* expressing the database size and the database structure, which have not so far been reported in the literature as costs for user satisfaction, are introduced. The performance functions for both Wizard-of-Oz systems lead to the conclusion that the system's knowledge representation is of great importance and that the database parameters are indispensable when evaluating the performance of information-providing dialogue systems.

1. Uvod

Z namenom omogočiti primerjavo različnih govornih vmesnikov, kjer nas zanima, v kolikšni meri posamezni dejavniki vplivajo na učinkovitost in kako strategija vodenja dialoga vpliva na zadovoljstvo uporabnikov, je bilo leta 1997 (Walker et al., 1997a) kot potencialna splošna metodologija vrednotenja učinkovitosti govornih vmesnikov predlagano ogrodje PARADISE (PARAdigm for DIalogue System Evaluation). Ogrodje PARADISE omogoča izpeljavo ocene učinkovitosti sistema kot uteženo linearno kombinacijo od domene odvisnih *parametrov uspešnosti naloge* in *cen dialoga* (tj. *parametri učinkovitosti dialoga* in *parametri kakovosti dialoga*), zajema pa model učinkovitosti sistema, ki za osnovni cilj postavlja maksimiranje zadovoljstva uporabnikov.

Model učinkovitosti sistema, ki ga zajema ogrodje PARADISE, trdi, da lahko funkcijo učinkovitosti sistema določimo z multiplo linearno regresijo (Seber, 1977) z zadovoljstvom uporabnikov kot odvisno spremenljivko ter parametri uspešnosti naloge in cen dialoga kot neodvisnimi spremenljivkami:

$$\text{Učinkovitost} = \alpha * \mathcal{N}(\kappa) - \sum_{i=1}^{n} w_i * \mathcal{N}(c_i).$$

Pri tem je α utež Kappa koeficienta κ, w_i so uteži cen dialoga c_i, $\mathcal{N}$ pa je funkcija normalizacije (Hajdinjak in Mihelič, 2006c). Z normalizacijo parametrov κ in c_i dosežemo relevantnost in primerljivost uteži preslikanih parametrov $\mathcal{N}(\kappa)$ in $\mathcal{N}(c_1), \ldots, \mathcal{N}(c_n)$.

Funkcija učinkovitosti tedaj omogoča napovedovanje zadovoljstva uporabnikov, vrednotenje učinkovitosti in izboljševanje sistema, primerjavo sistemov z istimi ali različnimi domenami, samodejno iskanje problematičnih dialogov in spreminjanje strategije vodenja dialoga že med interakcijo.

Ogrodje PARADISE smo uporabili pri vrednotenju učinkovitosti dveh nedograjenih sistemov za podajanje informacij o vremenu in vremenski napovedi (Žibert et al., 2004), s katerima smo izvajali eksperiment Čarovnik iz Oza (Hajdinjak in Mihelič, 2004). Eksperiment Čarovnik iz Oza je trenutno najboljša alternativa za zbiranje podatkov, ki izražajo jezik komunikacije človek–stroj. V teh eksperimentih so uporabniki prepričani, da se pogovarjajo s strojem – računalnikom, kar pa ni res. V resnici za računalnikom sedi človek (čarovnik), ki vsaj delno simulira delovanje sistema za dialog.

V skladu z našo trditvijo (Hajdinjak in Mihelič, 2006a), da je treba vplive samodejnega razpoznavanja govora iz sistema odstraniti, če želimo vrednotiti učinkovitost kakšnega drugega modula (v našem primeru modula za vodenje dialoga), je človek čarovnik v prvem sistemu simuliral razumevanje govora (razpoznavanje govora in razumevanje naravnega jezika) ter vodenje dialoga, v drugem sistemu pa le razumevanje govora.

Oba sistema sta se poleg načina vodenja dialoga razlikovala še v vrsti podatkovne zbirke – v prvem eksperimentu je sistem dostopal do relacijske zbirke vremenskih podatkov, v drugem pa do posebne sodelujoče podatkovne zbirke (Hajdinjak, 2006b). Ko bomo govorili o strukturi dialoga, bomo uporabljali pojma *konverzacijskih iger* in *konverzacijskih potez*. *Konverzacijske igre* povezujemo z željami oz. konverzacijskimi cilji, kot je na primer cilj pridobiti določeno informacijo, in so sestavljene iz zaporedja izjav, ki se začnejo s pobudo in končajo, ko je cilj igre dosežen ali igra prekinjena. Sestavne dele konverzacijskih

iger imenujemo *konverzacijske poteze*. To so izjave, deli izjav ali množice izjav, ki izražajo isto namero, kot je na primer potrditev ali preverjanje.

2. Izbira regresijskih parametrov

Tako kot avtorice ogrodja PARADISE (Walker et al., 1997a) smo izbrali en sam parameter uspešnosti naloge:

- **Kappa koeficient** (κ) meri uspešnost sistema pri reševanju nalog, ki mu jih naloži uporabnik. Napake, do katerih pride pri razumevanju govora in jih sistem v tekoči konverzacijski igri odpravi, ne znižajo vrednosti tega koeficienta. Ker je v naših eksperimentih razumevanje govora simuliral čarovnik, koeficient κ, izračunan iz podatkov prvega eksperimenta, kaže uspešnost oz. spretnost čarovnika in fleksibilnost grafičnega vmesnika, ki je čarovniku pomagal voditi dialog, pri reševanju navideznih nesporazumov med uporabnikom in čarovnikom. V drugem eksperimentu, ko je vodenje dialoga prevzel posebej za to nalogo zgrajen modul (Hajdinjak, 2006b), koeficient κ kaže uspešnost tega modula za vodenje dialoga pri reševanju navideznih nesporazumov med uporabnikom in čarovnikom, ki so nastali ali zaradi tipkarskih napak čarovnika ali zaradi neavtoriziranih posegov čarovnika v pomenske predstavitve uporabnikovih izjav.

Za parametre učinkovitosti dialoga smo izbrali:

- **Povprečni čas dialoga** (MET) meri povprečni čas trajanja informacijskih konverzacijskih iger, katerih namen je pridobiti določeno informacijo in jih uporabnik vpelje v času svoje interakcije s sistemom.

- **Povprečno število potez** (MUM) meri povprečno število konverzacijskih potez, ki jih uporabnik potrebuje za izvedbo ali prekinitev vpeljanih informacijskih iger.

Čeprav so cene dialoga definirane kot parametri, katerih minimiranje ugodno vpliva na zadovoljstvo uporabnikov, je včasih naravneje vzeti količine, katerih učinek je ravno obraten. Izbrali smo naslednje parametre kakovosti dialoga:

- **Izpolnitev naloge** (Comp) se nanaša na mnenje uporabnika o tem, ali je od sistema dobil odgovor na prvo vprašanje oz. prvo nalogo, ki smo mu jo v eksperimentu zastavili (Hajdinjak in Mihelič, 2004). Parameter Comp zavzame vrednost 0, če uporabnik meni, da ni dobil odgovora na svoje vprašanje, in vrednost 1 v nasprotnem primeru.

- **Število uporabnikovih iniciativ** (NUI) šteje začetne konverzacijske poteze, s katerimi uporabnik vpelje informacijske igre.

- **Povprečno število besed** (MWT) meri povprečno število besed, vsebovanih v konverzacijskih potezah uporabnika.

- **Povprečni čas odziva** (MRT) meri povprečni čas, ki ga sistem porabi, da se odzove. V prvem eksperimentu je bil ta čas povezan z izbiro odgovorov na grafičnem vmesniku, v drugem pa s tipkanjem pomenskih predstavitev uporabnikovih potez.

- **Število manjkajočih odzivov** (NMR) meri razliko med številom potez sistema in številom potez uporabnika. Ta parameter izraža tako število potez, ki sledijo, ko sistem v vnaprej določenem času ne zazna govora, kakor tudi nepripravljenost uporabnika, da bi sistem odzdravil.

- **Število neprimernih iniciativ** (NUR) in **delež neprimernih iniciativ** (URR) merita število oz. delež začetnih potez uporabnika, katerih vsebina ne ustreza domeni sistema.

- **Število neprimernih odzivov** (NIR) in **delež neprimernih odzivov** (IRR) merita število oz. delež kontekstno neprimernih potez sistema. Sem štejemo tudi poteze, s katerimi sistem uporabnika prosi, naj ponovi zadnjo izjavo.

- **Število napak** (Error) meri napake sistema, kamor štejemo prekinitve telefonske povezave, neustrezno oblikovane povedi in nasprotujoče si odgovore.

- **Število pomoči** (NHM) in **delež pomoči** (HMR) merita število oz. delež potez sistema, ki uporabniku pomagajo nadaljevati dialog.

- **Število preverjanj** (NCM) in **delež preverjanj** (CMR) merita število oz. delež potez, s katerimi sistem prosi za potrditev informacij, ki jih pridobi na osnovi zgodovine dialoga. V prvem eksperimentu čarovnik ni izvajal potez tega tipa. Čarovnik, ki je simuliral popolno razumevanje govora, je sicer na podlagi zgodovine dialoga sklepal o nenavedenih podatkih, za katere pa uporabnika ni prosil, da jih potrdi.

- **Število podanih informacij** (NGD) in **delež podanih informacij** (GDR) merita število oz. delež potez, s katerimi sistem uporabniku poda iskane informacije, ki jih najde v podatkovni zbirki.

- **Število relevantnih informacij** (NRD) in **delež relevantnih informacij** (RDR) merita število oz. delež potez sistema, ki uporabnika usmerjajo k izbiri relevantnih, dosegljivih podatkov.

- **Število nepodanih informacij** (NND) in **delež nepodanih informacij** (NDR) merita število oz. delež potez, s katerimi sistem uporabniku sporoča, da nima zahtevanega podatka in ga pri tem ne usmerja k izbiri relevantnih, dosegljivih podatkov. V prvem eksperimentu so to poteze, ki pravijo, da sistem zahtevane informacije trenutno nima ali je sploh ne ponuja. V drugem eksperimentu pusti sistem to vprašanje odprto.

- **Število prekinjenih zahtev** (NAR) in **delež prekinjenih zahtev** (ARR) merita število oz. delež informacijskih iger, ki jih uporabnik prekine še preden se končajo.

Tabela 1: Srednje vrednosti izbranih regresijskih parametrov v prvem (WOZ1) in drugem (WOZ2) eksperimentu Čarovnik iz Oza.

		WOZ1	WOZ2	p
uspešnost naloge	**Kappa koeficient** (κ)	0.94	0.98	
učinkovitost dialoga	**povprečni čas dialoga** (MET)	13.76 s	17.39 s	0.000
	povprečno število potez (MUM)	1.48 s	1.68 s	0.047
	izpolnitev naloge (Comp)	0.97	0.96	
	število uporabnikovih iniciativ (NUI)	6.49	7.51	0.005
	povprečno število besed (MWT)	9.32 s	7.56 s	0.000
	povprečni čas odziva (MRT)	5.13 s	6.38 s	0.000
	število manjkajočih odzivov (NMR)	0.60	0.75	
	število neprimernih iniciativ (NUR)	0.48	0.13	0.011
	delež neprimernih iniciativ (URR)	0.08	0.02	
	število neprimernih odzivov (NIR)	0.41	0.90	0.009
	delež neprimernih odzivov (IRR)	0.04	0.06	
kakovost dialoga	**število napak** (Error)	0.12	0.06	
	število pomoči (NHM)	0.32	0.40	
	delež pomoči (HMR)	0.03	0.03	
	število preverjanj (NCM)*	-	2.19	
	delež preverjanj (CMR)*	-	0.16	
	število podanih informacij (NGD)	4.07	4.35	
	delež podanih informacij (GDR)	0.67	0.58	
	število relevantnih informacij (NRD)	0.70	2.06	0.000
	delež relevantnih informacij (RDR)	0.10	0.28	0.005
	število nepodanih informacij (NND)	1.67	0.94	0.000
	delež nepodanih informacij (NDR)	0.22	0.12	
	število prekinjenih zahtev (NAR)	0.05	0.16	
	delež prekinjenih zahtev (ARR)	0.01	0.02	
	zadovoljstvo uporabnika (US)	34.08	31.96	0.015

Zgoraj uporabljene kratice za imena parametrov se nanašajo na angleške besedne zveze.

Izbrane parametre je treba določiti samodejno, če je to mogoče, v skrajnem primeru pa jih ročno označiti. Zavedati se namreč moramo, da neodvisne spremenljivke funkcije učinkovitosti, ki niso samodejno določljive, skrčijo uporabnost ogrodja PARADISE – samodejno iskanje problematičnih dialogov in spreminjanje strategije vodenja dialoga med interakcijo tedaj nista več mogoča.

V prvem eksperimentu Čarovnik iz Oza smo morali večino parametrov določiti ročno. Šele modul za vodenje dialoga (Hajdinjak, 2006b), vključen v drugi sistem Čarovnik iz Oza, ki je potek dialoga zelo dobro strukturiral, je omogočil samodejno določljivost velike večine izbranih parametrov. Še vedno je bilo samodejno nemogoče določiti naslednje parametre: **Kappa koeficient** (κ), **izpolnitev naloge** (Comp), **število neprimernih iniciativ** (NUR) in **število napak** (Error).

Zanimivo je, da se **število podanih informacij** (NGD) in **delež podanih informacij** (GDR), **število relevantnih informacij** (NRD) in **delež relevantnih informacij** (RDR) ter **število nepodanih informacij** (NND) in **delež nepodanih informacij** (NDR), ki jih imenujemo *parametri podatkovne zbirke*, v literaturi o vrednotenju učinkovitosti sistemov za dialog ne pojavljajo. Razlog je verjetno ta, da imajo razvijalci sistemov za dialog le redko na razpolago podatkovno zbirko, katere struktura bi bila tako zelo časovno odvisna in skopa, kot je naša. Omenjen tip parametrov pa vseeno ni ostal popolnoma neopažen. Walker, Litman, Kamm in Abella (Walker et al., 1998) razmišljajo, da bi velikost podatkovne zbirke lahko značilno vplivala na učinkovitost sistema za dialog.

Srednje vrednosti izbranih regresijskih parametrov v obeh eksperimentih Čarovnik iz Oza so podane v tabeli 1. Vrstice s parametri, katerih razlika srednjih vrednosti v obeh eksperimentih je statistično značilna (Studentov primerjalni test; $p < 0.05$), so potemnjene in navedena je pripadajoča p vrednost.

3. Izbira regresijskih parametrov

V obeh eksperimentih Čarovnik iz Oza so uporabniki ocenili svoje zadovoljstvo tako, da so podali stopnjo strinjanja z izjavami o obnašanju oz. učinkovitosti sistema (Hajdinjak in Mihelič, 2006c). Splošno **zadovoljstvo uporabnika** (US) smo dobili kot vsoto ocen, zbranih z vprašalnikom, ki ga predlaga ogrodje PARADISE (Hajdinjak in Mihelič, 2006c). Vrednosti parametra US zato ležijo med 8 in 40. Srednja vrednost US za prvi eksperiment je enaka 34.08 (s standardnim odklonom 5.07), za drugega pa 31.96 (s standardnim odklonom 4.99). Obe srednji vrednosti zadovoljstva uporabnikov se statistično značilno razlikujeta ($p < 0.015$). Glej tabelo 1.

Ker smo želeli poiskati razlike med obema različicama sistemov Čarovnik iz Oza, za odvisno spremenljivko MLR modela učinkovitosti nismo vzeli US, ampak le seštevek ocen, dodeljenih vprašanjem, ki se nanašajo na razlike med sistemoma. Menimo, da so vprašanja, ki te spremembe (tj. vodenje dialoga v povezavi s predstavitvijo znanja) najbolje merijo, naslednja:

2. *Ali vas je sistem razumel?* (ASR)
   Vprašanje naj bi merilo učinkovitost razumevanja govora. Ker pa je v naših eksperimentih čarovnik simuliral tako rekoč popolno razumevanje govora, to ni bilo tako. V drugem eksperimentu, ko čarovnik, v nasprotju s prvim eksperimentom, v pomenske predstavitve uporabnikovih potez ni dodajal podatkov, na katere se je dalo sklepati iz zgodovine dialoga, se to vprašanje nanaša predvsem na modul za vodenje dialoga oz. njegovo učinkovitost pri polnjenju predalčkov.

3. *Ali ste brez težav prišli do odgovorov na vaša vprašanja?* (TE)
   Vprašanje naj bi merilo težavnost pridobivanja informacij. Nedvomno se nanaša na uspešnost čarovnika pri uravnavanju dialoga oz. učinkovitost modula za vodenje dialoga. Pri tem ima pomembno vlogo tudi predstavitev znanja.

6. *Ali se je sistem na vaše izjave odzival hitro (brez pojasnilnih vprašanj)?* (SR)
   Vprašanje naj bi merilo ustreznost sistemovih odzivov. Uporabnike sprašuje po mnenju o strategiji vodenja dialoga, ki je bila v drugem eksperimentu del modula za vodenje dialoga.

7. *Ali se je sistem obnašal tako, kot ste med dialogom od njega pričakovali?* (EB)
   Vprašanje naj bi merilo ujemanje med pričakovanim in dejanskim obnašanjem sistema. Vsekakor je tesno povezano z načinom vodenja dialoga in predstavitvijo znanja, ki je predpogoj sodelujočega načina odgovarjanja.

Vsoto ocen, dodeljenih naštetim vprašanjem, smo imenovali **zadovoljstvo uporabnika z vodenjem dialoga in ravnijo sodelujočega odgovarjanja** (DM). Ta spremenljivka zavzame vrednosti med 4 in 20.

Tiste neodvisne spremenljivke, ki so bile z odvisno spremenljivko DM v zelo nizki korelaciji ($p > 0.05$), smo iz modela odstranili (Hajdinjak in Mihelič, 2006c). Z uporabo Studentovega testa z $n-2$ prostostnimi stopnjami, kjer je n velikost učne množice, tj. $n = 73$ v prvem eksperimentu in $n = 68$ v drugem eksperimentu, smo tako prišli do ugotovitve, da je v prvem eksperimentu z neodvisno spremenljivko DM značilno koreliralo 10 parametrov (in sicer MUM, Comp, NUI, NIR, IRR, NGD, GDR, NRD, NND in NDR), v drugem pa 8 (in sicer κ, MET, MUM, IRR, CMR, GDR, RDR in ARR). Iz teh množic smo odstranili še parametre, ki bi lahko povzročali multikolinearnost modelov.

4. Funkcije učinkovitosti

Po postopku vzvratne eliminacije (Seber, 1977) za delno F statistiko $F_{out} = 4$ pri p-vrednosti približno enaki 0.05 na celotni učni množici, pridobljeni v prvem eksperimentu Čarovnik iz Oza, z DM kot odvisno spremenljivko,

- ⇝ **povprečno število potez** (MUM),
- ⇝ **izpolnitev naloge** (Comp),
- ⇝ **število neprimernih odzivov** (NIR),
- ⇝ **število relevantnih informacij** (NRD) in
- ⇝ **število nepodanih informacij** (NND)

pa kot neodvisnimi spremenljivkami, smo identificirali in odstranili slabih 10% vzorcev osamelcev (Hajdinjak, 2006b). To so meritve, ki se nenavadno razlikujejo od velike večine ostalih meritev in zato nepredvidljivo vplivajo na natančnost modela (Tabachnick in Fidell, 1996).

Postopek vzvratne eliminacije smo ponovili na zmanjšani učni množici vzorcev. Tabela 2 podaja dobljene delne F statistike, pripadajoče koeficiente determinacije R^2 (Johnson in Wichern, 2002) ter parametre, ki jih v posameznih korakih iz modela učinkovitosti prvega sistema Čarovnik iz Oza odstranimo. Postopek vzvratne eliminacije ustavimo pred 4. korakom, ko delna F statistika preseže vrednost 4.

	F_i	R^2	odstranjen parameter
poln model	-	0.59	-
1. korak ($i = 1$)	0.00	0.59	NIR
2. korak ($i = 2$)	0.21	0.59	MUM
3. korak ($i = 3$)	3.32	0.57	NRD
4. korak ($i = 4$)	9.01	0.51	Comp

Tabela 2: Tabela vzvratne eliminacije za prvi sistem Čarovnik iz Oza in odvisno spremenljivko DM.

Iz začetnega MLR modela z vzvratno eliminacijo odstranimo tri parametre, in sicer NIR, MUM in NRD. Funkcija učinkovitosti za prvi sistem Čarovnik iz Oza in odvisno spremenljivko DM_1, ki se nanaša na podatke, pridobljene v prvem eksperimentu Čarovnik iz Oza, je zato taka:

$$\widehat{\mathcal{N}(\mathrm{DM}_1)} = 0.25 * \mathcal{N}(\mathrm{Comp}) - 0.65 * \mathcal{N}(\mathrm{NND}).$$

Dobljena funkcija učinkovitosti pojasnjuje 57% variance, tj. $R^2 = 0.57$. Najizrazitejši parameter, ki negativno vpliva na DM_1, je parameter podatkovne zbirke NND.

Po postopku vzvratne eliminacije za $F_{out} = 4$ pri p-vrednosti približno enaki 0.05 na celotni učni množici, pridobljeni v drugem eksperimentu Čarovnik iz Oza, z DM kot odvisno spremenljivko,

- ⇝ **Kappa koeficient** (κ),
- ⇝ **povprečni čas dialoga** (MET),
- ⇝ **delež preverjanj** (CMR),

$\rightsquigarrow$ **delež podanih informacij** (GDR) in

$\rightsquigarrow$ **delež prekinjenih zahtev** (ARR)

pa kot neodvisnimi spremenljivkami, smo identificirali in odstranili dobrih 7% vzorcev osamelcev.

Postopek vzvratne eliminacije smo ponovili na zmanjšani učni množici vzorcev. Tabela 3 podaja dobljene delne F statistike, pripadajoče koeficiente determinacije R^2 ter parametre, ki jih v posameznih korakih iz modela učinkovitosti drugega sistema Čarovnik iz Oza odstranimo. Postopek vzvratne eliminacije ustavimo pred $3.$ korakom, ko delna F statistika preseže vrednost 4.

	F_i	R^2	odstranjen parameter
poln model	-	0.48	-
1. korak ($i = 1$)	1.71	0.46	MET
2. korak ($i = 2$)	2.84	0.44	ARR
3. korak ($i = 3$)	12.59	0.32	κ

Tabela 3: Tabela vzvratne eliminacije za drugi sistem Čarovnik iz Oza in odvisno spremenljivko DM.

Iz začetnega MLR modela z vzvratno eliminacijo odstranimo dva parametra, in sicer MET in ARR. Funkcija učinkovitosti za drugi sistem Čarovnik iz Oza in odvisno spremenljivko DM_2, ki se nanaša na podatke, pridobljene v drugem eksperimentu Čarovnik iz Oza, je zato taka:

$$\widehat{\mathcal{N}(\text{DM}_2)} = 0.36{*}\mathcal{N}(\kappa) - 0.38{*}\mathcal{N}(\text{CMR}) + 0.40{*}\mathcal{N}(\text{GDR}).$$

Dobljena funkcija učinkovitosti pojasnjuje 44% variance, tj. $R^2 = 0.44$, in ima tri parametre – **Kappa koeficient** (κ) in **delež podanih informacij** (GDR) pozitivno vplivata na DM_2, **delež preverjanj** (CMR) pa negativno vpliva na DM_2.

Nobeden od parametrov, ki jih vsebuje $\widehat{\mathcal{N}(\text{DM}_1)}$, ni značilen za $\widehat{\mathcal{N}(\text{DM}_2)}$. Obratno pa ni res. Parameter podatkovne zbirke GDR, ki ima zelo velik pozitivni vpliv na DM_2, sicer ni vsebovan v $\widehat{\mathcal{N}(\text{DM}_1)}$, je pa visoko (negativno) koreliran s parametrom podatkovne zbirke NND, tj. najmočnejšim (negativnim) parametrom funkcije $\widehat{\mathcal{N}(\text{DM}_1)}$ (Hajdinjak, 2006b).

Analiza obeh funkcij učinkovitosti za DM omogoča vrednotenje učinkovitosti modula za vodenje dialoga, povezanega s sodelujočo podatkovno zbirko:

- Edini parameter, ki nastopa v funkciji učinkovitosti za DM_2 in je statistično značilen tudi za DM_1 ($p < 0.004$), je parameter podatkovne zbirke **delež podanih informacij** (GDR). V funkciji učinkovitosti za DM_1 namesto GDR sicer nastopa parameter podatkovne zbirke **število nepodanih informacij** (NND), ki je z njim visoko negativno koreliran in hkrati bolj značilen za DM_1 ($p < 0.0005$). Torej, parametri podatkovne zbirke predstavljajo edino podobnost med funkcijama učinkovitosti obeh sistemov Čarovnik iz Oza. Ta ugotovitev kaže na izjemno pomembnost predstavitve znanja oz. zgradbe podatkovne zbirke sistema za dialog. Pridemo do spoznanja, da so parametri podatkovne zbirke nepogrešljivi pri vrednotenju učinkovitosti sistemov za dialog, še posebej pa pri vrednotenju učinkovitosti sistemov za podajanje informacij.

- Medtem ko je parameter podatkovne zbirke **število nepodanih informacij** (NND) v prvem eksperimentu pomembno (negativno) vplival na zadovoljstvo uporabnikov, je njegov (negativni) vpliv v drugem eksperimentu izjemno splahnel. Vemo že (tabela 1), da se je srednja vrednost parametra **število relevantnih informacij** (NRD) v drugem eksperimentu značilno povečala, srednja vrednost NND pa zato značilno zmanjšala. Vse torej kaže na to, da zmanjšanje števila odzivov, s katerimi sistem uporabniku sporoča, da zahtevane informacije nima, hkrati pa mu ne ponudi nobenih dosegljivih, relevantnih informacij, negativno vpliva na zadovoljstvo uporabnika. Razvijalci sistemov za dialog morajo zato težiti k zmanjšanju števila takih odzivov oz. povečanju stopnje sodelujočega odgovarjanja. Sklepamo lahko tudi, da strategija usmerjanja uporabnika k izbiri dosegljivih, relevantnih podatkov, ki je implementirana v modulu za samodejno vodenje dialoga, na zadovoljstvo uporabnikov ne vpliva negativno.

- Ugotovili smo, da so bili uporabniki v prvem eksperimentu bolj dojemljivi za kvantitativne parametre (tj. NUR, NIR, NHM, NGD, NRD, NND, NAR), uporabniki v drugem eksperimentu pa za njim pripadajoče proporcionalne parametre (tj. URR, IRR, HMR, GDR, RDR, NDR, ARR). Funkcija učinovitosti za DM_1 vsebuje, poleg parametra Comp, še kvantitativni parameter **število nepodanih informacij** (NND). Funkcija učinkovitosti za DM_2 pa vsebuje, poleg parametra κ, še dva proporcionalna parametra, namreč **delež preverjanj** (CMR) in **delež podanih informacij** (GDR). Menimo, da je to posledica konsistentno povečanega ponujanja relevantnih informacij v drugem eksperimentu, ki je vodilo do več novih informacijskih iger in s tem do večje dojemljivosti uporabnikov za proporcionalne količine. Vsekakor so glede tega potrebne nadaljnje raziskave.

- Parametra **Kappa koeficient** (κ) in **izpolnitev naloge** (Comp) sta bila v naših eksperimentih nekorelirana. V prvem eksperimentu je na zadovoljstvo uporabnikov DM_1 močno (pozitivno) vplival Comp, κ ni imel statistično značilnega vpliva. V drugem eksperimentu je bilo ravno obratno – na zadovoljstvo uporabnikov DM_2 je močno (pozitivno) vplival κ, Comp pa ni imel statistično značilnega vpliva. Ugotovitev, do katere so prišle Walker, Litman, Kamm in Abella (Walker et al., 1998), da **izpolnitev naloge** (Comp) močneje vpliva na zadovoljstvo uporabnika kot **Kappa koeficient** (κ), torej ni vedno resnična. Le parameter Comp, katerega vrednost mora posredovati uporabnik, za vrednotenje učinkovitosti sistemov za dialog zato ni dovolj. Še vedno je dobro meriti tudi κ, ki pa ga na žalost prav tako ni mogoče določiti samodejno.

- Parameter, ki na zadovoljstvo uporabnikov DM_2

najmočneje negativno vpliva, je **delež preverjanj** (CMR). Sistem za dialog lahko torej izboljšamo, če zmanjšamo delež potez, ki preverjajo točnost podatkov, pridobljenih na osnovi zgodovine dialoga, ki jih uporabnik v svoji izjavi ne poda ali jih sistem ne razume. Vpliv parametra CMR v sistemih za dialog ni mogoče popolnoma odpraviti, zato ker je določeno število preverjanj nujno vsakič, ko imamo opravka s samodejnim razumevanjem govora. Napake, ki se pojavljajo pri samodejnem razumevanju govora, sistem namreč prisilijo, da svoje razumevanje uporabnikovih izjav preveri vsakič, ko o njihovi pravilnosti ni popolnoma prepričan. Če tega ne bi počel, bi nekontrolirano podajal napačne odgovore. To bi povečalo srednjo vrednost parametra **delež neprimernih odzivov** (IRR) in tako zelo verjetno vodilo do večjega nezadovoljstva s sistemom.

Funkciji učinkovitosti obeh sistemov Čarovnik iz Oza z **zadovoljstvom uporabnika** (US) kot odvisno spremenljivko sta se zelo razlikovali v natančnosti ($R^2 = 0.58$ proti $R^2 = 0.24$) (Hajdinjak, 2006b). Potem ko smo za odvisno spremenljivko vzeli **zadovoljstvo uporabnika z vodenjem dialoga in ravnijo sodelujočega odgovarjanja** (DM), nam je uspelo razliko v natančnosti izjemno zmanjšati ($R^2 = 0.57$ proti $R^2 = 0.44$). Upravičeno lahko torej trdimo, da se da DM veliko bolje modelirati kot US.

Povejmo še, da literatura o vrednotenju učinkovitosti sistemov za dialog z ogrodjem PARADISE v glavnem poroča o koeficientih determinacije R^2, ki so blizu mejne vrednosti 0.5, pogosto precej nižje (Walker et al., 1997b; Walker et al., 1998; Walker et al., 2001; Möller, 2005), le redko pa presežejo vrednost 0.6 (Litman in Shimei, 2002).

5. Sklep

Ogrodje PARADISE smo uporabili pri vrednotenju učinkovitosti dveh nedograjenih sistemov za podajanje informacij o vremenu in vremenski napovedi, s katerima smo izvajali eksperiment Čarovnik iz Oza. Za namene vrednotenja smo izbrali in določili 25 regresijskih parametrov. Pri vrednotenju učinkovitosti sistemov za podajanje informacij smo predlagali še neuveljavljene parametre podatkovne zbirke, ki izražajo velikost in sestavo podatkovne zbirke. V raziskave smo vključili kvantitativne in proporcionalne parametre podatkovne zbirke. Ugotovili smo, da so bili uporabniki prvega sistema bolj dojemljivi za kvantitativne parametre, v drugem pa za proporcionalne parametre.

Ker smo želeli poiskati razlike med dvema sistemoma Čarovnik iz Oza, ki sta se razlikovala le v načinu vodenja dialoga in predstavitvi znanja, smo mero zadovoljstva uporabnikov definirali kot vsoto ocen, ki se nanašajo na vpeljane spremembe. Po vzvratni eliminaciji smo dobili funkciji učinkovitosti, ki ne vsebujeta nobenega skupnega parametra. Edini parameter, ki nastopa v funkciji učinkovitosti drugega sistema in je bil statistično značilen tudi v prvem eksperimentu, je eden od parametrov podatkovne zbirke. Prišli smo do spoznanja, da so parametri podatkovne zbirke edina podobnost med funkcijama učinkovitosti obeh sistemov Čarovnik iz Oza in da ima predstavitev znanja v sistemih za podajanje informacij velik pomen.

6. Literatura

M. Hajdinjak in F. Mihelič. 2004. Conducting the wizard-of-oz experiment. *Informatica*, 28(4):425–430.

M. Hajdinjak in F. Mihelič. 2006a. The paradise evaluation framework: Issues and findings. *Computational Linguistics*, 32.

M. Hajdinjak. 2006b. *Predstavitev znanja in vrednotenje učinkovitosti sodelujočih samodejnih sistemov za dialog, Doktorska disertacija*. Fakulteta za elektrotehniko, Univerza v Ljubljani, Ljubljana.

M. Hajdinjak in F. Mihelič. 2006c. Vrednotenje govornih vmesnikov z ogrodjem paradise. V: *Zbornik IS-LTC 2006 9. mednarodne multikonference Informacijska družba IS'2006*. Ljubljana, Slovenija.

R. A. Johnson in D. W. Wichern. 2002. *Applied multivariate statistical analysis*. Prentice-Hall, Upper Saddle River (NJ).

D. J. Litman in P. Shimei. 2002. Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12(2-3):111–137.

S. Möller. 2005. Evaluating telephone-based interactive systems. V: *Proceedings of the COST278 Final Workshop and ISCA Tutorial and Research Workshop (ITRW) on Applied Spoken Language Interaction in Distributed Environments*. Aalborg, Danska.

G. A. F. Seber. 1977. *Linear Regression Analysis*. John Wiley & Sons, New York.

B. G. Tabachnick in L. S. Fidell. 1996. *Using Multivariate Statistics, Third Edition*. Harper Collins, New York.

J. Žibert, S. Martinčić-Ipšić, M. Hajdinjak, I. Ipšić, in F. Mihelič. 2004. Development of a bilingual spoken dialog system for weather information retrieval. V: *Proceedings of the 8th European Conference on Speech Communication and Technology*, str. 1917–1920. Ženeva, Švica.

M. A. Walker, D. Litman, C. A. Kamm, in A. Abella. 1997a. Paradise: A framework for evaluating spoken dialogue agents. V: *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics*, str. 271–280. Madrid, Španija.

M. A. Walker, D. Hindle, J. Fromer, G. Di Fabbrizio, in C. Mestel. 1997b. Evaluating competing agent strategies for a voice email agent. V: *Proceedings of the 5th European Conference on Speech Communication and Technology*, str. 2219–2222. Rodos, Grčija.

M. A. Walker, D. J. Litman, C. A. Kamm, in A. Abella. 1998. Evaluating spoken dialogue agents with paradise: Two case studies. *Computer Speech and Language*, 12(3):317–347.

M. A. Walker, R. Passonneau, in J. E. Boland. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. V: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, str. 515–522. Toulouse, Francija.

Vrednotenje govornih vmesnikov z ogrodjem PARADISE

Melita Hajdinjak, France Mihelič

Laboratorij za umetno zaznavanje, sisteme in kibernetiko,
Fakulteta za elektrotehniko, Univerza v Ljubljani
Tržaška 25, 1000 Ljubljana
{melita.hajdinjak, france.mihelic}@fe.uni-lj.si

Povzetek

Opišemo potencialno splošno metodologijo vrednotenja učinkovitosti sistemov za dialog, imenovano ogrodje PARADISE (PARAdigm for DIalogue System Evaluation). Ogrodje PARADISE omogoča izpeljavo ocene učinkovitosti sistema kot uteženo linearno kombinacijo od domene odvisnih parametrov uspešnosti naloge in cen dialoga, zajema pa model učinkovitosti sistema, katerega osnovni cilj je maksimirati zadovoljstvo uporabnikov. Model učinkovitosti sistema, ki ga zajema ogrodje PARADISE, trdi, da lahko funkcijo učinkovitosti sistema določimo z multiplo linearno regresijo. Osredotočimo se na nekatere pomanjkljivosti, težave in nerešena vprašanja ogrodja PARADISE – opišemo vpliv normalizacije na natančnost napovedovanja zadovoljstva uporabnikov, navedemo regresijske predpostavke in poudarimo pomembnost dobre izbire regresijskih parametrov. Nakažemo tudi morebitne učinke razpoznavanja govora na rezultate vrednotenja in analiziramo vprašalnik, na podlagi katerega ogrodje PARADISE določa zadovoljstvo uporabnikov (tj. odvisno spremenljivko funkcije učinkovitosti). V dosedanjih raziskavah so bile nekatere izmed naštetih točk premalo poudarjene, zanemarjene ali celo napačno interpretirane.

Speech-interface evaluation using the PARADISE framework

We introduce the PARADISE (PARAdigm for DIalogue System Evaluation) framework, a potential general methodology for evaluating spoken-language dialogue systems. The PARADISE framework maintains that the system's primary objective is to maximize user satisfaction, and it derives a combined performance metric for a dialogue system as a weighted linear combination of task-success measures and dialogue costs. The PARADISE model of performance posits that a performance function can be derived by applying multivariate linear regression with user satisfaction as the dependent variable and task-success measures and dialogue costs as the independent variables. We focus on some PARADISE issues (with most of them arising from the application of multivariate linear regression) that have, up to now, not been sufficiently emphasized or have even been neglected by the dialogue-system community. These include considerations regarding the selection of appropriate regression parameters, normalization effects on the accuracy of the prediction, the influence of speech-recognition errors on the performance function, and the selection of an appropriate user-satisfaction measure.

1. Uvod

Avtomatizacija sporazumevanja z govorom je še vedno eden izmed največjih raziskovalnih izzivov. Razlogov je več:

- Govor je naraven – govoriti se naučimo, še preden znamo brati in pisati.
- Govor je učinkovit – večina ljudi je sposobnih govoriti petkrat hitreje kot tipkati in verjetno celo desetkrat hitreje kot pisati.
- Govor je fleksibilen – med sporazumevanjem z govorom se nam ni treba ničesar dotikati in ne opazovati.

Računalniški sistem, ki uporabniku omogoča, da z govorom dostopa do določenih aplikacij, imenujemo *sistem za dialog* ali *govorni vmesnik*.

Z razvojem sistemov za dialog se pojavljajo tudi potrebe po vrednotenju učinkovitosti in medsebojni primerjavi takih sistemov. Težava, ki se pojavi, je ta, da vrednotenja učinkovitosti sistema za dialog ni mogoče omejiti na primerjave z referenčnimi odgovori oz. referenčnimi poteki dialogov (Bates in Ayuso, 1991; Polifroni et al., 1992; Price et al., 1992). Množica sprejemljivih dialogov je namreč lahko zelo velika. Naslednja težava je veliko število potencialnih metrik dialoga. Sistem za dialog lahko npr. vrednotimo glede na njegovo sposobnost pomagati uporabnikom pri doseganju ciljev, glede na njegovo robustnost odkrivanja in premagovanja napak, ki se pojavljajo pri razpoznavanju oz. razumevanju govora, ali glede na skupno kakovost interakcije (Polifroni et al., 1992; Price et al., 1992; Danieli in Gerbino, 1995), ki jo omogoča.

Predlogi vrednotenja učinkovitosti sistemov za dialog, ki so se pojavili v zadnjih dveh desetletjih dvajsetega stoletja, se osredotočajo na razvoj različnih metrik dialoga. Predlagani so bili številni *objektivni parametri dialoga* (Price et al., 1992; Danieli in Gerbino, 1995) kot npr. število izjav, čas dialoga, povprečni čas odziva uporabnika, povprečni čas odziva sistema, delež izjav, sestavljenih iz več kot ene besede, ter povprečna dolžina izjav, sestavljenih iz več kot ene besede, ki jih lahko določimo brez mnenja človeka, in parametri, ki temeljijo na mnenju človeka, namreč *subjektivni parametri dialoga* (Shriberg et al., 1992; Danieli in Gerbino, 1995) kot npr. delež izjav, s katerimi sistem popravlja napake, delež kontekstno primernih izjav sistema, hevristično vrednotenje stopnje sodelovanja sistema na podlagi Griceovih maksim (Grice, 1975), delež pravilnih in delno pravilnih odgovorov, delež primernih in neprimernih izjav, s katerimi sistem usmerja uporabnika, ter zadovoljstvo uporabnika (Shriberg et al., 1992).

Z namenom omogočiti primerjavo sistemov z različnimi domenami, kjer je pomembno vedeti, v kolikšni meri posamezni parametri vplivajo na učinkovitost in kako strategija vodenja dialoga vpliva na zadovoljstvo uporabnikov, je bilo leta 1997 kot potencialna splošna metodologija vrednotenja učinkovitosti sistemov za dialog predlagano *ogrodje PARADISE* (PARAdigm for DIalogue System Evaluation) (Walker et al., 1997a).

Ko bomo govorili o strukturi dialoga, bomo uporabljali pojma *konverzacijskih iger* in *konverzacijskih potez*. *Konverzacijske igre* povezujemo z željami oz. konverzacijskimi cilji, kot je na primer cilj pridobiti določeno informacijo, in so sestavljene iz zaporedja izjav, ki se začnejo s pobudo in končajo, ko je cilj igre dosežen ali igra prekinjena. Sestavne dele konverzacijskih iger imenujemo *konverzacijske poteze*. To so izjave, deli izjav ali množice izjav, ki izražajo isto namero, kot je na primer potrditev ali preverjanje.

2. Ogrodje PARADISE

Ogrodje PARADISE omogoča izpeljavo ocene učinkovitosti sistema kot uteženo linearno kombinacijo *parametrov uspešnosti naloge* in *cen dialoga*, zajema pa model učinkovitosti sistema, ki za osnovni cilj postavlja maksimirati zadovoljstvo uporabnikov, kar doseže z maksimiranjem parametrov uspešnosti naloge in minimiranjem cen dialoga.

Zadovoljstvo uporabnikov ponavadi merimo z vprašalniki, v katerih uporabniki podajo stopnjo strinjanja z izjavami o različnih vidikih svoje interakcije s sistemom za dialog. Avtorice ogrodja PARADISE (Walker et al., 1997a) v ta namen uporabljajo vprašalnik, podan v tabeli 1. Vprašanja (v naštetem vrstnem redu) sprašujejo po učinku modula za tvorjenje govora, učinku modula za razpoznavanje govora, težavnosti pridobivanja informacij, hitrosti interakcije, izkušenosti uporabnikov, ustreznosti odzivov sistema, pričakovanem obnašanju sistema in načrtovani rabi sistema v prihodnosti. Večino odgovorov podajo opisno kot *skoraj nikoli*, *redko*, *včasih*, *pogosto* in *skoraj vedno*, nekatere pa le z *da*, *ne* in *mogoče*. Te potem preslikajo v množico naravnih števil od 1 do 5, pri čemer 1 pomeni najmanjšo, 5 pa največjo stopnjo strinjanja. Parameter, ki ocenjuje zadovoljstvo uporabnikov, dobijo kot vsoto vseh ocen in ga poimenujejo **zadovoljstvo uporabnika** (US).

Cene dialoga, tj. parametre dialoga, katerih minimizacija ugodno vpliva na zadovoljstvo uporabnikov, razdelimo v dve kategoriji: *parametri učinkovitosti* dialoga in *parametri kakovosti* dialoga. Parametri učinkovitosti dialoga (npr. število izjav, ki jih uporabnik potrebuje, da uresniči svojo namero, ali čas dialoga) merijo, kako učinkovito sistem uporabniku pomaga pri doseganju njegove namere. Parametri kakovosti dialoga (npr. kolikokrat mora uporabnik ponoviti svojo izjavo, da ga sistem razume, ali kakšen je čas čakanja na odziv sistema) pa zajemajo ostale vidike, ki lahko na zadovoljstvo uporabnika prav tako močno vplivajo. Ker vnaprej ni jasno, katere cene dialoga bodo najmočneje vplivale na zadovoljstvo uporabnikov, je pomembno, da v empiričnih raziskavah uporabljamo širok spekter teh parametrov (Walker et al., 1998).

Tabela 1: Vprašalnik za ocenjevanje zadovoljstva uporabnikov, ki ga predlaga ogrodje PARADISE.

1.	Ali ste sistem brez težav razumeli?
2.	Ali vas je sistem razumel?
3.	Ali ste brez težav prišli do odgovorov na vaša vprašanja?
4.	Ali je bila hitrost interakcije s sistemom primerna?
5.	Ali ste na vsakem koraku dialoga vedeli, kaj morate povedati?
6.	Ali se je sistem na vaše izjave odzival hitro (brez pojasnilnih vprašanj)?
7.	Ali se je sistem obnašal tako, kot ste med dialogom od njega pričakovali?
8.	Glede na vašo trenutno izkušnjo s sistemom, ali mislite, da boste sistem še kdaj poklicali?

Uspešnost naloge, ki se lahko nanaša na celoten dialog ali del dialoga, ki predstavlja zaključeno celoto, pomeni stopnjo ujemanja med vsebino zahtev uporabnika in dojemanjem te s strani sistema za dialog. Ogrodje PARADISE uporablja en sam parameter uspešnosti naloge, namreč **Kappa koeficient** (Carletta, 1996). **Kappa koeficient** (κ) izračunamo z uporabo Cohenove metode (Di Eugenio in Glass, 2004) in kontingenčne tabele, ki podaja ujemanje med vsebino zahtev uporabnika in dojemanjem te s strani sistema.

2.1. Model učinkovitosti

Če želimo sistem za dialog vrednotiti z ogrodjem PARADISE, moramo podatke zbrati v eksperimentu, v katerem bodo uporabniki ocenili svoje zadovoljstvo. Ostale parametre modela (parametri uspešnosti naloge, cene dialoga) pa je treba določiti samodejno ali jih ročno označiti.

Model učinkovitosti sistema, ki ga zajema ogrodje PARADISE, trdi, da lahko funkcijo učinkovitosti sistema določimo z uporabo *multiple linearne regresije* (MLR) z zadovoljstvom uporabnikov kot neodvisno spremenljivko ter parametri uspešnosti naloge, parametri učinkovitosti dialoga in parametri kakovosti dialoga kot neodvisnimi spremenljivkami:

$$\text{Učinkovitost} = \alpha\mathcal{N}(\kappa) - \sum_{i=1}^{n} w_i\mathcal{N}(c_i)$$

Pri tem je α utež edinega parametra uspešnosti naloge, namreč Kappa koeficienta κ, w_i so uteži cen dialoga c_i,

$\mathcal{N}$ pa je funkcija normalizacije:

$$\mathcal{N}(x) = \frac{x - \overline{x_0}}{\sigma_{x_0}}$$

Z $\overline{x_0}$ in σ_{x_0} smo označili srednjo vrednost in standardni odklon spremenljivke x_0 v učni množici, pridobljeni v ustreznem eksperimentu. Srednja vrednost s funkcijo normalizacije $\mathcal{N}$ preslikanih parametrov učne množice je 0, varianca in standardni odklon pa 1. Tako se znebimo težav, ki se pojavijo, če primerjamo vrednosti parametrov, ki se raztezajo na različnih intervalih in/ali so njihove vrednosti različno razpršene. Z normalizacijo parametrov κ in c_i dosežemo relevantnost in primerljivost uteži preslikanih parametrov $\mathcal{N}(\kappa)$ in $\mathcal{N}(c_1), \ldots, \mathcal{N}(c_n)$.

Rezultat multiple linearne regresije na učni množici parametrov, ki praviloma tvorijo predoločen sistem, je torej množica uteži, ki pomenijo sorazmeren prispevek teh parametrov k učinkovitosti sistema. Funkcija učinkovitosti, ki jo uvaja ogrodje PARADISE, zato omogoča:

- napovedovanje zadovoljstva uporabnikov,
- vrednotenje učinkovitosti sistema za dialog, tj. ugotavljanje vpliva posameznih parametrov na zadovoljstvo uporabnikov,
- izboljšanje sistema za dialog, tj. odpravljanje ali zmanjšanje vpliva parametrov, ki imajo najbolj negativne uteži in povečanje vpliva parametrov, ki imajo najbolj pozitivne uteži,
- primerjavo različnih sistemov za dialog, tj. primerjavo vplivov posameznih parametrov v pripadajočih funkcijah učinkovitosti, iz katerih lahko razberemo razlike med sistemi,
- samodejno iskanje problematičnih dialogov, tj. iskanje dialogov, katerih napovedano zadovoljstvo uporabnikov negativno izstopa, ter
- spreminjanje strategije vodenja dialoga med samo interakcijo, tj. spreminjanje načina sporazumevanja na osnovi napovedanega zadovoljstva uporabnika v že izvedenem delu interakcije.

V zadnjih letih je bilo opravljenih veliko študij učinkovitosti sistemov za dialog, ki so uporabljale ogrodje PARADISE (Walker et al., 1998; Kamm et al., 1998; Walker, 2000; Litman in Shimei, 2002; Larsen, 2003; Hajdinjak, 2006b). Ogrodje PARADISE je postala celo najbolj citirana metoda vrednotenja učinkovitosti sistemov za dialog.

3. Analiza ogrodja PARADISE

Osredotočili se bomo na nekatere pomanjkljivosti, težave in nerešena vprašanja ogrodja PARADISE (Hajdinjak in Mihelič, 2006a). Večina jih izvira ravno iz uporabe multiple linearne regresije.

3.1. Vpliv normalizacije na natančnost napovedovanja zadovoljstva uporabnikov

Multipla linearna regresija temelji na metodi najmanjših kvadratov, tj. minimira vsoto kvadratov razlik med v eksperimentu pridobljenimi vrednostmi (tj. učne množice) in napovedanimi vrednostmi zadovoljstva uporabnikov. Za dano vrednost zadovoljstva uporabnika US torej velja

$$\mathcal{N}(US) = \widehat{\mathcal{N}(US)} + \epsilon,$$

kjer je $\mathcal{N}(US)$ normalizirana pridobljena vrednost zadovoljstva uporabnika, $\widehat{\mathcal{N}(US)}$ napovedana normalizirana vrednost zadovoljstva uporabnika, ϵ pa napaka napovedi. Ker je srednja vrednost napake ϵ enaka 0, sta srednji vrednosti odvisne spremenljivke in njene napovedi enaki. Nenormalizirano zadovoljstvo uporabnika US lahko tedaj ocenimo kot

$$US = \widehat{\mathcal{N}(US)}\sigma_{US_0} + \overline{US}_0 + \epsilon\sigma_{US_0} = \widehat{US} + \epsilon\sigma_{US_0},$$

kjer sta $\overline{US}_0$ in σ_{US_0} srednja vrednost in standardni odklon v eksperimentu pridobljenih vrednosti zadovoljstva uporabnikov. Vidimo, da se napaka ocene normaliziranega zadovoljstva uporabnika $\mathcal{N}(US)$ pri tem poveča za faktor σ_{US_0}.

Kako dobro $\widehat{US}$ napoveduje US, kaže razmerje absolutnih vrednosti njune razlike in pridobljene vrednosti zadovoljstva uporabnika US:

$$q(US, \widehat{US}) = \frac{|US - \widehat{US}|}{|US|}$$

Naslednje razmerje pa kaže, da ocena normalizirane vrednosti zadovoljstva uporabnika $\widehat{\mathcal{N}(US)}$ ni vedno tako dobra kot ocena nenormalizirane vrednosti $\widehat{US}$:

$$\frac{q(\mathcal{N}(US), \widehat{\mathcal{N}(US)})}{q(US, \widehat{US})} = \frac{\frac{|\mathcal{N}(US) - \widehat{\mathcal{N}(US)}|}{|\mathcal{N}(US)|}}{\frac{|US - \widehat{US}|}{|US|}} = \frac{|US|}{|US - \overline{US}_0|}$$

Za $US > \frac{\overline{US}_0}{2}$ namreč velja:

$$\frac{q(\mathcal{N}(US), \widehat{\mathcal{N}(US)})}{q(US, \widehat{US})} = \frac{|US|}{|US - \overline{US}_0|} > 1$$

Iz prikazanega sledi, da je napovedano normalizirano vrednost zadovoljstva uporabnika $\widehat{\mathcal{N}(US)}$ treba transformirati nazaj na začetni interval, saj je ocena nenormalizirane vrednosti zadovoljstva uporabnika $\widehat{US}$ v večini primerov veliko boljša. To naredimo s transformacijo

$$\widehat{US} = \widehat{\mathcal{N}(US)}\sigma_{US_0} + \overline{US}_0,$$

ki je inverzna normalizaciji.

Ne samo da ustrezna literatura (Walker et al., 1997a; Walker et al., 1998; Walker, 2000; Litman in Shimei, 2002; Larsen, 2003) vplivu normalizacije ne posveča pozornosti, ampak tudi ne omenja, da je treba vrednosti zadovoljstva uporabnikov, preden začnemo izpeljavo modela učinkovitosti, normalizirati, če želimo preprečiti prevelike napake ocen (Hajdinjak, 2006b).

Obstaja več načinov merjenja natančnosti MLR modelov. Najpogosteje se uporablja *koeficient (multiple) determinacije*,

$$R^2 = \frac{\sum_{i=1}^{m}(\widehat{\mathbf{X}}_i - \overline{\mathbf{X}})^2}{\sum_{i=1}^{m}(\mathbf{X}_i - \overline{\mathbf{X}})^2},$$

tj. razmerje pojasnjene variance in celotne variance $var(\mathbf{X})$, pri čemer smo z m označili število enačb učne množice. Celotna varianca je vsota pojasnjene variance in nepojasnjene variance:

$$var(\mathbf{X}) = \frac{1}{m}\sum_{i=1}^{m}(\widehat{\mathbf{X}}_i - \overline{\mathbf{X}})^2 + \frac{1}{m}\sum_{i=1}^{m}(\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2$$

Nepojasnjena varianca oz. srednja kvadratna napaka,

$$\overline{\epsilon^2} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{X}_i - \widehat{\mathbf{X}}_i)^2,$$

je ravno količina, ki jo multipla linearna regresija minimira. Koeficient determinacije zavzame vrednosti med 0 in 1. Vrednosti, ki so bližje 1, pomenijo večjo natančnost modela, tj. boljšo linearno zvezo med odvisno spremenljivko in neodvisnimi spremenljivkami. Če koeficient determinacije R^2 pomnožimo s faktorjem 100, rezultat imenujemo *odstotek pojasnjene variance*.

Izkaže se, da je v MLR modelu z normaliziranimi spremenljivkami koeficient determinacije R^2 enak varianci napovedanih vrednosti:

$$R^2 = \frac{\sum_{i=1}^{m}\widehat{\mathcal{N}(\mathbf{US}_i)}^2}{m} = var(\widehat{\mathcal{N}(\mathbf{US})})$$

Pri tem smo z $\mathbf{US}_i$ označili i-to komponento vektorja $\mathbf{US}$ pridobljenih vrednosti zadovoljstva uporabnikov, z $\widehat{\mathcal{N}(\mathbf{US}_i)}$ pa i-to komponento vektorja $\widehat{\mathcal{N}(\mathbf{US})}$ napovedanih normaliziranih vrednosti zadovoljstva uporabnikov. Zadnja enakost velja zato, ker je $\overline{\widehat{\mathcal{N}(\mathbf{US})}}$, torej srednja vrednost napovedanih normaliziranih vrednosti zadovoljstva uporabnikov, enaka $\overline{\mathcal{N}(\mathbf{US})} = 0$. Zanimiva posledica te ugotovitve je, da so uteži funkcije učinkovitosti po absolutni vrednosti navzgor omejene z 1. Za MLR model

$$\widehat{\mathbf{X}} = \sum_{i=1}^{n}\alpha_i\mathbf{X_i}$$

namreč velja naslednje:

$$var(\widehat{\mathbf{X}}) = \sum_{i=1}^{n}\alpha_i^2 + 2\sum_{i=1}^{n}\sum_{j=i+1}^{n}\alpha_i\alpha_j corr(\mathbf{X_i}, \mathbf{X_j}),$$

pri čemer je

$$corr(\mathbf{X_i}, \mathbf{X_j}) = \frac{\frac{1}{m}\sum_{k=1}^{m}(\mathbf{X_i}_k - \overline{\mathbf{X_i}})(\mathbf{X_j}_k - \overline{\mathbf{X_j}})}{\sigma_{\mathbf{X_i}}\sigma_{\mathbf{X_j}}}$$

korelacija oz. *korelacijski koeficient* spremenljivk $\mathbf{X_i}$ in $\mathbf{X_j}$. Ker ima dvojna vsota v $var(\widehat{\mathbf{X}})$ same nenegativne člene, sledi

$$1 \geq R^2 = var(\widehat{\mathcal{N}(\mathbf{US})}) \geq \alpha^2 + \sum_{i=1}^{n} w_i^2$$

in zato napovedan rezultat za uteži funkcije učinkovitosti:

$$|\alpha| \leq 1$$

$$|w_i| \leq 1 \text{ za } i = 1, \ldots, n$$

Velja, da je koren koeficienta determinacije R enak korelaciji spremenljivke $\mathbf{X}$ z njeno oceno $\widehat{\mathbf{X}}$ (Seber, 1977):

$$R = corr(\mathbf{X}, \widehat{\mathbf{X}})$$

Če upoštevamo običajno pojmovanje visoke koreliranosti, tj. korelacijski koeficient, ki je po absolutni vrednosti večji od 0.7, to pomeni, da lahko šele pri $R^2 \geq 0.5$ govorimo o zadovoljivi natančnosti MLR modela.

3.2. Regresijske predpostavke

Uporaba multiple linearne regresije pri reševanju predoločenega linearnega sistema zahteva izpolnitev naslednjih pogojev (Johnson in Wichern, 2002):

1. LINEARNOST SPREMENLJIVK: Obstajati mora približno linearna zveza med odvisno spremenljivko $\mathbf{X}$ na eni strani in neodvisnimi spremenljivkami $\mathbf{X_1}, \ldots, \mathbf{X_n}$ na drugi strani, tj. pričakovana vrednost oz. matematično upanje odvisne spremenljivke mora biti linearna funkcija neodvisnih spremenljivk. Indikator linearnosti med odvisno spremenljivko in neodvisnimi spremenljivkami modela je velik koeficient determinacije R^2. Literatura o vrednotenju učinkovitosti sistemov za dialog z ogrodjem PARADISE v glavnem poroča o koeficientih determinacije R^2, ki so blizu mejne vrednosti 0.5 (Walker, 2000; Larsen, 2003), pogosto precej nižje (Walker et al., 1997b; Walker et al., 1998; Walker et al., 1999), le redko pa presežejo vrednost 0.6 (Litman in Shimei, 2002).

2. NEODVISNOST SPREMENLJIVK: Noben par neodvisnih spremenljivk $\mathbf{X_1}, \ldots, \mathbf{X_n}$ ne sme biti preveč koreliran, tj. korelacijski koeficienti $corr(\mathbf{X_i}, \mathbf{X_j})$ morajo biti po absolutni vrednosti manjši od 0.7. Če to ni tako, je dobljen model lahko zelo občutljiv na majhne merske napake ali spremembe vrednosti neodvisnih spremenljivk. Temu pojavu rečemo *multikolinearnost*. Odvečne neodvisne spremenljivke je zato treba odstraniti iz MLR modela. Zaradi težnje k čim večji natančnosti modela je smiselno odstraniti tiste spremenljivke, ki so z odvisno spremenljivko v nižji korelaciji.

Za napake dobljenega modela napovedovanja odvisne spremenljivke pa velja oz. mora veljati še naslednje:

3. NEPOŠEVNOST NAPAK: Srednja vrednost napake ϵ je enaka 0. To je posledica metode najmanjših kvadratov, na kateri temelji linearna regresija.

4. HOMOSKEDASTIČNOST NAPAK: Varianca napake ϵ mora biti po celotni učni množici enaka. V nasprotnem primeru je korelacija med odvisno spremenljivko in parametri modela lahko zavajajoče povprečje vzorcev višje in nižje korelacije.

5. NORMALNOST NAPAK: Napaka ϵ mora biti normalno porazdeljena slučajna spremenljivka.

Zanimivo vrsto vzorcev predstavljajo t. i. *osamelci*. Tako imenujemo meritve, ki se nenavadno razlikujejo od velike večine ostalih meritev in zato nepredvidljivo vplivajo na natančnost modela (Tabachnick in Fidell, 1996). Odstranitev osamelcev iz učne množice MLR modela je eden od običajnih regresijskih postopkov.

3.3. Pomembnost izbire regresijskih parametrov

Ko izbiramo podmnožico parametrov oz. neodvisnih spremenljivk MLR modela, se zastavi vprašanje, zakaj ne bi vzeli vseh parametrov, ki jih lahko pridobimo. To se zdi smiselno predvsem zato, ker koeficient determinacije R^2 s številom parametrov narašča. Izkaže pa se, da je uporaba vseh parametrov lahko neprimerna iz več razlogov:

- ⇝ Pridobiti celotno množico parametrov je včasih težko, časovno zahtevno in/ali samodejno nemogoče.
- ⇝ Če se omejimo na manjštevilnejšo množico parametrov, lahko to včasih bolj natančno določimo.
- ⇝ Varčnost je pomembna lastnost dobrih modelov – modeli z manj parametri omogočajo boljši vpogled v odnose med regresijskimi spremenljivkami.
- ⇝ Izračuni regresijskih koeficientov so v modelih z veliko spremenljivkami zaradi multikolinearnosti pogosto nestabilni.
- ⇝ Pokazati se da, da lahko neodvisne spremenljivke, ki so z odvisno spremenljivko v zelo nizki korelaciji (po absolutni vrednosti pod 0.1), povečajo srednjo kvadratno napako. Če take spremenljivke iz modela odstranimo, zmanjšamo napako napovedi.

  Za preizkus hipoteze o nekoreliranosti neodvisne spremenljivke $\mathbf{X_i}$ z odvisno spremenljivko $\mathbf{X}$ lahko uporabimo testno statistiko, ki temelji na Studentovi porazdelitvi.
- ⇝ Pokazati se tudi da, da lahko neodvisne spremenljivke, ki imajo v MLR modelu majhne neničelne (regresijske) koeficiente oz. uteži, povečajo srednjo kvadratno napako. Če takšne spremenljivke iz modela odstranimo, zmanjšamo napako napovedi.

V statistiki obstaja več načinov izbire 'dobre' podmnožice MLR parametrov, od katerih ima vsak svoje prednosti in slabosti. Najpogosteje se uporabljajo: *sprednja izbira*, *vzvratna eliminacija* in *postopna regresija* (Seber, 1977).

3.4. Merjenje zadovoljstva uporabnikov

Hone in Graham (Hone in Graham, 2000) sta opozorila na dejstvo, da vprašalnik (Tabela 1), s katerim avtorice ogrodja PARADISE merijo zadovoljstvo uporabnikov, ne temelji niti na teoriji niti na ustreznih empiričnih raziskavah in da je seštevanje ocen, ki naj bi merile popolnoma različne količine, sporno. Vsota naj bi bila smiselna le, če vsa vprašanja merijo isto količino.

Da bi bila vsota ali celo povprečje ocen, ki se nanašajo na učinkovitost katerega izmed modulov sistema za dialog, popolnoma nesmiselna, ni čisto res. Na izbran modul lahko gledamo kot na merjeno količino. Res je sicer, da lahko opazujemo različne vidike obnašanja tega modula, vendar nas ponavadi ne zanimajo le izolirane lastnosti, temveč tudi uspešnost modula kot celote. S tem v zvezi menimo, da tudi seštevanje ocen, dodeljenih vprašanjem za določanje zadovoljstva uporabnikov z različnimi vidiki delovanja sistema za dialog, ni popolnoma nesmiselno. Res je, da metoda ni dodelana, je pa lahko dober kazalec učinkovitosti sistema za dialog.

Ker za nobeno od obstoječih tehnik merjenja zadovoljstva uporabnikov sistemov za dialog ni dokazano, da izpolnjuje pogoje za veljaven psihometrični instrument, je treba vse sklepe, ki zajemajo zadovoljstvo uporabnikov, obravnavati zelo previdno. Žal je bil prvi resen poskus razvoja vprašalnika, ki bi zanesljivo, veljavno, objektivno in diskriminativno meril zadovoljstvo uporabnikov sistemov za dialog, (začasno) prekinjen (Hone in Graham, 2000).

Če se pojavi želja po vrednotenju katerega od modulov danega sistema za dialog (npr. modula za vodenje dialoga ali modula za razpoznavanje govora), pa je bolj smiselno sešteti ocene, dodeljene le tistim vprašanjem, ki se nanašajo na učinkovitost oz. obnašanje izbranega modula (Hajdinjak, 2006b).

3.5. Vplivi razpoznavanja govora na rezultate vrednotenja

Parameter, ki ima v funkciji učinkovitosti zaradi (po absolutni vrednosti) največje uteži najpogosteje najpomembnejšo vlogo, je parameter, ki meri učinkovitost modula za razpoznavanje govora (Walker et al., 1997b; Walker et al., 1998; Litman in Shimei, 2002; Larsen, 2003). To je, kakovost razpoznavanja govora ključno vpliva na zadovoljstvo uporabnikov – ob povečani učinkovitosti razpoznavanja govora se poveča tudi zadovoljstvo uporabnikov.

Kaj pa, če nas zanima npr. učinkovitost modula za vodenje dialoga ali razumevanje naravnega jezika? Na osnovi rezultatov, ki jih podaja literatura, smo prišli do sklepa, da bo vrednotenje učinkovitosti posameznih modulov zelo verjetno zanesljivejše in natančnejše, če odstranimo vpliv razpoznavanja govora, torej simuliramo tako rekoč popolno razpoznavanje. To lahko naredimo tako, da za pridobivanje regresijskih podatkov uporabimo eksperiment Čarovnik iz Oza (Hajdinjak in Mihelič, 2004), v katerem vlogo razpoznavalnika govora ali celo vlogo modulov za razumevanje govora prevzame človek. Ugotovili smo (Hajdinjak, 2006b), da pridejo v tem primeru v ospredje tudi tisti parametri modela učinkovitosti, ki jim zaradi izjemnega vpliva učinkovitosti razpoznavanja govora svoje vloge v preteklih študijah ni uspelo dokazati. Trdimo, da tako dobljene uteži funkcije učinkovitosti realneje izražajo vpliv parametrov na zadovoljstvo uporabnikov.

V skladu z našimi sklepi so tudi ugotovitve, do katerih so prišle Walker, Boland in Kamm (Walker et al., 1999). Ugotovile so, da se značilnosti in uteži parametrov modela učinkovitosti lahko spremenijo, če izboljšamo razpoznavanje govora.

4. Sklep

Podrobno smo preučili ogrodje PARADISE, ki velja za potencialno splošno metodologijo vrednotenja učinkovitosti sistemov za dialog. Opozorili smo na nekatere pomanjkljivosti in omejitve te metode ter predlagali morebitne rešitve (Hajdinjak in Mihelič, 2006a).

Prvič, opozorili smo na dejstvo, da je treba, če se želimo izogniti prevelikim napakam ocen, normalizirati tudi odvisno spremenljivko funkcije učinkovitosti, ki izraža zadovoljstvo uporabnikov, ter napovedano normalizirano vrednost zadovoljstva uporabnika transformirati nazaj na začetni interval.

Drugič, poudarili smo, da vprašalnik, s katerim avtorice ogrodja PARADISE merijo zadovoljstvo uporabnikov, ne temelji niti na teoriji niti na ustreznih empiričnih raziskavah, in zato ne more šteti za veljaven psihometrični instrument.

Tretjič, prišli smo do sklepa, da bo vrednotenje učinkovitosti posameznih modulov zelo verjetno zanesljivejše in natančnejše, če odstranimo vpliv razpoznavanja govora, torej simuliramo tako rekoč popolno razpoznavanje. Omenili smo, da v tem primeru pridejo v ospredje tudi tisti parametri modela učinkovitosti, ki jim zaradi izjemnega vpliva učinkovitosti razpoznavanja govora svoje vloge v preteklih študijah ni uspelo dokazati, in trdili, da tako dobljene uteži funkcije učinkovitosti realneje izražajo vpliv parametrov na zadovoljstvo uporabnikov.

5. Literatura

M. Bates in D. Ayuso. 1991. A proposal for incremental dialogue evaluation. V: *Proceedings of DARPA Speech and Natural Language Workshop*, str. 319–322. Pacific Grove, ZDA.

J. C. Carletta. 1996. Assessing the reliability of subjective codings. *Computational Linguistics*, 22(2):249–254.

M. Danieli in E. Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. V: *Proceedings of the 1995 AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, str. 34–39. Stanford, ZDA.

B. Di Eugenio in M. Glass. 2004. The kappa statistic: A second look. *Computational Linguistics*, 30(1):95–101.

H. Grice. 1975. *Logic and Conversation (Syntax and Semantics, Speech Acts, Vol. 3)*. Academic Press, New York.

M. Hajdinjak in F. Mihelič. 2004. Conducting the wizard-of-oz experiment. *Informatica*, 28(4):425–430.

M. Hajdinjak in F. Mihelič. 2006a. The paradise evaluation framework: Issues and findings. *Computational Linguistics*, 32.

M. Hajdinjak. 2006b. *Predstavitev znanja in vrednotenje učinkovitosti sodelujočih samodejnih sistemov za dialog, Doktorska disertacija*. Fakulteta za elektrotehniko, Univerza v Ljubljani, Ljubljana.

K. S. Hone in R. Graham. 2000. Towards a tool for the subjective assessment of speech system interfaces (sassi). *Natural Language Engineering*, 6(3/4):287–303.

R. A. Johnson in D. W. Wichern. 2002. *Applied multivariate statistical analysis*. Prentice-Hall, Upper Saddle River (NJ).

C. A. Kamm, D. J. Litman, in M A. Walker. 1998. From novice to expert: The effect of tutorials on user expertise with spoken dialogue systems. V: *Proceedings of the 5th International Conference on Spoken Language Processing*, str. 1211–1214. Rundle Mall, Avstralija.

L. B. Larsen. 2003. Issues in the evaluation of spoken dialogue systems using objective and subjective measures. V: *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, str. 209–214. St. Thomas, ZDA.

D. J. Litman in P. Shimei. 2002. Designing and evaluating an adaptive spoken dialogue system. *User Modeling and User-Adapted Interaction*, 12.

J. Polifroni, L. Hirschman, S. Seneff, in V. Zue. 1992. Experiments in evaluating interactive spoken language systems. V: *Proceedings of DARPA Speech and Natural Language Workshop*, str. 28–33. Harriman, ZDA.

P. Price, L. Hirschman, E. Shriberg, in E. Wade. 1992. Subject-based evaluation measures for interactive spoken language systems. V: *Proceedings of the DARPA Speech and Natural Language Workshop*, str. 34–39. Harriman, ZDA.

G. A. F. Seber. 1977. *Linear Regression Analysis*. John Wiley & Sons, New York.

E. Shriberg, E. Wade, in P. Price. 1992. Human-machine problem solving using spoken language systems (sls): Factors affecting performance and user satisfaction. V: *Proceedings of the DARPA Speech and Natural Language Workshop*, str. 49–54. Harriman, ZDA.

B. G. Tabachnick in L. S. Fidell. 1996. *Using Multivariate Statistics, Third Edition*. Harper Collins, New York.

M. A. Walker, D. Litman, C. A. Kamm, in A. Abella. 1997a. Paradise: A framework for evaluating spoken dialogue agents. V: *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics*, str. 271–280. Madrid, Španija.

M. A. Walker, D. Hindle, J. Fromer, G. Di Fabbrizio, in C. Mestel. 1997b. Evaluating competing agent strategies for a voice email agent. V: *Proceedings of the 5th European Conference on Speech Communication and Technology*, str. 2219–2222. Rodos, Grčija.

M. A. Walker, D. J. Litman, C. A. Kamm, in A. Abella. 1998. Evaluating spoken dialogue agents with paradise: Two case studies. *Computer Speech and Language*, 12(3):317–347.

M. A. Walker, J. Boland, in C. Kamm. 1999. The utility of elapsed time as a usability metric for spoken dialogue systems. V: *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, str. 317–320. Keystone, ZDA.

M. A. Walker. 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. *Journal of Artificial Intelligence Research*, 12:387–416.

Slovenska govorna in tekstovna baza parlamentarnih razprav za avtomatsko razpoznavanje govora

Andrej Žgank, Tomaž Rotovnik, Matej Grašič, Marko Kos, Damjan Vlaj in Zdravko Kačič

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
Smetanova ul. 17, SI-2000 Maribor, Slovenija
andrej.zgank@uni-mb.si http://www.dsplab.uni-mb.si

Povzetek

V članku bomo predstavili nov slovenski jezikovni vir – bazo SloParl. Sestavlja jo material zajet v okviru parlamentarnih razprav v Državnem zboru Republike Slovenije. Glavno vodilo projekta je bilo na stroškovno učinkovit način izdelati nov slovenski jezikovni vir za avtomatsko razpoznavanje tekočega govora. V govornem korpusu SloParl se tako nahaja skupno 100 ur govornega materiala. Le-ta je namenjen nenadzorovanemu oziroma rahlo nadzorovanemu učenju akustičnih modelov. V skladu s tem je potekala tudi priprava transkripcij govornega materiala. Drugi del baze SloParl tvori tekstovni korpus, ki vsebuje obdelane magnetograme parlamentarnih razprav iz obdobja 1996 - 2005. Tekstovni korpus vsebuje 23M besed. Primerjava z obstoječimi slovenskimi jezikovnimi viri je pokazala, da baza SloParl uspešno pokriva nove vidike na področju modeliranja slovenskega govora.

Slovenian parliamentary debates speech and text database for automatic speech recognition

A novel Slovenian language resource – the SloParl database – will be presented in this paper. It consists from spoken material acquired in the Slovenian Parliament. The main goal of the project was to cost-effectively collect a new Slovenian language resource that could be used to augment the available Slovenian speech corpora for developing a large vocabulary continuous speech recognition system. The SloParl speech corpus has a total length of 100 hours. The SloParl speech corpus will be used for lightly supervised or unsupervised acoustic models' training. In accordance with this, the accompanying transcriptions were prepared. The second part of the SloParl database is the text corpus, which covers text of all debates between years 1996 - 2005. It consists from 23M words. Comparison with other Slovenian language resources showed that SloParl database adds new aspects to the modelling of Slovenian language.

1. Uvod

Področje avtomatskega razpoznavanja govora je neločljivo povezano z izdelavo jezikovnih virov, ki so potrebni za izdelavo modelov vključenih v razpoznavalnik govora. Osnovne tri vrste jezikovnih virov so:

- *Govorna baza*: v obliki transkribiranega govora je potrebna za učenje akustičnih modelov razpoznavalnika govora.
- *Tekstovni korpus*: se uporablja za izdelavo jezikovnih modelov pri razpoznavanju tekočega govora.
- *Fonetični slovar*: služi za povezavo med obema zgornjima jezikovnima viroma.

Avtomatsko razpoznavanje govora dosega najboljše rezultate takrat, kadar so jezikovni viri, ki jih uporabimo za učenje modelov, kar najbolj podobni govornemu materialu s katerim se bo razpoznavalnik govora dejansko srečal. Posledica te zahteve je, da je velikokrat potrebno za novo področje uporabe razpoznavalnika govora pripraviti nove jezikovne vire.

Postopek izdelava jezikovnih virov za avtomatsko razpoznavanje govora je običajno zelo drag in dolgotrajen, saj je pri izdelavi kvalitetne govorne baze potrebno veliko ročnega dela. To se odraža v številu jezikovnih virov, ki so dostopni za posamezni jezik. Tukaj prevladujejo predvsem jeziki z velikim številom govorcev, za katere je izražen močan ekonomski interes za razvoj razpoznavalnikov govora.

S stališča kompleksnosti razpoznavalnikov govora so najobsežnejši jezikovni viri potrebni za razpoznavalnike tekočega govora. Če želimo pravilno oceniti parametre akustičnih in jezikovnih modelov, potrebujemo za izpeljavo postopka učenja velike količine učnih podatkov. V zadnjem obdobju lahko opazujemo razvoj razpoznavalnikov tekočega govora tudi za slovanske jezike (Byrne et al., 1999; Nouza et al., 2004; Žgank et al., 2001), kamor sodi tudi slovenščina. Le-ta je zaradi svojih značilnosti še posebej zahtevna za razpoznavanje tekočega govora.

Na področju razpoznavanja tekočega slovenskega govora je bila prva dostopna govorna baza SNABI (Dreo, 1995). Njena slabost je, da je omejena na posamezne domene. Razvoj slovenskega razpoznavalnika govora za neomejeno domeno omogoča govorna in tekstovna baza BNSI Broadcast News (Žgank et al., 2004) in govorna baza SiBN Broadcast News (Žibert in Mihelič, 2004). Kot posebni dodatek k bazi BNSI Broadcast News je bila razvita govorna baza SINOD, ki pokriva slovenski govor tujih govorcev (Žgank et al., 2006). Če primerjamo obseg obstoječih govornih baz za slovenski jezik s količino transkribiranega govornega materiala za druge jezike lahko vidimo, da slovenščina na tem področju razvoja jezikovnih tehnologij še vedno zaostaja.

Predstavljene težave in omejitve so bile vzpodbuda za razvoj novega slovenskega jezikovnega vira, baze SloParl[1]. Predstavljeni jezikovni vir vsebuje parlamentarne razprave iz Državnega zbora Republike Slovenije in je sestavljen iz

[1]Delo je bilo delno financirano s strani Agencije za raziskovalno dejavnost Republike Slovenije po pogodbi št. P2-0069.

govorne baze s transkripcijami in iz tekstovnega korpusa.

Projekt izdelave jezikovnega vira SloParl se je začel konec leta 2005 v sodelovanju med Fakulteto za elektrotehniko, računalništvo in informatiko Univerze v Mariboru in Državnim zborom Republike Slovenije. Osnovno vodilo pri izdelavi baze SloParl je bilo v kratkem času ter z minimalnim ročnim delom zagotoviti nov slovenski jezikovni vir za razpoznavanje tekočega govora. V primeru govorne baze SloParl smo tako namesto namenskih ročno tvorjenih transkripcij, ki natanko pokrivajo vse vidike govornega signala (govor, mašila, zvoke iz ozadja, šum, lastnosti govorca,...) uporabili magnetograme, ki so nastali v državnem zboru in vsebujejo samo prepis govora. Več avtorjev (Kemp in Waibel, 1999; Wessel in Ney, 2001; Lamel et al., 2002) je takšne govorne baze uspešno uporabilo za nenadzorovan oziroma rahlo nadzorovan postopek učenja akustičnih modelov. Takšen pristop je možno učinkovito uporabiti tudi z netranskribiranim govornim materialom. Dodatna motivacija za razvoj baze SloParl je bilo dejstvo, da se baze parlamentarnih razprav (Gollan et al., 2005; Biatov in Köhler, 2002) v zadnjem času pogosto uporabljajo za razvoj najsodobnejših razpoznavalnikov tekočega govora, kot so tisti vključeni v prevajalnike govora v govor ali v multimodalne aplikacije za zajem informacij.

V nadaljevanju članka bomo v drugem poglavju opisali postopek priprave baze SloParl ter na kakšen način je bil zajet material. Lastnosti in statistiko govornega korpusa bomo predstavili v tretjem poglavju, tekstovnega pa v četrtem. Zaključek bomo podali v petem poglavju.

2. Priprava baze SloParl in zajemanje materiala

Zasedanja Državnega zbora Republike Slovenije lahko razdelimo na dva dela. V prvem delu so redne seje, ki so praviloma na sporedu enkrat na mesec in trajajo več dni. Teme sej so običajno različne, pri tem pa lahko pokrivajo zelo širok nabor vsebin. V drugi del delovanja državnega zbora sodijo izredne seje, ki so sklicane v primeru obravnave kakšne nujne teme. Izredne seje praviloma pokrivajo ozko tematiko, vezano na posamično zadevo. Predstavljena razlika med obema vrstama sej državnega zbora je pomembna pri izdelavi jezikovnega vira.

Državni zbor sestavlja 90 poslank in poslancev, ki so izvoljeni za obdobje štirih let. Sejo vodi predsednik ali podpredsednik zbora. Hkrati s poslanci sodelujejo na sejah kot govorci tudi različni drugi ljudje. Praviloma gre tukaj za člane Vlade Republike Slovenije, ki odgovarjajo na poslanska vprašanja in pobude oziroma predstavljajo predloge zakonov.

Vsaka seja državnega zbora je hkrati snemana na dva načina. Zvok in slika sta v digitalni obliki zajeta na DVD medij, hkrati pa je samo zvok v analogni obliki posnet tudi na profesionalni magnetofonski trak. Pri izbiri medija za presnemavanje govornega materiala za bazo izgovorjav je bilo prvo vodilo kvaliteta zajetega materiala. Ker v državnem zboru pri snemanju na DVD medije uporabljajo kodek z izgubno kompresijo, smo za izvor govornega materiala v bazi SloParl izbrali analogne magnetofonske trakove. Analogni zvočni signal smo zajemali neposredno na osebni računalnik v katerega je bila vgrajena zvočna kartica visoke ločljivosti SoundBlaster Audigy. Izvorni analogni govorni signal smo digitalizirali s 16 bitno ločljivostjo pri 48 kHz vzorčenju. Kasneje smo zajeti govor pretvorili v signal s 16 bitno ločljivostjo in 16 kHz vzorčenjem.

Med zajemanjem govornega signala so se občasno pojavile težave z nastavitvijo glasnosti, saj imajo različni govorci različen stil govora. Praviloma je bil med glasnejšimi govorci predsedujoči, ki smo ga tako uporabili za nastavitev referenčne vrednosti. Akustično okolje sej državnega zbora je s stališča avtomatskega razpoznavalnika govora zelo kompleksno, saj se pogosto pojavljata šum ter govor v ozadju. Velika parlamentarna dvorana lahko v določenih primerih generira odmev, kar dodatno oteži akustično okolje. Ozvočenje govorcev na sejah državnega zbora je izvedeno s konferenčnimi mikrofoni, ki so nameščeni na mizah pred parlamentarci.

V govorni del baze SloParl smo izbrali posnetke iz obdobja 2000 - 2005. Pri izbiri posnetkov parlamentarnih sej smo prednostno izbrali tiste seje, katerih datumi se prekrivajo s televizijskimi oddajami vključenimi v slovensko bazo BNSI Broadcast News (Žgank et al., 2004). Takšno prekrivanje govornega materiala je še posebej pomembno, kadar želimo zožiti domeno akustičnih in jezikovnih modelov avtomatskega razpoznavalnika govora.

Drugi del baze SloParl sestavlja tekstovni korpus. Magnetogrami parlamentarnih razprav so prosto dostopni na domači strani Državnega zbora RS za obdobje od leta 1996 naprej. Uporabljen je HTML format datotek. Magnetogrami sej vsebujejo zapise razprav, ki so jih pripravili v državnem zboru. Pripravljeni tekstovni korpus bomo uporabili predvsem za učenje jezikovnih modelov avtomatskega razpoznavalnika tekočega govora.

3. Govorni korpus

Govorni korpus slovenske baze SloParl vsebuje posnetke 100 ur govornega materiala. Celotno govorno bazo SloParl smo razdelili na tri nabore: učnega, razvojnega in testnega. Največji izmed teh treh je učni nabor, ki obsega 92 ur govornega materiala. V razvojnem naboru (redna seja državnega zbora iz junija 2001), ki je namenjen nastavljanju parametrov razpoznavalnika tekočega govora, se nahajajo 4 ure posnetkov. Preostale 4 ure govornega materiala (redna seja iz februarja 2002) smo dodelili v testni nabor, ki je namenjen vrednotenju razpoznavalnika govora. Osnovna statistika govornega korpusa baze SloParl je predstavljena v tabeli 1.

Parameter	Vrednost
Število sej	20
Redne seje	13
Izredne seje	7
Povprečna dolžina (učni set)	5:05

Tabela 1: Statistika govornega korpus slovenske baze SloParl.

Govorni korpus sestavlja 20 sej državnega zbora, od tega 13 rednih ter 7 izrednih. Povprečna dolžina seje, ki

je vključena v učni nabor, znaša 5:05 ure. Običajno vsebuje vsaka seja državnega zbora različne prekinitve. Takšne prekinitve smo izrezali iz posnetkov, saj so nepotrebne za učenje akustičnih modelov. V nadaljevanju smo analizirali razlike med rednimi in izrednimi sejami državnega zbora. Potrdila se je napoved, da redne seje pokrivajo širši spekter tematik, ter so običajno napovedane za daljše obdobje vnaprej. Nasprotno je vsebina izrednih sej osredotočena na eno samo tematiko, običajno pa so sklicane v krajšem časovnem roku. Slovar izrednih sej je praviloma manjši in bolj homogen, kot pri rednih sejah. To dejstvo olajša razvoj razpoznavalnika govora, saj je tako lažje adaptirati slovar razpoznavalnika, ter s tem zmanjšati delež besed izven slovarja. Le-ta predstavlja za pregibne jezike eno izmed glavnih ovir za dosego dobrega rezultata. Redne seje državnega zbora, vključene v učni set baze SloParl so v povprečju za približno 12% daljše kot izredne seje.

Bistveni del govorne baze za učenje akustičnih modelov so transkripcije izgovorjenega. Osnovno vodilo, ki smo mu sledili pri izdelavi transkripcij za bazo SloParl, je bilo čimbolj zmanjšati količino potrebnega ročnega dela. Tako smo kot osnovo vzeli magnetograme sej, ki so dostopni na domači strani parlamenta. Le-ti v veliki meri vsebujejo prepis govora iz razprav. V magnetogramih manjkajo oznake za efekte spontanega govora (npr.: mašila, ponovni štarti, zatikanja,...), katerih modeliranje lahko izboljša kvaliteto akustičnih modelov. Po drugi strani pa magnetogrami vsebujejo različne dodatne meta informacije (npr.: ime govorca, rezultati glasovanje, časovne oznake,...), ki niso neposredno povezane z izgovorjenim besedilom. Ime govorca, datum, številko in tip seje smo obdržali v glavi transkripcije kot meta informacijo, medtem ko smo ostanek takšnih informacij izločili iz transkripcij.

Učni nabor govornega korpusa SloParl smo razdelili v dva enako velika dela. Vsak je velik 46 ur. V prvem delu smo transkripcije pustili v takšni neobdelani obliki. Drugi polovici transkripcij smo ročno dodali časovne meje za vsako menjavo govorcev. Dodatna časovna informacija lahko izboljša kvaliteto akustičnih modelov (Lamel et al., 2002), in jo je možno relativno hitro in preprosto dodati v transkripcije.

Če želimo uporabljati razvojni in testni nabor govornega korpusa za razvoj sistema za razpoznavanje tekočega govora, potrebujemo popolne transkripcije izgovorjenega. Tako je potrebno oba nabor ročno transkribirati. Pri tem smo uporabili tri fazni pristop, ki smo ga uporabili že pri izdelavi slovenske baze BNSI Broadcast News (Žgank et al., 2004). Magnetogrami sej so služili za tvorjenje inicialne verzije transkripcij. Za obdelavo transkripcij smo uporabili pravila zapisovanja in delovno okolje (program Transcriber (Barras et al., 2001)), ki smo jih uporabljali že v projektu BNSI Broadcast News.

Da bi pokazali obseg in kompleksnost govornega korpusa slovenske baze SloParl smo opravili analizo transkripcij. Rezultati statistike so podani v tabeli 2.

Dvajset sej vključenih v slovensko bazo SloParl vsebuje 3665 menjav govorcev. V celotnem govornem korpusu so izgovorjave 255 različnih govorcev. Transkripcije 100 ur govornega materiala vsebujejo skupaj 655k besed, kjer je 37k besed različnih. Za primerjavo poglejmo obseg

Parameter	Vrednost
Menjave govorca	3665
Število govorcev	255
Število besed	655k
Število različnih besed	37k

Tabela 2: Analiza transkripcij govornega korpusa slovenske baze SloParl.

slovenske govorne baze BNSI Broadcast News (Žgank et al., 2004): ta v 36 urah govornega materiala pokriva 1565 govorcev, ki so izgovorili 268k besed (37k različnih). Približno isto število različnih besed v bazah SloParl in BNSI je verjetno posledica dejstva, da je tematika parlamentarnih razprav praviloma ožja, kot je tematika dnevnoinformativnih oddaj. Ker so politiki pogosto gostje v dnevnoinformativnih oddajah, smo primerjali oba nabora govorcev. V obeh se pojavlja 89 govorcev, kar predstavlja 34,9% govorcev v bazi SloParl. Analizirali smo tudi prekrivanje med slovarjem govornega korpusa SloParl in BNSI Broadcast News. Prekrivanje je bilo 46,3%. Glede na relativno nizek nivo prekrivanja slovarjev je možno sklepati, da bo bazo SloParl možno učinkovito uporabiti kot dopolnilo k bazi BNSI Broadcast News.

4. Tekstovni korpus

Drugi – tekstovni – del baze SloParl je namenjen učenju jezikovnih modelov razpoznavalnika tekočega govora. Kvaliteta jezikovnih modelov, ki jih uporabimo v razpoznavalniku tekočega govora za pregibne jezike je še posebej pomembna, saj ocena jezikovnega modela bistveno vpliva na doseženi rezultat. Neobdelan tekst s parlamentarnimi razpravami smo zajeli na domači strani Državnega zbora Republike Slovenije. V tekstovni korpus za izdelavo jezikovnih modelov smo vključili vse redne in izredne seje iz obdobja 1996 - 2005.

Tekst razprav dostopen na domači strani državnega zbora uporablja kodno tabelo UTF-8, ki smo jo zaradi združljivosti z ostalimi slovenskimi jezikovnimi viri spremenili v kodno tabelo ISO 8859-2. Kot je že bilo omenjeno, vsebujejo magnetogrami nekatere dodatne informacije, kot je na primer: ime govorca, rezultat glasovanja, časovne meje... Analiza, katere izmed teh informacij je smiselno ohraniti v tekstovnem korpusu za učenje jezikovnih modelov, je pokazala na sledeče parametre: ime govorca, datum, številka seje in tip seje. Te parametre smo ohranili v tekstovnem korpusu baze SloParl kot dodatno informacijo. Pričakujemo lahko, da bomo te informacije koristno uporabili med postopkom razvoja razpoznavalnika govora. Eden izmed možnih načinov uporabe teh parametrov je združevanje različnih tipov sej za zožanje tematike jezikovnih modelov. Vse ostale dodatne informacije, ki so se nahajale v magnetogramih in niso predstavljale izgovorjenega, smo izločili iz tekstovnega korpusa.

V nadaljevanju smo opravili analizo tekstovnega korpusa baze SloParl – rezultati analize so predstavljeni v tabeli 3.

V tekstovni korpus slovenske baze SloParl smo vključili 10 let parlamentarnih razprav. V tem obdobju se je odvi-

Parameter	Vrednost
Letniki	10
Število sej	188
Redne seje	69
Izredne seje	119
Število razprav	781
Število besed	23M
Število različnih besed	182k

Tabela 3: Statistika tekstovnega korpusa slovenske baze SloParl.

jalo 188 sej, od tega jih je bilo 69 rednih, preostalih 119 pa izrednih. Posamezna seja traja običajno več kot en dan, zato je skupno število razprav bistveno višje. V tem obdobju je tako bilo 781 razprav, ki smo jih vključili v tekstovni korpus. Le-ta vsebuje skupaj 23M besed, kjer je 182k besed različnih. Tiste seje državnega zbora, ki smo jih vključili v razvojni in testni nabor govornega korpusa, smo v celoti izločili iz tekstovnega korpusa.

Predstavitev novega slovenskega jezikovnega vira, baze SloParl, bomo zaključili s primerjavo z drugimi slovenskimi tekstovnimi korpusi. Tukaj je kot posebej pomembno dejstvo potrebno izpostaviti, da tekstovni korpus baze SloParl vsebuje znaten delež zapisov govorjenega jezika, ki se po svojih tipičnih lastnostih bistveno loči od pisanega besedila v časopisnih korpusih, ki so običajno v uporabi. To je še posebej pomembno pri razpoznavanju spontanega tekočega govora, kjer želimo doseči čim večjo skladnost med akustičnimi in jezikovnimi modeli. V bazi SloParl tako najdemo spontan govora v primeru poslanskih replik. Po drugi strani je bran govor prisoten v primerih predstavitve predlogov zakonov. Primerjava s časopisnim korpusom Večer pokaže, da le-ta vsebuje 105M besed, od tega 660k različnih. Če sedaj primerjamo razmerje med skupnim številom besed in številom različnih besed za oba tekstovna korpusa, lahko vidimo, da je le-to podobno, kar kaže na primerljivo kompleksnost obeh tekstovnih korpusov.

5. Zaključek

V članku smo predstavili nov slovenski jezikovni vir, bazo SloParl, ki vsebuje parlamentarne razprave Državnega zbora Republike Slovenije. Sestavljata jo govorni in tekstovni korpus. Osnovno vodilo pri izdelavi baze je bilo na stroškovno učinkovit način povečati število slovenskih jezikovnih virov, ki so namenjeni avtomatskemu razpoznavanju tekočega govora z velikim slovarjem besed.

Govorni korpus SloParl bomo uporabljali za nenadzorovano oziroma rahlo nadzorovano učenje akustičnih modelov. Tekstovni korpus je v kombinaciji s časopisnimi korpusi namenjen učenju jezikovnih modelov. Pri nadaljnjem delu se bomo osredotočili na razvoj razpoznavalnika govora za področje parlamentarnih razprav.

Zahvala

Avtorji članka se zahvaljujejo osebju Državnega zbora Republike Slovenije, ki je sodelovalo pri izvedbi projekta SloParl.

6. Literatura

Barras, C., Geoffrois, E., Wu, Z. and Liberman, M., "Transcriber: Development and use of a tool for assisting speech corpora production", Speech Communication, Vol. 33, Issues 1-2, 5-22, 2001.

Biatov, K., Köhler, J., "Methods and Tools for Speech Data Acquisition exploiting a Database of German Parliamentary Speeches and Transcripts from the Internet", Proc. LREC 2002, Las Palmas, Španija, junij 2002.

Byrne, W., Hajic, J., Ircing, P., Khudanpur, F., McDonough, J., Peterek, N., and Psutka, J., "Large vocabulary speech recognition for read and broadcast Czech", Proc. Workshop on Text Speech and Dialog, Plzen, Češka, 1999, Lecture Notes in Artificial Intelligence, Vol. 1692.

Dreo, D., "Slovene speech data base SNABI", Dialog Man-Machine : second International Workshop, Maribor, Slovenija, 1995.

Gollan, C., Biasni, M., Kanthak, S., Schlüter R., Ney, H., "Cross Domain Automatic Transcription on the TC-STAR EPPS Corpus", Proc. ICASSP 2005, Philadelphia.

Kemp, T., Waibel, A., "Unsupervised Training Of A Speech Recognizer: Recent Experiments", Proc. Eurospeech 1999, Budimpešta, Madžarska.

Lamel, L., Gauvain, J., and Adda, G., "Lightly supervised and unsupervised acoustic model training", Computer Speech & Language, Volume 16, Issue 1, , januar 2002, 115–129.

Nouza, J., Nejedlova, D., Zdansky, J., Kolorenc, J., "Very Large Vocabulary Speech Recognition System for Automatic Transcription of Czech Broadcast Programs", Proc. ICSLP 2004, Jeju Island, Koreja.

Wessel, F., Ney, H., "Unsupervised training of acoustic models for large vocabulary continuous speech recognition". In Proc. IEEE Automatic Speech Recognition and Understanding Workshop, Madonna di Campiglio, Italija, december 2001.

Žgank, A., Kačič, Z., and Horvat, B., "Large vocabulary continuous speech recognizer for Slovenian language", Proc. Text, speech and dialogue : 4th international conference, TSD 2001, Železna Ruda, Češka, Lecture notes in Artificial Intelligence, Vol. 2166, 242–248, Springer 2001.

Žgank, A., Rotovnik, T., Sepesy Maučec, M., Verdonik, D., Kitak, J., Vlaj, D., Hozjan, V., Kačič, Z., Horvat, B., "Acquisition and annotation of Slovenian Broadcast News database", Fourth international conference on language resources and evaluation, Lizbona, Portugalska. LREC 2004, Vol. 6, 2103–2106, 2004.

Žgank, A., Verdonik, D., Zögling Markuš, A., Kačič, Z., "SINOD - Slovenian non-native speech database", Proc. LREC 2006, Genova, Italija.

Žibert, J., Mihelič, F., (2004) Development of Slovenian broadcast news speech database. *Fourth International Conference on Language Resources and Evaluation*, Lizbona, Portugalska.

Načelo večjezičnosti ali večjezični korpus iz manjše množice dvojezičnih

Jasna Belc, Miran Željko

Vlada Republike Slovenije
Generalni sekretariat Vlade Republike Slovenije
Služba za prevajanje, tolmačenje, redakcijo in terminologijo
Gregorčičeva 20, 1000 LJUBLJANA, Slovenija
Tel.: +386 1 478 25 44; faks: +386 1 478 15 62
e-naslov: jasna.belc@gov.si, miran.zeljko@gov.si

Povzetek

Članek prikazuje zamisel, kako nastane večjezični korpus na podlagi več dvojezičnih in njegovo uresničitev. Prototipni model je zamišljen na 3 korpusih s tremi različnimi jeziki, ki vsebujejo kot eno od sestavin slovenščino. Uresničitev zamisli kot zastavljeni projekt je na dosegu nekaj pretvorbnih programov in uporabi arhitekture terminološke zbirke kot lupine, ki daje zavetje večjezični korpusni zbirki (4-jezični), njen spletni prikaz pa se uresničuje z uporabo konkordančnega programa, ki omogoča iskanje poljubnih delcev segmentne celote (za preverjanje prevodnih ustreznic) znotraj 4 mogočih jezikov. Projekt je prav tako zanimiv zaradi priprav Slovenije na predsedovanje EU v drugi polovici leta 2008.

The Multilinguality Principle or a Multilingual Corpus Derived from Several Bilingual Corpora

The article presents the idea how we can build a multilingual corpus from the various bilingual corpora and its realisation. The conception of the prototype model is carried out from three bilingual corpora containing three different languages where the other component is Slovene. The idea has been carried out as a project which is attainable within the scope of some transformational programmes and the use of architecture of the terminological database as a shell for hosting the new multilingual corpus collection (4-lingual), its web representation is realised with the concordancer that allows for the searching of any part of the whole segments (to find the translation matches) within 4 available languages. The project is interesting also within the scope of the preparations of Slovenia to the EU presidency in the second half of 2008.

1. Uvod

Cilj tega projekta so jezikovne tehnologije, ki so usmerjene v izdelavo produktov uporabne vrednosti, ki so uporabni pri iskanju večjezičnih informacij bodisi zaradi običajnega prevajanja, iskanja strokovnega izrazja v več jezikih hkrati, kot podlaga za strojno prevajanje in za raznovrstne raziskave, ki lahko temeljijo ali pa se dopolnjujejo s preverjanjem na več jezikih ipd.

Glavni cilj projekta, ki že poteka, obsega:

– pripravo več dvojezičnih korpusov,
– pripravo omejenega štirijezičnega korpusa, nastalega iz 3 dvojezičnih.

Dosedanji produkti:

– terminološka zbirka s temi poglavitnimi lastnostmi: urejena in strukturirana po vsebini in obliki (metajezikovno označevanje), uporabna in splošno dostopna na spletnih straneh;
– korpusna zbirka – doslej dvojezična, v pripravi za objavo pa še več dvojezičnih zbirk, v katerih je eden od jezikov slovenščina, drugi pa (za zdaj) trije pomembni evropski in svetovni jeziki.

2. Uresničitev projekta

Uporaba obstoječih orodij nemške znamke Trados nam omogoča naslednje osnovne operacije:

– poravnavanje besedil v dvojezični različici z orodjem WinAlign;
– prevajanje in urejanje prevodnih zbirk v pomnilniku prevodov z orodjem Translator's Workbench;
– urejanje terminološke ali kakšne druge zbirke v okviru ali arhitekturi, ki služi kot ogrodje ali lupina za vnos, shranjevanje različno strukturiranih večjezičnih ali večaspektualnih podatkov glede na zamisel sestavljavca take podatkovne zbirke, hkrati tudi shranjevanje besedilnih (črkovno-številčnih podatkov), grafičnih ali kakšnih drugih podatkov – z orodjem MultiTerm.

Faze, potrebne za uresničitev projekta:

1) Poravnave (vzporeditve) dvojezičnih besedil, od katerih je eden od jezikov slovenščina, omogočijo nastanek dvojezičnega pomnilnika, v katerega se po zamišljeni zgradbeni predlogi uvozijo rezultati dobljenih poravnav. Poravnave z orodjem WinAlign lahko sicer po vnosu ustreznih preverjenih besedil, ki si medsebojno ustrezajo glede na vmesno dejavnost prevajanja med takima besediloma (ki je običajno enosmerna, npr. iz slovenščine v francoščino ali iz angleščine v slovenščino), potekajo samodejno, vendar je zaradi standardnih algoritmov takih poravnalnih programov bolje s sodelovanjem človeka omogočiti preverjeno poravnavo, ki zagotavlja kakovost poravnanih besedil, tj., da si po dva in dva segmenta iz različnih jezikov dejansko ustrezata po prevodu. Deloma to zagotavlja sama segmentacija besedil v vsakem posameznem jeziku, ki je vključena v program poravnave *(alignment programme),* vendar pa sredstva, na katera se opira sama poravnava, po jezikih niso enako razporejena. Gre za neke vrste pravopisna interpunkcijska znamenja (ločila), ki pa jih različni jeziki različno uporabljajo. Poleg ustaljenih jezikovnih interpunkcijskih znamenj v elektronskih besedilih najdemo še druga znamenja (npr. presledek, konec

vrstice, konec odstavka, alinejni zamik ali tabulator ipd.), ta protistavimo v dveh različnih jezikovnih besedilih, ki tako razpadeta na segmente. Nastali segmenti niso vedno enako dolgi, kakor tudi ne velja vedno – čeprav je pri določeni vrsti strokovnih besedil zaželeno prevajanje z upoštevanjem enakega zaključka bistvenih besedilnih delov, kot so npr. stavki, ki se končujejo običajno s piko (.), alineje, ki se zaključujejo običajno s prehodom v novo vrstico (¶ ipd.), včasih tudi z vejico (ki pa tu ni pomembna), uvajalni stavki, ki se končujejo z dvopičjem (:), členitev besedila na širše ali ožje stavčne ali besednozvezne enote, ki se končujejo pogosto s podpičjem (;) – da bi si dvojezični segmenti ustrezali homomorfno. Poravnalni algoritmi lahko povezujejo kvečjemu dva segmenta z nasprotnim edinim segmentom (v drugem jeziku) ali nasprotno, razporeditev načina vzporeditve pa je bolj ali manj prepuščena preračunavanju s statističnimi metodami v samem programu. Program poravnave omogoča uporabniku le nekaj izbir, npr. izbiro poravnave po odstavkih kot segmentnih delih ali pa izbiro »stavčnih segmentov«, opisanih ob rabi segmentacijsko-interpunkcijskih sredstev.

2) »Pretvorba« iz poravnav v pomnilnik prevodov: po izbranih oblikovnih in vsebinskih nastavitvah v 'praznem' pomnilniku prevodov orodja Translator's Workbench (Database Setup in Project Settings) v pomnilnik uvozimo vse želene poravnave, ki smo jih uresničili s poravnalnikom dvojezičnih besedil. Pomnilnike ločimo glede na prevodni jezikovni par (npr. angleško-slovenski, slovensko-francoski ali nemško-slovenski ipd.), dobimo vsaj tri različne pomnilnike prevodov, ki jih pozneje pretvorimo v korpus zaradi objave in javnega dostopa na spletu.

3) »Pretvorba iz pomnilnika v korpus: ne glede na prejšnjo smer prevajanja iz pomnilnika prevodov izvozimo njegovo vsebino in jo z ustreznim programom za predstavitev na spletu (pretvorba v html zapis ipd.) prikažemo na spletnih straneh. Take pretvorbe so bile prikazane že v prejšnjih objavah M. Željka ob nastanku prvega korpusa besedil, zajetih pri programu prevajanja zakonodaje EU iz angleščine v slovenščino (glej vir 7). Ob postavitvi več dvojezičnih korpusov pa nastane vprašanje, ali jih lahko med seboj povežemo. Na voljo imamo namreč po en skupen sestavnik vseh dvojezičnih korpusov, tj. slovenski del prevodov oz polovico vsakega jezikovnega para. Kaže pa, da pri obstoječih pomnilnikih prevodov ali dodanih dvojezičnih korpusih taka pretvorba ni mogoča, ker so segmenti zelo različni in samo označevanje z vsebinskimi atributi brez oznake zaporednega segmenta znotraj nekega besedila oz. v samem korpusu, pomnilniku ali dvojezični zbirki ni mogoče. S pretvorbo v »večjezično zbirko« pa bi dobili želeni »večjezični korpus«.

4) Kaj storiti? Ker so si izvozne in uvozne datoteke iz dveh (če ne celo vseh treh) orodij Tradosovega paketa za prevajanje in urejanje terminologije zelo podobne (vsebujejo namreč enote, ki so lahko sestavni del kake standardno kodirane xml datoteke), jih lahko združimo oz. pretvorimo eno v drugo, le da dobimo iz WinAlignovega poravnalnika in Translator's Workbencha samo dvojezične prevodne enote. Tu nam na pomoč priskoči kot ogrodje ali lupina, namenjena strukturiranemu shranjevanju podatkov, Tradosova aplikacija Multiterm. Potrebno je še nekaj pretvorb, preverjanja rezultatov, prilagajanja lastnim potrebam in zamislim in rezultati prototipne večjezične prevodne zbirke oz. korpusa so lahko pred nami.

Podobnosti med izvozom iz obeh Tradosovih orodij si lahko ogledamo na kratkem izseku segmentov, ki jih omogočajo vpisi v »terminološko zbirko« Multiterm ali prevodne enote iz Translator's Workbencha. Za vnos v štirijezični korpus prevodov (v lupini Multiterma) je treba pripraviti ustrezen program pretvorb med obema zapisoma v izvoznih datotekah (xml ali sorodnih vrst) s preambulo in ustreznim zaključkom.

Segmenti iz Multiterma:

```
**
<Creation Date>25.07.2001 - 18:04:00
<Created By>super
<Change Date>25.07.2001 - 18:04:00
<Changed By>super
<Entry Class>1
<Graphic>
<Entry Number>20251
<Subject>informatics AUL
<Subj>informatika
<SourceDoc&Lang>Sklep Sveta 92/242/EGS
<EN>standardization activities
<SL>dejavnosti standardiziranja
<Reliability>4
<DE>Normungstätigkeiten
<FR>activités de normalisation
<TermRef>uvod priloge
**
<Creation Date>25.07.2001 - 18:04:00
<Created By>super
<Change Date>25.07.2001 - 18:04:00
<Changed By>super
<Entry Class>1
<Graphic>
<Entry Number>20244
<Subject>informatics AUL
<Subj>informatika
<SourceDoc&Lang>Sklep Sveta 92/242/EGS
<EN>security of information systems
<SL>varnost informacijskih sistemov
<Reliability>4
<DE>Sicherheit von Informationssystemen
<FR>sécurité des systmes d'information
<TermRef>preambula
**
<Creation Date>25.07.2001 - 18:04:00
<Created By>super
<Change Date>25.07.2001 - 18:04:00
<Changed By>super
<Entry Class>1
<Graphic>
<Entry Number>20245
<Subject>informatics AUL
<Subj>informatika
<SourceDoc&Lang>Sklep Sveta 92/242/EGS
<EN>information market
<SL>informacijski trg
<Reliability>4
<DE>Informationsmarkt
<FR>marché de l'information
<TermRef>preambula
**
```

<Creation Date>16.07.1998 - 22:39:50
<Created By>super
<Change Date>10.09.2004 - 08:25:43
<Changed By>super
<Entry Class>1
<Graphic>
<Entry Number>2157
<Subj>informatika
<Subject>informatics AUL
<EN>data bank
<SL>banka podatkov
<FR>banque des données
<DE>Databank
<TermRef>Evropski sporazum, Ur. l. 44, 1997
**

Segmenti iz Translator's Workbencha:
**
<TrU>
<CrD>18052004, 16:58:32
<CrU>SVEZ
<Att L=Stanje>Pravna redakcija
<Att L=Področje>Znanost in kultura
<Txt L=Oznaka>32002D2367
<Seg L=EN-GB>Major projects
<Seg L=SL>Glavni projekti
</TrU>
<TrU>
<CrD>18052004, 16:58:32
<CrU>SVEZ
<Att L=Stanje>Pravna redakcija
<Att L=Področje>Znanost in kultura
<Txt L=Oznaka>32002D2367
<Seg L=EN-GB>These will be subject to a formal project management approach, as follows:
<Seg L=SL>Uradni pristop projektnega upravljanja bo veljal za naslednja področja:
</TrU>
<TrU>
<CrD>18052004, 16:58:32
<CrU>SVEZ
<Att L=Stanje>Pravna redakcija
<Att L=Področje>Znanost in kultura
<Txt L=Oznaka>32002D2367
<Seg L=EN-GB>Different instruments of cooperation between national statistical organisations and Eurostat will be put in place.
<Seg L=SL>Vzpostavljeni bodo različni instrumenti sodelovanja med nacionalnimi statističnimi organizacijami in Eurostatom.
</TrU>
<TrU>
<CrD>18052004, 16:58:33
<CrU>SVEZ
<Att L=Stanje>Pravna redakcija
<Att L=Področje>Znanost in kultura
<Txt L=Oznaka>32002D2367
<Seg L=EN-GB>Quality assurance and the scientific basis of Community statistics will be the result of close cooperation between official and academic statisticians.
<Seg L=SL>Zagotavljanje kakovosti in znanstvena podlaga statistike Skupnosti bosta posledica tesnega sodelovanja med uradnimi in akademskimi statističnimi krogi.
</TrU>
<TrU>
<CrD>18052004, 16:58:34
<CrU>SVEZ
<Att L=Stanje>Pravna redakcija
<Att L=Področje>Znanost in kultura
<Txt L=Oznaka>32002D2367
<Seg L=EN-GB>Specific projects
<Seg L=SL>Specifični projekti
</TrU>
**

Slika 1

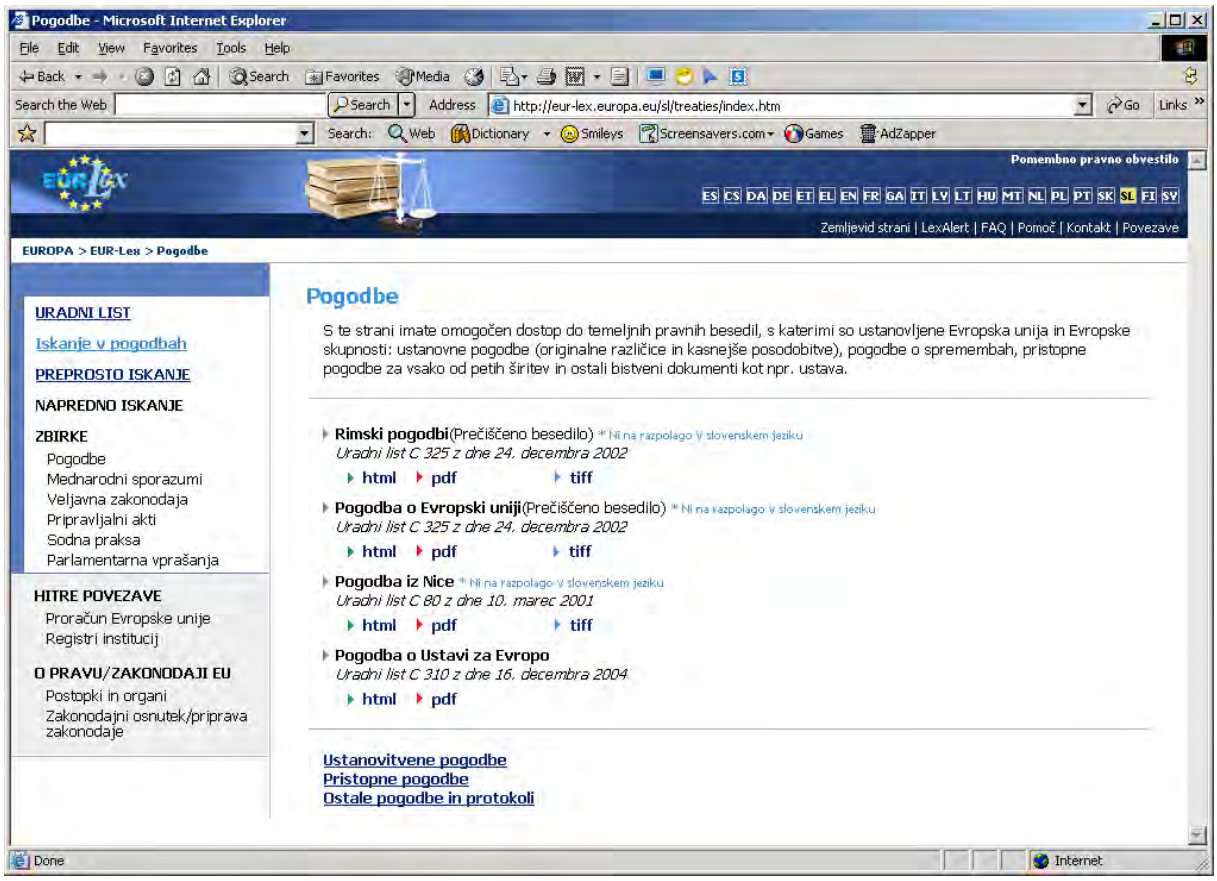
Viri za večjezični korpus:

Med slovenskimi viri, ki so primerni za vzporeditev z drugimi jeziki, so zlasti ustanovitvene in pristopne pogodbe Evropskih skupnosti, med temi tudi že ratificirane pogodbe o dveh novih članicah, ki bosta predvidoma vstopili v Evropsko unijo leta 2007, ter še neratificirana Pogodba o Ustavi za Evropo. Prevod historičnih različic teh pogodb je bil ena od obvez Slovenije za njeno polnopravno članstvo v Evropski uniji leta 2004.

Uresničitev projekta kot takega je povezana z naslednjimi koraki in predvidenimi rešitvami morebitnih težav:

a) prenos iz enega xml zapisa (izvoz dvojezičnega korpusa oz. prevodne zbirke iz Translator's Workbencha) v drug xml zapis zaradi že vgrajenih funkcij v Multitermu ne bi smel povzročati težav: dvojezične zapise iz dvojezičnih korpusov se tako lahko zaporedno prenese kot xml zapise v lupino Multiterma, ki mu vnaprej določimo izbrane atribute, npr. želene jezike in druge podatke, atrubute, ki jih vsebuje tudi dvojezična prevodna zbirka ali korpus (Področje, Stanje, Oznaka – ki se nanašajo na posamezen dokument in kot so zapisani
in poenoteni v vseh pomnilnikih prevodov);

b) zaporedni prenos (uvoz) xml zapisov dvojezičnih prevodnih zbirk v Multitermovo lupino utegne le podvojiti ali potrojiti vnos slovenskega segmenta v Multitermovo podatkovno bazo, kar se da rešiti s preprostim algoritmom iskanja in brisanja 'sinonimnih terminov', v našem primeru homografnih segmentov (stavkov ali besednih zvez). Ta poseg v samem Multitermu – glede na številne funkcije, ki omogočajo redakcijo, zlivanje enot ali zlivanje in brisanje odvečnih ali enakopisnih segmentov v programu Multiterm - ali pa ob ponovnem izvozu dobljene vsebine iz Multiterma ob ustvarjanju nove večjezične zbirke ni posebno zapleten, predvsem zaradi številnih možnosti, ki jih nudi sam Multiterm ali pa homogenost zapisa v formatu xml. Celoten potek uskladitve (poenotenja in normalizacije) novonastale večjezične zbirke zato ne predstavlja velikih težav, le preudaren razmislek in načrtovanje zaporednih korakov urejanja same zbirke, ki se jo pripravi za nov izvoz podatkovne baze v format xml, ki je podlaga vsebinskega vira podatkov, po katerih se 'sprehaja' konkordančni program med iskanjem želenih iskanih nizov (zadetkov) iz enega od jezikov v večjezičnem korpusu, postavljenem na splet.



Slika 2

3. Uporabna vrednost projekta

Prednosti takega večjezičnega »korpusa«: poravnave so narejene po »stavčnih segmentih« in ne po odstavkih, kar omogoča večjo verjetnost zadetkov v pomnilniku prevodov in tudi v korpusu. Poznavanje najpomembnejših dokumentov zakonodaje EU je bistvenega pomena za vse državljane držav članic Evropske unije. Prikaz takih segmentov v okolju različnih tujih spremnih jezikov omogoča primerjavo slovenščine s še tremi tujimi jeziki, med katerimi lahko izberemo tistega, ki ga najbolje poznamo ali potrebujemo pri svojem delu (npr. prevajanju, , tolmačenju, lektoriranju oz. jezikovni redakciji, dejavnostih v zvezi s terminologijo, stroko oz. samimi pripravami na predsedovanje Slovenije Evropski uniji). Nadaljnja uporaba večjezičnega korpusa bo pokazala, v katero smer naj se še razvijajo dodatne aplikacije, ki jih je ponudila svojim uporabnikom Služba za prevajanje, tolmačenje, redakcijo in terminologijo v Generalnem sekretariatu Republike Slovenije (GSV).

GSV s svojim terminološkim in jezikovnotehnološkim delom ponuja vedno nove rešitve, hkrati pa daje svoje produkte na voljo tudi raziskovalnim ustanovam, ki z dodatnimi funkcijami (označevanja: besednovrstnega *(Part of Speech)*, skladenjskega (besednozveznega, vsaj z določanjem jeder in ujemanj) ali pomenoslovnega (dodajanje ontologij ali pomenoslovnih abstraktnih kategorij v smislu raziskovalnega dela J. Pustejovskega in drugih jezikoslovcev), tipiziranih pomenoslovnih kategorij, ki jih uporabljajo zlasti kategorialne slovnice itd.) ob skupnem sodelovanju omogočajo nastanek vedno novih produktov, namenjenih zlasti prevajalcem in jezikoslovcem v neposredno uporabo ali tudi za nadaljnje jezikoslovne, jezikovnotehnološke ali računalniške raziskave.

In še okvirni številčni izračun:

Najpomembnejši dokumenti, ki naj bi sestavljali prototipni projekt večjezičnega korpusa s stavčno poravnanimi segmenti (ustanovitvene in pristopne pogodbe, druge pogodbe in protokoli, glej sliko 2), štejejo skupaj okrog 20 000 strani uradneg lista EU. Povprečno dobimo iz ene strani vsaj 10 prevodnih enot, ena prevodna enota pa šteje približno 30 besed. Tako je mogoče v povprečju pričakovati vsaj 5 do 6 milijonov besed v celotnem osnovnem večjezičnem korpusu z osnovnimi poravnanimi dokumenti v 4 jezikih, in ta bo v jeseni na vpogled in uporabo na spletnih straneh naše službe.

4. Pomen večjezičnega korpusa za druge stroke in uporabnike

Poleg nekaj navedenih konkretnih primerov uporabe bo imel večjezični korpus in njegov spletni konkordančnik tudi pomembnejšo pravno-politično, geografsko prepoznavalno, kulturno in izobraževalno vlogo. Na pravno-političnem in geografskem področju gre za pomembno vlogo pri širjenju načel demokratičnosti uporabe uradnega jezika, ki ga lahko primerjamo z drugimi, zlasti delovnimi jeziki institucij EU, povečanje prepoznavnosti Slovenije in slovenskega

jezika, večje poznavanje Slovenije in slovenščine s pomočjo učenja jezika za potrebe prevajanja v ustanovah EU iz slovenščine v tuje jezike za tolmače in prevajalce slovenščine (med katerimi so tudi tujci), pa tudi zaradi želje po poznavanju slovenske kulture nasploh, ne le zgolj v pravnostrokovnih krogih in pravno-političnih besedilih. Omenjeni korpus je lahko tudi vzvod za širjenje tovrstnega vedenja, kar postavlja Slovenijo v sklop uresničitev prej nezavednih želja posameznikov kot sodržavljanov Evropske unije.

Viri

1) Hans van Halteren (1999): Syntactic Wordclass Tagging, Kluwer Academic Publishers, Dordrecht, Boston, London.
2) Laurent Romary (2000): TMF – Terminological Markup Framework, Laboratoire LORIA, (CNRS, INRIA, Univerza v Nancyju, ISO meeting, London, 2000)
http://www.loria.fr/projets/TMF/DOC/SLIDES/TMF-ISO_pres.ppt
3) Nancy Ide, Laurent Romary: A Common Framework for Syntactic Annotation
http://acl.ldc.upenn.edu/P/P01/P01-1040.pdf
4) Petr Sgall, Jarmila Panevová, Eva Hajičová (2004): Deep Syntactic Annotation: Tectogrammatical Representation and Beyond, hlt-naacl2004, (Human Language Technology conference / North American chapter of the Association for Computational Linguistics annual meeting).
http://acl.ldc.upenn.edu/hlt-naacl2004/frontiers/pdf/naacl04sph.pdf
5) Ray C. Dougherty (1994): Natural Language Computing, An English Generative Grammar in Prolog, Laurence Erlbaum Associates, Publishers, Hillsdale, New Jersey, Howe, UK.
6) Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R. (2004): Massive multilingual corpus compilation: Acquis Communautaire and totale. In Proc. of the Second Language Technology Conference. April 2004, Poznan.
7) Miran Željko (2002): Pripomočki na spletu za prevajalce zakonodaje EU. Zbornik mednarodne konference Informacijska družba 2002 – jezikovne tehnologije. Ljubljana, oktober 2002.
http://nl.ijs.si/isjt02/zbornik/sdjt02-05zeljko.pdf
8) Miran Željko, Adriana Krstič (2002): Web-based Trados Databases – an Alternative Approach. Kongres Mednarodne zveze prevajalcev. Vancouver, Kanada, avgust 2002.
9) Darja Erbič, Adriana Krstič Sedej, Jasna Belc, Nataša Zaviršek - Žorž, Nevenka Gajšek, Miran Željko) (2005): Slovenščina na spletu v dokumentih slovenske različice pravnega reda Evropske unije, terminološki zbirki in korpusu. Simpozij Obdobja 24: Razvoj slovenskega strokovnega jezika, Ljubljana, november 2005.
10) Jasna Belc (2002): Konferenci ob rob: Sodelovanje na področju terminologije in drugih sorodnih disciplin, zlasti jezikovnih tehnologij. Zbornik prispevkov s simpozija Terminologija v času globalizacije. Ljubljana, 5. in 6. junij 2003, str. 361–365.
11) Miran Željko (2003): Evroterm in Evrokorpus – terminološki slovar in korpus prevodov. Zbornik prispevkov s simpozija Terminologija v času globalizacije. Ljubljana, 5. in 6. junij 2003, str. 139–149.
12) Trados Manual for MultiTerm and Translator's Workbench, v. 7.0 (2005).
13) Tomaž Erjavec (2005): Foundational Course: Annotation of Language Resources: XML, TEI, OWL:od http://nl.ijs.si/et/teach/esslli05/esslli05-1.html
do http://nl.ijs.si/et/teach/esslli05/esslli05-5.html.
14) James Pustejovsky (2005): Type Selection and the Semantics of Local Context, Lectures at ESSLLI 2005, Edinburgh: http://www.macs.hw.ac.uk/esslli05/giveabs.php?30
15) Glyn Morrill (1994): Type Logical Grammar: Categorial Logic of Signs, Kluwer Academic, Dordrecht.

Korpus govorjene slovenščine

Jana Zemljarič Miklavčič

Center za slovenščino kot drugi/tuji jezik, Filozofska fakulteta, Univerza v Ljubljani
Aškerčeva 2, 1000 Ljubljana

Povzetek

Učni korpus govorjene slovenščine je bil zgrajen kot teoretična in praktična podlaga za gradnjo večjega govornega korpusa slovenščine, ki naj bi dopolnjeval korpus pisnih besedil FidaPlus. Učni korpus sestavljajo digitalni posnetki spontanega govora, ki so bili zbrani po različnih taksonomskih in demografskih kriterijih. Posnetki so bili transkribirani v razširjeni ortografski transkripciji. Med transkribiranjem so bila določena načela transkribiranja in označevanja. Učni korpus je v obliki, ki omogoča iskanje po korpusu, z geslom dostopen na spletnem naslovu http://torvald.aksis.uib.no/talem/jana/s9.html; dostopna so cela besedila ter konkordance in kolokacije posameznih besed. Za iskanje po korpusu so na voljo različni kriteriji, transkripcije pa so povezane z zvočnimi posnetki.

Spoken Corpus of Slovene

A pilot corpus of spoken Slovene has been compiled to establish a theoretical and empirical foundation for building a large spoken corpus of Slovene, which is planned to complement the written FidaPLUS corpus. Pilot corpus is based on digital recordings of spontaneous speech, collected according to different contextual and demographic criteria. The recordings have been transcribed in enriched orthographic transcription. During actual transcription work, the transcription and annotation standards have been outlined. Pilot spoken corpus is available in searchable form at http://torvald.aksis.uib.no/talem/jana/s9.html (authorization needed); the whole texts are accessible, as well as concordances and collocations of single words. Different criteria could be used for searching the corpus. In each case, transcriptions are linked to sound files.

1. Uvod

V času mojega trimesečnega študijskega bivanja na Oddelku za kulturo, jezik in jezikovne tehnologije[1] na Univerzi v Bergnu na Norveškem je nastal manjši korpus govorjene slovenščine, ki naj bi služil kot učni korpus za gradnjo govorne komponente referenčnega korpusa. Namen gradnje učnega korpusa je bil spoznati metode zbiranja, shranjevanja in dokumentiranja govorjenih besedil, razviti in testirati načela transkribiranja, določiti in testirati korpusne oznake ter pokazati nekatere možnosti za uporabo govornega korpusa.

2. Korpusi govorjenega jezika

Korpusi govorjenega jezika so računalniške zbirke transkribiranih posnetkov spontanega govora; razlikujemo jih od korpusov govora, kjer gre običajno za studijske posnetke izoliranih izjav, ki nastajajo za potrebe fonetično-fonoloških raziskav in govornih tehnologij (Gorjanc 2005: 8). Korpusi govorjenega jezika so izrednega pomena za raziskovanje jezika, predvsem njegovih slovnično-leksikalnih lastnosti: uporabljajo se za jezikovne opise, za preverjanje hipotez o jeziku, kot jezikovni vir pri poučevanju in učenju tujega jezika, pa tudi pri raziskavah, ki zadevajo sintezo in razpoznavanje govora (Verdonik 2006: 7). Govorni korpusi so v nasprotju z realno jezikovno produkcijo mnogo manjši od pisnih, ker jih je izredno težko graditi; največji doslej zgrajeni govorni korpusi so govorna komponenta BNC, govorna komponenta BoE (obe velikosti okrog 10 milijonov besed) in Nizozemski govorni korpus (8,3 milijona besed), ki je najmlajši izmed njih (2004) in ima edini transkripcije povezane z zvočnimi posnetki.[2]

V ortografaskem prepisu govora se izgubijo številne značilnosti originalnega govornega dogodka. Izgubo teh informacij je mogoče vsaj deloma nadomestiti s prozdičnimi oznakami in fonetično transkripcijo, pomembna pa je tudi sočasna dostopnost zvočnih posnetkov. Stopnja transkripcije je odvisna od namembnosti korpusa; govorni korpusi so zaradi velikanske količine gradiva običajno transkribirani v ortografski transkripciji, na manjšem delu korpusa pa je lahko narejena tudi fonetična transkripcija.

3. Korpus govorjene slovenščine

3.1. Zajem besedil v korpus

Vprašanje reprezentativnosti korpusa se zastavlja na začetku vsakega razmišljanja o gradnji korpusa in je podlaga za vse kasnejše postopke pri uresničevanju gradnje. Če želimo iz korpusa dobiti relevantne podatke o jeziku, mora korpus kot vzorec izkazovati podobne lastnosti kot govorjeni jezik celotne populacije, ki jo želi predstavljati. Reprezentativnost govornih korpusov se običajno dosega s kombinacijo demografske in besedilnovrstne metode zbiranja (Crowdy 1993: 260); pri prvi izberemo reprezentativni vzorec govorcev celotne populacije (glede na demografske lastnosti, kot so spol, starost, izobrazba, regijski izvor), pri drugi pa besedila zajemamo na podlagi besedilnovrstne taksonomije govorjenih besedil.

Pri gradnji učnega korpusa zaradi časovnih in "človeških" omejitev ni bilo mogoče v celoti upoštevati načel za reprezentativni in uravnoteženi zajem besedil. Kljub temu sem se pri zbiranju gradiva trudila, da bi se besedila čim bolj razlikovala po lastnostih, na katerih je zgrajena taksonomija govorjenih besedil, in da so se govorci čim bolj razlikovali po demografskih lastnostih. Korpus sestavlja 7 različno dolgih posnetkov v skupni dolžini 89 minut (15.000 besed), v njem pa je sodelovalo 20 govorcev; popis demografskih lastnosti govorcev je podan v Tabeli 1.

[1] http://www.aksis.uib.no/

[2] http://lands.let.kun.nl/cgn/doc_English/topics/project/pro_info.htm#sample1

Kriterij	Porazdelitev glede na določene kategorije						?
Spol	ženske: 11			moški: 9			–
Starost	30 let ali manj: 8			30 let ali več: 11			1
Regija	LjO: 13	SZ: 0	SV: 3	Z: 0	J in JZ: 2	drugo: 1	1
Izobrazba	končana OŠ: 3		končana SŠ: 4		končana univ.: 12		–
Prvi jezik	slovenski: 19			drugo: 1			–

Tabela 1: Demografske lastnosti govorcev učnega korpusa

Razmerja znotraj posameznih kontekstualnih kriterijev so naslednja:
- dialogi (in multilogi) proti monologom: 94 % : 6 %,
- zasebna besedila proti javnim: 19,5 % : 80,5 %,
- neformalna besedila proti formalnim: 35,5 % : 64,5 %,
- besedila, posneta v osebnem stiku, proti besedilom s prenosnikom: 31 % : 69 %.

Velikost in struktura korpusa sta zadoščali za določitev korpusnih oznak, zadovoljivi pa sta bili tudi za določitev transkripcijskih standardov. Žal pa se pri tej velikosti ni bilo mogoče niti približati reprezentativnosti govornega korpusa; tega dejstva ne bi bistveno spremenila niti dvakratna ali celo trikratna količina gradiva. Glede na izkušnje pri gradnjah drugih govornih korpusov predvidevam, da bi bilo za slovenščino smiselno načrtovati govorni korpus velikosti 1 milijon besed, to pa je po mojem mnenju tudi največ, kar je v doglednem času mogoče doseči.

3.2. Transkribiranje učnega korpusa

Besedila učnega korpusa govorjene slovenščine so členjena na izjave, ki so omejene bodisi s premorom bodisi z menjavo govorcev. Izjave so transkribirane po priporočilih EAGLES (19996) v ortografski transkripciji brez ločil in brez velikih začetnic na začetku povedi. Velike začetnice imajo lastna imena, kadar ne gre za osebne podatke; ti so nadomeščeni z nevtralnimi oznakami. Besede, ki imajo znano (slovarsko) pisno obliko, so skoraj dosledno zapisane z upoštevanjem pisne norme; pri tem sem se zgledovala po drugih referenčnih govornih korpusih[3] ter sledila osnovni filozofiji delovne skupine EAGLES za govorjena besedila, ki temelji na načelu, naj bo pri transkribiranju spontanih govorjenih besedil v čim večji meri upoštevan standardni zapis besed, vse nestandardne oblike v transkripciji pa naj bodo jasno označene.[4] Zapisovanje govora z ortografsko transkripcijo postane problematično, kadar besede nimajo ustaljene pisne oblike ali od nje v govoru zelo odstopajo. Pri zapisovanju teh besed je treba iskati rešitve, ki morajo biti abstrahirane in transparentne. V nadaljevanju navajam nekaj primerov besed brez ustaljene pisne oblike (v kurzivu):

- *ene* tri ure smo čakali,
- a veš, *un* Michael, *uni* fjordi, *un* Grega,
- vsi *ta* glavni fjordi,
- a *pol* gremo,
- a *čmo* pogledat,
- to je *tle*, to je *tlele*,
- *ə* (polglasnik),
- citatne besede (*imamo mi posla i bez toga)*,
- narečne/žargonske/slengovske besede (*pležuh, tošel, bičikleta, jabčki).*

Sicer pa so bile za označevanje lastnosti govora in za opis nejezikovnih dogodkov v učnem korpusu uporabljene oznake podane v Tabeli 2

Oznaka	Pomen
<pavza>	premor
<ime>	nadomešča osebno lastno ime
[besedilo]	prekrivni govor
<neraz>	nerazumljivi govor
<?>*besedilo*</?>	nezanesljiva transkripcija
<repeat>	ponavljanje
=	napačni začetek
<tj:>*besedilo*</tj>	besedilo, izgovorjeno v tujem jeziku
<nst>*besedilo*</nst>	beseda brez ustaljene pisne oblike[5]
<nv>smeh</nv>	neverbalni dogodki
 besedilo</br>	brano besedilo
(*opis*)	neverbalni zvoki v ozadju

Tabela 2: Oznake učnega korpusa

Za transkribiranje učnega korpusa sta bili uporabljeni dve transkripcijski orodji, Transcriber in Praat; obe orodji ob nadaljnjem procesiranju omogočata neposredno povezavo transkripcij in zvočnih posnetkov. V primeru gradnje večjega govornega korpusa bi bilo verjetno bolje uporabljati Praat, ker omogoča tudi akustične analize govora.

3.3. Konvertiranje učnega korpusa

Transkripcije je v korpus s konkordančnikom in s povezavo med transkripcijami in zvočnimi posnetki konvertiral Knut Hofland na Univerzi v Bergnu. Kot vidimo v Sliki 1, obe transkripcijski orodji, Praat in Transcriber, vsaki transkribirani izjavi z veliko natančnostjo pripišeta začetno in končno časovno oznako.

[3] Npr. govorni komponenti BNC in BoE, Nizozemskem govornem korpusu itd.

[4] http://www.ilc.cnr.it/EAGLES96/spokentx/node24.html#csor

[5] Ta oznaka je bila v učnem korpusu večkrat po nepotrebnem uporabljena; žal na oznake korpusa, ki je na strežniku Univerze v Bergnu, po izteku štipendije ne morem več vplivati.

```
<Turn speaker="spk2" startTime="325.991"
endTime="331.468">
<Sync time="325.991"/>
Ru&lt;lz&gt; Rupel je ne vem on je tak
svetovljan on je brihten človek
</Turn>
<Turn speaker="spk1 spk2"
startTime="331.468" endTime="334.563">
<Sync time="331.468"/>
<Who nb="1"/>
&lt;nst&gt;kurca&lt;/ nst&gt; je [brihten
&lt;neraz&gt;]
<Who nb="2"/>
[je je&lt;repeat/&gt; je je&lt;repeat/&gt;]
Rupel je hud veš
</Turn>
<Turn speaker="spk1" startTime="334.563"
endTime="347.261">
<Sync time="334.563"/>
takrat ko so bili Pankrti je zapisal v eno
revijo
<Sync time="337.567"/>
```

Slika 1. Izjave s pripisanim začetnim in končnim časom

Program razdeli časovni odsek med začetkom in koncem izjave s številom besed in naredi interpolacijo časa za vsako besedo znotraj izjave; na ta način je dosežena dokaj natančna sinhronizacija zvoka in transkripcije. Razumljivo je, da pri tem prihaja tudi do zamikov, zato pri poslušanju izjav v konkordancah ne slišimo vedno samo tistega, kar bi želeli.

Iskalna platforma korpusa omogoča enostavno iskanje po demografskih in/ali kontekstualnih kriterijih, ki so bili predvideni ob načrtovanju korpusa: po spolu govorca, izobrazbi, regiji, prvem jeziku, odnosu med govorci, pa tudi po nekaterih kontekstualnih kriterijih (skrivaj posneta besedila, tip in struktura besedila, okoliščine in prenosnik).[6]

Iskalno okno omogoča iskanje ene, dveh ali treh sosednjih besed in izpis njihovih konkordanc. Iščemo lahko cele besede, lahko pa le začetne ali končne dele besed. Poleg tega okno omogoča tudi prilagajanje sobesedila v konkordančnem izpisu in spreminjanje dolžine predvajanega zvočnega posnetka levo in desno od besed(e) v konkordanci.

4. Iskanje po korpusu

Besedila UKGS so dostopna na dva načina: kot celote ali preko konkordančnika. Dostopnost besedil, ki jih je mogoče v celoti ali po delih tudi poslušati, je za nekatere jezikoslovne analize zelo pomembna. Slika 2 prikazuje primer transkribiranega in označenega besedila, segmentiranega na izjave.

Običajnejši dostop do gradiva v korpusu je preko konkordančnika, kjer kot rezultat iskanja dobimo konkordančni niz: primer vidimo v Sliki 3. Na začetku vsake vrstice je šifra posnetka, iz katerega je vzeta izjava (npr. R06), in šifra govorca izjave (G17). S klikom na šifro prikličemo glavo besedila, kjer lahko preberemo podatke o okoliščinah nastanka besedila in demografske podatke o govorcu.

L	G17:	[ja {neraz} Mobi je izgubil zdaj]
	G16:	[{neraz} zgubil] je to kar so že podpisali ne
L	G16:	in potem predsednik ta ə Vege [ne]
	G17:	[ja]
L	G16:	əm predsednik uprave jaz ne vem kdo {neraz} saj {nst} nima veze {/nst}
L		əm Američan je skratka ne pač ə
L		se pizdi ne kako lahko zdaj Mobitel əm se zoperstavi Združenim narodom ne
L	G17:	{shift=vpr} a [res] {/shift=vpr}
	G16:	[Američan] a veš pizda
L	G16:	ne [oni ki imajo] {pavza} [ne {neraz}] potem pa oni ves zgrožen {shift=vpr} kako lahko {/shift=vpr} ne
	G17:	[ja ja zastopim] ki se itak ne šmirglajo [Združenih narodov]
L	G17:	[{nv} smeh {/nv}]

Slika 2. Del transkribiranega besedila

R04--G15	imam celo <nst> kle </nst>	**ne vem**	+G03+ <pavza>
R05--G11	to so pa tako široke da	**ne vem**	+G11+ kot da bi
R05--G11	[pobiramo te besede] ne pa	**ne vem**	+G12+ [ja ja]
R05--G11	podvozja stanejo okoli	**ne vem**	+G12+ [pet
R07--G20	pa pa <repet/> sem pozabil	**ne vem**	+G19+ ja ja
R06--G16	ima ąe ali kaj saj	**ne vem**	[meni se zdi ja da
R03--G08	najhujša +G08+ najlepša	**ne vem**	auv= vedno
R02--G02	in in in <repet/>	**ne vem**	besede in
R02--G03	əm mogoče bomo naredili	**ne vem**	dve ali tri poletne
R02--G02	recimo ljudje ki so	**ne vem**	dve uri na na
R06--G17	</okr> ali kje l'e saj	**ne vem**	ja ə če bi
R02--G02	tudi druge oblike recimo	**ne vem**	glede seminarja
R06--G17	ə </shift=vpr> pač <pavza>	**ne vem**	jaz zdaj teh izraz
R06--G16	a veš +G17+ [<neraz>	**ne vem**	kako jim to
R06--G17	<nst> pizda </nst> to pa	**ne vem**	ja čisto so
R05--G11	[<nv> smeh </nv>] +G11+	**ne vem**	kaj ąe ja to je to

Slika 3. Konkordančni niz in kolokacija besed "ne vem"

Med najbolj dragocene podatke v referenčnih korpusih sodijo podatki o frekvenci pojavljanja in sopojavljanja besed. Nekateri statistični podatki so dostopni tudi za učni korpus: v korpusu je okrog 15.000 pojavnic in 3118 različnic (o pojavnicah in različnicah prim. Gorjanc 2005); od tega se jih dve tretjini pojavi samo enkrat, kar naj bi bilo za govorne korpuse značilno. Najvišjo frekvenco v učnem korpusu ima oblika glagola biti "je", in sicer skoraj 500. Slika 4 prikazuje besede z najvišjo frekvenco pojavljanja v učnem korpusu.

1	498	35.422	je
2	425	30.230	ne
3	358	25.464	ə
4	313	22.263	pa
5	297	21.125	in
6	284	20.201	se
7	270	19.205	da
8	268	19.063	to
9	265	18.849	ja
10	264	18.778	v

[6] Oznake kriterijev so v angleščini, ki je bila sporazumevalni jezik med menoj in Knutom Hofmanom na Univerzi v Bergnu.

11	186	13.230	na
12	143	10.171	tudi
13	130	9.247	za
14	115	8.180	ki
15	106	7.540	so
16	105	7.469	tako
17	105	7.469	mhm
18	98	6.971	kaj
19	88	6.259	a
20	86	6.117	še
21	84	5.975	če
22	78	5.548	zdaj

Slika 4. Seznam besed z najvišjo frekvenco v učnem korpusu

V prvem stolpcu seznama je zaporedna številka (glede na frekvanco pojavljanja), v drugem stolpcu absolutna frekvenca (število pojavitev v korpusu), v tretjem stolpcu pa relativna frekvenca (število pojavitev na 1000 besed).

5. Zaključek

Učni korpus govorjene slovenščine je zaenkrat z geslom dostopen na spletnem naslovu http://torvald.aksis.uib.no/talem/jana/s9.html (geslo je mogoče dobiti na naslovu jana.zemljaric@ff.uni-lj.si). Največji pomanjkljivosti korpusa sta njegova neuravnoteženost in nereprezentativnost, kar je treba upoštevati pri morebitni nadaljnji uporabi korpusnih podatkov. Namen gradnje učnega korpusa je bil vendarle dosežen, saj so bila ob gradnji razvita načela zajemanja govorjenih besedil, predstavljene so bile različne možnosti transkribiranja spontanega govora, izbrane in preizkušene možnosti označevanja, prikazati pa je mogoče tudi nekatere možnosti uporabe govornega korpusa. Za nekatere oznake in načela korpusa so bili že podani predlogi za izboljšave, ki bodo upoštevani ob morebitni gradnji večjega govornega korpusa.

6. Literatura

Burnard, L., 2000. *Where did we go wrong? A retrospective look at the design of the BNC*. SILFI 6th International Conference, "Spoken Italian", Congress Proccedings. Duisburg, 28. 6.–2. 7. 2000. <http://users.ox.ac.uk/~lou/wip/silfitalk.html>

Crowdy, S., 1993. Spoken Corpus Design. *Literary and Linguistics Computing 8/4*. Oxford University Press. 259–265.

Crowdy, S., 1994. Spoken Corpus Transcription. *Literary and Linguistics Computing 9/1*. Oxford University Press. 25–28.

EAGLES preliminary recommendations on Spoken Texts, 1996. EAGLES (Expert Advisory Group on Language Engineering Standards) Spoken Language Working Group. <http://www.ilc.cnr.it/EAGLES96/spokentx/spokentx.html>

Gorjanc, V., 2005. *Uvod v korpusno jezikoslovje*. Domžale, Izolit.

Leech, G., G. Myers in J. Thomas (ur.), 1995. *Spoken English on Computer. Transcription, mark-up and application*. New York: Longman Publishing.

Llisteri, J., 1996. *Preliminary recommendations on Spoken Texts*. EAGLES (Expert Advisory Group on Language Engineering Standards). Version of May, 1996. http://www.ilc.pi.cnr.it/EAGLES96/spokentx/spokentx.html

Stabej, M., in P. Vitez, 2000. KGB (korpus govorjenih besedil) v slovenščini. V: ERJAVEC in GROS (ur.), *Jezikovne tehnologije za slovenski jezik*. Ljubljana: 79–81.

The spoken Dutsch corpus project, 2004. <http://lands.let.kun.nl/cgn/doc_English/topics/project/pro_info.htm>

Verdonik, D., 2006: *Analiza diskurza kot podpora sistemom strojnega simultanega prevajanja govora.* Doktorska disertacija. Mentor Marko Stabej. Univerza v Ljubljani, Filozofska fakulteta, Oddelek za slovenistiko.

Zemljarič Miklavčič, J., 2004. Taksonomija besedilnih tipov za gradnjo govornega korpusa. V E. Kržišnik (ur.): *Aktualizacija jezikovnozvrstne teorije na slovenskem: Členitev jezikovne resničnosti. Obdobja 22*. Ljubljana: Center za slovenščino kot drugi/tuji jezik, Filozofska fakulteta Univerze v Ljubljani.

Iskanje pragmatičnih enot v neoznačenem korpusu: primer kažipotov

Agnes Pisanski Peterlin

Oddelek za prevajalstvo, Filozofska fakulteta, Univerza v Ljubljani
Aškerčeva 2, 1000 Ljubljana
agnes.pisanski@guest.arnes.si

Povzetek

V prispevku je predstavljen poskus uporabe korpusne analize za iskanje pragmatičnih enot v neoznačenem specializiranem korpusu 46 slovenskih poljudnoznanstvenih člankov; v korpusu so z elektronskim iskanjem identificirani kažipoti, ki so vrsta metabesedila. Iskanje generira 2782 zadetkov, od katerih jih je 8,1 % identificiranih kot kažipoti. Nadaljnje ročno iskanje identificira dodatne primere kažipotov; pokaže se, da je elektronsko iskanje uspešno identificiralo nekaj več kot 80 % vseh kažipotov. Predstavljene so možnosti izboljšav v iskanju, kot npr. redukcija seznama iskanih izrazov na tiste, ki so bili pred tem najdeni v poljudnoznanstvenih člankih, iskanje nizov besed namesto posameznih besed in razširitev seznama iskanih besed.

Search for Pragmatic Units in an Untagged Corpus: The Signpost Case

The paper presents an attempt to use corpus analysis to search for pragmatic units in an untagged specialized corpus of 46 Slovene popular science articles; the corpus is searched electronically for signposts, a type of metatext. The search yields 2782 hits of which 8,1% are identified as signposts. A subsequent manual search process identifies additional instances of signposting, revealing that the electronic search has successfully identified just over 80% of all the signposts. Possibilities for improving the search, such as reducing the list of search words to those previously found in popular science articles, using word combinations instead of single words and expanding the list of search words are discussed.

1 Uvod

Čeprav se je korpusni pristop sprva uporabljal zlasti v analizah leksiko-gramatikalne tematike, se je v zadnjih desetih letih začel razširjati tudi na področje besediloslovnih raziskav. Te so zastavljene na različne načine: Gorjanc (2005: 71) navaja, da je bilo v zadnjem času veliko energije vložene v besediloslovno označevanje, pri čemer posebej izpostavlja področje slovnične kohezije. Poleg slovnične kohezije pa se besediloslovne raziskave pogosto lotevajo tudi drugih vrst problematike, pri čemer pa enotnega in sistematičnega pristopa ni.

V nekaterih primerih se avtorji raziskav odločijo sami označiti korpus, ki so ga zbrali za raziskavo, z besediloslovnimi oznakami, ki so za njihovo raziskavo uporabne; Upton in Connor (2001) npr. ročno označita besedilo po retoričnih korakih. Po tako označenem korpusu je iskanje enostavno in učinkovito, vendar so takšen korpus in oznake običajno manj uporabne za druge raziskave. Nekateri raziskovalci poskušajo uporabljati korpusne oznake, ki označujejo nekoliko bolj splošne besediloslovne kategorije. Eden bolj posrečenih poskusov v tej smeri je RST, Teorija retorične strukture (angl. Rhetorical Structure Theory), ki sta jo osnovala Mann in Thompson (1988) in v okviru katere so nastale razmeroma splošne oznake za strukturo besedila. Primere korpusov označenih po načelih RST lahko najdemo na spletni strani RST, http://www.sfu.ca/rst. Zanimiva primera raziskav v korpusih označenih po načelih RST predstavljata Taboada (2006), ki analizira rabo diskurznih označevalcev, in Burstein in Marcu (2003), ki primerjata dva sistema avtomatične diskurzne analize, s katero identificirata tezne in zaključne stavke v študentskih esejih.

Poleg tovrstnega pristopa, ki predpostavlja, da je za kvalitetno besediloslovno korpusno raziskavo najkoristneje posebej označevati besedilne korpuse za izbrane besediloslovne kategorije, pa so danes mnoge besediloslovne raziskave, zlasti tiste, ki se osredotočajo na strokovni jezik, zasnovane na drugačnem pristopu: avtorji za potrebe svoje raziskave pripravijo lasten specializiran korpus, ki je največkrat neoznačen, nato pa si pri analizi pomagajo z obstoječimi programi za besedilno analizo in konkordance (prim. Flowerdew, 2002: 96, za opis stanja na področju angleščine v znanosti). V tem prispevku predstavljam primer tovrstne raziskave in ob dobljenih rezultatih opozarjam na nekatere težave, ki se v zvezi s takšnim iskanjem pojavljajo.

2 Kažipoti

Kažipoti so elementi, s katerimi tvorec besedila bralcu oziroma poslušalcu napoveduje vsebino besedila, ki sledi, ali pa se sklicuje na tisto, kar je bilo že povedano v istem besedilu, npr. *v nadaljevanju bomo pokazali*, *da*.... ali pa *kot smo že omenili*.... Kažipote uvrščamo med metabesedilne elemente; to so deli besedila, ki bralcu ali poslušalcu služijo za orientacijo po besedilu oziroma nakazujejo odnos avtorja do sprejemnika besedila, k vsebini besedila pa ne prispevajo.

2.1 Zakaj so pomembni?

Poleg tega, da so raziskovalcu retorične strukture zanimivi s teoretičnega vidika, kot vrsta metabesedila, ki predstavlja nadvse kompleksno, problematično in heterogeno kategorijo, so zanimivi tudi iz različnih praktičnih razlogov. Raziskave so pokazale, da se jeziki, vede in žanri v rabi metabesedilnih elementov med seboj razlikujejo (npr. Bäcklund, 1998, Dahl 2004); to je bilo pokazano tudi za slovenščino v primerjavi z angleščino in za različne žanre in stroke (Pisanski Peterlin, 2005). Glede na to, da se tovrstnih razlik pravzaprav slabo zavedamo, so možnosti napak pri oblikovanju besedila v tujem jeziku ali v žanru, ki piscu ni domač, precejšnje.

2.2 Zakaj so problematični za elektronsko iskanje?

Kot številne druge besediloslovne kategorije so kažipoti problematični, ker niso formalna, temveč pragmatična kategorija: identificiramo jih po njihovi vlogu v besedilu in ne po njihovi obliki. Za pravilno interpretacijo metabesedilnosti je kontekst ključnega pomena. Čeprav se pogosto pojavljajo v podobnih ali enakih oblikah (npr. *omenjeni primeri, pokazali bomo, kot že rečeno,...*), je včasih njihova oblika povsem netipična (npr. *presenečenj pa še ni konec*). Prav tako pojavitev tipične oblike še ne pomeni, da ta označuje kažipot. Besedna zveza *opisane vrste* je kažipot, če se nanaša na vrste, ki jih je avtor besedila predhodno opisal v istem besedilu (v smislu *zgoraj/prej opisane vrste*), če pa se nanaša na vrste, ki so jih različni raziskovalci opisali do sedaj (v nasprotju s takimi, ki jih še niso opisali), pa to ni kažipot, temveč del vsebine besedila.

A kljub temu, da metabesedilo ni formalna kategorije, se avtorji, ki metabesedilne elemente raziskujejo v angleščini (Hyland, 2004) in v drugih jezikih (npr. Dahl, 2004 v norveščini in francoščini), vse pogosteje odločajo za elektronsko iskanje po posameznih tipičnih izrazih, saj le tako lahko raziskujejo rabo v velikih vzorcih in so njihove ugotovitve lahko statistično veljavne. Za iskanje po korpusu nekateri avtorji uporabljajo različne obstoječe programe za besedilno analizo in konkordance, kot so WordPilot 2000 (uporabljen v Hyland, 2005 in Harwood, 2005), WordSmith Tools (uporabljen v Hewings, M. in Hewings, A., 2002), MonoConcPro (uporabljen v Hyland, 2004), drugi pa za iskanje uporabljajo program, narejen "po meri" prav za iskanje po njihovem korpusu (npr. Dahl, 2004).

Ob tem se odpira zanimivo vprašanje, kakšne so specifične značilnosti takega iskanja v slovenščini, ki je v nasprotju z angleščino morfološko mnogo bogatejša, kako uspešno je tovrstno iskanje in s katerimi težavami se pri njem srečujemo.

3 Korpus in metoda

3.1 Korpus

Analizirani korpus je bil sestavljen za potrebe te raziskave in obsega 46 slovenskih besedil. Vsa besedila so izvirni slovenski poljudnoznanstveni članki, objavljeni v reviji *Proteus* med letoma 1997 in 1999. Skupna dolžina korpusa je 341 486 besed.

3.2 Metoda

V okviru predpriprave na analizo sta bila sestavljena dva seznama iskanih izrazov. Seznama sta nastala na podlagi predhodne raziskave (Pisanski Peterlin, 2005), ki je med drugim obsegala ročno analizo 70 besedil in identifikacijo vseh pojavitev kažipotov v njih. Od omenjenih besedil je bilo 35 besedil slovenskih, v katerih je bilo identificiranih 223 kažipotov. Iz tega nabora kažipotov so bili v najprej izbrani vsi glagoli, razen tistih, ki so bili ocenjeni kot presplošni, da bi bili uporabni za iskanje (npr. *biti* v *namen je*). Izdelava seznama glagolov se zdi smiselna, kajti velika večina kažipotov vsebuje glagolsko obliko. Tako dobljeni seznam obsega 66 glagolov, to pa so:

analizirati, dokazati, govoriti, ilustrirati, interpretirati, izogniti, izpeljati, izračunati, končati, lotiti, narediti, navesti, obdelati, obravnavati, ogledati, omejiti, omeniti, omenjati, opazovati, opisati, opozoriti, opraviti, opredeliti, opustiti, orisati, podpirati, pogledati, poiskati, pojasniti, pokazati, poskusiti, posvetiti, potrebovati, poudariti, povedati, povrniti, povzeti, predstaviti, pretresti, preveriti, prevladovati, prezentirati, prihraniti, prikazati, primerjati, pripeljati, privzeti, razgrinjati, razkriti, razlagati, razložiti, reči, seznaniti, skicirati, spoznati, ubrati, ugotoviti, uporabiti, ustaviti, videti, vprašati, vrniti, začenjati, zajemati, zanimati, zapisati.

Ločeno je bil narejen popis vseh glagolskih slovničnih oblik, ki so se pojavljale v funkciji kažipotov (npr. velelnik za prvo osebo množine, kot npr. *poglejmo*, nedoločnik v povezavi z modalnim glagolom, npr. moramo *pogledati*, prva oseba množine v sedanjiku, npr. v članku *pokažemo* itd.); kjer so se pojavljale večbesedne oblike (npr. *bomo pokazali*), je bil v analizo zajet le polnopomenski glagol, torej deležnik *pokazali.* Takih oblike je bilo enajst, in sicer *nedoločnik, velelnik za prvo osebo množine, 1. oseba ednine v sedanjiku, 1. oseba množine v sedanjiku, 3. oseba ednine v sedanjiku, 3. oseba množine v sedanjiku, deležnik na -l, deležnik na -la, deležnik na -lo, deležnik na -li in deležnik na -le*, kar pomeni, da je potrebno za 66 glagolov generirati seznam 726 glagolskih oblik. Morfološko bogata slovenščina je v primerjavi z morfološko revno angleščino nekoliko bolj zahtevna pri predpripravi na analizo, a hkrati omogoča izločitev tistih oblik, ki se nikoli ne pojavljajo kot kažipoti (npr. druga oseba ednine), kar lahko pomeni nekoliko bolj osredotočeno iskanje z manj lažnimi zadetki. Žal v primeru kažipotov glagolskih oblik, ki bi jih lahko v celoti izločili, ni veliko, v primeru drugih metabesedilnih elementov ali drugih funkcijskih kategorij, pa je situacija nekoliko drugačna in je tovrstno sortiranje lahko zelo koristno.

Nato je bil na podlagi že omenjenega nabora kažipotov narejen še seznam drugih izrazov (samostalniki, pridevniki, deležniki na *-n/-t* in prislovi), ki bi utegnili služiti kot potencialni identifikatorji kažipotov. Kriteriji za izbor so bili pri tem seznamu mnogo ožji kot pri glagolih: pregled izvornega nabora kažipotov je pokazal, da je pomen kažipota najpogosteje vsebovan v glagolu, med tem ko večina samostalnikov in spremljevalnih besed nima nujno le metabesedilnega pomena. Vendar pa se tudi med neglagoli pojavljajo tipični deli kažipotov (npr. samostalniki, kot so *članek, namen, razdelek* itd., pridevniki, kot so *zgornji, naslednji, spodnji,...* in prislovi, kot so *najprej, tu, zgoraj*). Deležniki na *-n/-t* so v svojem izvoru seveda glagolske oblike, vendar jih po funkciji že Toporišič (2000) uvršča med pridevniško besedo. Prav zato in zaradi svojih morfoloških značilnosti so zbrani na seznamu neglagolskih oblik. Poleg tega jih. Zaradi razmeroma omejenega števila glagolov, ki so se v izvornem seznamu pojavljali v obliki deležnikov na *-n/-t*, se je zdelo smiselno takšno omejeno število deležnikov tudi obdržati. Seznam neglagolskih izrazov za iskanje kažipotov je tako obsegal naslednje besede:

- samostalnike: *analiza, faza, razprava, članek, način, namen, opis, podatek, postopek, primer, prispevek, razdelek, rezultat,*

sestavek, začetek, zapis, zgled, dejstvo, delo, nadaljevanje, področje, poglavje, vprašanje, pot, rešitev, ugotovitev,

- pridevnike in deležnike na -n/-t *gornji, naslednji, našteti, obravnavani, omenjani, omenjeni, opisani, prejšnji, pričujoči, uvodni, spodnji, zadnji, zgornji*, ter številnik *prvi* (po analogiji z *zadnji*)
- prislove *doslej, kmalu, najprej, nato, nazadnje, pozneje, pravkar, predhodno , prej, sedaj, spodaj, tu, tukaj, uvodoma, zdaj, zgoraj, že*

V nasprotju z glagolskimi oblikami na prvem seznamu na drugem seznamu ni mogoče izločiti morfoloških oblik samostalnikov in pridevnikov (vključno z deležniki na *-n/-t*), saj se v metabesedilni vlogi lahko pojavijo v vseh sklonskih oblikah in v vseh treh številih, pridevniki in deležniki pa tudi v vseh spolih. Tako se npr. samostalnik *članek* v vlogi kažipota sicer izrazito tipično pojavlja v kombinaciji *v+članku*, vendar ni nobenega razloga zakaj se ne bi pojavljal tudi v primerih, kot je npr. *članek predstavlja,... namen članka je..., s tem člankom želim prikazati...* Na enak način kot pri glagolih generiramo seznam neglagolskih oblik, ki obsega 401 obliko.

Analiza korpusa je bila narejena s programskim orodjem za konkordance WordSmith Tools (Scott, 1996), verzija 4.0. Sledila je ročna izločitev vseh zadetkov, ki niso imeli metabesedilnega pomena. Nato so bila vsa besedila v testnem korpusu še enkrat analizirana ročno, da bi identificirali vse primere kažipotov, ki jih elektronsko iskanje ni zajelo, s čemer bi oceni uspešnost uporabljene metode. Nazadnje so bila besedila še enkrat računalniško analizirana s programskim orodjem za konkordance Wordsmith Tools, tokrat z nekoliko modificiranim seznamom izrazov, z namenom izpopolnitve metode iskanja.

4 Rezultati

V **tabeli 1** so predstavljeni rezultati iskanja po obeh seznamih in skupni seštevek: v prvi vrstici je predstavljeno število zadetkov elektronskega iskanja, v drugi vrstici je navedeno število zadetkov, ki so ostali po ročni izločitvi lažnih pozitivnih zadetkov in so bili identificirani kot kažipoti, v tretji vrstici pa je razmerje med pravimi pozitivnimi zadetki in vsemi zadetki elektronskega iskanja izraženo v odstotkih.

	Glagoli	Neglagoli	Skupaj
Vsi zadetki	731	2051	2782
Kažipoti	54	172	226
% vseh zadetkov	7,4 %	8,4 %	8,1 %

Tabela 1: Rezultati elektronskega iskanja kažipotov in ročnega čiščenja dobljenih rezultatov

Iskanje po seznamu glagolov je prineslo 731 zadetkov, iskanje po seznamu drugih izrazov pa 2051 zadetkov. Od tega je bilo med identificiranimi glagolskimi oblikami 54 oziroma 7,4 % takih, ki so bili v resnici v vlogi kažipota, med ostalimi oblikami pa 172 oziroma 8,4 % vseh.

Pomemben je tudi podatek o številu primerov, ki so se podvajali na obeh seznamih; takih primerov je bilo 31. Nekateri primeri so se podvajali znotraj istega seznama, taka sta bila med glagoli 2 primera, med ostalimi pa 36 primerov. Končni seznam vseh kažipotov, ki jih je elektronsko iskanje po korpusu identificiralo, tako obsega 157 primerov.

Rezultati ročne analize so pokazali, da je bilo elektronsko iskanje kažipotov delno uspešno. Ročna analiza je namreč v korpusu identificirala še dodatnih 34 primerov, kar pomeni, da je elektronsko iskanje zajelo nekaj več kot 82 % vseh pojavitev. Skupno število vseh identificiranih kažipotov je 191, kar pomeni, da se v povprečju pojavlja 4,1 kažipot na besedilo, oziroma nekaj manj kot 0,6 na 1000 besed.

5 Diskusija

Pri avtomatskem iskanju kažipotov v korpusu sta se pojavila dva problema: prvi je veliko število lažnih pozitivnih zadetkov (pravih pozitivnih je manj kot 10 %), drugi pa je dejstvo, da z metodo uspešno najdemo le nekaj več kot 80 % kažipotov.

5.1 Lažni zadetki

Velik odstotek lažnih pozitivnih zadetkov v pričujoči raziskavi ni bil posebej problematičen, saj je uporabljeni korpus razmeroma majhen, zato je bilo ročno izločanje lažnih pozitivnih obvladljiva naloga; v večjem korpusu bi bilo to bistveno težje. Hyland (2004: 136–7) vprašanje lažnih pozitivnih zadetkov rešuje s pomočjo statistike: med številnimi zadetki jih naključno izbere 50 in med njimi identificira tiste, ki se pojavljajo v metabesedilni vlogi, nato pa izračuna isto razmerje za celoten korpus. Zdi se, da je takšen pristop primeren za grobo ugotavljanje deleža metabesedilnih elementov v celotnem korpusu, čeprav 50 primerov ni velik vzorec. Vsekakor pa s statistično posplošitvijo ne moremo ugotavljati subtilnih razlik v dveh podobnih korpusih, saj so raziskave pokazale, da so takšne razlike lahko razmeroma majhne. Prav tako podatki pridobljeni s statističnim preračunavanjem niso najbolj primerni za nadaljnje ugotavljanje drugih lastnosti kažipotov, npr. lokacije, referenčnega dosega, referenčne razdalje itd.

5.1.1 Krajšanje seznama

Ena od možnosti za izboljšavo iskanja je sprememba oziroma krajšanje seznama iskanih izrazov. Seznam bi lahko skrajšali v številu slovničnih oblik ali leksikalnih enot. Pregled ustreznosti nabora glagolskih slovničnih oblik pokaže, da od enajstih oblik tri niso generirale pravih pozitivnih zadetkov; te oblike so prva oseba ednine v sedanjiku ter deležnika na *-lo* in *-le*. Izpuščanje deležnikov na *-lo* in *-le* bi bilo nesmiselno, saj sta to le varianti deležnika na *-l*, ki je sicer v oblikah na *-l*, *-la* in *-li* generiral razmeroma velik delež pravih pozitivnih zadetkov in bi se v drugem naboru besedil lahko pojavljale tudi oblike na *-lo* in *-le*. Izpuščanje prve osebe ednine v sedanjiku prav tako ni smiselno: čeprav je očitno, da se v danem korpusu tudi v člankih, kjer je avtor en sam mnogo pogosteje pojavlja prva oseba množine, pa vendarle naletimo tudi primer kažipota v prvi osebi ednine v prihodniku, npr. *bom opisal*.

Pregled ustreznosti nabora leksikalnih enot daje boljše možnosti za krajšanje. Izhodiščna predpostavka pri

izdelavi seznama je bila, da je, glede na to, da so kažipoti pragmatična in ne formalna kategorija, smiselno na seznam uvrstiti čim več različnih izrazov, ki se v tej vlogi lahko pojavljajo. Podroben pregled zadetkov to predpostavko zanika: pokaže se, da veliko večino pravih pozitivnih zadetkov generirajo eni in isti izrazi, drugi pa sistematično nastopajo v nemetabesedilni vlogi.

Eden od možnih razlogov za to, da so se na seznamu pojavili nekateri izrazi, ki so generirali same nemetabesedile zadetke je morda v izhodiščnem naboru kažipotov. Ta je bil narejen na podlagi ročne analize 35 besedil iz treh različnih žanrov: znanstvenega članka, poljudnoznanstvenega članka in univerzitetnega učbenika. Predhodne raziskave so sicer pokazale, da se žanri med seboj močno razlikujejo v rabi metabesedilnih elementov (Crismore in Farnsworth, 1990 za angleščino, Bäcklund, 1998 za angleščino, nemščino in švedščino, in Pisanski Peterlin, 2005 za angleščino in slovenščino), z vprašanjem oblik v posameznih žanrih pa se ni ukvarjala nobena od njih. Primerjava lažnih in pravih zadetkov pokaže, da je številne lažne pozitivne zadetke mogoče pripisati dejstvu, da so bili na izvorni seznam iskanih izrazov uvrščeni iz univerzitetnega učbenika ali iz znanstvenega članka.

Primer izraza, ki generira številne lažne pozitivne zadetke je *delo*. Ta podobno kot *poglavje* izhaja iz kažipotov, ki se pojavljajo v univerzitetnih učbenikih, v poljudnoznanstvenih člankih pa ga pravzaprav ne pričakujemo, čeprav v resnici lahko naletimo na primere, v katerih avtorji *razdelke* v članku poimenujejo *poglavja* (npr. *kot smo pokazali v prejšnjem poglavju*). Podobno je tudi razmeroma neverjetno, čeprav ne nemogoče, da bi avtor članek poimenoval *delo*, (npr. *v tem delu bomo pokazali*). Pregled lažnih pozitivnih zadetkov pa pokaže, da se v veliki večini pojavljajo različne sklonske oblike samostalnika *del*, ki se prekrivajo s samostalnikom *delo* (*dela, delu, del* ipd.), samostalnik *del* pa je v poljudnoznanstvenih člankih razmeroma pogost.

Druga vrsta primerov lažnih pozitivnih zadetkov izhaja iz primerov glagolov, ki se pojavljajo v kažipotih v znanstvenih člankih. Kažipoti kot so *v članku bomo analizirali, zgoraj smo dokazali, v naslednjem razdelku bomo izračunali, izpeljali smo,...* se v naravoslovnih znanstvenih člankih pogosto pojavljajo. Narava poljudnoznanstvenih člankov pa je drugačna: v teh besedilih avtorji tipično ne analizirajo, dokazujejo, izračunavajo ali izpeljujejo itd., temveč predstavljajo, povzemajo, poročajo. Zelo pogosto pa poročajo o raziskavah ali analizah drugih raziskovalcev, npr. *na univerzi v Cambridgeu so to analizirali, dokazali, izračunali, izpeljali,...* Iskanje po izrazih kot so *analizirali, dokazali, izračunali, izpeljali* tako seveda pripelje do številnih lažnih pozitivnih zadetkov.

Kot možna izboljšava seznama se torej ponuja krajšanje seznama na zgolj tiste izraze, ki so bili identificirani v poljudnoznanstvenih člankih, brez dvoma pa to pomeni tudi izgubo nekaterih pravih pozitivnih zadetkov. Novi seznam je precej krajši in obsega 31 izrazov, to pa so:

- glagoli *izračunati, narediti, navesti, omeniti, omenjati, opredeliti, pogledati, povedati, prikazati, razgrinjati, seznaniti, spoznati, ugotoviti, uporabiti, ustaviti,*
- samostalniki *članek, način, področje, razdelek, vprašanje,*
- pridevniki in deležniki *našteti, omenjeni, opisani, prejšnji, uvodni, zgornji,*
- prislovi *najprej, prej, tu, tukaj, zgoraj, že.*

Tudi v tem primeru so za vsak izraz generirane ustrezne slovnične oblike po enakih načelih kot zgoraj.

Ponovna analiza pokaže, da so rezultati takšnega iskanja bolj obvladljivi, vendar manj natančni. Iskanje s skupnim seznamom iskanih besed, ki vsebuje glagole in ostale izraze iz poljudnoznanstvenih člankov, da 990 zadetkov, med njimi je pravih pozitivnih 128, kar je nekaj manj kot 13 %. Med identificiranimi je bilo 29 ponovitev, kar pomeni, da je bilo zares identificiranih 99 kažipotov, torej le dobro polovico vseh kažipotov (191). V resnici torej ne moremo govoriti o posebni izboljšavi.

Toda modifikacija tega seznama ponuja nekaj smernic za nadaljnje izboljšave: med iskanimi besedami izstopa prislov *že*, ki generira 222 zadetkov, med njimi številne kažipote, večina pa je lažnih pozitivnih. Pregled kažipotov pokaže, da se *že* največkrat pojavlja v povezavi z glagoloma *opisati/omeniti* ali z njunimi deležniki, to pa pomeni, da se zadetki večinoma podvajajo. V resnici prislov *že* samostojno prispeva le k identifikaciji dveh zadetkov; če bi ga izpustili iz seznama iskanih besed, bi bili rezultati drugačni: med zadetki bi bilo 16,4 % kažipotov. Tudi pregled izgubljenih zadetkov pokaže zanimivo sliko: z dodatkom samo štirih izrazov (*primer, prvi, naslednji* in *opisati*) z vsemi njihovimi oblikami, bi zajeli še 23 pojavitev, to pa bi pomenilo, da je metoda uspešna skoraj 64 %. Ti rezultati nakazujejo možno smer za izboljšave pri iskanju: nekoliko modificiran seznam, ki bi bil vezan na žanr, bi lahko nekoliko zmanjšal število lažnih pozitivnih zadetkov.

5.1.2 Iskanje nizov besed

Ena od možnosti za izboljšavo iskanja se zdi iskanje po nizih besed, ki se v vlogi kažipotov pogosto pojavljajo skupaj. Dobljeni rezultati delno potrjujejo smiselnost takega iskanja: skoraj 60 % glagolov je uporabljenih v kombinacijah z nekim drugim izrazom z metabesedilnim pomenom, med ostalimi izrazi je ta odstotek nekoliko nižji, a vedno dosega skoraj 40 % K temu dodamo še možnost iskanja nizov besed v kateri ima le ena metabesedilni pomen, druge pa so njene slovnične sopojavnice (npr. predlogi, pomožni glagoli,...). Namesto iskanja za obliko *članku*, bi tako iskali niz *v+članku*, ki zelo pogosto nastopa v metabesedilni vlogi, izognili pa bi se lažnim pozitivnim v smislu *njegovemu članku ne moremo očitati...* Podobno bi dosegli s kombinacijo *bomo+pokazali* namesto *pokazali*: izognili bi se vrsti primerov, kot je *pokazali so...* ipd.

Težave pri takšnem iskanju pa povzroča fleksibilnost besednega vrstnega reda v slovenščini:

V članku bomo pokazali...
V tem članku bomo pokazali...
Pokazali bomo...
Pokazali pa bomo...
V tem prvem članku bomo sedaj pokazali...
Zdaj pa bomo, ne glede na vse že prej izražene pomisleke, pokazali...

Z resno zastavljeno raziskavo najpogostejših besednih kombinacij in sopojavitev bi lahko izdelali zelo uporaben seznam iskalnih nizov, toda ob tem se postavlja vprašanje o smiselnosti takšne študije. Iskanje po besednih nizih

namreč ne more biti dokončna rešitev: nekateri kažipoti ne vsebujejo tipičnih kombinacij besed.

Drugo možnost za zmanjšanje lažnih pozitivnih zadetkov predstavlja izločanje nizov, ki se tipično pojavljajo v nemetabesedilnem pomenu. Dahl (2004: 1816) navaja uporabnost avtomatičnega izključevanja kombinacij besed kot so *they+glagol* ali *he+glagol* (ta v nasprotju z *I/we+glagol* navadno ne nastopa v vlogi kažipotov) v angleščini, francoščini in norveščini. V slovenščini takšno avtomatično izločanje ni mogoče: ker osebka ni treba izraziti: namesto *in his paper, he* ***shows*** se pojavi *v članku* ***pokaže***, oblika *pokaže* pa je seveda lahko kažipot (v smislu *rezultat* ***pokaže*** *naslednja odstopanja*).

5.2 Izgubljeni kažipoti

Čeprav so lažni pozitivni zadetki problematični zaradi zamudnega ročnega čiščenja, pa ne ogrožajo točnosti dobljenega rezultata. Nasprotno izgubljeni kažipoti, torej tisti, ki jih elektronsko iskanje ne identificira, rezultat izkrivljajo. Zanimivo je, da se avtorji, ki opisujejo elektronsko iskanje tistih vrst metabesedilnih elementov, ki so izrazito neformalni in pragmatični (npr. Hyland, 2004, Dahl, 2004) in za katere pričakujemo, da jih elektronsko iskanje ne bo v celoti zajelo, ne ukvarjajo z vprašanjem izpuščenih pojavitev.

Kot rečeno, je bilo v pričujoči raziskavi pri elektronskem iskanju nekaj manj kot 20 % kažipotov izgubljenih. Pregled teh primerov pokaže, zakaj jih elektronske iskanje ni zajelo. Nekateri razlogi za izgubo posameznih primerov so zelo banalni in jih lahko takoj odpravimo (npr. napačen zapis besede, *ommenjena*), vendar je tovrstnih primerov malo. Za del izgubljenih kažipotov ugotovimo, da so tako netipičnih oblik in tako močno vezani na kontekst, da jih ne bi bilo smiselno predvideti za elektronski iskanje. Navajam nekaj primerov:

.... so zelo primerne za ponazoritev

Poleg te podobnosti... je še nekaj, česar ne smemo prezreti.

...prav tako zasluži kratko predstavitev

V nekaterih drugih primerih lahko identificiramo izraze, s katerimi bi lahko kažipot našli tudi elektronsko, vendar so ti zelo pogosti in v veliki večini primerov rabljeni nemetabesedilno, kar pomeni, da jih verjetno ne bi bilo smiselno vključevati med iskalne izraze. Tak je zlasti kazalni zaimek *ta*, ki se običajno navezuje neposredno na predhodni stavek, kar pomeni, da ni pravi kažipot. V nekaterih primerih pa je vendarle uporabljen kot kažipot (npr. *te metode...* v smislu v *prejšnjem odstavku omenjene metode*). Podobna težava se pojavlja s števniki (glavnimi in vrstilnimi): v besedilih so zelo pogosti kot kažipoti pa razmeroma redki. Če torej izključimo netipične kažipote, primerke s kazalnimi zaimkom in s števniki, ugotovimo, da v uporabljenem korpusu približno polovice izgubljenih kažipotov z elektronskim iskanjem ne bi našli.

Drugo polovico izgubljenih kažipotov bi v danem korpusu lahko našli z izboljšanim seznamom iskanih izrazov. (V konkretnem primeru bi dopolnitev z izrazi, kot so velelnik *spomnimo se*, deležniki *opazovan, prikazan, sledeč, naveden, povedano* in prislov *zdaj* uspešnost iskanja približala 90 %). Jasno pa je, da to ne pomeni, da je seznam izboljšan do največje možne mere: šele večje število novih korpusnih raziskav in ročnih preverjanj v istem žanru bi pokazalo, kakšen je optimalen seznam za iskanje, to pa za raziskavo omejenega obsega ne pride v poštev.

6 Sklep

Iskanje kažipotov, metabesedilnih elementov in drugih pragmatičnih enot v elektronskem korpusu omogoča obdelavo mnogo večjih besedilnih zbirk kot ročno iskanje, zato so dobljeni rezultati statistično bolj relevantni. Toda ob njih se postavlja vprašanje veljavnosti: v pričujoči raziskave je bilo pri iskanju izgubljenih skoraj 20 % pojavitev. S tem vprašanjem se sorodne raziskave (npr. Hyland, 2004, Dahl, 2004) ne obremenjujejo. Obenem se pojavi problem številnih lažnih pozitivnih zadetkov: iskanje z daljšim seznamom iskanih izrazov je generiralo seznam primerov ,v katerem več kot 90 % zadetkov ni bilo v vlogi kažipotov. Pregled razlogov za take rezultate privede do nekaterih idej za izboljšave: redukcija seznama besed na tiste, ki so se v prejšnji raziskavi (Pisanski Peterlin, 2005) pojavile v poljudnoznanstvenih besedilih, se izkaže za preveč radikalen poseg, saj se z njim močno zmanjša uspešnost iskanja, vseeno pa nakazuje smer, v kateri bi bilo mogoče delovati za optimizacijo seznama iskanih besed. Iskanje nizov besed namesto posameznih besed bi prav tako generiralo natančnejši seznam, vendar bi povečalo delež izgubljenih kažipotov. Najboljša rešitev se zdi kombinacija vseh treh pristopov: izboljšanje seznama, krajšanje seznama v povezavi z žanrom in uporaba iskanja po nizih besed za avtomatično iskanje pravih pozitivnih zadetkov med vsemi zadetki.

Iskanje v neoznačenem korpusu je v slovenščini bolj zapleteno kot v angleščini: kjer bi v angleščini iskali s seznamom 400 oblik, je potrebno za iskanje v slovenščini izdelati seznam več kot 1000 oblik. Hkrati pa izdelava tega seznama omogoča izpuščanje tistih oblik, za katere je mogoče domnevati, da se ne pojavljajo v kažipotih in je tozadevno lahko nekoliko bolj natančno kot v angleščini. Po drugi strani pa je možnost avtomatičnega izključevanja lažnih pozitivnih zadetkov na podlagi tipičnih kombinacij, kot jo za angleščino, francoščino in norveščino uporablja Dahl (2004: 1816), v slovenščini zelo omejena.

Dobljeni rezultati pokažejo, da je iskanje pragmatičnih enot, kot so kažipoti, mogoče tudi v neoznačenem korpusu v slovenščini z obstoječim programskim orodjem za besedilno analizo in izdelavo konkordanc; pri tem pa sta predpriprave na analizo in obdelave podatkov nekoliko kompleksnejši kot v angleščini. Kot alternativa se seveda ponuja tudi možnost avtomatičnega iskanja v posebej označenem korpusu in s posebej narejenim programom.

7 Literatura

J. Burstein in D. Marcu. 2003. A Machine Learning Approach for Identification of Thesis and Conclusion Statements in Student Essays. *Computers and the Humanities*, 37: 455–467.

I. Bäcklund. 1998. Metatext in professional writing. A contrastive study of English, German and Swedish. *TEFA* 25: 1–42.

A. Crismore, and R. Farnsworth. 1990. Metadiscourse in popular and professional science discourse. V W. Nash

ur. *The Writing Scholar*. (118–136). Newbury Park, CA: Sage.
T. Dahl. 2004. Textual metadiscourse in research articles: A marker of national culture or of academic discipline? *Journal of Pragmatics*, 36: 1807–1825.
V. Gorjanc. 2005. *Uvod v korpusno jezikoslovje*. Domžale: Izolit.
L. Flowerdew. 2002. Corpus-Based Analyses in EAP. V J. Flowerdew, ur. *Academic Discourse*. (95–114) Harlow: Longman.
N. Harwood. 'Nowhere has anyone attempted... In this article I aim to do just that': A corpus-based study of self-promotional I and we in academic writing across four disciplines. *Journal of Pragmatics*, 7: 1207–1231.
M. Hewings in A. Hewings: "It is interesting to note that..." a comparative study of anticipatory 'it' in student and published writing. *English for Specific Purposes*, 21: 367–383.
K. Hyland. 2004. Disciplinary interactions: Metadiscourse in L2 postgraduate writing. *Journal of Second Language Writing*, 13: 133–151.
K. Hyland. 2005. Stance and engagement: a model of interaction in academic discourse. *Discourse Studies*, 7 (2) 173–192.
W. Mann in S. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8 (3): 243-281
A. Pisanski Peterlin. 2005. *Konvencije rabe metabesedilnih elementov*. Doktorska disertacija. Filozofska fakulteta, Ljubljana.
M. Scott. 1996. *WordSmith Tools*. Oxford: Oxford English Software.
M. Taboada. 2006. Discourse markers as signals (or not) of rhetorical relations. *Journal of Pragmatics*, 38: 567–592.
J. Toporišič. 2000. *Slovenska slovnica*. (4. razširjena izdaja.) Maribor: Obzorja.
T. Upton in U. Connor. 2001. Using computerized corpus analysis to investigate the textlinguistic discourse moves of a genre. *English for Specific Purposes* 20, 313–329.

Oblikovanje korpusa usvajanja slovenščine kot tujega jezika

Mojca Stritar*

* Center za slovenščino kot drugi/tuji jezik, Filozofska fakulteta, Univerza v Ljubljani
Kongresni trg 12, 1000 Ljubljana
mojca.stritar@ff.uni-lj.si

Povzetek

Prispevek podaja razmislek o temeljnih vprašanjih načrtovanja korpusa usvajanja slovenščine kot tujega jezika, povezanih z jezikom, prenosnikom, velikostjo, vrsto in tematiko besedil, tvorci ter kontrolnim korpusom. Ustavi se ob različnih vidikih označevanja napak, predlagane pa so tudi rešitve za pilotski korpus usvajanja slovenščine kot tujega jezika.

Slovene learner corpus design

In the article different problems considering the design of Slovene learner corpus are being discussed, such as the language, written and spoken corpora, size, text types, subjects, authors and control corpora. Various aspects of error mark-up are analyzed and finally the basic design of a pilot Slovene learner corpus is being proposed.

1. Uvod

Zaradi svoje teoretične in praktične uporabnosti vedno nujnejši del jezikovnega načrtovanja postajajo korpusi usvajanja tujega jezika, ki predstavljajo jezik, kot ga pišejo ali govorijo tisti, ki niso njegovi rojeni govorci. V pričujočem prispevku bodo pregledane in ovrednotene nekatere rešitve že obstoječih tujih tovrstnih korpusov, hkrati pa bo podan predlog za oblikovanje korpusa usvajanja slovenščine kot tujega[1] jezika (v nadaljevanju KUST).

Pregledala sem dostopne podatke o 26 obstoječih korpusih, sicer pa ves čas nastajajo še novi. Kar 23 od teh korpusov je bilo narejenih s ciljnim jezikom angleščino, katere jezikoslovna in sociolingvistična realnost sta temeljno drugačni od slovenske. Zato so neposredne vzporednice med obstoječimi in bodočim slovenskim korpusom usvajanja nemogoče in je treba njihove rešitve upoštevati kot izhodišče, ne pa kot nujen zgled.

Ključno vprašanje pri snovanju in oblikovanju vsakega korpusa je njegov namen. Korpusi usvajanja tujega jezika so uporabni tako za teoretične raziskave procesa usvajanja in opisovanje vmesnega jezika učečih se kot tudi za različne praktične aplikacije, kot so slovarji, slovnice, učbeniki ali programska orodja. Danes je izdelava slovarjev, pa tudi slovnic, kompromis med podatki rojenih govorcev iz nespecializiranih korpusov, ki pokažejo tipično v ciljnem jeziku, in podatki iz korpusov usvajanja, ki povedo, katere težave so tipične za učeče se (Granger, 1998). Pri izdelavi učbenikov so korpusi vir za realne primere in nudijo gradivo za vaje prepoznavanja in odpravljanja napak (Pravec, 2002). Pregled obstoječih korpusov usvajanja pokaže, da je večinoma pomembna pedagoška uporabnost, predvsem diagnosticiranje ter odpravljanje najpogostejših napak in težav tujih govorcev (Axelsson, 2000; Lin 1999; Uzar 1998), medtem ko nekaj korpusov poleg tega deklarira tudi bolj teoretične cilje (Kennedy, 1998; Shih, 2000; Pravec, 2002; Dagneaux et al. 2001). Ker se raziskovalci slovenščine kot tujega jezika s tem praviloma ukvarjajo tako na abstraktni, teoretični kot tudi na praktični ravni, bi moral KUST nuditi relevantne in uporabniku prijazne informacije za raziskovalne in praktično-aplikativne namene.

2. Jezik

Pri korpusih usvajanja tujega jezika sta pomembna ciljni jezik, torej jezik, "ki se ga nekdo uči z namenom, da bi ga obvladal bodisi kot svoj prvi, drugi ali tuji jezik" (Pirih Svetina 2005), in izhodiščni oziroma prvi jezik, "iz katerega se nekdo uči vse druge ali tuje jezike" (navedeno delo). Velika večina obstoječih korpusov je usmerjena iz enega izhodiščnega jezika v en ciljni jezik, praviloma angleščino (Atwell et al., 2003; Axelsson, 2000; Cheng, Warren, 1999; De Cock et al., 1999; Granger, 2001; Horváth, 2003; Izumi et al., 2004; Kennedy, 1998; Lin, 1999; Pravec, 2002; Shih, 2000; Sugiura, 2000; Tenfjord et al., 2004; Tono, 2003; Uzar, 1998). Tujih govorcev slovenščine ne moremo omejiti na skupino z enim samim prvim jezikom, zato bi se bilo pri KUST-u bolje zgledovati po korpusih manj globalno razširjenih jezikov. Taka sta ASK, korpus usvajanja norveščine kot tujega jezika, ali FRIDA, korpus francoščine. V obeh je izhodiščnih jezikov več in predstavljajo največje skupine tujih govorcev oziroma priseljencev.

3. Prenosnik

Za referenčne korpuse je vprašanje prenosnika bistveno, korpusi usvajanja pa nikoli ne morejo biti reprezentativni za celotno populacijo tujih govorcev z vsemi prvimi jeziki in stopnjami znanja. Zato referenčnost ostaja utopična želja, graditelji pa zaenkrat v glavnem delajo pisne korpuse.

Od 26 pregledanih korpusov je 18 samo pisnih, pet govornih, trije pa imajo govorni in pisni del, pri čemer je pisni vedno večji od govornega.

	Pisni korpusi	**Govorni korpusi**	**Korpusi s pisnim in govornim delom**
Število besed	60.640.000	1.600.000	835.000

Tabela 1: Število besed v korpusih usvajanja tujega jezika glede na prenosnik.

[1] *Tuji jezik* se učimo "v okolju, kjer ta jezik običajno ni v uporabi" (Pirih Svetina 2005), *drugi jezik* pa je nematerni jezik, ki se v govorčevi skupnosti redno uporablja, torej prevladujoči jezik okolja, v katerem živi. Zaradi elegantnejšega poimenovanja uporabljam izraz *korpus usvajanja slovenščine kot tujega jezika* kot krovni pojem za tuji in drugi jezik.

Razlogi za tako velik delež pisnih korpusov so praktični – zajem besedil je že pri pisnih tekstih relativno zamuden, saj jih je treba pretipkati, transkripcija govorjenih tekstov pa bi bila še počasnejša. Za slovenščino še ni referenčnega govornega korpusa, ki bi reševal načelna vprašanja tega tipa, zato bi bil tudi KUST na začetku izključno pisni.

4. Velikost

Prav zaradi pretipkavanja posameznih sestavnih besedil se korpusi usvajanja po velikosti težko primerjajo z nespecializiranimi korpusi.

Korpus	Ciljni jezik	Število besed
HKUST	angleščina	25.000.000
CLC	angleščina	15.000.000
LCLE	angleščina	10.000.000
TELEC	angleščina	3.000.000
ICLE	angleščina	2.000.000
TELC	angleščina	1.300.000
SST	angleščina	1.000.000
USE	angleščina	1.000.000
CEJL	angleščina	1.000.000
CLEC	angleščina	1.000.000
Taiwanese Corpus	angleščina	730.000
HKCCE	angleščina	500.000
PELCRA	angleščina	500.000
ASK	norveščina	500.000
JPU	angleščina	400.000
POLY U	angleščina	400.000
JEFLL	angleščina	250.000
FRIDA	francoščina	200.000
LINDSEI	angleščina	100.000
MELD	angleščina	100.000
EVA	angleščina	85.000
PELE	angleščina	10.000

Tabela 2: Velikost korpusov usvajanja tujega jezika (Atwell et al., 2003; Axelsson, 2000; Cheng, Warren, 1999; De Cock et al., 1999; Granger, 2001; Horváth, 2003; Izumi et al., 2004; Kennedy, 1998; Lin, 1999; Pravec, 2002; Shih, 2000; Sugiura, 2000; Tenfjord et al., 2004; Tono, 2003; Uzar, 1998).

Zgornja tabela kaže, da je glavnina korpusov velika med pol milijona in milijonom besed. Očitno je to vsaj za angleščino srednja mera med zahtevnostjo gradnje in relevantnostjo rezultatov. Zato bi bilo to tudi smiselno izhodišče za končno verzijo KUST-a; verjetno bo že uporaba pilotske verzije pokazala, ali ne bi bila za tako fleksijski jezik, kot je slovenščina, relevantnejša kaka druga velikost.

S tem je neposredno povezana velikost sestavnih besedil, ki nikjer ne presegajo tisoč besed, sicer pa ima največ korpusov tekste z okrog petsto besedami. Japonski JEFLL izstopa z minimumom dvajset besed pri besedilih najmlajših tvorcev, starih 12 in 13 let (Pravec, 2002). Besedila z izpitov iz znanja slovenščine, ki so možen vir za KUST, imajo po dvesto besed, in tudi spisi s tečajev po izkušnjah učiteljev redko presegajo eno pisano stran. Za polmilijonski korpus bi torej potrebovali 2500 tekstov s po 200 besedami.

5. Vrsta besedil in tema

Vrste besedil in funkcijske zvrsti so v pisnih korpusih usvajanja tujega jezika zelo raznolike. Največ je spisov in esejev, pojavljajo se še pisma, dopisi, dnevniki, poročila, članki, govori, seminarske naloge in podobno. Mnogi korpusi, npr. tajski TELC, hongkonški HKUST, britanski CLC in poljska PELCRA, vključujejo tekste z jezikovnih izpitov, predvsem zaradi enostavne dosegljivosti v večjih količinah, kontroliranih pogojev tvorjenja in približno enake stopnje jezikovne zmožnosti tvorcev. Vendar se spisi, nastali v razredu, besedila z izpitov in naloge, napisane doma, razlikujejo po spodbudi pri nastanku besedil (ali so nastala spontano ali s predhodno pripravo), uporabi referenc (ali so tvorci uporabljali slovar, učne pripomočke ali se sklicevali na že napisane tekste), ter časovni omejenosti pri pisanju. Kaj verodostojneje izraža vmesni jezik? Po eni strani stres na izpitih zmanjšuje jezikovno performanco, po drugi pa smo v dejanski komunikaciji le redko brez časovnih in drugih omejitev, saj npr. sogovorniki niso pripravljeni poljubno čakati na odgovor. Zato samo redki korpusi, HKUST, britanski LCLE, hongkonški TELEC, japonski CEJL, madžarski JPU in mednarodni ICLE (Pravec, 2002; Granger, 2001), vključujejo besedila, napisana doma brez časovnih omejitev, in znotraj korpusov ti predstavljajo manjši delež besedil. Vsi ostali po dostopnih podatkih sodeč vključujejo samo tekste z različnih izpitov in testov. Tovrstni pogoji nastanka so očitno primernejši za vključevanje, zato bi bilo to smiselno upoštevati tudi v KUST-u.

Tematika sestavnih besedil je tvorcem dana vnaprej in zelo pestra, pomembna pa je, ker vpliva na izbiro besedišča. Splošne teme sprožajo uporabo drugačnega besedišča in slovničnih struktur kot bolj specifične. Ker pa še noben korpus usvajanja ni zajel reprezentativnega deleža vseh polnopomenskih besed, se raziskave bolj osredotočajo na slovnične besede in strukture, torej sama tematika ni tako ključna. Vendarle so primernejše argumentativne teme kot opisne, pripovedne, strokovne ali tehnične: priljubljeni so aktualni dogodki in družbeni problemi, služba, potovanja in konjički, razpravljanje o odnosu do ciljnega jezika ali izobraževanja, obnove prebranega in videnega.

Na rezultate vpliva tudi način zbiranja besedil. Lahko so zbrana longitudinalno, z več prispevki istega tvorca iz različnih časovnih obdobij, ali presečno, s hkratnim zajemom besedil več tvorcev. Zaradi relativne mladosti korpusov usvajanja tujega jezika in enostavnosti izvedbe so vsi pregledani korpusi presečni, tak pa bo tudi KUST.

6. Tvorci

Pogoj za uravnoteženost korpusa so tvorci besedil. Njihove notranje kriterije, kot je npr. motiviranost, je težko nadzorovati, graditelji pa pazijo na uravnoteženost in konsistentnost zunanjih dejavnikov: starosti, spola, izobrazbe, prvega jezika, učnega okolja ciljnega jezika in stopnje jezikovne zmožnosti v ciljnem jeziku. Pomembna sta tudi znanje ostalih tujih jezikov ter praktične izkušnje, ki so povezane z bivanjem učečega se v državah, kjer je ciljni jezik prvi jezik (Granger, 1998).

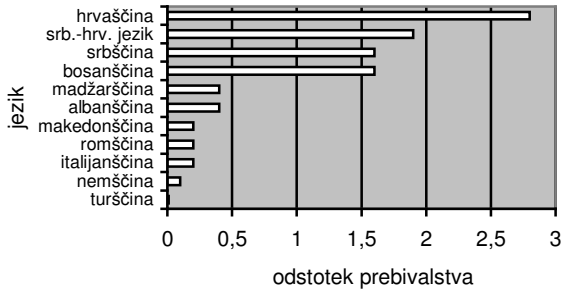
Tvorci v obstoječih korpusih so večinoma še sredi šolanja, v glavnem univerzitetni študentje ali srednješolci (Axelsson, 2000; Granger, 2001; Pravec, 2002; Tenfjord et al., 2004; Uzar, 1998). Čeprav ni nujno, da tisti, ki se še šolajo, pišejo več kot ljudje, ki so šolanje že končali, pa raziskovalci, ki so pogosto zaposleni na univerzah, lažje

pridejo do njihovih besedil. Seveda so primerni tvorci tudi starejši udeleženci tečajev tujega jezika – v vsakem primeru gre ponavadi za ljudi, ki se jezik učijo institucionalizirano, saj so le tako njihova besedila dostopna graditeljem korpusov.

Načini zbiranja podatkov o tvorcih so različni. Vir za korpusa CLC in PELCRA so izpitne pole, za korpusa ICLE in FRIDA tvorci izpolnijo vprašalnik (Pravec, 2002; Granger, 2001). Tudi načini vključevanja podatkov v korpus se razlikujejo, povsod pa so seveda anonimni. Švedski USE ima posebno bazo podatkov o tvorcih v excelovi datoteki (Axelsson, 2000), toda največkrat je to vključeno v glavo TEI vsakega besedila.

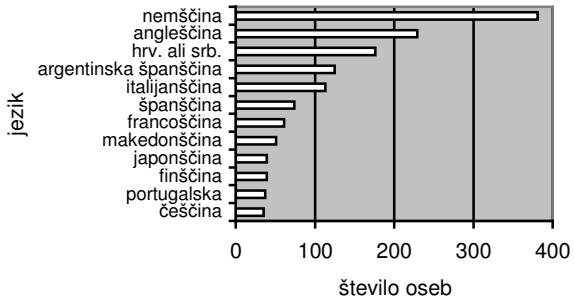
6.1. Prvi jezik

Ciljni jezik večine korpusov je angleščina, ki je tako razširjena po vsem svetu, da lahko tuji govorci angleščine z določenim prvim jezikom oblikujejo svoj korpus. Pri manjših jezikih, kjer je tudi manj konkurence, je situacija drugačna. Francoska FRIDA in norveški ASK imata več izhodiščnih jezikov, ki naj bi ustrezali jezikom največjih skupin priseljencev v tej državi. V Sloveniji je ob popisu leta 2002 87,7 odstotka prebivalstva za svoj materni jezik navedlo slovenščino (Šircelj, 2003), torej je bilo 12,3 odstotka govorcev, za katere slovenščina ni prvi jezik. Smiseln se zdi sklep, da večina govori slovenščino kot drugi jezik. Prav njihovi najpogostejši prvi jeziki bi morali biti tudi prvi jeziki v KUST-u, saj naj bi predstavljali pomemben del populacije govorcev slovenščine kot drugega jezika.



Slika 1. Materni jeziki (razen slovenščine) prebivalstva Slovenije ob popisu leta 2002 (Šircelj, 2003).

Sklepanje o govorcih slovenščine kot tujega jezika bo temeljilo na podatkih o udeležbi na tečajih slovenščine za tujce v letih od 2003 do 2005.[2] Ob tem se je treba zavedati, da bi bili za uporabnike KUST-a relevantnejši tisti tuji govorci, ki se učijo prek organiziranega učnega procesa, kar avtomatično daje prednost govorcem slovenščine kot tujega jezika.



Slika 2. Prvi jeziki udeležencev tečajev slovenščine.[3]

Glede na obe sliki bi bilo treba v KUST-u upoštevati vsaj naslednje skupine izhodiščnih jezikov:

1. Govorci hrvaškega, srbskega oziroma bošnjaškega jezika so najštevilčnejši med govorci slovenščine kot drugega jezika (za 7,9 odstotkov prebivalstva je eden od teh jezikov materni), pa tudi med govorci slovenščine kot tujega jezika jih je 10 odstotkov. Zaradi jezikovnih razlik bi bilo najbolje za vsakega od teh treh jezikov oblikovati ločen podkorpus.

2. 0,2 odstotka prebivalcev sta leta 2002 kot svoj materni jezik navedla makedonščino, torej je za večino med njimi slovenščina drugi jezik. Tudi slabi 3 odstotki udeležencev tečajev govorijo makedonsko kot prvi jezik.

3. Nemščina kot materni jezik je med slovenskim prebivalstvom relativno redka (0,1 odstotek) in za Nemce slovenščina ni pogosto drugi jezik, zato pa je na tečajih slovenščine največ, kar 22 odstotkov udeležencev iz Nemčije, Avstrije in Švice.

4. Tudi za angleško govoreče osebe je slovenščina pomembna predvsem kot tuji jezik. Dobrih 13 odstotkov udeležencev tečajev slovenščine prihaja iz Velike Britanije, Irske, ZDA, Kanade, Avstralije in Nove Zelandije.

5. Govorci slovenščine iz Argentine, ki so večinoma potomci slovenskih izseljencev, imajo poseben status, saj je zanje slovenščina dejansko prvi jezik v družini, ne pa tudi v okolju. Njihov delež na tečajih slovenščine je 7-odstoten. Glede na jezikovno ozadje bi k njim lahko prikljućili tudi ostale govorce španščine (4 odstotke), ki prihajajo iz Španije, Mehike, Venezuele, Urugvaja, Kube, Peruja, Čila, Bolivije, Dominikanske republike in drugih držav. Korpusni podatki bodo pokazali, v kolikšni meri se njihova slovenščina razlikuje od slovenščine argentinskih izseljencev.

6. Zadnja večja skupina so govorci italijanščine (0,2 odstotka prebivalstva, dobrih 6 odstotkov na tečajih slovenščine), vendar gre tudi pri teh za precejšen delež zamejskih Slovencev, ki obiskujejo tečaje.

Druge izrazitejše skupine slovensko govorečih tujcev na tečajih so še tiste s francoščino, japonščino, finščino, portugalščino in češčino kot prvim jezikom.

Če izhajamo iz omenjene situacije, bi KUST lahko razdelili na podkorpuse, ki bi se ločili po izhodiščnih jezikih. Ti naj bi bili hrvaški, srbski in bošnjaški, makedonski, nemški, angleški, španski in italijanski. Smiselno je, da so med seboj po velikosti primerljivi, saj razmerja

[2] Podatki so bili pridobljeni na Centru za slovenščino kot drugi/tuji jezik Filozofske fakultete Univerze v Ljubljani.

[3] Hrvaški, srbski in bošnjaški jezik so združeni v isto kategorijo, ker se udeleženci tečajev slovenščine pri izpolnjevanju prijavnic pogosto ne opredelijo za enega od jezikov in podatki torej niso točni.

med populacijami niso konstantna. Tako bi imel v polmilijonskem korpusu vsak podkorpus v končni fazi 100.000 besed oziroma 400 besedil s po 250 besedami. Poleg tega bi bilo smiselno oblikovati še podkorpus ostalih izhodiščnih jezikov, ki bi bil bolj kot zanimivost, informacija ali primerjava.

6.2. Stopnja jezikovne zmožnosti

Le malo korpusov vključuje besedila začetnikov, saj ti tvorijo kratke tekste z mnogo odkloni od norme, popolnoma tujimi strukturami in malo koherentne vsebine. Precej primernejši so zato nadaljevalci ali izpopolnjevalci. Največ je korpusov, kjer so avtorji nadaljevalci (Granger, 2001; Kennedy, 1998; Pravec, 2002; Shih, 2000; Tenfjord et al., 2004). Ker je tudi na tečajih in izpitih iz slovenščine več nadaljevalcev kot izpopolnjevalcev,[4] bi se bilo v KUST-u smiselno osredotočiti na nadaljevalno stopnjo.

7. Kontrolni korpus

Uporabnost korpusa usvajanja tujega jezika poveča vzporedni kontrolni korpus jezika rojenih govorcev, ki omogoča primerjave. Tako dobimo kvantitativne podatke o pogostnosti določenih besed, besednih vrst, skladenjskih struktur in značilnostih diskurza (Tono, 2003). Nekateri za kontrolo uporabijo že obstoječe korpuse ali njihove dele, drugi pa zgradijo poseben podkorpus. Od 18 pisnih korpusov usvajanja tujega jezika, ki so relevantni ob razmišljanju o KUST-u, jih ima le šest kontrolni korpus (Granger, 2002; Tenfjord et al., 2004; Uzar, 1998; Izumi et al., 2004; Pravec, 2002). V obeh primerih si morajo biti načela gradnje približno podobna. Ker so kontrolni korpusi, ki zahtevajo precej truda graditeljev, skromno razširjeni, in ker za slovenščino obstaja referenčni korpus FIDA, bi lahko bil KUST vsaj na začetku brez posebnega kontrolnega korpusa.

8. Označevanje napak

Ker korpus usvajanja tujega jezika za razliko od običajnih korpusov nudi odklon od norme, je posebej koristno, če so v njem označene tudi napake, ki nastajajo pri jezikovni produkciji. Prav z analizo napak se je začelo raziskovanje vmesnega jezika učečih se, in čeprav danes to še zdaleč ni več edini vidik raziskav, je še vedno eden izmed najpomembnejših.

Za razliko od bolj razvitih in predvidljivih ravni označevanja, kot sta oblikoslovno označevanje in lematiziranje, je označevanje napak izjemno zamudno. Jezikovne rešitve učečih so večkrat tako nenavadne in ustvarjalne, da jih je nemogoče vnaprej predvideti. Za japonski korpus SST so skušali razviti sistem za avtomatično prepoznavanje napak. Program, ki je za uspešno delovanje potreboval dvaintrideset različnih podatkov,[5] so naučili na stopetdesetih dokumentih in ga preizkusili na šestnajstih, vendar so bili rezultati nezadovoljivi. Dodali so pravilne izjave iz podkorpusa rojenih govorcev, izjave izpraševalca in popravljene stavke iz korpusa. Rezultati so se nekoliko izboljšali, vendar so bili še vedno zgolj 43-odstotno natančni. Nazadnje so dodali umetne napake in s tem natančnost programa nekoliko zvišali, vendar bi potrebovali še več označenega gradiva za učenje, da bi bila natančnost zadovoljiva (Izumi et al., 2004). Zato označevanje poteka ročno in ni prav verjetno, da bo kmalu avtomatizirano.

Zaradi časovne potratnosti označevanja so napake zaenkrat označene le na manjšem delu korpusov. Od pregledanih korpusov usvajanja tujega jezika le v 14 označujejo napake, vendar tudi tu niso označene na vseh besedah, temveč zgolj na omejenem vzorcu, na primer na tretjini ali petini besed. Pri več kot desetmilijonskih korpusih to popolnoma zadostuje. V celoti je tako označenih 22 % vseh besed.

Za kakovostno in informativno obdelavo je treba napake klasificirati glede na določeno taksonomijo, ki naj bi omogočala opis in hkrati kvantitativno analizo. Klasifikacija je naporno, sporno in razmeroma brezplodno delo, kajti razdelimo jih lahko na različne načine glede na cilj raziskave in jezikoslovno teorijo, iz katere izhajamo. Tako celo za angleščino kot ciljni jezik ni ustaljenega načina, temveč vsak korpus deli napake po svoje. Zato so v 25-milijonskem HKUST-u označili pet milijonov besed samo s kategorijama "napaka" in "ne-napaka" (Pravec, 2002).

Možen način klasificiranja je glede na izvor napak. Delitev je več (Pirih Svetina, 2005), vendar je o izvoru brez dvosmerne komunikacije s tvorcem besedila in brez poznavanja njegovega ozadja dejansko mogoče samo ugibati. Oblikovalci korpusov tovrstnih klasifikacij zato ne uporabljajo. Edini, ki jih deli na omenjeni način, je Japanese Learners' Corpus (Sugiura, 2000).

Uporabnejše so tipologije, ki napake razvrščajo glede na spremembo predvidene ciljne oblike oziroma način odklona od nje. Tu ločimo zgrešitve ali napačne izbore, kjer je jezikovna značilnost narobe enkodirana, izpuste, kjer ni enkodirana, in dodajanja oziroma vstavitve, kjer je enkodirana odvečna jezikovna značilnost (Ragan, 2001). Včasih sta posebni kategoriji še neustrezen besedni red in sestava neobstoječe oblike iz dveh pravilnih.

V korpusih usvajanja tujega jezika se pojavlja tudi klasifikacija, ki napake razvršča znotraj posameznih jezikoslovnih ravnin in natančnejših kategorij: lahko so fonetične, oblikoslovne, oblikoskladenjske, skladenjske, pravopisne, leksikalne, povezane s samostalniki, glagoli, pridevniki in podobno. Take oznake so relativno objektivne, čeprav še vedno odvisne od interpretacije označevalca.

Najbolj priljubljene in tudi najuporabnejše so kombinacije klasifikacij. V japonskem korpusu SST ima vsaka napaka označene tri vrste podatkov: oblikoslovno umestitev, slovnično pravilo in pravilno obliko. Nabor skupaj vsebuje petinštirideset oznak. Poleg tega jih delijo tudi glede na spremembo ciljne oblike na izpuste, zamenjave in vstavitve (Izumi et al., 2004).

*I belong to two baseball **<n_num crr="teams">** team**</n_num>**.

Primer 1: Označen stavek z napako v korpusu SST: *n* označuje samostalnik, *num* pove, da gre za napako v številu, atribut *crr*= pa vsebuje pravilno obliko.

V korpusu ICLE vsaki napaki določijo krovno kategorijo, ali gre za formalno, slovnično, leksiko-slovnično, leksikalno napako, napako registra, odvečne/manjkajoče besede/napačen besedni red ali slog. Tej sledi vrsta podkategorij (Pravec, 2002). V korpusu FRIDA dobi vsaka napa-

[4] Podatki so bili pridobljeni na Centru za slovenščino kot drugi/tuji jezik Filozofske fakultete Univerze v Ljubljani.

[5] To so ciljna beseda, dve besedi prej in dve potem, njihove besedne vrste in slovarske oblike, pet različnih kombinacij omenjenega ter prve in zadnje črke ciljne besede.

ka tri oznake: področje (slovnica, slovar, zapis itn.), kategorijo (vrsta, število itn.) in slovnično kategorijo (samostalnik, pridevnik itn.) (Granger, 2001). Norveški ASK ima zelo razčlenjeno taksonomijo na leksemske, morfološke, sintaktične, interpunkcijske in nerazvrščene napake, ki se nato še natančneje delijo (Tenfjord et al., 2004).

Zaradi težavnosti in spornosti klasificiranja napak se nekateri celo sprašujejo o smislu tega početja (Tono, 2003). Delo je pogosto intuitivno, možne so različne interpretacije, nobena analiza pa ne zajame vseh razlag (Ragan, 2001). Yukio Tono, ki je sodeloval pri več japonskih korpusih usvajanja angleščine, zaradi heterogenosti taksonomij predlaga, da bi se vsi držali splošnega nabora, kjer bi določili samo jezikovno kategorijo in način odklona od ciljne oblike, nato pa bi ga prilagajali potrebam svoje raziskave (2003). Da brez prilagajanja ne gre, je ugotovil, ko je pri ročnem označevanju shemo nehote adaptiral ciljem svoje raziskave.

Predlog klasifikacije v KUST-u je bil narejen po analizi manjšega nabora besedil tujih tvorcev[6] in napake deli na dveh ravneh.[7] Čeprav se zdi privlačna možnost, da je prvi kriterij sprememba ciljne oblike, to verjetno ni prvi podatek, ki bi zanimal profesionalne uporabnike, zato je prva raven kombinacija jezikovne ravnine in spremembe ciljne oblike. Tako so tu kategorije pravopis, besedišče, besedotvorje, oblikoslovje, besedni red, skladnja, izpust in vstavitev. K pravopisu spadajo vse besede z manjkajočimi ali odvečnimi črkami, napačna raba male oziroma velike začetnice in narobe zapisane besede. Besedišče pokriva napake, kjer so uporabljene besede, ki v slovenščini ne obstajajo ali pa so uporabljene v napačnem kontekstu. Napake zaradi napačnega tvorjenja besed so označene kot napake v besedotvorju. V oblikoslovje se uvrščajo napake zaradi nepravilnega pregibanja. Od skladenjskih napak se zaradi pogostnosti ločijo napake v besednem redu, tu gre lahko za napake znotraj samostalniških besednih zvez ali v naslonskem nizu. Izpusti in vstavitve se nanašajo na spremembo ciljne oblike in nimajo nadaljnjih atributov, ker bi bilo to glede na relativno redkost njihovega pojavljanja neekonomično. Poleg tega ugibanje o tem, kaj je v besedilu izpuščeno, zmanjšuje objektivnost in relevantnost oznak. Gotovo pa bi bilo smiselno uvesti tudi kategorijo nerazvrščenih napak za vse pojave, ki ne sodijo v nobeno drugo kategorijo.

Na drugi ravni je pomembna besedna vrsta. Strukturalistična razdelitev je pri tem nekoliko prilagojena pogostnosti pojavljanja, tako so kategorije samostalnik, pridevnik, glagol, prislov, števnik, predlog, veznik, členek in medmet. Prav veliko napak členkov in medmetov pri tujcih sicer ne pričakujemo, vprašanje pa je, kam uvrstiti samostalniške in pridevniške zaimke – ali narediti posebno kategorijo ali jih dati k samostalnikom oziroma pridevnikom? V izhodišču bi bila lahko izbrana slednja rešitev, nekoliko večji korpus učnih primerov pa bo pokazal njeno ustreznost.

1. raven	**2. raven**	**Primer**
pravopis	glagol	Ljudje so jo uzeli na piko.
besedišče	pridevnik	nisem rabil prvih luči
besedišče	členek	ta oseba ki je domneva prišla, ni res
besedotvorje	samostalnik	V ponedeljek sem poučevala francoskoščino.
oblikoslovje	glagol	V soboto in nedeljo sem šla na sprehod, sem berala
besedni red		Zjutraj zgodaj sem vstal.
skladnja		Na prvem vtisu je tema filma politična.
vstavitev[8]		
izpust		Včeraj sem zbudila zelo pozno.

Tabela 2: Primeri napak za posamezne kategorije[9]

Poskusi še natančnejšega klasificiranja, npr. zaradi napačnega spola, sklona ali števila, so pokazali, da se je pri tem nemogoče izogniti dvoumnostim in subjektivni interpretaciji. Čim bi imele tovrstne napake eno oznako, bi bila analiza otežkočena, saj bi bile druge interpretacije avtomatsko izključene.

V KUST-u poleg napak ne bodo napisane pravilne oblike. Na prvi pogled se to zdi možnost za dodatna pojasnila, posebej, kadar označevalec dobi že popravljen spis. Vendar je pogosto težko določiti, kaj je pravilna oblika očitno napačne jezikovne značilnosti. Domnevno pravilne oblike skrčijo možne interpretacije, ker označevalci od tvorcev besedil ne morejo izvedeti, kaj so v resnici hoteli povedati. Navajanje postane bolj ali manj posrečeno ugibanje, ki vsiljuje en sam vidik, zmanjšuje objektivnost označevanja in oteži priklic drugih možnosti.

Opisana klasifikacija je primerna za govorce najrazličnejših prvih jezikov, saj ni smiselno oblikovati ločenih klasifikacij za različne prve jezike. Seveda je že doživela prilagoditve in jih bo ob dejanskem označevanju učnega KUST-a nedvomno spet, saj bo treba sproti reševati probleme, kot je, kam uvrstiti uporabo napačne besedne vrste. Vendar je ne glede na (ne)popolnost klasifikacije pomembno predvsem to, da je označevanje konsistentno znotraj korpusa.

Zaradi omejene dostopnosti podatkov o obstoječih korpusih je težko podati natančen pregled tehnične plati označevanja napak, vendar jih večinoma seveda označujejo z jezikom SGML (HKUST) ali XML (LCLE, FRIDA, JEFLL, SST) (Pravec, 2002; Granger, 2001; Izumi et al., 2004). To je tudi edini smotrn označevalni jezik za KUST.

```
V soboto in nedeljo sem šla na sprehod, sem <sic
ana="TKUST.Obl.glag">berala</sic>
```

Primer 2: Označena napaka v poskusnem KUST-u (prim. op. 4); *sic* pomeni, da gre za napako, *TKUST.Obl.glag* pa, da je napaka oblikoslovna, in sicer glagola.

[6] Besedila petih tvorcev na nižji nadaljevalni stopnji s skupaj 600 besedami so nastala na tečaju slovenščine za tujce v študijskem letu 2004/2005.

[7] Klasifikacija je primerna za korpus, ki ni oblikoslovno označen. Ker sta v načrtu tudi lematizacija in oblikoslovno označevanje, bo treba klasifikacijo takrat ustrezno prilagoditi.

[8] V poskusno zbranem KUST-u (prim. op. 4) ni bilo primera vstavitve.

[9] Primeri so iz poskusno zbranega KUST-a (prim. op. 4).

9. Zaključek

Po analizi različnih že obstoječih tujih korpusov usvajanja, ki so delno lahko zgled za slovenski korpus, je nastala v članku predlagana zasnova korpusa usvajanja slovenščine kot tujega jezika, po kateri gre za pol- do enomilijonski korpus s ciljnim jezikom slovenščino in izhodiščnimi jeziki srbščino oziroma hrvaščino, makedonščino, angleščino, nemščino in italijanščino ter skupino vseh ostalih jezikov. Korpus je samo pisni. Tvorci so na nadaljevalni stopnji znanja, o njih so znani osnovni sociolingvistični podatki, podpišejo pa tudi privolitveni obrazec. Vir besedil so spisi z izpitov iz znanja slovenščine in s tečajev slovenščine, tematika je čim splošnejša. V korpusu ni besedil, napisanih doma, saj so tako pogoji nastanka relativno izenačeni za vse tekste. Kontrolnega korpusa ni, saj lahko kot to funkcionira FIDA, v njem pa so ročno označene napake po razdelani klasifikaciji, ki zajema način odklona od pričakovane oblike, jezikovno ravnino in besedno vrsto.

Naštete značilnosti in lastnosti bodo preizkušene na pilotskem korpusu usvajanja tujega jezika. Predvsem pri označevanju napak, kjer za slovenščino v ta namen še ni bilo razdelanega in preizkušenega sistema, pa je pričakovati največ sprememb in prilagoditev, ki bodo končno verzijo naredile prijaznejšo in uporabnejšo tako za znanstvene raziskovalce kot tudi za bolj praktično usmerjene pedagoge.

10. Literatura

Aston, Guy, 1997. *Small and Large Corpora in Language Learning.* http://home.sslmit.unibo.it/~guy/wudj1.htm.

Atwell, Eric, Howarth, Peter, Souter, Clive, 2003. The ISLE Corpus: Italian and German Spoken Learners' English. *ICAME Journal*. 27:5–18.

Axelsson, Margareta Westergreen, 2000. USE – The Uppsala Student English Corpus: An instrument for needs analysis. *ICAME Journal*. 24:155–157.

Cheng, Winnie, Warren, Martin, 1999. Facilitating a description of intercultural conversations: the Hong Kong Corpus of Conversational English. *ICAME Journal.* 23:5–20.

Dagneaux, Estelle, Granger, Sylviane, Meunier, Fanny, Petch-Tyson, Stephanie, Vilret, Xavier, 2001. *A web interface to the International Corpus of Learner English*. http://juppiter.fltr.ucl.ac.be/FLTR/GERM/ETAN/CECL/Events/icamepr.htm#interface.

De Cock, S., Granger, Sylviane, Petch-Tyson, S., 1999. *The Louvain International Database of Spoken English Interlanguage: The LINDSEI Project.* http://juppiter.fltr.ucl.ac.be/FLTR/GERM/ETAN/CECL/Cecl-Projects/Lindsei/download/lindsei.pdf.

Erjavec, Tomaž, 2004. *Uvod v korpusno jezikoslovje*. http://nl.ijs.si/et/talks/korpus/korpusno.html.

Gillard, Patrick, Gadsby, Adam, 1998. Using a learners' corpus in compiling ELT dicitionaries. V S. Granger (ur.), *Learner English on Computer*. London, New York: Longman.

Gorjanc, Vojko, 2005. *Uvod v korpusno jezikoslovje*. Ljubljana: Izolit.

Granger, Sylviane, Tribble, Chris, 1998. Learner corpus data in the foreign language classroom: form-focused instruction and data-driven learning. V S. Granger (ur.), *Learner English on Computer*. London, New York: Longman.

Granger, Sylviane, 2001. *International Corpus of Learner English: The ICLE Project.* http://juppiter.fltr.ucl.ac.be/FLTR/GERM/ETAN/CECL/Cecl-Projects/Icle/download/icle.pdf.

Horváth, József, 1999. *Advanced Writing in English as a Foreign Language: A Corpus-Based Study of Processes and Products.* http://www.geocities.com/writing_site/thesis/index.html.

Izumi, Emi, Uchimoto, Kiyotaka, in Isahara, Hitoshi, 2004. SST speech corpus of Japanese Learners' English and automatic detection of learners' errors. *ICAME Journal*. 28:31–48.

Kennedy, Graeme, 1998. *An Introduction to Corpus Linguistics*. London, New York: Longman.

Lin, Linda H. F., 1999. *Applying Information Technology to a corpus of student report writing to help students write better reports*. http://elc.polyu.edu.hk/conference/papers/Lin.htm.

Milton, John, 1998. Exploiting L1 and interlanguage corpora in the design of an electronic language learning and production environment. V S. Granger (ur.), *Learner English on Computer*. London, New York: Longman.

Pirih Svetina, Nataša, 2003. Napaka v ogledalu procesa učenja tujega jezika. *Jezik in slovstvo* 2:17–26.

Pirih Svetina, Nataša, 2005. *Slovenščina kot tuji jezik*. Ljubljana: Izolit.

Pravec, Norma, 2002. Survey of learner corpora. *ICAME Journal.* 26:81–114.

Ragan, Peter H., 2001. Classroom Use of a Systemic Functional Small Learner Corpus. V M. Ghadessy, A. Henry, in R. L. Roseberry (ur.), *Small Corpus Studies and ELT*. Amsterdam, Philadelphia: John Benjamins Publishing Co.

Shih, Rebecca Hsue-Hueh, 2000. Compiling Taiwanese Learner Corpus of English. *Computational Linguistic and Chinese Language Processing*. 2: 89–102.

Sugiura, Masatoshi, 2000. *On Enhancing the Writing Skill of EFL Learners in Japan: Utilizing the Insights and Technologies of Corpus Linguistics*. http://oscar.lang.nagoya-u.ac.jp/~sugiura/hawaii/680p/corpuswriting.html.

Šircelj, Milivoja, 2003. *Verska, jezikovna in narodna sestava prebivalstva Slovenije: Popisi 1921-2002.* Ljubljana: Statistični urad republike Slovenije-

Tenfjord, Kari, Meurer, Paul, Hofland, Knut, 2004. *The ASK corpus – a language learner corpus of Norweigan as a second language (Poster).* http://www.ugr.es/~talc6/talc_search/proceedings/60.html.

Tono, Yukio, 2003. Learner corpora: design, development and applications. *Proceedings of the Corpus Linguistics 2003 Conference*. Lancaster: University.

Uzar, Rafal S., 1998. *The PELE Project: A New Perspective for Learner Language Corpora.* http://members.fortunecity.com/pelcra/pele.htm.

Learning rules for morphological analysis and synthesis of Macedonian nouns, adjectives and verbs

Aneta Ivanovska*, Katerina Zdravkova†, Tomaž Erjavec*, Sašo Džeroski*

* Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
{aneta.ivanovska, tomaz.erjavec, saso.dzeroski}@ijs.si

†Institute of Informatics, Faculty of Natural Sciences and Mathematics
Arhimedova 5, 1000 Skopje, Macedonia
keti@ii.edu.mk

Abstract

This paper presents a machine learning approach to morphological analysis and synthesis of Macedonian nouns, adjectives and verbs. An inductive logic programming (ILP) system, Clog, was used to learn the inflectional paradigms of Macedonian words. Clog learns first-order decision lists, i.e., ordered sets of rules. Training and testing of the rules was performed on the words originating from Orwell's "1984". High accuracies (over 90%) were achieved, which are encouraging both for further research in annotation of Macedonian language resources as well as practical use (eg. in web search engines).

Učenje pravil za oblikoslovno analizo in sintezo makedonskih samostalnikov, pridevnikov in glagolov

V prispevku predstavimo metodo za strojno učenje oblikoslovne analize in sinteze makedonskih samostalnikov, pridevnikov in glagolov. Za učenje pregibnih paradigem makedonskih besed smo uporabili sistem Clog, ki temelji na induktivnem logičnem programiranju. Clog se nauči odločitvenih seznamov prvega reda, to je urejenih seznamov pravil. Za učenje in testiranje pravil smo uporabili besede, ki jih vsebuje roman "1984" G. Orwella. Dosegli smo visoko točnost (preko 90%), kar je spodbudno tako za nadaljnje raziskave pri označevanju makedonskih jezikovnih virov, kot za praktično uporabi, npr. pri mrežnih iskalnikih.

1. Introduction

The Macedonian language belongs to the South-Slavic family of languages and with the other Slavic languages shares a rich system of inflections. Although morphological rules for word formation in Macedonian have been exhaustively studied by linguists for decades (Koneski, 1952, 2004), they have not been systematized until recently (Petrovski, 2005). This paper is concerned with the problem of machine learning of morphological rules for producing the inflectional forms of nouns, adjectives and verbs given the base form (lemma), and of deducing the base form from those inflectional forms, i.e., of learning rules for morphological synthesis and analysis.

This task has been previously addressed for other Slavic languages, such as Slovenian (Erjavec and Džeroski, 2004), Czech, Bulgarian, etc. The first attempt to do this for Macedonian was by learning rules for morphological analysis and synthesis of nouns (Ivanovska et al., 2005). In this study the same approach was applied to the Macedonian adjectives and verbs.

The examples (word-forms) used in the process of learning rules for morphological analysis and synthesis were taken from the Macedonian translation of George Orwell's "1984", which itself is meant to become a part of the Multext-East (Multilingual Text Tools and Corpora for Eastern and Central European Languages) language resources (Erjavec, 2004).

In this paper, we first describe the preprocessing of the data, which consists of the annotation of words made in line with Multext-East notation and transforming the data in a format for running the ILP system Clog. Particular attention in the paper is paid to the process of learning rules for morphological analysis and synthesis of Macedonian words using the ILP system Clog (Manandhar et al., 1998). Section 2 explains in details the preprocessing of the data. The experiments and the experimental results are presented in Section 3. Conclusions discuss the results and directions for future work.

2. Preparing the lexicon

The morphosyntactic annotation of words was made according to the Multext-East specification (Erjavec, 2004), where each word-form is associated with a morphosyntactic description (MSD) presented as a packed string. Its first character, always uppercase, represents the part-of-speech (grammatical category). It is followed by a list of character values corresponding to the part-of-speech dependent attributes. For instance, the MSD Ncmsnn expands to PoS (part-of-speech): Noun; Type: *common*; Gender: *masculine*; Number: *singular*; Case: *nominative*; and Definiteness: *no*.

Nouns in Macedonian are marked for type (2 values), gender (3 values), number (3 values), case (3 values) and definiteness (4 values); adjectives are marked for type (3), degree (4), gender (3), number (2) and definiteness (4); verbs are marked for type (4), form (3), tense (3), person (3), number (2), gender (3), negative (2) and aspect (2) (Figures 1, 2, 3).

The annotation of Macedonian words was facilitated with a specially designed program (Fig. 5). PoS tagging was automatically performed according to word prefixes and suffixes. Correction of misclassified words was done during MSD tagging. This process was also performed automatically, and polished manually (Ivanovska et al

2005). However, even with manual correction, annotation of the words was not error free, as noticed during the process of learning morphological rules, and was afterwards corrected. In parallel with the MSD annotation, lemmas were added in a separate column. The structure of the resulting lexicon is illustrated in Figure 4.

P	ATT	VAL	
1	Type	common	c
		proper	p
2	Gender	masculine	m
		feminine	f
		neuter	n
3	Number	singular	s
		plural	p
		count	t
4	Case	nominative	n
		vocative	v
		oblique	o
5	Definiteness	no	n
		yes	y
		distal	d
		proximal	p

Figure 1. Attributes and their possible values of the grammatical category Nouns

P	ATT	VAL	
1	Type	qualificative	f
		ordinal	o
2	Degree	positive	p
		comparative	c
		superlative	s
		elative	e
3	Gender	masculine	m
		feminine	f
		neuter	n
4	Number	singular	s
		plural	p
5	Definiteness	no	n
		yes	y
		distal	d
		proximal	p

Figure 2. Attributes and their possible values of the adjectives

P	ATT	VAL	
1	Type	main	m
		auxiliary	a
		modal	o
		copula	c
2	VForm	indicative	i
		imperative	m
		participle	p
3	Tense	present	p
		imperfect	i
		aorist	a
4	Person	first	1
		second	2
		third	3
5	Number	singular	s
		plural	p
6	Gender	masculine	m
		feminine	f
		neuter	n
7	Negative	no	n
		yes	y
8	Aspect	progressive	p
		perfective	e

Figure 3. Attributes and their possible values of the verbs

Next, the lexicon had to be transformed into the format suitable for running Clog. The tables were split into three documents (one for every analyzed grammatical category), and the text was (for easier inspection) transliterated into the Latin alphabet.

```
авион    авион  Ncmsnn
авиони   авион  Ncmpnn
автентични  автентичен  Afpmp-n
автоматска  автоматски  Aopfs-n
асимилира  асимилира  Vmip3s--n-----p
асоцираа  асоцира  Vmii3p--n-----p
```

Figure 4. An example of the structure of the lexicon

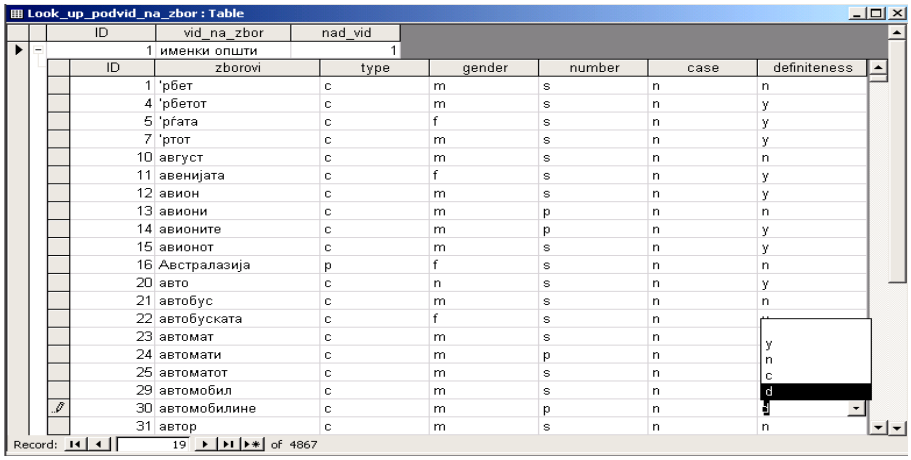


Figure 5. An example of the structure of the lexicon

3. Experiments and results

This section describes how the morphological analysis and synthesis of nouns, adjectives and verbs was carried out using the inductive logic programming system Clog.

We will fist explain the notion of first-order decision lists on the problem of synthesis of the past tense of English verbs, one of the first examples of learning morphology with ILP, using the ILP system FOIDL (Mooney and Califf, 1995).

The ILP formulation of the problem is as follows. A logic program has to be learned defining the relation past(PresentVerb, PastVerb), where PresentVerb is an orthographic representation of the present tense of a verb and PastVerb is an orthographic representation of its past tense. PresentVerb is the input and PastVerb the output argument. Given are examples of input/output pairs, such as past([b,a,r,k], [b,a,r,k,e,d]) and past([g,o], [w,e,n,t]). The program for the relation past uses the predicate split(A, B, C) as background knowledge: this predicate splits a list of letters A into two lists B and C, e.g., split([b,a,r,k,e,d], [b,a,r,k], [e,d]).

Given examples and background knowledge, FOIDL (Mooney and Califf, 1995) learns a first-order decision list defining the predicate past. And example of such list is given in Figure 6.

```
past([g,o], [w,e,n,t]) :- !.
past(A, B) :- split(A, C, [e,p]), split(B, C, [p,t]), !.
past(A, B) :- split(B, A, [d]), split(A, _, [e]), !.
past(A, B) :- split(B, A, [e,d]), !.
```

Figure 6. A first-order decision list

Clog is similar to FOIDL in the sense that it also learns first-order decision lists, i.e., ordered sets of rules, from positive examples only (Manandhar et al., 1998). For the process of learning rules, triplets from the training data, presented earlier, were used. Each triplet was an example of analysis of the form *msd(orth, lemma)*, where *orth* and *lemma* are the orthographic representations of the word-form and the lemma, respectively. Within the learning setting of inductive logic programming, *msd(Orth, Lemma)* is a relation or predicate that consists of all pairs (word-form, lemma) that have the same morphosyntactic description. *Orth* is the input and *Lemma* is the output argument. In Clog the predicate *mate* is used as background knowledge instead of the predicate *split*. *mate* generalizes *split* to deal also with prefixes (useful for analyzing superlative forms of Macedonian adjectives).

A set of rules was learned for each of the *msd* predicates. The rules were encoded as PROLOG facts. An example of rules for the analysis of Macedonian qualificative indefinite feminine adjectives is given in Figure 7.

```
afpfs_n(A,B):-mate(A,B,[],[],[r,n,a],[r,e,n]),!.
afpfs_n(A,B):-mate(A,B,[],[],[d,n,a],[d,e,n]),!.
afpfs_n(A,B):-mate(A,B,[],[],[t,n,a],[t,e,n]),!.
afpfs_n(A,B):-mate(A,B,[],[],[v,n,a],[v,e,n]),!.
afpfs_n(A,B):-mate(A,B,[],[],[b,n,a],[b,e,n]),!.
```

Figure 7. An example of PROLOG rules for analysis the Macedonian qualificative indefinite feminine adjectives

3.1. Morphological analysis and synthesis

As said previously, morphological analysis is the process of deducing the base form (lemma) from the inflectional forms of the words (word-forms), while morphological synthesis is the process of producing the inflectional forms given the base form. In this section, these learnt rules are described for the three grammatical categories –adjectives, verbs and nouns.

3.1.1. Adjectives

The morphological analysis and synthesis were carried out over 5,078 word-forms of adjectives. A set of rules was learned for every MSD that has more than 100 examples. MSDs with less than 100 examples do not provide enough data to induce good rules. The rule sets vary in size and complexity over different MSDs and refer to how the suffix or/and the prefix of the word changes to obtain the base form. An example of induced exceptions and rules for analyzing the singular of Macedonian qualificative definite feminine adjectives is given in Figure 8.

```
afpfs_y([z,e,m,j,i,n,a,t,a],[z,e,m,j,i,n]):- !.
afpfs_y ([t,o,p,l,a,t,a],[t,o,p,o,l]):- !.
afpfs_y ([t,e,n,k,a,t,a],[t,e,n,o,k]):- !.
afpfs_y ([s,t,a,r,a,t,a],[s,t,a,r]):- !.

afpfs_y (A,B):-mate(A,B,,[],[],[t,r,a,t,a],[t,a,r]),!.
afpfs_y (A,B):-mate(A,B,[],[],[s,a,t,a],[s]),!.
afpfs_y (A,B):-mate(A,B,[],[],[v,a,t,a],[v]),!.
afpfs_y (A,B):-mate(A,B,[],[],[e,t,a],[e,t]),!.
afpfs_y (A,B):-mate(A,B,[],[],[m,a,t,a],[m]),!.
afpfs_y (A,B):-mate(A,B,[],[],[g,a,t,a],[g]),!.
afpfs_y (A,B):-mate(A,B,[],[],[o,k,a,t,a],[o,k]),!.
```

Figure 8. Exceptions and rules in analysis of Macedonian qualificative definite feminine adjectives in the singular

Some of the words were not correctly analyzed and this happens because of three reasons: a) the rule applied to the word-form generates a word that is not equal to the lemma, b) there is no rule that corresponds to that combination of word-form – msd, and c) there is an error due to the manual annotation of the words. An example of incorrectly lemmatized adjectives is given in Figure 9.

```
slatka  Afpfs-n  sladok  |slatk|  ERR
dobra  Afpfs-n  dobar  |dobr|  ERR
mrtvi  Afpmp-n  mrtov  |mrtv|  ERR
polna  Afpfs-n  poln  |polen|  ERR
mekoto  Afpns-y  mek  |???|  ERR
kratko  Afpns-n  kratok  |???|  ERR
slabite  Afpns-y  slab  |???|  ERR
negibnata  Afpfs-y  negibnat  |???|  ERR
blagoto  Afpns-n  blag  |???|  ERR
spokojni  Afpns-n  spokoen  |spokojen|  ERR
```

Figure 9. Incorrectly lemmatized adjectives

In Figure 9 the adjectives *slatka (sweet)*, *dobra (good)*, *mrtvi (dead)* and *polna (full)* were incorrectly lemmatized because the rules applied to them generated words that are not equal with their lemmas. For the adjectives *mekoto (soft)*, *kratko (short)* and *slabite (thin)* there were no rules that could be applied so it did not generate any word. And the errors in the last three adjectives – *negibnata (untouched)*, *blagoto (sweet)* and *spokojni (calm)*, were due to the incorrect annotation (errors in the MSDs).

To test the accuracy of the obtained rules, 10-fold cross validation was performed. After correcting the errors, which were due to the manual annotation of the adjectives, we have achieved average accuracy of 93.12%. The obtained average accuracy of the rules for the analysis is given in Table 1.

Morphological synthesis was carried out over the same set of adjectives, only the structure of the set was changed, i.e., the columns of lemmas and word-forms were swapped. The data consisted of triplets lemma-wordform-msd. Again, rules were learned only for those MSDs that have more than 100 examples. Examples of exceptions and rules for synthesizing the neuter singular form of the qualificative adjectives are given in Figure 10.

```
afpns_n([o,d,g,o,v,o,r,e,n],[o,d,g,o,v,o,r,e,n,o]):- !.
afpns_n ([u,t,v,r,d,e,n],[u,t,v,r,d,e,n,o]):- !.
afpns_n ([v,i,s,o,k],[v,i,s,o,k,o]):- !.
afpns_n ([s,t,u,d,e,n],[s,t,u,d,e,n,o]):- !.

afpns_n(A,B):-mate(A,B,[i],[i],[l,e,n],[l,n,o]),!.
afpns_n(A,B):-mate(A,B,[v,o],[v,o],[e,n],[e,n,o]),!.
afpns_n (A,B):-mate(A,B,[],[],[p],[p,o]),!.
afpns_n(A,B):-mate(A,B,[n,a],[n,a],[l,e,n],[l,e,n,o]),!.
afpns_n (A,B):-mate(A,B,[],[],[a,r],[r,o]),!.
afpns_n (A,B):-mate(A,B,[],[],[t],[t,o]),!.
afpns_n (A,B):-mate(A,B,[],[],[a,n],[a,n,o]),!.
```

Figure 10. Exceptions and rules in synthesizing the neuter singular form of the Macedonian qualificative adjectives

The accuracy of the synthesis rules was tested using 10-fold cross validation and the average accuracy is 82.77%, which is lower than the accuracy of the rules obtained in the morphological analysis, mostly because of the large number of inflectional forms in which one adjective can be found. The average accuracy and standard deviation for the morphological synthesis of the adjectives are given in Table 1.

	Analysis	Synthesis
Accuracy (%)	93.12	82.77
Standard deviation	1.30610915	1.066208235

Table 1. Average accuracy and standard deviation of the rules for analysis and synthesis of Macedonian adjectives

3.1.2. Verbs

Verbs are the most complex grammatical category in Macedonian language (they have the largest number of attributes and MSDs). The morphological analysis and synthesis of verbs were carried out over 5483 word-forms of verbs. The process of learning rules is the same as described earlier. Only the MSDs that have more than 100 examples were included in the process of learning rules. An example of exceptions and rules for analyzing the participle form of the Macedonian verbs is given in Figure 11.

```
vmpa3sm_n__e([d,o,b,i,l],[d,o,b,i,e]):- !.
vmpa3sm_n__e ([z,e,l],[z,e,m,e]):- !.
vmpa3sm_n__e ([r,a,z,b,r,a,l],[r,a,z,b,e,r,e]):- !.
vmpa3sm_n__e ([i,s,p,i,l],[i,s,p,i,e]):- !.

vmpa3sm_n__e(A,B):-mate(A,B,[p],[p],[z,n,a,l],[z,n,a,e]),!.
vmpa3sm_n__e (A,B):-mate(A,B,[],[],[k,a,l],[k,a]),!.
vmpa3sm_n__e(A,B):-mate(A,B,[],[],[l,e,g,o,l],[l,e,z,e]),!.
vmpa3sm_n__e(A,B):-mate(A,B,[],[],[d,a,l],[d,a,d,e]),!.
vmpa3sm_n__e (A,B):-mate(A,B,[],[],[n,a,l],[n,e]),!.
vmpa3sm_n__e (A,B):-mate(A,B,[],[],[i,l],[i]),!.
```

Figure 11. Exceptions and rules in analysis of participle form of Macedonian verbs

Again, there were incorrectly lemmatized verbs and the reasons for that are the same as for the adjectives. Figure 12 shows some incorrectly lemmatized verbs.

```
dobija  Vmia3p--n-----e  dobie  |dobi|  ERR
razberat  Vmip3p--n-----e  razbere  |razberi|  ERR
dozna  Vmia2s--n-----e  doznae  |dozne|  ERR
umrat  Vmip3p--n-----e  umre  |umri|  ERR
zaprea  Vmia3p--n-----e  zapre  |???|  ERR
odigraa  Vmia3p--n-----e  odigra  |???|  ERR
zagriza  Vmia3s--n-----e  zagrize  |???|  ERR
sporedat  Vmia3p--n-----e  sporedi  |???|  ERR
nadvladea  Vmii2s--n-----p  nadvladee  |???| ERR
nagovorila  Vmii3s--n-----p  nagovara  |???|  ERR
```

Figure 12. Incorrectly lemmatized verbs

The verbs *dobija (to get)*, *razberat (to understand)*, *dozna (to find out)* and *umrat (to die)* were incorrectly lemmatized because the rules applied to them did not generate the right base-forms. For the verbs *zaprea (to stop)*, *odigraa (to play)* and *zagriza (to bite)* there were no rules that could be applied and the errors in the last three verbs – *sporedat (to compare)*, *nadvladea (to dominate)*, *nagovorila (to persuade)* – were due to errors in the annotation (wrong MSDs).

The average accuracy of the obtained rules for morphological analysis of verbs, tested with 10-fold cross validation, is 91.65% (Table 2).

For the process of morphological synthesis the data were transformed into triplets lemma-wordform-msd and rules were learned over it. Some of the rules and exceptions for synthesizing the participle form of the Macedonian verbs are presented in Figure 13.

```
vmpa2sm_n__e([s,v,r,t,i],[s,v,r,t,e,l]):- !.
vmpa2sm_n__e ([r,a,z,b,e,r,e],[r,a,z,b,r,a,l]):- !.
vmpa2sm_n__e ([p,r,i,v,r,z,e],[p,r,i,v,r,z,a,l]):- !.
vmpa2sm_n__e ([p,o,s,t,o,i],[p,o,s,t,o,e,l]):- !.

vmpa2sm_n__e(A,B):-mate(A,B,[],[],[z,n,a,e],[z,n,a,l]),!.
vmpa2sm_n__e(A,B):-mate(A,B,[],[],[v,i,d,i],[v,i,d,e,l]),!.
vmpa2sm_n__e (A,B):-mate(A,B,[],[],[e,e],[e,a,l]),!.
vmpa2sm_n__e(A,B):-mate(A,B,[],[],[l,e,z,e],[l,e,g,o,l]),!.
vmpa2sm_n__e(A,B):-mate(A,B,[],[],[d,a,d,e],[d,a,l]),!.
vmpa2sm_n__e (A,B):-mate(A,B,[],[],[a],[a,l]),!.
vmpa2sm_n__e (A,B):-mate(A,B,[],[],[n,e],[n,a,l]),!.
vmpa2sm_n__e (A,B):-mate(A,B,[],[],[i],[i,l]),!.
vmpa2sm_n__e (A,B):-mate(A,B,[],[],[i,e],[i,l]),!.
```

Figure 13. Exceptions and rules in synthesizing the participle form of Macedonian verbs

The average accuracy of the obtained rules for the synthesis of verbs, tested with 10-fold cross validation, is 95.71%. (Table 2).

	Analysis	Synthesis
Accuracy (%)	91.65	95.71
Standard deviation	1.383468	1.424956

Table 2. Average accuracy and standard deviation of the rules for analysis and synthesis of Macedonian verbs

3.1.3. Nouns

The process of morphological analysis and synthesis of nouns was the same as for adjectives and verbs. Some exceptions and rules for analyzing the common nouns of feminine gender are presented in Figure 14.

10-fold cross validation was used to test the accuracy of the rules for morphological analysis and synthesis of nouns. The average accuracy obtained was 97.01% (Ivanovska et al., 2005). Morphological synthesis of the nouns resulted with slightly smaller average accuracy of 94.81%.

```
ncfpnn([r,a,s,p,r,a,v,i,i],[r,a,s,p,r,a,v,a]):- !.
ncfpnn([s,t,r,u,i],[s,t,r,u,j,a]):- !.
ncfpnn([r,a,c,e],[r,a,k,a]):- !.
ncfpnn([n,o,z,e],[n,o,g,a]):- !.

ncfpnn(A,B):-mate(A,B,[],[],[s,t,i],[s,t]),!.
ncfpnn(A,B):-mate(A,B,[],[],[i,i],[i,j,a]),!.
ncfpnn(A,B):-mate(A,B,[i,d],[i,d],[i],[j,a]),!.
ncfpnn(A,B):-mate(A,B,[],[],[i],[a]),!.
```

Figure 14. Exceptions and rules in analysis of common Macedonian nouns of feminine gender

Some exceptions and rules for synthesis of common neuter nouns in plural are presented in Figure 15.

```
ncnpny([d,e,t,e],[d,e,c,a,t,a]):- !.
ncnpny([z,i,v,o,t,n,o],[z,i,v,o,t,n,i,t,e]):- !.
ncnpny([b,e,b,e],[b,e,b,i,n,j,a,t,a]):- !.

ncnpny(A,B):-mate(A,B,[p,o],[p,o],[e],[i,n,j,a,t,a]),!.
ncnpny(A,B):-mate(A,B,[],[],[c,e],[c,a,t,a]),!.
```

Figure 15. Exceptions and rules in synthesizing the common neuter nouns in plural

4. Conclusions

In this paper we have addressed the problem of morphological analysis and synthesis of Macedonian adjectives, verbs and nouns. To learn rules for morphological analysis and synthesis we used word-forms from the Macedonian translation of Orwell's "1984".

We successfully applied the ILP system Clog for learning rules for analysis and synthesis of Macedonian words. The obtained average accuracies of the learned rules for analysis and synthesis of adjectives are 93.12% and 82.77%, respectively; for verbs – 91.65% and 95.71%; and for nouns - 97.01% and 94.81%.

The high accuracies achieved are encouraging for further research as well as practical use. For example, morphological analysis/synthesis could be used in web search engines.

To this end the rules for morphological analysis and synthesis need to be connected with PoS tagging to perform lemmatization. Although preliminary research has been done on learning PoS tagging for Macedonian (Vojnovski, 2005), further work with a larger and better annotated corpus is needed. Once we have a PoS tagger we can use the rules learned in this work for lemmatization, which is relevant both for practical use as well as for example another syntactic annotation of Macedonian language resources.

5. References

Dimitrova, L., T. Erjavec, N. Ide, H.-J. Kaalep, V. Petkevič, and D. Tufis. 1998. Multext-east: Parallel and comparable corpora and lexicons for six central and eastern european languages. In *Proceedings of the COLING-ACL '98*, pages 315–319, Montreal, Quebec, Canada. http://nl.ijs.si/ME/.

Erjavec, T. 2004. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004)*, ELRA, Paris., pp. 1535-1538.

Erjavec, T., and S.Džeroski, 2004. Machine Learning of Morphosyntactic Structure: Lemmatizing unknown Slovene words. *Applied Artificial Intelligence* 18(1), 2004, pp. 17-41.

Ivanovska, A., K.Zdravkova, S.Džeroski, and T.Erjavec, 2005. Learning rules for morphological analysis and synthesis of Macedonian nouns. In *Proceedings of SiKDD 2005 (Conference on Data Mining and Data Warehouses)*, Ljubljana, Slovenia, pp. 195-198.

Koneski, B., 1952. Grammar of Standard Macedonian. *Prosvetno Delo*, first edition, in Macedonian

Koneski, B., 2004. Grammar of Macedonian Language. *Prosvetno Delo*, in Macedonian

Manandhar, S., S.Džeroski, and T.Erjavec, 1998. Learning multilingual morphology with CLOG. In *Proceedings of Inductive Logic Programming: 8th International Workshop (ILP-98)*, Number 1446 in Lecture Notes in Artificial Intelligence, ed. D. Page, pages135-144, Berlin, Springer-Verlag

Mooney, R. J., and M. E. Califf. 1995. Induction of first-order decision lists: Results on learning the past tense of English verbs. *Journal of Artificial Intelligence Research* 3(1):1–24.

Petrovski, A., 2005. About Macedonian Computational Dictionary. In *Proceedings of BCI2005*, Ohrid, Macedonia, pp. 76-81

Vojnovski, V., S. Džeroski, and T. Erjavec, 2005. Learning PoS tagging from a tagged Macedonian text corpus. In *Proceedings of SiKDD 2005 (Conference on Data Mining and Data Warehouses)*, Ljubljana, Slovenia, pp. 199-202.

Dodatne dvoumnosti zaradi popustljivosti analizatorja pri analizi slovenskih stavkov

Peter Holozan

Amebis d. o. o.
Bakovnik 3, 1241 Kamnik
peter.holozan@amebis.si

Povzetek

Zaradi neknjižnih oblik, ki se pojavljajo v besedilih, je smiselno, da analizator pri stavčni analizi predvidi, da bi lahko bila v stavku tudi kakšna od pogostih neknjižnih oblik. Ta popustljivost analizatorja pa lahko pripelje do dvoumnosti, kjer je potem možno, da v stavku je neknjižna oblika ali pa je ni in gre za drug pomen. V takih primerih je treba določiti, katera od možnosti je bolj verjetna.

Additional ambiguities caused by the Slovenian analyser's permissiveness

Due to non-standard forms occurring in texts, it makes sense for an analyser to anticipate that a sentence could contain a common non-standard form. However, the analyser's permissiveness can lead to ambiguity where it is possible to analyse a sentence either as containing a non-standard form and having one meaning or as not containing a non-standard form and having another meaning. In such cases, it is necessary to choose which is the likelier possibility.

1 Uvod

Pri stavčni analizi se pojavi problem, da se v besedilu, ki ga je treba analizirati, lahko pojavijo neknjižne oblike. Pri slovničnem pregledovalniku je zelo pomembno, da te oblike čim bolje ugotovimo, da lahko pregledovalnik nanje opozori, pa tudi pri strojnem prevajanju je zaželeno, da je čim odpornejše na neknjižne oblike in napake.

Analizator, ki ga uporabljamo v podjetju Amebis, tako v slovarju vsebuje najpogostejše neknjižne oblike (*življenski*, *nebo* namesto *ne bo*, *jest* namesto *jaz*), pri manj pogostih pa jih označevalnik ugiba sproti, če je beseda neznana (npr. vezava *ne* s povednim sedanjikom glagola). Označevalnik tudi dodaja tipično zamenjane sklone (dajalnik in mestnik) in števila (dvojina in množina).

Težava se pojavi, kadar je nek stavek mogoče analizirati z uporabo popustljivosti ali pa brez tega, ker se mora potem analizator odločiti, katere možnost je verjetnejša.

2 Namen članka

V članku bodo najprej opisani nekateri tipični neknjižni pojavi, ki se pojavljajo v slovenskih besedilih, in načini, kako jih analizator lahko upošteva.

Sledil bo opis iskanja stavkov, pri katerih pride zaradi upoštevanja neknjižnih oblik do dvoumnosti.

Te stavke bom razvrstil po značilnih skupinah in poskusil najti metode, kako izbrati analizo, ki je bolj verjetna.

Na koncu bo predstavljeno, kako pri teh primerih delujeta prevajalni sistem Presis in slovnični pregledovalnik BesAna.

3 Nekatere tipične neknjižne oblike v slovenskih besedilih

3.1 Neknjižne besede

Te besede (npr. *življenski*, *cuker*, *jest*) so rešene tako, da so vnesene v slovar s posebno oznako in povezane na ustrezne pomene. Ta način ima dodatno prednost, ker zmanjšuje verjetnost, da bi se neknjižna beseda po pomoti vnesla v slovar kot knjižna, saj je tam že vnesena, le da ima posebno oznako.

3.2 Neknjižno sklanjanje in spreganje

Najpogostejše take oblike (*zadanemo*, *otroci* za orodnik množine) so vnesene že v slovar. Dodatno označevalnik pri označevanju išče še nekatere oblike, kot so na primer pogovorne oblike opisnih deležnikov na -l (*gledu* namesto *gledal*) in podaljševanje osnove pri sklanjanju imen (*Mirota*, *Lukata*).

3.3 Zanikanje povednega sedanjika

Pogosto postaja tudi, da se povedni sedanjik glagola zanika tako, da se predpona ne- prilepi k glagolu, podobno kot je to pri pridevnikih in samostalnikih (*nevem*). To je zelo pogosto pri oblikah za prihodnjik in pogojnik glagola *biti*: *nebom*, *nebo*, *nebi*. Te oblike so za glagol *biti* vnesene že v slovar, za druge glagole pa jih pri neznanih besedah poskusi najti označevalnik.

3.4 Neknjižna uporaba povratnih svojilnih zaimkov

V knjižni slovenščini je treba v primeru, ko se svojina nanaša na osebek (izjema je splošno lastništvo (Herrity, 2000)), uporabiti povratni svojilni zaimek, kar pa se pogosto opušča (*Popravljam moj članek.*).

V teh primerih sam analizator nima težav, dodatno le označi, kadar se nepovratni svojilni zaimek ujema z osebkom, da lahko potem na podlagi tega slovnični pregledovalnik opozori uporabnika na morebitno neknjižno uporabo.

3.5 Napol vikanje

Pri napol vikanju je pomožni glagol v množini kot pri vikanju, opisni deležnik je pa v ednini kot pri tikanju (*Kdaj boste prišla?*, *Kdaj boste prišel?*).

3.6 Zanikanje s tožilnikom

Predmet v tožilniku se pri zanikanju stavka spremeni v rodilnik (pri glagolu *biti* v pomenu *obstajati* preide v

rodilnik iz imenovalnika tudi osebek (*Sosede ni doma.*)). To se pogosto opušča.

Amebisov analizator pri zanikanih stavkih sprejema tako predmete v tožilniku kot v rodilniku, vendar prve posebej označi v analizi, da lahko slovnični pregledovalnik potem ugotovi, da gre za možno neknjižno uporabo.

3.7 Neuporaba dvojine

Do tega lahko pride že pri samem številku (*pred dvemi leti*), te neknjižne oblike so vnesene v slovar. Druga možnost pa je, da je v neknjižnem številu samostalnik oz. pridevnik (*dve ure*). Da bi bil analizator v takih primerih lahko uspešen, označevalnik vse oblike za množino podvoji, da lahko pomenijo tudi neknjižno dvojino.

3.8 Zamenjevanje mestnika z dajalnikom

Do tega prihaja pri moškem spolu: *na velikemu vrtu*. Označevalnik dajalniku ednine moškega spola pripiše še možnost, da gre za neknjižni mestnik, s čimer potem analizator uspešno analizira take primere.

3.9 Zamenjevanje nedoločnika in namenilnika

V knjižni slovenščini mora biti ob glagolih premikanja namenilnik, ob drugih glagolih pa nedoločnik. V pogovornem jeziku se nedoločnik pogosto zamenja z namenilnikom (*Moram delat.*), včasih pa zaradi hiperkorektnosti (ker se posebej trudi, da ne bi pozabil na nedoločnike) kdo uporabi tudi nedoločnik namesto namenilnika ob glagolih premikanja (*Grem pisati.*).

4 Iskanje stavkov, pri katerih pride do dvoumnosti

Vhodni korpus, ki smo ga uporabili pri iskanju stavkov, je seznam primerov, ki so jih prevajali uporabniki spletne različice slovensko-angleškega strojnega prevajalnika Presis. Ker so ti primeri popolnoma nelektorirani (in so vnašalci pogosto tudi namerno uporabljali pogovorni jezik), je med njimi zelo veliko neknjižnih oblik. Dopolnjen je še z različnimi primeri, na katere smo naleteli pri ročnem preverjanju delovanja strojnega prevajanja in iskanja slovničnih napak.

Za te primere smo se odločili, ker smo želeli v besedilih čim več neknjižnih oblik, ki se pojavljajo pri pisanju, zato smo se izognili uporabi korpusa Fida, ki sicer vsebuje tudi precej nelektoriranih besedil, vendar so tudi pri teh besedilih pisci večinoma izobraženi in se trudijo pisati čim bolj knjižno, kar zelo zmanjša število neknjižnih oblik.

Pokazalo se je, da je zaradi same zasnove analizatorja relativno zapleteno najti take pare analiz, kjer pride do dodatne dvoumnosti zaradi upoštevanja neknjižnih oblik. Najenostavnejša rešitev bi bila, da bi za vsak stavek izbrali vsakič po eno možno analizo in potem uporabili slovnični pregledovalnik. Dvoumni stavki bi bili tisti, pri katerih bi se pojavila vsaj po ena analiza z in brez pripomb pregledovalnika. Žal se je pokazalo, da je že število analiz včasih zelo veliko, zato jih analizator potem poreže. Pri zaključku analize se izgubijo tudi nekateri vmesni rezultati, zato bi bilo najlaže ponoviti kar celotno analizo. Vendar je analizator časovno precej zahteven in bi tak način dela bil zelo počasen.

Zato je bil s pomočjo pogojnega prevajanja raje dodan v analizator del kode, ki sproti ugotavlja, ali je bila v analizi uporabljena kakšna označena neknjižna oblika ali ne. Če se pri analizi stavka pojavijo analize obeh vrst, je stavek dvoumen.

Pri napol vikanju in svojilnih zaimkih na možno dvoumnost opozarja že slovnični pregledovalnik, zato smo tam uporabili kar njegov rezultat.

5 Analiza primerov

Seznam možnih dvoumnosti je bil ročno pregledan in dvoumnosti razvrščene po tipičnih skupinah.

5.1 Možna neuporaba povratnega svojilnega zaimka

Pri prvi in drugi osebi je sklepanje o manjkajočem povratnem svojilnem zaimku precej zanesljivo (*Poklical sem mojo prijateljico.*). Problematične dvoumnosti se pojavijo le pri pogojnih stavkih z izpuščenim osebkom, kjer ne moremo biti prepričani o osebi.

Rad bi opravičil mojega brata.
Rad vam bi predstavil mojega soseda Ivana.
Rad bi vam predstavil moj hobi.
V spomin na mojega deda bi ga želel obnoviti.
Na koncu bi se rad opravičil zaradi moje angleščine.

Primer 1. Primeri pogojnih stavkov z možno neuporabo povratnega svojilnega zaimka

Če bi se zanesli na to, da v besedilu ni neknjižnih uporab, bi lahko v teh stavkih sklepali na to, da osebek ni v prvi osebi. Kot pa kaže primer 1, je največkrat bolj verjetno, da manjka povratni svojilni zaimek.

Nekoliko drugačen je položaj pri tretji osebi.

Janez je ljubil njegovo ženo.
Okrog njenega grebena je krožila cela jata vodnih ptičev.
Irski raziskovalci so raziskali pomembnost zajtrka pri njihovih študentih
Vstopila je v njene hlačke.
Uporabnik lahko vnaprej definira opozorilni klic na njegovo telefonsko številko.
lagal ji je glede njegovega premoženja
da bo tako dober kot njegov oče

Primer 2. Primeri svojilnega zaimka in osebka v tretji osebi

V primeru 2 se vidi, da je pri tretji osebi veliko več možnosti, da je stavek res napisan knjižno (kar pa bi bilo največkrat mogoče ugotoviti šele iz sobesedila). To je potem treba upoštevati tudi pri slovničnem pregledovalniku, ki mora v teh primerih veliko manj opozarjati kot pri prvi in drugi osebi.

5.2 Napol vikanje

Pri napol vikanju za ženski spol pride do dvoumnosti zaradi tega, ker se oblika prekrije z drugo osebo dvojine za srednji spol. Na srečo pa je (razen morda v kakšni znanstveni fantastiki) zelo malo verjetno, da bi kdo

ogovarjal skupino bitij srednjega spola, tako da se da zelo zanesljivo sklepati, da gre vedno za napol vikanje.

Kam ste šla potem?
ki ste mi ga sporočila
S katero pošiljko ste poslala sete.
Nič mi niste pisala.
Ali ste že kdaj obiskala mladinski hotel?
Boste ponudbo potrdila?
vi ste dolgočasna

Primer 3. Primeri napol vikanja

Vendar pa tudi tukaj lahko pride do dodatnih dvoumnosti:

Vi ste ženska.

Primer 4. Primer napačno ugotovljenega napol vikanja

V primeru 4 pride do dvoumnosti zaradi tega, ker besedo *ženska* prepozna tudi kot pridevnik, kar bi pomenilo napol vikanje. V teh primerih mora analizator zato dati prednost analizi s samostalnikom.

5.3 Analize z odvečno dvojino

Zaradi dodatne dvojine v označevalniku se analize lahko povečajo še za (neknjižno) dvojino. Pri imenovalniku do tega ne pride, ker se mora ujemati še z glagolom, se pa to zgodi pri predložnih zvezah.

Pred leti me je povozil traktor.

Primer 5. Nepotrebno dodana dvojina

V primeru 5 tako analizator najde tudi možnost, da bi se to zgodilo pred dvema letoma.

Te možnosti skoraj vedno odvečne, kadar pa niso, je to nemogoče ugotoviti brez sobesedila. Na srečo pri prevajanju v angleščino potem to dvoumnost izgubimo, slovnični pregledovalnik pa tudi ne opozarja nanjo.

5.4 Dvoumnost zaradi zanikanja s tožilnikom

Popustljivost analizatorja pri zanikanju s tožilnikom prinese težave pri ženskem spolu.

Ne gledam slike.

Primer 6. Zanikanje s tožilnikom ali ne

Pri primeru 6 je tako število slik odvisno od tega, ali je predmet v rodilniku ali v tožilniku. Tega brez sobesedila ni mogoče ugotoviti, Amebisov analizator zato da prednost knjižni obliki.

Možna rešitev te težave bi bila, da bi program pogledal celotno besedilo in preštel vse nedvoumno zanikane predmete in iz tega sklepal, kaj pisec večkrat uporablja.

5.5 Nebo

Lepljenje nikalnice *ne* ob obliko glagola *biti* za prihodnjik je vedno pogostejši pojav. V večini primerov jo lahko opazi že črkovalnik, zaplete pa se pri tretji osebi ednine, kjer se neknjižna oblika prekrije z relativno pogostim samostalnikom *nebo*.

To nebo poceni.
Oblačno nebo.
lepo nebo v odboju
oljnato nebo
Nad nami je nebo.
nebo v bolečini

Primer 7. Dvoumnosti z besedo *nebo*

V večini primerov je pravi pomen nebo, nekatere primere pa je mogoče razrešiti tudi z dodatnimi pravili. Tako ima glagol *poceniti* označeno, da je vezava s predmetom *nebo* malo verjetna, tako da pride v tem primeru na prvo mesto glagol *biti*.

Zamenjava pri uporabnikih nebo problem.

Primer 8. Zanimiva dvoumnost

V primeru 8 je stavek, kjer na prvi pogled ni bilo videti, da bi analiza lahko bila dvoumna. Pokaže pa se, da je mogoče stavek razumeti kot: *Nebo zamenjava problem pri uporabnikih.*, kjer *zamenjava* oblika glagola *zamenjavati* (drugi možnosti sta še glagol *zamenjati* in samostalnik *zamenjava*). Možna rešitev za ta stavek bi bila, da bi se zmanjšale verjetnosti za analize, ki vsebujejo glagol *zamenjavati*, ki je relativno redek.

5.6 Jest

V pogovornem jeziku se *jest* pogosto pojavlja kot nadomestek za *jaz*. Dvoumnosti povzroča dejstvo, da je beseda tudi namenilnik glagola *jesti*, kar povzroči, da zelo pogosto postanejo dvoumni stavki, ki vsebujejo glagole premikanja.

Jest grem.
Jest grem na ples.
jest grem domov
grem samo jest
grem malo jest
jest grem pa v kvalifikacije
šel sem jest domov

Primer 9. Dvoumnosti z besedo *jest*

Te dvoumnosti je žal brez sobesedila zelo težko razrešiti. Tudi tukaj bi bil verjetno pravi način to, da bi analizator iz okoliškega besedila ocenil, koliko pisec uporablja pogovorni jezik.

5.7 Dvoumnosti zaradi domnevnih namenilnikov namesto nedoločnikov

Kar nekaj pogostih besed se prekriva z možnimi namenilniki (*spet*, *izpit*, *past*, *rit*, *pet*, …). V kombinaciji z glagoli, ki se lahko vežejo tudi nedoločniki (ob modalnih glagolih, kjer je dvoumnosti manj, ker se običajno ne vežejo s čim drugim, sta taka primera pogosto *imeti* (*Imam pisati članek.*) in *biti* (*Delati je naporno.*, *Živeti je pesem.*)).

Danes imamo spet računalništvo.
Jutri moram spet v službo
Jutri imam izpit.
Daj pet sto.
Ima veliko rit.
moja prijateljica ima božansko rit
spet imam moje zobe
tu je past domovine

Primer 10. Možni neknjižni namenilniki

V veliki večini primerov je prava analiza tista brez namenilnika (na splošno se *imeti* in *biti* ne vežeta pogosto z nedoločnikom), tako da je znižanje verjetnosti analizam z njim dovolj zanesljiva metoda za ugotavljanje prave analize.

Ta statut je veljavno sprejet.
Dobro razvit je turizem.

Primer 11. Prekrivanje namenilnika in deležnika na -t

Včasih se z namenilnikom prekrijejo tudi deležniki na -t. Tudi v teh primerih je verjetnost, da gre res za neknjižno obliko, po navadi majhna.

5.8 Mam in mamo

V pogovornem jeziku se pojavlja krajšanje glagola *imeti* v *meti*. Do dvoumnosti lahko pri povednem sedanjiku pride zaradi prekrivanja z oblikami samostalnika *mama*.

poškodovan prst mam
štiri ure mam časa
in mamo Ivano

Primer 12. Pogovorna oblika *imeti*

Ugotavljanje pravega pomena je tukaj precej zoprno. Možna rešitev je, da se prepove, da je *mama* desni prilastek samostalnika *ura*, kar pa razreši le delček primerov.

5.9 Radio in radij

V pogovornem jeziku se samostalnik *radio* pogosto piše kot *radijo*. Na to tudi ne (z izjemo imenovalnika in tožilnika ednine) opozarja črkovalnik, ker se oblike prekrivajo z oblikami samostalnika *radij*.

popravilo radijev
Težava je pri rezanju radija.
Pri krivljenju kabla je upoštevati minimalne radije krivljenja.
Oglašujemo preko radija in časopisov.
veliki radiji
nekotirani notranji radiji so taki
kar se trenutno vrti na radiju
Vsi nekotirani radiji.
dobre dogovore smo sklenili tudi z radiji in časopisnimi hišami
Zanima me, če si lahko na svojo spletno stran dodam povezavo do vašega radija?
Že cel mesec oglašuje po radiju.
slišal sem na radiju

Primer 13. Prekrivanje oblik radia in radija

Te dvoumnosti je zelo težko razrešiti, edini možen način je z uporabo pomenov. Tako se na primer da dodati pravilo, da se *popravilo* ne veže s pomenoma *radij (element)* in *polmer*.

5.10 Vso

Relativno pogosta je neknjižna uporaba oblike *vso* namesto *vse* za tožilnik ednine srednjega spola pridevniškega zaimka *ves*.

V prostorih je vso pohištvo
, da ponudi vso bogastvo vonjav.
ruši vso dosedanje delo

Primer 14. Dvoumnosti z besedo *vso*

Težavo povzroči to, da je *vso* lahko tudi posamostaljen tožilnik ednine ženskega spola. Stavke iz primera 14 analizator zato razume tudi tako, da je *vso* v njih predmet v tožilniku, resnični predmet pa osebek.

Vsi trije primeri, ki so bili najdeni v vzorcu, so taki, da bi bilo bolje dati prednost analizi, kjer je v besedilu neknjižna oblika. Delno bi se dalo to reševati po posameznih primerih glagolov (ni posebno verjetno, da bi pohištvo kaj pojedlo), smiselno bi bilo pa preizkusiti tudi splošnejše pravilo, da v primerih, kjer besedi *vso* takoj sledi osebek v ednini srednjega spola, zmanjšamo verjetnost tej analizi. Nadaljnje preizkušanje na daljših primerih bo pokazalo, če je tako pravilo dovolj zanesljivo za uporabo.

6 Primeri uporabe

Analizator v prevajalnem sistemu Presis že zna razrešiti nekatere od zgoraj naštetih dvoumnosti.

Vhod	Izhod
To nebo poceni.	This won't be cheap.
Oblačno nebo.	Cloudy sky.
Jest grem.	I go to eat.
Jest gledam televizijo.	I watch television.

Tabela 1: Primeri strojnih prevodov prevajalnika Presis

Tudi slovnični pregledovalnik BesAna uporablja isti analizator. Zato tudi ta v prvem primeru takoj svetuje, da bi bilo bolj knjižno napisati *To ne bo poceni.*

7 Literatura

Herrity, Peter, 2000. *Slovene: A Comprehensive Grammar*. Routledge.

Žagar, France, 1987. *Pouk slovenske slovnice in pravopisa v višjih razredih osnovne šole.* Založba Obzorja Maribor, prvi natis.

Žagar, France, 1991. *Slovenska slovnica in jezikovna vadnica*. Založba Obzorja Maribor, šesta dopolnjena in razširjena izdaja.

AVTOMATIČNO PREPOZNAVANJE LASTNIH IMEN

Mihael Arčan*, Špela Vintar‡

*mihael_arcan@yahoo.de
‡spela.vintar@guest.arnes.si

Filozofska fakulteta, Univerza v Ljubljani
Arškerčeva 2, 1000 Ljubljana

Named Entity Recognition

The paper deals with Named Entities in German and Slovene texts. We first describe Named Entities from a linguistic viewpoint, where a typology of Named Entities and a theoretical framework is given. The second part of the paper describes different methods of Named Entity Recognition (NER). Since names, like all nouns, are capitalized in German, the task of automatic recognition is not trivial. In Slovene, capitalization is a useful indicator for names. Two main methods for NER are employed: The grammatical rule method utilizes both the internal structure of names as well as their context. The second method is statistical, where frequencies of Named Entities are compared to other parts-of-speech. In the described experiment both methods, but mainly the first, were put to use. While simple or single word names, especially geographical names, are conventionally recognized via lists, multi-word names are more difficult to recognize. The system was evaluated with F-Measure with the result 0,64 for German and 0,74 for Slovene.

Povzetek

Prispevek se ukvarja z lastnimi imeni v nemščini in slovenščini. Na začetku bodo lastna imena obravnavana z lingvističnega vidika, kjer v uvodnih razdelkih podamo razčlenitev lastnih imen in podroben opis. Drugi del prispevka se posveti različnim metodam prepoznavanja lastnih imen. Kot podskupino samostalnikov jih je v nemškem jeziku vizualno težko prepoznati, ker se tudi samostalniki pišejo z veliko začetnico. V slovenskem jeziku pa se pišejo z veliko začetnico le lastna imena, kar olajša njihovo prepoznavanje. Za prepoznavanje lastnih imen se uporabljata dve metodi. Metoda s slovničnimi pravili se poslužuje tako notranje strukture kot tudi okolice lastnih imen. Pri drugi, statistični metodi pride do uporabe statističnih računov, in sicer pogostost nastopa lastnih imen v relaciji z ostalimi besednimi vrstami. V pričujoči raziskavi sta bili uporabljeni obe metodi, poudarek pa je bil na metodi pravil. Izkazalo se je, da največ težav povzročajo večbesedna lastna imena. Prepoznavanje preprostih ali enobesednih lastnih imen, posebno zemljepisnih, se pri vseh sistemih rešuje s pomočjo seznamov. Na koncu je bil izračunan uspeh raziskave z vrednostjo F, ki je za nemško besedilo znašala 0,64 in za slovensko besedilo 0,74 točke.

1. Uvod

Dandanes si sveta ne moremo več predstavljati brez računalnikov. Veliko ljudi se niti ne zaveda, da so jezikovne tehnologije del vsakdanjega življenja. Pri pisanju besedila nam pri prepoznavanju napak pomaga črkovalnik, ki napake tudi avtomatično popravi. Uporaba elektronskih slovarjev nam lajša iskanje neznanih besed. Strojni prevajalniki, kot sta BabelFish[1] in Presis[2], prevedejo želeno besedilo v različne jezike. Vsem tem pripomočkom je skupno to, da uporabljajo izčrpne baze podatkov.

Iskanje lastnih imen v nemškem jeziku predstavlja večji izziv, ker se v nemščini vsi samostalniki pišejo z veliko začetnico. Zato smo se v tej raziskavi osredotočili na prepoznavanje lastnih imen v nemškem jeziku. Čeprav se v slovenskem jeziku pišejo z veliko začetnico le lastna imena, ta lastnost za prepoznavnost lastnih imen velikokrat ne zadošča.

2. Klasifikacija lastnih imen

V slovenskem in nemškem jeziku se lastna imena delijo na dve glavni skupini:

- osebna lastna imena,
- zemljepisna lastna imena.

Med lastna imena uvrščamo tudi imena stvaritev, organizacij, delovnih skupnosti ipd. (Slovenski pravopis 1994, 22-24.)

Posebna skupina lastnih imen so imena proizvodov. Ker so jim imena podeljena, imajo lastnosti lastnih imen, ker pa hkrati razvrščajo proizvode v različne kategorije, jih lahko uvrščamo tudi med občna imena (Shippan 1992, 65).

Ker so se prvi programi za prepoznavanje lastnih imen razvijali za prepoznavanje imen v angleškem jeziku, je pomembno tudi dejstvo, da v angleškem jeziku v skupino lastnih imen sodijo tudi imena dni (Monday), mesecev (May) in praznikov (Christmas). Zanimivo je, da v angleščini med lastna imena sodijo tudi poimenovanja valut, datumi in odstotki (Roth 2002, 5; Bekavac 2005, 19).

[1] Babelfisch Altavista, http://babelfisch.altavista.com.
[2] Presis, http://presis.amebis.si.

3. Prepoznavanje lastnih imen

Na mednarodni konferenci o razumevanju sporočil MUC-7[3] leta 1998 so se med sabo pomerili programi, ki delujejo na osnovi slovničnih pravil (PNF, BSEE, LaSie, LT TTT), in programi, ki za prepoznavanje lastnih imen uporabljajo statistične metode (IdentiFinder, MENE) (Roth 2002, 25-82).

Za prepoznavanje lastnih imen lahko torej uporabljamo dve osnovni metodi.

Prva temelji na slovničnih pravilih, vanjo pa sodita interna in eksterna evidenca. Pri interni evidenci gre za tipične lastnosti in zgradbo imen (*d. o. o., d. d., Inštitut X, X vrh, X jezero*). K njej pripisujemo tudi začetnice imen (*George W. Bush*). Eksterna evidenca pa uporablja sobesedila oz. okolice lastnih imen (*Prof. X Y, Gospa X Y ali X Y ml.*). Pri eksterni evidenci se lahko uporabljajo tudi seznami indikatorjev, ki napovedujejo lastna imena.

Druga metoda je statistična in uporablja izključno pojavitve besed ali besednih vrst v besedilu.

Na prvi pogled se zdi, da bi bil idealen sistem sinteza obeh prijemov, vendar se na konferenci MUC-7 ni izkazalo tako. Pokazalo se je, da so za hitro vzpostavitev sistema primernejše statistične metode. Velika prednost tovrstnih sistemov je tudi ta, da so jezikovno neodvisni. V primerjavi z metodo pravil, pri kateri moramo napisati vsa slovnična pravila in seznam indikatorjev za vsak jezik posebej, se zdi statističen pristop ustreznejši. Njegova slabost pa je velika odvisnost od označenega korpusa, zaradi česar se prednosti in slabost obeh metod izenačita.

4. Uporabljene metode za prepoznavanje lastnih imen

Za raziskavo smo izdelali manjši slovenski in nemški korpus turističnih besedil s številnimi zemljepisnimi imeni. Izbrali smo turistični vodnik *Pozdrav iz Slovenije* in njegov prevod v nemški jezik *Grüße aus Slowenien*. S pomočjo korpusa smo iskali lastna imena v slovenskem in nemškem jeziku, stavčna poravnava besedil s programom Déjà Vu[4] pa je omogočila še iskanje imen po vzporednem korpusu.

<stavek><sl>"Čez Julijske Alpe teče razvodje med Jadranskim in Črnim morjem."</sl><ge>"Über die Julischen Alpen verläuft auch die Wasserscheide zwischen der Adria und dem Schwarzen Meer."</ge></stavek>

Tabela 1: Primer vzporednega korpusa.

Korpusu smo dodali morfološko-sintaktične oznake. Za označevanje slovenskega dela je poskrbelo podjetje Amebis, nemški del pa so označili na univerzi v Stuttgartu. Za izdelavo sistema za avtomatično prepoznavanje lastnih imen smo uporabili programski jezik Perl.

[3] MUC - Message Understanding Conference, http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/overview.html.

[4] Déjà Vu: http://www.atril.com.

4.1. Velika začetnica

Glavni poudarek v raziskavi je bil na veliki začetnici. V slovenskem jeziku nam je ta v veliko pomoč pri iskanju lastnih imen, v nemškem jeziku pa nastanejo težave, ker se tudi ostali samostalniki pišejo z veliko začetnico. Tabela 2 prikazuje razmerje med obema jezikoma glede vseh besed v izdelanem korpusu in besed, zapisanih z veliko začetnico.

	slov.	nem.
pojavnica (ang. token)	14.975	13.874
velika začetnica	2.324	4.832
velika zač. v [%]	15,92	34,83

Tabela 2: Razmerje med pojavnicami in besedami z veliko začetnico v slovenščini in nemščini.

Iz tabele 2 je razvidno, kako samostalniki v nemškem jeziku povečujejo delež besed, zapisanih z veliko začetnico. Drugače je v slovenskem jeziku, kjer se z veliko pišejo le besede na začetku stavka in lastna imena.

V tabeli 3 je seznam vseh besed z veliko začetnico, najdenih v dveh vzporednih stavkih iz korpusa.

n	sl.	nem.	n	nem.
1	Povojni*	Die*	15	Einwohner*
2	Nagla*	Entwicklung*	16	*Italien*
3	*Kopra*	Kriegsende*	17	Gebiet*
4	*Izole*	Positivem*	18	Verlust*
5	*Piranu*	Folgen*	19	Identität*
6	Ker*	Das*	20	Mit*
7	*Istre*	Wachstum*	21	Willen*
8	Z*	Bebauung*	22	Staates*
9		Stadtbild*	23	Atmosphäre*
10		*Koper*	24	Toleranz*
11		*Izola*	25	Gemeinschaft*
12		*Piran*	26	Verbindungen*
13		Weil*	27	Staatsgrenze*
14		Nachkriegsjahren*	* - ni lastno ime	

Tabela 3: Seznam vseh besed z veliko začetnico v dveh vzporednih stavkih iz korpusa.

4.2. Abeceda

Ena od možnosti za prepoznavanje imen je bilo iskanje po nestandardnih črkah (ang. diacritics) v posameznem jeziku. Ker je bilo slovensko besedilo prevedeno v nemški jezik in ker se lastna imena praviloma ne prevajajo (Grah 2000, 1), se v nemškem

besedilu pojavljajo tudi slovenski šumniki. Tako smo v nemškem besedilu hitro našli slovenska lastna imena.

Soča-Tal, Soča, Krško, Kočevje, Snežnik, Goričko, Vače, Hrušica, Mežica-Tal, Lož, Solčava, Boč, Lašče, Brežice, Ormož, Rateče, Stična, Divača, Veržej, Ajdovščina, Sečovlje, Konjiška, Otočec, Sežana, Donačka, Loški, Tržič, Višnja, Košenjak, Paški, Rogaška, Olševa, Mežica, Stržen, Križna, Dobriča, Komarča ….

Tabela 4: V nemškem besedilu najdena slovenska imena.

4.3. Rodilnik in lastna imena

V nemškem jeziku se samostalniku v rodilniku doda končnica -s (des Tische**s**). Lastnim imenom pa se doda končnica -s tudi, kadar ta stojijo brez člena (die Landschaft Slowenien**s**). Če pa pred imenom stoji določni člen, se sklanja samo člen in lastno ime ne dobi končnice.

Südgrenze des *Pohorje*
Besiedlung des *Prekmurje*
Bergland des *Snežnik*
Bergrücken des Boč
Erlebnis des Kucelj

Tabela 5: Rodilnik in lastna imena.

4.4. Nemški predlog *von*

Čeprav je rodilnik v nemškem jeziku precej pogost, samostalnik v rodilniku nima vedno končnice -s. Če se samostalnik konča s sičnikom ali če se želimo izogniti hiatu, uporabimo nemški predlog *von* (Grah 2000: 9, 19-20).

Tudi s tem modulom smo našli kar nekaj lastnih imen.

Bischöfe von *Freising* belehnt
Gebiet von *Idrijsko* und
Zeiten von *Tavčar* und
Ebenen von *Radovljica* und
Wasserscheide von *Rateče* ist
Talkessels von *Ljubljana*
Gegend von Bled übergehend
Grenzen von Slowenien
Eibenbaum von Solčava
Tal von Vitanje

Tabela 6: Predlog *von.*

4.5. Seznam zemljepisnih imen

Kot je za sisteme za prepoznavanje lastnih imen običajno, smo tudi mi uporabili seznam zemljepisnih imen (ang. gazetteer). Ta so bila izluščena iz digitalne enciklopedije Wikipedia (različica na DVD-ju, 2005). Seznam je vseboval 59 imen v slovenskem jeziku in njihove prevode v nemškem jeziku, k temu pa smo dodali še 193 lastnih imen v slovenščini.

4.6. Seznam osebnih imen

Osebna imena smo enako kot zemljepisna izluščili iz enciklopedije Wikipedia. Za razliko od zemljepisnih imen smo vključili vsa osebna imena, zgodovinska in sodobna, kar predstavlja približno 35.000 različnih osebnih imen.

S pomočjo izdelanega seznama je sistem pravilno prepoznal imena kot npr.: H. Freyer, (Kaiser) Augustus, (Kaiser) Karl VI., (Erzherzog) Rudolf IV. ali (König) Heinrich II.

Kljub navedenim rezultatom se je ta možnost izkazala za zelo slabo. Ker so bila imena izluščena iz nemške enciklopedije in ker je bilo v seznam zajetih veliko imen, je prišlo do ogromnega prekrivanja med splošnimi samostalniki in lastnimi imeni. Ta prekrivanja so bila npr.: (George) Moor, (Simon) Koper, (Dieter) Jahr, (Peet) Stol, (Viktor) Klima oder (Friedrich) Wetter. Vsi ti primeri so v korpusu nastopali kot samostalniki, zaradi velike baze Wikipedie pa jih je ta modul (napačno) označil kot lastna imena.

4.7. Iztočnice iz sobesedila

Druga možnost prepoznavanja lastnih imen je že omenjena eksterna evidenca. S to metodo analiziramo okolico lastnih imen in najdene vzorce uporabimo za napovedovanje imen. Čeprav lastnih imen načeloma ne prevajamo, jih prevajalci pogosto prevedejo ali razložijo v oklepajih ali v narekovajih.

Capris (Koper), Vrhnika (Nauportus), Kočevje (Gotschee), Otok (Gutenwerth), Unterkrain (Dolenjsko), Igla (Nadel), Savario (Szombathely), Nauportus (Vrhnika), Poetovio (Ptuj), Siscia (Sisak), Rog (Hornwald)

Tabela 7: Prepoznavanje imen z opisom.

Druga možnost prepoznavanja imen s pomočjo okolice je opisovanje gora z nadmorsko višino. Tako je v besedilu nekaj primerov, kjer za imenom nastopa oklepaj z navedbo nadmorske višine.

Križ (2429 m)	*Brezje* (538 m)
Slavnik (1028 m)	*Macelj* (718 m)
Grmada (887 m)	*Peca-Berg* (2126 m)
Mrzlica (1122 m)	*Stenica* (1091 m)
Gozdnik (1090 m)	*Boč* (979 m)
Rogla (1517 m)	*Ratitovec* (1678 m)

Tabela 8: Prepoznavanje z višinsko oznako.

4.8. Zaporedje črk

Drug način prepoznavanja lastnih imen je različno zaporedje črk v besedi v posameznem jeziku.

Za ugotavljanje zaporedja črk v nemškem jeziku je bil uporabljen seznam osnovnega besedišča nemškega jezika,

za slovensko zaporedje črk pa smo uporabili izvorno besedilo.

sch che hen ich gen ten ste ter ein ver cht cke ken ung ach sen rei lic ren ben nde lle len sse ern den tte aus fen nge der … … … … …. … rop gpl ägl rnr nap önl röh rok önn rog ukt uku rrü nbe nfä	pre sta ske ega ski sko ove nsk ost nje dol rav lin let jsk red kra lov anj sto eni oli pri ven ran slo …… lam bes tah bev gič lar a-f iol ion lii naz lif zcv mda lau žer hol lib nbe lay lfs

Tabela 9: Zaporedje črk v posameznem jeziku.

Modul je primerjal nemško besedilo s tipičnim slovenskim zaporedjem črk in če je imela beseda v prevodu netipično zaporedje črk, jo je označil kot slovensko lastno ime.

Gajus, Pokljuka, Sava, Krn, Soča, Boka, Muzci, Celje, Kozjak, Ptuj, Mežica, Košuta, Pavla, Osp, Sečovlje, Piave, Zasavje, Cezlak ...

Tabela 10: Rezultati iz modula zaporedja črk.

4.9. Končnica -ska/-ske/-sko

Ta modul je uporaben za prepoznavanje večbesednih lastnih imen. Pri analizi besedila in rezultatov je bilo razvidno, da se pri številnih slovenskih večbesednih lastnih imenih prvi del konča s končnico -ska/-ske/-sko. To je bil tudi edini modul, ki je iskal večbesedna lastna imena, ostali moduli so iskali le po enobesednih imenih.

Šaleška dolina, Osapska Reka, Radgonsko-Kapelske gorice, Kamniška Bistrica, Mučka Bistrica, Dolenjske Toplice, Matarsko podolje, Kočevska Mala, Potočka zijalka, Sorško polje, Baška grapa, Selška dolina, Polhograjska Gora, Slatenska plošča

Tabela 11: Imena s končnico -ska/-ske/-sko.

4.10. Statistika

Kot smo že omenili, se za prepoznavanje imen uporabljajo tudi statistične metode. Čeprav smo se v prispevku bolj posvetili imenom in njihovi okolici, statističnih metod nismo povsem zanemarili.

Ta modul se je posvetil zaporedju različnih besednih vrst pred lastnimi imeni, pri čemer smo uporabili oblikoskladenjsko označen korpus. V tem modulu ni bila pomembna struktura imena, temveč besedne vrste in njihova pojavnost.

Tabela 12 prikazuje najpogostejše kombinacije dveh sosednjih besednih vrst. V tej tabeli je razvidno, da je v nemškem jeziku najpogostejša kombinacija člena in samostalnika. Lastna imena najpogosteje nastopajo s predlogi, ki stojijo pred njimi. Takšni primeri so bili: in *Železniki*, von *Škofje? Loka*, in *Slowenien*, is *Bohinj*, bis *Bovec* …

Z nadaljnjo analizo smo se osredotočili na zaporedje besednih vrst do sedem mest pred imenom. Te kombinacije prikazuje spodnja tabela.

ART|NN
ADJA|NN
NN|PUNCT
NN|APPR
ART|ADJA
APPR|ART|
NN|ART
PUNCT|ART
APPR|NE
NN|VVFIN
PUNCT|APPR

Tabela 12: Kombinacija besednih vrst.

APPR|ART|NN|ART|ADJA|NN|+NE (APPR)an (ART)den (NN)Stellen (ART)der (ADJA)heutigen (NN)Städte (NE)Črnomelj
NN|APPR|ART|NE|APPR|NE|+NE (NN)Brücke (APPR)über (ART)die (NN)Save (APPR)bei (NE)Zidani (NE)Most
ADJA|NN|APPR|ART|ADJA|NN|+NE (ADJA)entscheidende (NN)Schlacht (APPR)zwischen (ART)den (ADJA)konkurrierenden (NN)Kaisern (NE)Theodosius
PUNCT|APPR|ART|NN|APPR|NE|+NE (PUNCT). (APPR)Unter (ART)dem (NN)Blegoš (APPR)in (NE)Poljanska (NE)dolina
NN|APPR|ART|NN|APPR|+NE (NN)Gegend (APPR)an (ART)der (NN)Save (APPR)zwischen (NE)Litija

Tabela 13: Kombinacija besednih vrst pred lastnimi imeni.

Razlika med statističnimi metodami in metodami z uporabo pravil je ta, da statistični sistem poda vrednost v odstotkih, ki pa ni zavezujoča. Pri opisu gora z nadmorsko višino je npr. zelo velika verjetnost, da pred opisom stoji lastno ime. Statistična tabela pa nam poda samo verjetnost (npr. 66 %), ne izključuje pa možnosti, da najdena beseda ni lastno ime.

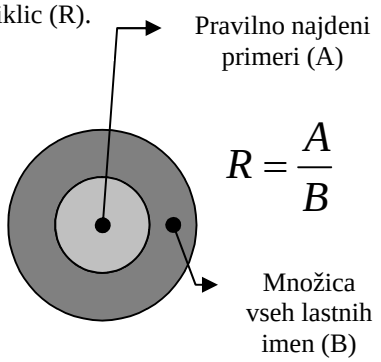
Za popoln statistični sistem pa seveda ni dovolj, da gledamo samo, kaj stoji pred imeni, temveč moramo opazovati celotno okolico imen.

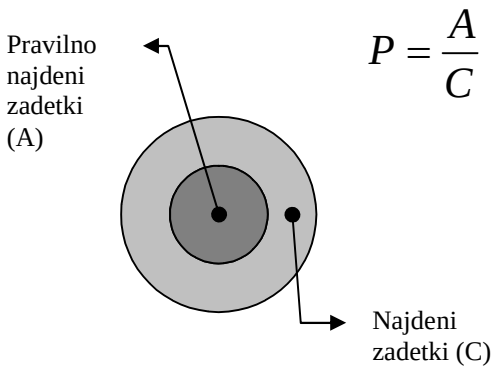
5. Evalvacija

Učinkovitost sistema smo izračunali z vrednostjo F (ang. F-Measure), ki primerja dve vrednosti, in sicer priklic (ang. recall, R) in natančnost (ang. precision, P).

Priklic (R) predstavlja razmerje med pravilno najdenimi imeni in množico vseh lastnih imen v korpusu. Natančnost (P) pa predstavlja razmerje med pravilno najdenimi zadetki in najdenimi zadetki v korpusu (Kemayou Yamga 2006).

Slika 1: Priklic (R).





Slika 2: Natančnost (P).

Za posamezne module je bilo mogoče izračunati samo natančnost (P), ker se posamezen modul nanaša na specifično lastnost lastnih imen in ne more zajeti vseh ostalih imen, ki te lastnosti nimajo.

	A	C	P
abeceda	153	210	0,73
rodilnik	14	19	0,74
predlog »von«	90	109	0,83
seznam zemljep. imen	112	112	1,00
seznam osebnih imen	12	138	0,09
iztočnice in sobesedila	22	24	0,91
zaporedje črk	45	59	0,76
končnica -ska/-ske/-sko	38	42	0,90

Tabela 14: Vrednosti P za posamezne module.

Na koncu raziskave smo vse module povezali in izračunali celotno vrednost F po enačbi:

$$F = \frac{2 \times (R \times P)}{R + P}$$

Za izračun vrednosti F smo morali celotno besedilo z lastnimi imeni ročno označiti.

pravilno najdeni primeri	402
najdeni zadetki	592
natančnost (P)	0,68

Tabela 15: Natančnost (P) za celotni korpus.

pravilno najdeni primeri	402
množica lastnih imen	660
priklic (R)	0,61

Tabela 16: Priklic (R) za celotno besedilo.

Vrednost F za nemško besedilo:

$$F_{de} = \frac{2 \times (0{,}61 \times 0{,}68)}{0{,}61 + 0{,}68} = 0{,}643$$

Za primerjavo smo izračunali še vrednost F za slovensko besedilo. Tudi za to besedilo smo uporabili iste module, z manjšimi modifikacijami.

pravilno najdeni primeri	849
najdeni zadetki	1283
natančnost (P)	0,66

Tabela 17: Natančnost (P) za celotno besedilo.

pravilno najdeni primeri	849
množica lastnih imen	1006
priklic (R)	0,84

Tabela 18: Priklic (R) za celotno besedilo.

Vrednost F za slovensko besedilo:

$$F_{sl} = \frac{2 \times (0{,}84 \times 0{,}66)}{0{,}84 + 0{,}66} = 0{,}739$$

6. Zaključek

Prispevek prikazuje, da obstajajo različne možnosti za luščenje lastnih imen iz tekočega besedila. Velika začetnica sicer zelo pomaga pri prepoznavanju lastnih imen, ampak samo, če so imena enobesedna. Pri večbesednih imenih označuje velika začetnica v slovenskem jeziku samo začetek lastnega imena. Te pomoči pa v nemškem jeziku ni, ker vsaka velika začetnica ne označuje nujno lastnega imena.

Učinkovitost smo izračunali s pomočjo vrednosti F, ki je za nemško besedilo znašala 0,64, za slovensko pa 0,74 točke. Pri tem je razvidno, kako lahko velika začetnica v slovenskem jeziku poveča učinkovitost sistema.

Nadaljnji korak bi bila sintaktična analiza (ang. parsing) za prepoznavanje večbesednih lastnih imen in označevanje teh kot eno samo enoto.

V veliko pomoč bi bil tudi večji učni korpus, na katerem bi se sistem lahko učil. Z večjim korpusom bi se tudi povečalo število različnih lastnih imen.

Ena večjih pomanjkljivosti sistema je pomanjkanje slovničnih pravil, ki jih je v izdelanem sistemu zaenkrat deset, medtem ko razvitejši sistemi uporabljajo več tisoč takšnih pravil.

Ugotavljamo, da je luščenje lastnih imen z uporabljenimi postopki učinkovito, vendar sistem kot vsak drug potrebuje čas za rast in učenje.

7. Literatura

Bekavac, Božo (2005): Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima, Sveučilište u Zagrebu Filozofski fakultet, Zagreb

Grah, Käthe (2002): Slovenska stvarna imena v nemških besedilih – Slowenische Sachnamen in deutschen Texten, Ljubljana, Znanstveni inštitut Filozofske fakultete

Kemayou Yamga, Syriane (2006): Informationsextraktion http://kontext.fraunhofer.de/haenelt/kurs/Referate/KemayouYamga_W05/Informationsextraktion.pdf (15. 5. 2006)

Peterlin, Janez (2003a): Pozdravljena, Slovenija, prenovljena izdaja, Mladinska knjiga, Ljubljana

Peterlin, Janez (2003b): Grüsse aus Slowenien, überarbeitete Ausgabe, Mladinska knjiga, Ljubljana

Roth, Jaennette (2002): Der Stand der Kunst in der Eigennamen-Erkennung, Mit einem Fokus auf Produktenamen-Erkennung, Lizentiatsarbeit der Philosophischen Fakultät der Universität Zürich, Zürich http://www.ifi.unizh.ch/cl/study/lizarbeiten/lizjeannetteroth.pdf (15. 5. 2006)

Schippan, Thea (1992): Lexikologie der deutsche Gegenwartsfrage, Tübingen, Niemeyer Verlag

Slovenski pravopis 1: Pravila (1994): 4., pregledana izdaja (s stvarnim kazalom), Ljubljana, DZS

Uporaba korpusa pri urejanju spletnega terminološkega slovarja

Katarina Puc,[*] Tomaž Erjavec[‡]
*Ljubljana, katarina.puc@drustvo-informatika.si
‡Odsek za tehnologije znanja, Institut Jožef Stefan
Jamova 39, 1000 Ljubljana
tomaz.erjavec@ijs.si

Povzetek

V članku opišemo, kako se pri urejanju spletnega terminološkega slovarja Islovar uporablja korpus informatike, ki se oblikuje iz zbornikov posvetovanja Dnevi slovenske informatike. Islovar vsebuje izrazje s področja informatike in je odprt za uporabo, pa tudi za prispevke uporabnikov. Zbranih izrazov je veliko več kot urejenih, ker je končno urejanje zahteven postopek. Pri urejanju uredništvo upošteva vse ugotovljene sinonime izrazov, ki jih hkrati ovrednoti glede na uporabo. Korpus informatike je odličen vir, ker zajema prispevke številnih, različnih avtorjev. V prispevku predstavimo izdelavo in značilnosti korpusa, podamo primere uporabe korpusa ter drugih elektronskih virov in zaključimo z načrti za prihodnost.

Using a corpus for editing an on-line terminological dictionary

In the paper the use of DSI corpus for editing the on-line terminological dictionary Islovar is described. DSI is a specialized corpus, created from proceedings of Slovene Informatics Conferences. The Islovar dictionary includes terms relating to the field of informatics, with open access for users and also for contributors of new words and commentaries. Because compiling the final edition of the dictionary is a demanding process, the dictionary still contains more collected than edited entries. The editors take into consideration all the discovered synonyms which are evaluated and labelled according the usage. Originating from articles by many authors, the corpus is an excellent source of informatics terms. In the paper, the creation and the characteristics of the corpus are described. Some examples of use in editing Islovar and a comparison with reference corpora and other electronic sources are given. Finally, we mention some plans for the future.

1. Uvod

Islovar, http://www.islovar.org/, je slovenski spletni terminološki slovar informatike, ki ga ureja Slovensko društvo INFORMATIKA od aprila 2001. Slovar navaja tudi angleške ustreznice, tako da lahko iščejo uporabniki slovenske izraze tudi iz angleških, urejeni slovarski sestavki pa vsebujejo slovensko razlago in kvalifikatorje.

Ker so njegovi avtorji informatiki, je naravno, da se pri uporabi in pri urejanju tega slovarja izkoriščajo vse prednosti informacijskih tehnologij. Prav pri izdajanju slovarjev so se te prednosti v zadnjih letih posebej izkazale, o čemer priča veliko število spletnih slovarjev v vseh jezikih. Posebnost Islovarja je v njegovi odprtosti ne samo za branje, temveč tudi za zbiranje in urejanje.Uporabniki slovarja lahko nove izraze prispevajo, dodajajo razlage in obstoječe sestavke komentirajo.

Odprtost, značilna za objave na svetovnem spletu, je velika prednost, ker omogoča sodelovanje velike populacije. Po drugi strani pa je lahko tudi problem: ker lahko vsak objavlja karkoli, sta kakovost in zanesljivost spletnih dokumentov pogosto vprašljivi. Uredništvo Islovarja to rešuje z dogovorjenim uredniškim postopkom in tako zagotavlja, da so pri urejanju vsi zapisi v Islovar večkrat pregledani.

Namen Islovarja je poenotenje informacijskega izrazja in opremljanje novih pojmov s splošno privzetimi slovenskimi ustreznicami. Zato uredništvo deluje v skladu z načeli upoštevanja zanesljivosti, kakovosti in ažurnosti vsebine. Islovar je informativni in normativni slovar, beleži vse pojave izraza, jih ovrednoti in ustrezno označi.

Za zbiranje in analiziranje izrazja se je v slovaropisju uveljavila jezikovna tehnologija korpusov. Za raziskave specializiranih besedil nastajajo specializirani korpusi, ki so navadno manjši in vsebujejo jezik v točno določeni rabi. Pogosto jih uporabnik izdela sam, s točno določenim namenom (Arhar, 2006).

Tak specializirani korpus je korpus informatike (v nadaljevanju korpus DSI), ki vsebuje članke iz zbornikov posvetovanja Dnevi slovenske informatike iz let 2003−2006. Ta posvetovanja so letna in množično obiskana, obravnavajo pa različno tematiko, od strateških vidikov informatike, do operacijskih raziskav. Prispevki avtorjev se objavljajo v zbornikih. Tematika je pretežno aktualna in se menja iz leta v leto, avtorji besedil pa so informatiki iz prakse in z univerz. Prispevki so lektorirani, tako da žargonski izrazi zvečine niso zajeti.

V članku bomo opisali uredniški postopek v Islovarju, zgradbo korpusa DSI in kako ta korpus uporabljamo pri slovaropisju.

2. Uredniški postopek v Islovarju

Islovar je spletni računalniški program. Od novembra 2004 deluje že v drugi izdaji, ki je vnesla mnoge izboljšave v uporabniškem in v uredniškem vmesniku. Zlasti je bil izboljšan iskalnik, ki omogoča iskanje tudi po podobnih izrazih in zato uporabnikom dovoljuje tudi manjše napake pri vnosu iskanih iztrazov.

Uredniški vmesnik ima številne nove možnosti: iskanje po Islovarju po raznih kriterijih, tudi po avtorjih sestavkov, vpogled v novo zapisane sestavke in v zgodovino sprememb, delo z zbirkami.

Delo urednikov poteka neposredno v spletnem slovarju po dogovorjenem uredniškem postopku. Slovarski sestavki imajo posebno oznako, ki kaže na stopnjo obdelave sestavka in zanesljivost vsebine. Oznako *predlog* prejmejo izrazi takoj po vnosu v slovar, po prvem pregledu, ko se preverja vsebinska primernost za področje informatike, pa so izrazi označeni kot *pregledani*.

Naslednja značilnost uredniškega postopka je analiza zbirke izrazov, ki jo pripravi urednik ali strokovna skupina. Zbirka je vsebinsko povezana družina, ki zajema tudi vse izraze, ki so bili uporabljeni v razlagah v tej zbirki. Po javni razpravi o predlagani zbirki, ki se je udeležujejo vsi uredniki Islovarja, sledijo končni popravki in opredelitev *strokovno pregledano*.

Do tu se uredniški postopek usmerja zlasti na določitev vsebinskega obsega, strokovne ustreznosti strokovnega izraza in natančnosti razlage. Sledi slovaropisna obravnava, kjer se poskrbi za formalno pravilnost slovarskih sestavkov glede na slovaropisna priporočila in uskladitev z že urejenimi slovarskimi sestavki. Po slovaropisni obravnavi strokovna skupina oziroma urednik, ki je pripravila oziroma pripravil zbirko, ponovno preveri pravilnost vsebine, sledi natančen formalni pregled, nakar se slovarski sestavki označijo kot *urejeni*. Tudi ti sestavki se kasneje lahko še spreminjajo.

Vse te razprave potekajo delno na sestankih, delno neposredno v Islovarju (kot komentarji k sestavkom ali v okviru razprave v forumu) ali po elektronski pošti. Udeleženci na sestankih uporabljajo osebne računalnike z neposredno povezavo na spletno stran Islovarja, tako da je vse delo neposredno zabeleženo in vidno vsem urednikom. Izpis na papirju je nujno potreben šele pri končnem formalnem pregledu.

V tem postopku uredniki poleg izdaj v knjižni obliki izkoriščajo vse vire, ki so dostopni v elektronski obliki na spletu. V slovenščini so na voljo številni slovarčki, glosarji in Leksikon računalništva in informatike, uporabljajo pa se predvsem angleški spletni slovarji in leksikoni ter drugi spletni dokumenti, ki jih najdemo s spletnima iskalnikoma najdi.si in Google.

Ker poimenovanje pojmov na tem področju nikakor ni poenoteno in uporabljajo v slovenščini razni avtorji različne izraze za isti pojem, se mora uredništvo odločati, kako ovrednotiti posamezne ustreznice in ali predlagana razlaga res ustreza sodobni uporabi. Pri tem odločanju so pomembni dostopnost, zanesljivost in ažurnost virov.

Sorazmerno lahko dostopen je podatek o pogostosti uporabe. Zelo priročna pri tem sta iskalnika najdi.si in Google v slovenščini. Rezultati takega iskanja pa so žal pogosto nezanesljivi. Številni pomembni dokumenti na spletu sploh niso objavljeni, nesorazmerno pa so zastopani prispevki raznih posameznikov, razprave v forumih, in seveda objave prodajalcev opreme. Zato je rezultate spletnega iskanja treba temeljito pretehtati. Kot zanesljiv spletni vir obravnava uredništvo seminarska in diplomska dela, objavljena na spletu, pa tudi programe fakultet in reviji Monitor in Moj mikro.

Zanesljiv vir je tudi terminološka zbirka Evroterm, kjer so koristni primeri uporabe in prevodi v druge jezike. Žal pa je obdelana predvsem zakonodaja Evropske unije in zato informatika samo obrobno.

Zaradi ažurnosti, tako pomembne pri informatiki, so samo delno uporabni dokumenti, ki so stari 5 let in več. Zato skoraj ni uporaben referenčni korpus Nova beseda, ki strokovne izraze navaja pretežno iz dnevnika Delo, iz Monitorja pa iz 2000 do leta 2002. Tudi Leksikon (Pahor, 2002) je žal že v marsičem zastarel.

Glede ažurnosti in zanesljivosti je korpus DSI za uporabo uredništva Islovarja daleč najboljši. Nudi nam vpogled v primere uporabe in pogostost uporabe. Ker posvetovanja Dnevi slovenske informatike obravnavajo široko, zlasti aktualno tematiko, je korpus uporaben v velikem številu primerov.

3. Korpus DSI

Ker zborniki pokrivajo isto področje kot slovar, obenem pa so strokovni prispevki dragocen vir svežega slovenskega izrazja, smo se že leta 2003 dogovorili, da se zbornike pretvori v korpus, ki bi nato lahko služil kot podpora pri izdelavi slovarja (Erjavec in Vintar, 2004).

Korpus DSI je bil narejen iz digitalnih originalov (Microsoft Word) konferenčnih prispevkov. Ti so zapisani v skladu s predlogo konference, kar v marsičem olajša nadaljnji postopek pretvorbe. Dokumente smo najprej pretvorili v XML, nato pa s filtrom XSLT ta XML pretvorili v besedilo korpusa. Filter iz dokumentov izloči nebesedilne elemente in tiste razdelke, ki so pisani v angleškem jeziku (angleški povzetek, bibliografija). Tu je potrebno omeniti, da se je v vsakem od zbornikov pojavilo par prispevkov, ki zaradi napak v formatu Word niso bili pretvorjeni; zato je število besedil v korpusu nekaj manjše od števila prispevkov v zbornikih, število besed pa zaradi tega, in zaradi omenjenih izpustov delov besedil tudi manjše kot število besed v zbornikih.

V drugi fazi se besedilo korpusa jezikoslovno označi. To smo storili s pomočjo programa *totale* (Erjavec et al., 2005), ki besedilo naprej tokenizira (razdeli besedilo na besede, ločila in povedi), nato oblikoslovno označi (vsaki besedi pripiše njeno oblikoslovno oznako iz nabora MULTEXT-East za slovenski jezik) in lematizira (določi besedam njihovo osnovno obliko). Program sicer označi vse besede v korpusu, tako znane kot neznane, vendar pa pri označevanju dela tudi napake; največ problemov povzročajo angleške besede in okrajšave.

Korpus trenutno obsega štiri zbornike (2003–2006), velikost korpusa po posameznih letnikih in skupno pa je podana v Tabeli 1.

Letnik	*Besedil*	*Odstavkov*	*Stavkov*	*Besed*
2003	111	3.164	9.791	196.883
2004	109	2.893	9.273	200.287
2005	123	3.546	10.474	223.635
2006	137	4.277	12.022	262.260
Σ	480	13.880	41.560	883.065

Tabela 1. Velikost korpusa DSI 2003−2006

Poglejmo še besedišče korpusa. V Tabeli 2 je podano število različnih lem, torej osnovnih oblik besed po treh najbolj zanimivih besednih vrstah, ter za vse besedne vrste. Kot rečeno, prihaja pri lematizaciji tudi do napak, zato so razmeroma zanesljive samo leme, ki se v korpusi pojavijo večkrat; tabela poda števila za vse leme, ter za tiste, ki se pojavijo vsaj dvakrat oz. trikrat.

Besedna vrsta	≥ 3	≥ 2	≥ 1
Samostalnik	6.010	8.273	16.987
Pridevnik	2.873	3.794	7.466
Glagol	1.943	2.567	5.079
Vse	11.633	15.528	30.828

Tabela 2: Število različnih lem v korpusu DSI 2003−2006

3.1. Dostop do korpusa

Za uporabo korpusa potrebujemo programska orodja, predvsem konkordančnik. Dobri konkordančniki omogočajo iskanje po kombinacijah različnih kriterijev in znajo rezultate poizvedb prikazati na več načinov. Najbolj udobni in za resno delo najbolj primerni so konkordančniki, ki si jih instaliramo na lasten računalnik, vanje uvozimo korpus in ga analiziramo. Pri našem delu smo testno uporabili orodje Wordsmith (Scott, 2006), ki sicer ponuja poleg izdelave konkordanc tudi frekvenčne sezname, sezname ključnih besed in kolokacij, vendar pa deluje samo nad izvornim besedilom, kar pomeni, da ne moremo iskati po lemah ali oblikoslovnih oznakah, niti po oznakah zahtevati izpisa. Druga "slabost" orodja je, da je na voljo samo proti plačilu (obstaja pa tudi 40-dnevna evaluacijska verzija), korpus pa bo uporaben samo tistim, ki program kupijo in instalirajo.

Mrežni konkordančniki sicer nudijo manj možnosti, so pa dostopni vsem in uporabni brez posebne lokalne programske opreme. Na IJS že vrsto let obstaja konkordančnik na naslovu http://nl2.ijs.si/. Tu je na voljo več dvo- in enojezičnih korpusov slovenskega jezika, tudi (vsako leto obnavljani) DSI. Spletni vmesnik omogoča iskanje po korpusu in lahko prikaže rezultate na tri načine: kot spisek konkordanc, kot seznam zadetkov s frekvencami in, za dvojezične korpuse, kot seznam poravnanih segmentov. Primer rezultatov poizvedbe za pridevniki, ki jim sledi lema »aplikacija«, ki je izpisan kot frekvenčni seznam, je podan v Sliki 1.

Spletni vmesnik za iskanje uporablja strežnik korpusov CQP (Christ 1998), ki ima zelo bogat iskalni jezik, saj lahko prek regularnih izrazov (npr. »*aplikacij.**«) iščemo po kombinaciji pojavnic iz besedila, ali pa po njihovih oznakah (v našem primeru leme in oblikoslovne oznake MULTEXT-East). Ker je računalnik, na katerem teče servis, razmeroma močan, CQP pa optimiran na hitrost, tudi kompleksne poizvedbe vrnejo rezultat v razmeroma kratkem času.

Slabosti trenutne implementacije, ki se jih sicer zavedamo, bi pa v njihovo odpravljanje bilo treba vložiti nekaj dela, je zahteven jezik poizvedb,[1] ki bi ga bilo bolje prevesti v bolj strukturiran obrazec HTML, ter majhna možnost izbire oblike izpisa.

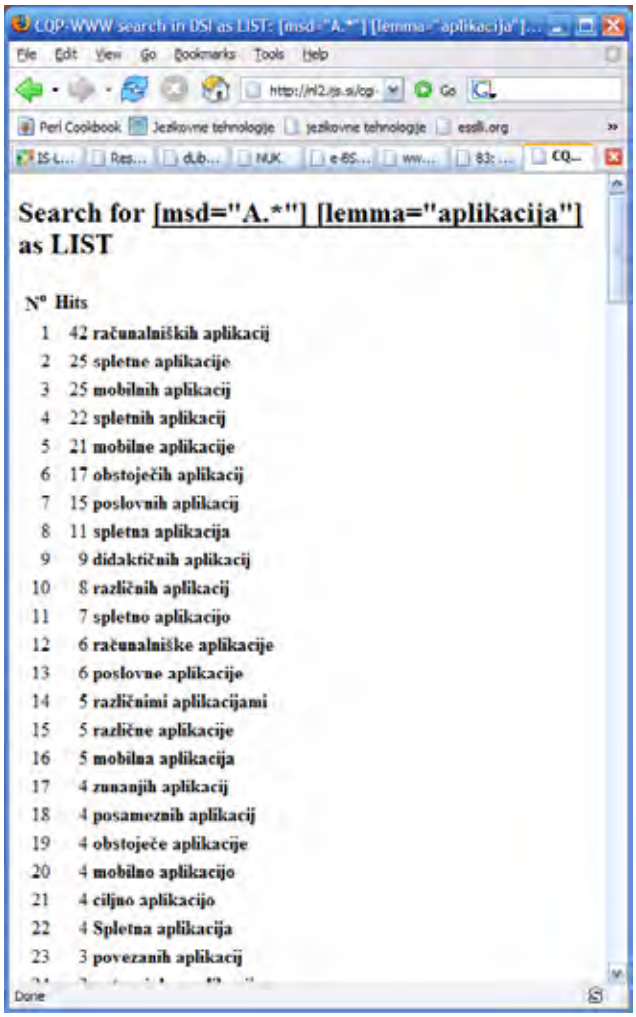
4. Uporaba korpusa DSI

Korpus DSI lahko uporabljamo za številne analize, ki nam pomagajo pri urejanju Islovarja. Pri pregledovanju primerov uporabe, ki so razvidni iz kolokacij, se odločamo o izboru slovenskih ustreznic in vrednotenju sinonimov, podatki o pogostosti posameznih izrazov pa nam pomagajo pri odločanju, katere izraze urediti prioritetno in katere sploh uvrstiti v Islovar.

4.1. Izbor slovenskih ustreznic

Prikazali bomo dva primera, kjer smo se pri odločanju o zapisu v Islovar odločali na podlagi korpusa DSI. Gre za izraze, ki se v informatiki pogosto uporabljajo, zlasti v besednih zvezah, pojavljajo pa se v več sinonimih.

[1] Poizvedba, ki poišče pridevnike v primerniku in besedne oblike od leme, ki se začne na *man* oz. *men* je *[msd="A.c.*"] [lemma="m[ae]n.*"]*



Slika 1. Primer rezultatov poizvedbe v korpusu DSI

Ker je za področje informatike osnovni jezik angleščina, se med domačim izrazjem pojavljajo tudi besede, privzete iz angleščine in v angleški pisavi. Slovenščina je sorazmerno negostoljubna do takih pojavov. Vprašanje je, ali take izraze zajeti v terminološki slovar in kako.

4.1.1. Online v slovenščini

Online se uporablja v angleškem jeziku kot pridevnik ali kot prislov, v inačicah *on-line*, *on line*, *online*. Ima širok pomen. Beseda v angleški pisavi obstaja v številnih slovenskih besedilih, tudi uglednih institucij, kot je COBISS (online informacijski sistem in servisi), Centralna tehniška knjižnica (online informacijski servis). Angleško-slovenski slovar bibliotekarske terminologije pa besedo zapiše *onlajn* (onlajn servis).

V elektronskih virih se online, on-line pojavljata v slovenskih besedilih z naslednjo pogostostjo:

Nova beseda	*najdi.si*	*korpus DSI*
352	517.000	102

Tabela 3: Pogostost online, on-line v elektronskih virih

Primeri iz korpusa Nova beseda niso uporabni, zvečine gre za imena časopisov, institucij. Iz Monitorja izvirata *on-line* (6) in *on line* (159), vendar gre za zapise iz let 2000 in 2002. Taki viri so za danes tako aktualen izraz zastareli.

Primeri uporabe v najdi.si so številni in zelo različni. Sicer pa je pregled takšne množice, tudi če jo skrčimo po raznih kriterijih, praktično nemogoč. Iz teh podatkov lahko sklepamo samo to, da je beseda *online* v slovenščini zelo pogosta, po vsej verjetnosti zato, ker ji nismo našli prave ustreznice.

Korpus DSI nam omogoča boljšo preglednost uporabe in navaja pojavnosti, kot so:

- online učenje na daljavo, online literarna revija, online storitve, online katalog, online prijava;
- on-line prodaja, on-line sestanek, on-line nakup, on-line poslovanje.

Očitno je pomen angleške besede *online* tako širok, da ga je slovenščini nemogoče nadomestiti z eno samo ustreznico. Izraz se je iz strokovne rabe prenesel tudi v splošno besedišče. Sorazmerno redko se nadomešča z drugimi slovenskimi ustreznicami. Torej smo se odločili, da ga vključimo v Islovar v angleški pisavi, urejenega v naslednji slovarski sestavek:

online neskl. *(angl. online, on-line)* žarg.
1. ki je dostopen v oddaljenem računalniškem sistemu, npr. online podatkovna baza
2. ki je dostopen po telekomunikacijskem omrežju, npr. online banka
3. gl. povezan in priključen in priklopljen
4. gl. spleten
5. gl. sproten

V Islovarju imamo zdaj urejenih 25 slovarskih sestavkov, ki se nanašajo na angleški *online*. Ti sestavki se bodo z leti brez dvoma pomnožili in tudi spremenili, ko bo slovenščina ta izraz povsem absorbirala ali pa ga izločila. Za take spremembe je Islovar odlično opremljen, saj omogoča takojšnje posodabljanje.

4.1.2. Management v slovenščini

Management je beseda, ki se je uveljavila v slovenščini v zadnjih petnajstih letih v pomenih: poslovodstvo, upravljanje, vodenje. Slovar slovenskega knjižnega jezika (SSKJ) pozna besedo še v angleški pisavi, Slovenski pravopis iz leta 2000 pa samo še kot *menedžment*, kar usmerja na vodenje, upravljanje.

V informatiki se ta izraz pojavlja v številnih besednih zvezah, v angleški pisavi pa tudi kot razne slovenske ustreznice, npr. upravljanje, obvladovanje, ravnanje. Za isti pojem se pojavlja več inačic, npr.: avtorji slovenijo *business process management* kot upravljanje poslovnih procesov, management poslovnih procesov, obvladovanje poslovnih procesov, sistem za procesno ravnanje.

V Islovarju smo imeli pred končnim urejanjem zbranih 53 iztočnic z ustreznicami za *management*, od teh številne sinonime ali po našem mnenju nepravilne ustreznice. Pri urejanju slovarskih sestavkov smo se naslonili na opredelitev splošnih pomenov v SSKJ in na primere uporabe in na pogostost v korpusu DSI.

Če s funkcijo »wordlist« izpišemo število vseh pojavitev v korpusu DSI, se nam potrdi domneva, da je *upravljanje* v informatiki povsem uveljavljen izraz. Zato smo ga razen v nekaterih primerih, zlasti pri izrazu *obvladovanje*, prevzeli tudi v Islovarju kot nadrejeni sinonim. *Menedžment* se v korpusu DSI pojavlja pretežno v pomenu poslovodstvo.

Izraz	*pogostost*
upravljanje	1207
obvladovanje	314
management	203
ravnanje	55
menedžment	8

Tabela 4: pogostost ustreznic za *management* v korpusu DSI

Pri zvezah z *upravljanje* smo opazili pogosto (346 krat) napačno rabo z orodnikom, npr. upravljanje z vsebinami, s tveganji, z znanjem, kar po SSKJ in Slovenskem pravopisu ni pravilno. V Islovarju te rabe zdaj sicer ne priporočamo, se bo pa morda v informatiki uveljavila.

V Islovarju imamo torej urejena naslednja slovarska sestavka:

management -a m *(angl. management)*
1. gl. menedžment (1) in upravljanje (3)
2. gl. vodenje
3. gl. poslovodstvo in uprava

upravljanje -a s *(angl. management)*
1. načrtovanje, nadziranje in vzdrževanje informacijske tehnologije, npr. upravljanje podatkovnih baz, upravljanje dokumentov
2. usmerjanje procesov, postopkov v organizaciji z uporabo informacijske tehnologije, npr. upravljanje znanja; sin. ravnanje (3)
3. organizacijska funkcija, katere osrednje naloge so načrtovanje, organiziranje, nadziranje dejavnosti; sin. management (1), menedžment (1)
4. računalniško usmerjanje delovanja sistema, naprave; prim. krmiljenje.

Beseda *upravljanje* je iz splošnega jezika prešla v strokovni jezik informatike, kjer se je uveljavila bolj kot angleška beseda *management* ali *menedžment.* Privzela je številne pomene. *Management* in *menedžment* pa se uporabljata v splošnem jeziku najpogosteje v pomenu poslovodstvo, uprava.

4.2. Pogostost izrazov kot osnova za urejanje

Analiza pogostosti informacijskih izrazov je zelo koristna, zlasti pri pregledu samostalniških besed. Kot smo že omenili, smo pri analizi korpusa DSI uporabili tudi orodje Wordsmith (Scott, 2006), ki poleg izdelave konkordanc in frekvenčnega seznama ponuja tudi izdelavo seznama ključnih besed. Te dobimo tako, da izberemo frekvenčna seznama našega (specializiranega) korpusa in korpusa splošnega jezika (v našem primeru je bil to vzorec iz korpusa FIDA), nato pa Wordsmith prek statističnih mer določi, katere besede (in s kakšno mero »ključnosti«) se v specializiranem korpusu pojavijo večkrat kot pričakovano glede na splošno besedišče.

Začetek seznam ključnih besedah korpusa DSI, urejenega po ključnosti, je podan v Sliki 2. V prvem stolpcu je ključna beseda, v drugem absolutna frekvenca v korpusu DSI, v četrtem v vzorcu FIDA, v petem pa vrednost ključnosti.

V seznamu sicer vidimo, da se v korpusu večkrat uporabljajo nekatere funkcijske besede, verjetno zaradi drugačnosti stila člankov od pretežno neznanstvenih besedil v FIDI, ter dosti več angleškega izrazja, za nas bolj zanimivi pa so samostalniki, kjer je seznam lepo razporejen. Izrazi, ki so na vrhu, so zares temeljni za informatiko. Bi pa seznam seveda bil bistveno bolj uporaben, če bi v njem lahko opazovali leme namesto besednih oblik.

Ključna beseda	Frekv. DSI	Frekv. Ref.	Ključnost
podatkov	2954	461	3238,9
sistema	1941	154	2652,98
procesov	1426	22	2437,37
storitev	1598	89	2354,81
sistem	1782	212	2165,91
poslovnih	1377	55	2141,23
podjetja	1757	331	1764,5
it	1019	22	1697,38
of	1277	118	1677,04
potrebno	1547	292	1551,94
informacijske	878	8	1543,88
poslovanja	983	48	1481,91
rešitev	1285	182	1464,9
and	959	60	1381,42
uporabnikov	816	20	1343,46
sistemov	897	48	1331,06
omogoča	1142	153	1329,91
rešitve	978	86	1301,59
informacij	1031	111	1294,21
informacijskih	713	3	1285,4
upravljanje	827	39	1253,79
uporabo	1014	118	1241,22
procesa	790	34	1214,86
projekta	992	117	1208,82
programske	730	26	1152,6
is	752	35	1142,46
tehnologije	705	27	1102,41
opreme	781	57	1088,24
ter	2924	1676	1075,16
uporabe	791	66	1067,11
ikt	570	0	1056,18
uporabniki	670	27	1040,24
informacijski	586	15	960,614
informatike	551	7	952,813
informacijskega	537	5	943,249
npr	813	111	939,883
spletnih	556	11	932,802
znanja	748	84	926,303

Slika 2. Ključne besede v korpusu DSI

Izrazi, ki se najpogosteje uporabljajo v pisani obliki, bi morali biti v Islovarju prioritetno urejeni. Pri pregledu opažamo, da so nekateri sicer urejeni, niso pa urejene vse besedne zveze, ki so že v Islovarju, verjetno bo treba še nekatere dodati. Prav te lahko najdemo iz primerov uporabe, ki jih najdemo s konkordančnikom v korpusu DSI.

V primerjavi s korpusom DSI iz leta 2005 se je rang nekaterih izrazov spremenil, kar dokazuje dinamičnost njihove uporabe. Upravljanje se je pojavilo 911 krat v letu 2005, v letu 2006 pa kar 1316 krat. Povečala se je tudi uporaba besede *management* s 303 krat na 450 krat. Zelo veliko se uporablja *aplikacija* (ki je Islovar ne priporoča, temveč usmerja na uporabniški program): v korpusu 2003−2005 729 krat, korpusu 2003−2006 pa kar 1412 krat. Na temelju teh ugotovitev bo verjetno treba popraviti slovarski sestavek za *aplikacijo*.

Pridevniške besede nastopajo v terminološkem slovarju predvsem v besednih zvezah. Iz korpusa DSI smo izluščili 142 tipičnih najpogostejših pridevniških besed, pri katerih opažamo podobno razporeditev kot pri samostalnikih. Na vrhu so: *informacijski* (3562 krat), *spletni* (1752 krat), *elektronski* (1670 krat), *podatkovni* (1200 krat). S temi pridevniškimi besedami lahko poiščemo vse možne besedne zveze, ki nastopajo v korpusu in jih primerjamo z vsebino Islovarja.

V Islovarju že najdemo 28 različnih zvez s *podatkovni*, npr. *podatkovna baza*, *podatkovno skladišče*, *podatkovno rudarjenje*. V korpusu DSI pa so še številne druge (skupno 142), npr. *podatkovni agregat*, *podatkovni element*, *podatkovni strežnik*, *podatkovno upravljanje*. Z analizo uporabe presodimo, katere od teh še vključiti v Islovar. Na ta način dopolnjujemo vsebino Islovarja z aktualnimi izrazi.

5. Uporaba drugih elektronskih virov

Zborniki DSI zajemajo predvsem organizacijsko informacijske teme, ki so v času odvijanja posvetovanja aktualne. Zato korpusa DSI ne moremo uporabiti za odločanje o številnem izrazju, ki tudi sodi v Islovar.

Navedimo kot primer izraz *pomnilnik*, ki se zaradi razvoja informacijske tehnologije pojavlja v številnih različicah in zvezah, v Islovarju kot iztočnica ali del besedne zveze kar 111 krat, v razlagah pa 146 krat. V korpusu DSI ga najdemo samo 37 krat v nekaterih splošnih pomenih. Tukaj se moramo usmeriti na druge elektronske vire, predvsem na iskalnika najdi.si in Google. Pregledovanje je v tem primeru zelo zamudno, ker oba iskalnika navajata veliko število primerov, med katerimi so pretežno reklamna besedila prodajalcev.

Korpus Nova beseda navaja pomnilnik 2619 krat, primeri pa so iz časnika Delo in revije Monitor. Delo navaja *pomnilnik* v najbolj splošnem pomenu. Monitor je zanesljiv vir, vendar v tem primeru slabo uporaben, ker so primeri iz leta 2000 – torej očitno zastareli.

Nova zbirka Besede slovenskega jezika najde primere za *pomnilnik* 13 krat, pretežno iz najdi.si. Med drugim navaja *brisljiv*, *nebrisljiv pomnilnik* iz istega, vendar nezanesljivega vira.

Monitor izdaja zdaj tudi skrajšano, spletno inačico revije, ki pa žal še nima velikega arhiva. Podobno revija Moj mikro. Zato smo morali pri zbiranju izrazov uporabiti številne, tudi knjižne vire. Ker pa se tehnologija razvija, se bo družina *pomnilnik* v Islovarju brez dvoma še razširjala.

Ugotavljamo, da sta Google in najdi.si dobri orodji predvsem za ugotavljanje pojavnosti nekega izraza v spletnih dokumentih. Če ugotovimo, da se neki izraz pojavlja večkrat v zanesljivih virih, potem lahko presodimo, da ta izraz sodi v Islovar kot iztočnica, kot enakovreden ali nadrejeni sinonim.

6. Sklepne ugotovitve

Pri zbiranju in urejanju izrazja za terminološki slovar informatike se uredniki naslanjajo predvsem na dostopne elektronske informacijske vire. Pomembna kriterija pri odločanju o uporabi teh virov sta ažurnost in zanesljivost.

Pri iskanju pogostosti uporabe posameznih izrazov in opredeljevanju pomena pri oblikovanju razlage si uredniki pomagajo s slovarji in dokumenti, ki so dostopni na svetovnem spletu. Pri teh raziskavah je v veliko pomoč tudi specializirani korpus DSI, ki omogoča različne vpoglede v uporabo in se posodablja vsako leto.

Kot smo pokazali na dveh primerih urejanja, nam uporaba tega korpusa zanesljivo pokaže pogostost in pomene posameznih izrazov v informatiki v sedanjem času. Namen spletnega terminološkega slovarja Islovar pa je med drugim tudi zasledovati in beležiti razvoj informacijskega izrazja v slovenščini.

Korpus DSI za zdaj zajema zbornike posvetovanj Dnevi slovenske informatike. Lahko bi ga razširili z različnimi besedili, npr. s članki strokovnih revij in z besedili učbenikov ter drugih publikacij, tako da bi postal bolj uravnotežen in še bolj uporaben. To pa bi bil zahtevnejši projekt, ki bi potreboval širšo podporo.

Literatura in viri

Arhar, Š.(2006): Gradnja specializiranega korpusa. Jezik in slovstvo, 2006, št. 1

Christ, O. (1994): A modular and flexible architecture for an integrated corpus query system. Proceedings of the 3rd Conference of Computational Lexicography and Text Research (COMPLEX'94), Budimpešta. *http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/*

Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R. (2005). Massive multilingual corpus compilation: ACQUIS Communautaire and totale. *Proceedings of the Second Language Technology Conference*, april 2004, Poznan.

Erjavec, T., Vintar, Š. (2004). Korpus kot podpora slovarju informacijskega izrazja slovenskega jezika. *Uporabna informatika. (Ljubljana)*, 12/2 97-106.

Pahor, D., et al.(2002): Leksikon računalništva in informatike, Ljubljana, Založba Pasadena

Scott, M. (2006): WordSmith Tools, Version 4. Oxford University Press. *http://www.lexically.net/wordsmith/*

Turk, T., Puc, K. (2006): Islovar kot model spletnega terminološkega slovarja. Razvoj slovenskega knjižnega jezika (Obdobja 24 – Metode in zvrsti) (v tisku)

Slovarski knjižni viri:

Slovenski pravopis, Ljubljana: Založba ZRC, ZRC SAZU, 2001

Angleško-slovenski slovar bibliotekarske terminologije, Ljubljana: Narodna in univerzitetna knjižnica, 2002

Spletni viri:

Islovar *http://islovar.org*

Slovar slovenskega knjižnega jezika *http://bos.zrc-sazu.si/sskj.html*

Besede slovenskega jezika *http://bos.zrc-sazu.si/besede.html*

Nova beseda *http://bos.zrc-sazu.si/s_beseda.html*

Spletni slovarji *http://www.sigov.si/slovar.html*

Zbirka tujih slovarjev One Look *http://onelook.com*

Konkordančnik za korpus DSI: A WWW Concordance Service *http://nl2.ijs.si/index-mono.html*

Evroterm *http://www.gov.si/evroterm/*

Spletne strani:

Najdi.si *http://www.najdi.si*

Google *http://google.com/*

Monitor *http://www.monitor.si/*

Moj mikro *http://www.mojmikro.si/index.plus*

Slovenska odvisnostna drevesnica: prvi rezultati

Tomaž Erjavec*, Nina Ledinek†

*Odsek za tehnologije znanja, Institut "Jožef Stefan"
Jamova 39, 1000 Ljubljana
tomaz.erjavec@ijs.si
†Šoštanj, Slovenija
nina.ledinek@siol.net

Povzetek

Članek obravnava rezultate prve faze gradnje korpusa Slovenske odvisnostne drevesnice (Slovene Dependency Treebank, SDT), ki trenutno obsega 2.000 povedi oz. približno 30.000 besed. Zaradi skladenjskih sorodnosti med češčino in slovenščino, dostopnosti izčrpnega priročnika za površinskoskladenjsko označevanje češčine in inteligentnega urejevalnika dreves je označevalni sistem SDT oblikovan po modelu korpusa Prague Dependency Treebank (PDT). Korpus SDT zaenkrat sestavlja del oblikoskladenjsko označenega vzporednega korpusa MULTEXT-East, tj. prvi del prevoda romana *1984* Georgea Orwella. Korpus je bil najprej označen avtomatsko, nato pa so bile jezikovnoanalitične skladenjske oznake s pomočjo urejevalnika TrEd popravljene še ročno. Vzporedno je potekalo tudi prilagajanje češkega priročnika za označevanje za slovenščino. Trenutna verzija korpusa je dostopna v formatih XML/TEI in izpeljanih formatih in je že bila vključena v več raziskav, predvsem tisto v okviru CoNLL-X, ki je evalvirala natančnost dvajsetih dependenčnih razčlenjevalnikov na drevesnicah trinajstih jezikov. Prispevek predstavlja tudi načrte za nadaljnje delo, zlasti v zvezi s korpusom, ki bo Slovenski odvisnostni drevesnici dodan v prihodnje.

Slovene Dependency Treebank: first results

The paper presents the first release of the Slovene Dependency Treebank, currently containing 2,000 sentences or 30,000 words. Our approach to annotation is based on the Prague Dependency Treebank, which serves as an excellent model due to the similarity of the languages, the existence of a detailed annotation guide and an annotation editor. The initial treebank contains a portion of the MULTEXT-East parallel word-level annotated corpus, namely the first part of the Slovene translation of Orwell's "1984". This corpus was first parsed automatically, to arrive at the initial analytic level dependency trees. These were then hand corrected using the tree editor TrEd. The current version is available in XML/TEI, as well as derived formats, and has been used in several comparative evaluations, e.g. as part of the dataset for the CoNLL-X shared task on dependency parsing. Further work, in the first instance the composition of the corpus to be annotated next is also discussed.

1. Uvod

Skladenjsko označeni korpusi[1] postajajo pomembni jezikovni viri, saj omogočajo statističen pregled distribucije skladenjskih kategorij na velikem vzorcu besedil dejanske jezikovne rabe – pri čemer skladenjsko analizo (relativno) velikega vzorca realnih besedil navadno predpostavljajo in jo, predvsem, napovedujejo – in tako olajšujejo raziskave teoretičnega jezikoslovja ter skladnje posameznih jezikov. Poleg tega potrebujemo podatke, ki jih skladenjsko označeni korpusi nudijo, tudi za razvoj jezikovnih tehnologij, saj je na njih mogoče testirati in predvsem šolati avtomatske skladenjske označevalnike, pri čemer dajejo v zadnjem času zelo obetavne rezultate zlasti statistični označevalniki.

Zaenkrat obstaja kar nekaj problemsko zamejenih opisov skladnje slovenščine, ki sledijo različnim jezikoslovnim usmeritvam, in (nesočasnih) slovnic slovenskega jezika.[2] Najbolj celovita je Slovenska slovnica (Toporišič, 1984), najizčrpnejša opisa različnih vidikov slovenske skladnje pa sta Nova slovenska skladnja (Toporišič, 1982) ter Vezljivost v slovenskem jeziku (s poudarkom na glagolu) (Žele, 2001), poleg tega pa so bili številni skladenjski fenomeni raziskani zlasti v okviru generativne paradigme. Vendar pa za slovenščino še vedno ne obstaja nobena strogo formalna, računalniška (tj. primerna za računalniško obravnavo jezika) in obenem izčrpna slovnica. Do nedavnega tudi skladenjsko označenega korpusa slovenskega jezika še nismo imeli, saj so bili dostopni samo oblikoskladenjsko označeni in lematizirani korpusi slovenščine (Jakopin in Bizjak, 1996; Lönneker, 2005; Erjavec et al., 1998; Erjavec, 2006).

Z gradnjo korpusa Slovene Dependency Treebank[3] smo začeli leta 2003. V prvi fazi smo izbrali teoretični model označevanja, usposobili programsko platformo in pripravili korpus, tako da smo ga avtomatsko skladenjsko označili. V naslednji fazi smo se posvetili ročnemu površinskoskladenjskemu označevanju 2.000 povedi oz. 30.000 besed korpusa in pripravi priročnika za površinskoskladenjsko označevanje slovenskega jezika. Rezultati dela, ki je bilo zaključeno pred kratkim (Džeroski et al., 2006), so že vidni, saj je bil korpus že uporabljen v dveh raziskavah.

Čeprav je zaenkrat označen relativno majhen nabor povedi, smo v času od začetka projekta uspeli ustvariti za skladenjsko označevanje potrebno infrastrukturo. V razdelku 2 bomo zato prikazali, kakšen teoretični model smo za gradnjo korpusa izbrali, razdelek 3 pojasnjuje, kako smo ga operacionalizirali in prilagodili za slovenščino, v razdelku 4 pa bomo predstavili rezultate tekmovanja CoNLL-X v zvezi s korpusom SDT. Razdelek 5 prinaša nekaj zaključkov.

[1] Za dobronamerne pripombe in komentarje, ki so pripomogli k izboljšanju prvotne verzije besedila, se avtorja zahvaljujeta anonimnemu recenzentu. Za morebitne napake, ki se v članku še vedno pojavljajo, sta odgovorna sama.

[2] Npr. Breznikova (1916–1934), čitankarska (Bajec et al.; 1940–1956, 1964), Toporišičeva (1976–2004) ipd. Pogosto so nastajale (tudi) kot nekakšni nadomestki za srednješolske učbenike ali pa kot njihova nadgradnja.

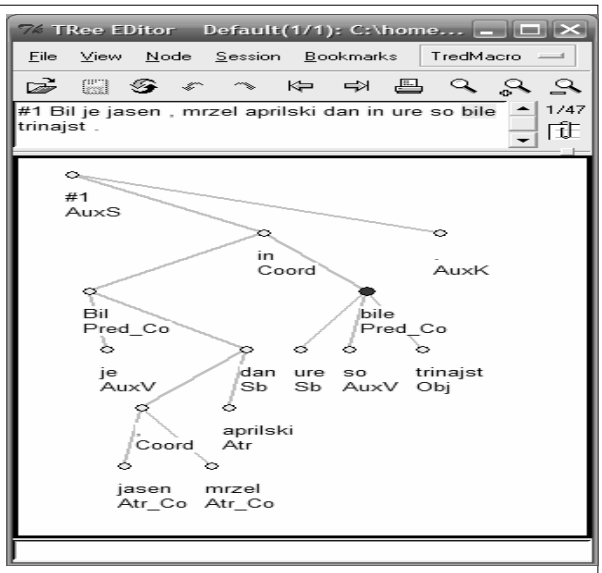
[3] http://nl.ijs.si/sdt/

2. Ozadje

Pri gradnji skladenjsko označenega korpusa, zlasti za jezik, ki takšnega korpusa še nima, je odločilnega pomena, kakšen teoretičen (in praktičen) model označevanja razvijemo oz. prevzamemo. Projekt Slovenska odvisnostna drevesnica namenskega vira financiranja zaenkrat nima, zato razvijanje lastnega modela označevanja ni bilo mogoče, hkrati pa bi bila priprava takšnega sistema otežena zaradi kadrovskih in časovnih omejitev. Glede na tipološke sorodnosti med jeziki in teoretične modele njihove obravnave smo torej med javno dostopnimi označevalnimi modeli izbrali najbolj ustreznega in prevzete jezikoslovne rešitve prilagodili za slovenščino.

Čeprav za veliko germanskih ter romanskih jezikov in za nekatere slovanske jezike, npr. bolgarščino (Simov et al., 2002) in ruščino (Boguslavsky et al. 2000), skladenjsko označeni korpusi in programska orodja za njihovo analizo že obstajajo, smo za slovenščino našli vzornika v projektu Prague Dependency Treebank[4] (LDC, 2001). Gre za enega najbolj ambicioznih in najbolje dokumentiranih projektov skladenjskega označevanja morfološko bogatih jezikov s prostim besednim redom, poleg tega pa je za komparativno analizo že na voljo zelo obsežen, na dveh ravneh označen korpus. Projekt PDT je za slovenščino posebej relevanten še zlasti zato, ker smo lahko zaradi pomensko-, funkcijsko- in strukturnoskladenjske podobnosti med slovenščino in češčino poleg teoretičnega modela (dependenčna skladnja, funkcijska generativna slovnica) v prvi fazi dela neposredno prevzeli tudi sistem ročnega površinskoskladenjskega označevanja korpusa, definiranega v priročniku *Annotations at Analytical Level: Instructions for Annotators* (Bémová et al., 1999) (v nadaljevanju: *AAL*), poleg tega pa smo imeli na voljo še urejevalnik dreves, ki zelo olajšuje ročno označevanje korpusa in omogoča njegovo vizualizacijo.

Na analitični, tj. površinskoskladenjski ravni, pri čemer so upoštevana zlasti funkcijskoskladenjska razmerja, je struktura vsake povedi v korpusu PDT predstavljena s skladenjskim drevesom, v katerem je razločevalno opredeljen tip površinskoskladenjske odvisnosti vsake pojavnice v povedi v razmerju do njenega neposredno nadrejenega elementa. Tektogramatična raven oz. pomenskoskladenjska raven že vključuje globlja, semantična razmerja med besedami, upoštevana pa so tudi koreferenčna razmerja, rematsko-tematska struktura povedi oz. členitev po aktualnosti. Korpus SDT je zaenkrat označen le na površinskoskladenjski ravni.



Slika 1: Skladenjsko drevo iz korpusa SDT v TrEdu.

2.1. Priročnik za površinskoskladenjsko označevanje

Ena od prednostnih nalog pri gradnji in označevanju skladenjsko označenega korpusa, zlasti če je označevalcev korpusa več, je pripraviti priročnik za označevanje, ki bo predpisoval način označevanja za čim več skladenjskih struktur. Opis skladenjskih razmerij mora biti zaradi zahtev avtomatske obdelave jezika skrajno formaliziran in izčrpen, poleg tega pa mora ilustrirati označevalne konvencije tudi z zgledi (delov) skladenjskih dreves. Priprava takšnega priročnika je zelo zahtevna in zamudna, potekati pa mora hkrati z ročnim označevanjem, kot nekakšna sinteza analize, opisa in, posledično, "izčiščenja" označevalnega sistema na podlagi pridobljenih izkušenj.

Za korpus PDT so javno dostopen priročnik za površinskoskladenjsko označevanje (v češčini) pripravili leta 1997 (Hajič et al., 1997). Zaradi skladenjske sorodnosti med češčino in slovenščino smo začeli korpus SDT označevati po pravilih priročnika *AAL*, istočasno pa smo ga začeli primerjati z obstoječimi opisi slovenske skladnje in ga nato za potrebe slovenščine prilagajati glede na izkušnje pri ročnem označevanju ter glede na razumevanje pomensko-, funkcijsko- ter strukturnoskladenjske vloge struktur v novejšem slovenskem jezikoslovju. Proces prilagajanja sistema za površinskoskladenjsko označevanje razumemo kot permanenten proces, ki bo, glede na to, da korpusi kažejo, da je jezik v skladenjskem smislu veliko bolj raznolik, pravila pa mnogo manj določljiva, kot kažejo mnogi sedanji jezikoslovni opisi, trajal še kar nekaj časa. Definirati bo treba način označevanja struktur, specifičnih za slovenščino,[5] češke zglede bo treba nadomestiti s

[4] The Prague Dependency Treebank, 1.0 in 2.0β, http://ufal.mff.cuni.cz/pdt/

[5] Te strukture je večinoma mogoče odkriti le naključno, pri primerjavi opisov in ročnem označevanju povedi (ko naletimo na strukturo, za katero menimo, da v slovenskem jezikoslovju v

slovenskimi, vse druge spremembe priročnika *AAL*, zlasti tiste, ki so posledica razlik (v interpretaciji) jezikovnih sistemov, pa bomo morali zelo natančno dokumentirati.

V trenutni fazi dela prilagajamo samo označevanje površinskoskladenjske ravni, načrtujemo pa, da bomo korpus kasneje označili tudi pomenskoskladenjsko. Glede na spremembe na površinskoskladenjski ravni bo treba prilagoditi tudi nekatere pomenskoskladenjske oznake, vendar pričakujemo, da bo zaradi večje stopnje "univerzalnosti" te ravni prilagoditev manj kot sprememb na površinskoskladenjski ravni.[6]

Vpeljani označevalni sistem je primeren predvsem za skladenjsko označevanje pisnih tekstov, za analizo govornih korpusov pa ga bo treba dodatno spremeniti (ali morda vpeljati novega). Skladnja govornega teksta je namreč veliko bolj kompleksna oz. manj "urejena" kot skladnja pisnih besedil (Halliday, 1989), njenih pojavnih oblik zato ne smemo obravnavati kot "odstopov" od pisne norme. Ker za slovenščino pregleden in izčrpen opis skladnje govornih tekstov še ne obstaja in ker niti dileme v zvezi z označevanjem govornih korpusov na "nižjih" ravneh, ki so predpogoj za skladenjsko označevanje oz. njegova predstopnja, večinoma še niso razrešene, priprave skladenjsko označenega govornega korpusa pri projektu Slovenska odvisnostna drevesnica zaenkrat ne načrtujemo,[7] kljub temu da se glede na tendenco razvoja korpusnega jezikoslovja pomembnosti področja zavedamo.

2.2. Urejevalnik dreves TrEd in skladenjski razčlenjevalnik za slovenščino

Pomembno orodje pri gradnji skladenjsko označenega korpusa je urejevalnik dreves, ki omogoča vizualizacijo in ročno korekcijo (avtomatsko že označenih) skladenjskih dreves. Dober urejevalnik označevalcu korpusa označevanje zelo olajša, hitrost dela se poveča, število napak pri označevanju pa je znatno manjše.

Za delo s korpusom PDT je bil razvit urejevalnik TrEd (Hajič et al., 2001), ki je javno dostopen, zato ga uporabljamo tudi pri projektu Slovenska odvisnostna drevesnica. Napisan je v programskem jeziku Perl/Tk in deluje tako na operacijskem sistemu Linux kot na sistemu Windows. Omogoča navigacijo med datotekami in povedmi, označevanje struktur z operacijo "primi in spusti" ter hitro izbiro analitičnih oznak s seznamov. Program je zelo konfigurabilen ter podpira precejšnje število vhodnih in izhodnih formatov (npr. XML/TEI, omogoča pa tudi prikaz skladenjskih dreves v formatu GIF). Slika skladenjskega drevesa, kot jo vidimo na računalniškem ekranu v programu TrEd, je prikazana na Sliki 1.

Da bi se označevalcem korpusa delo olajšalo, je bil razvit skladenjski razčlenjevalnik za slovenščino (Džeroski et al., 2006), ki deluje na osnovi majhnega števila ročno napisanih pravil. Program izkorišča oznake, ki so bile prvotnim besednim oblikam pripisane na oblikoskladenjski ravni in ki dajejo sorazmerno dobro informacijo o (potencialni) skladenjski strukturi povedi, in s pomočjo te informacije prvotnim besednim oblikam pripiše analitične oznake in odvisnostna razmerja.

3. Korpus SDT

V prvi fazi gradnje korpusa smo izvorne datoteke v formatu XML (glej razdelek 3.1) pretvorili v format programa TrEd. Razdelili smo jih na manjše datoteke, ki vsebujejo približno 50 povedi.[8] Te so bile nato najprej označene avtomatsko, nato pa ročno pregledane in popravljene. Kasneje smo analitične oznake TrEdovih datotek pridružili izvornim podatkom, nastali korpus pa je bil nato zopet pretvorjen v dokument XML (glej razdelek 3.2) in trenutno inačico korpusa SDT. Nadaljnji načrti za razširitev korpusa so predstavljeni v razdelku 3.3.

3.1. Korpus MULTEXT-East "1984"

V prvi fazi dela se nam je zdelo najpomembneje, da je korpus, ki naj bi služil kot osnova za skladenjsko označevanje, čim bolje dokumentiran in da je oblikoskladenjsko čim natančneje označen. Kot izvorni korpus za površinskoskladenjsko označevanje smo zato izbrali slovenski del oblikoskladenjsko označenega vzporednega korpusa MULTEXT-East (Erjavec, 2004), ki vsebuje oblikoskladenjsko označen prevod romana *1984* Georgea Orwella.

Korpus MULTEXT-East je zapisan v formatu XML, upošteva priporočila iniciative TEI P4 (Sperberg-McQueen in Burnard, 2002) ter je stavčno poravnan z angleškim originalom in prevodi romana v nekatere druge jezike. Razdvoumljanje oblikoskladenjskih oznak in lem glede na kontekst je potekalo v dveh fazah, najprej avtomatsko, nato pa so bile oznake pregledane še ročno. Njihova definicija je bila prevzeta po načelih projekta MULTEXT in oblikovana v sodelovanju z iniciativo EAGLES, kar omogoča večjo izmenljivost podatkov, poleg tega pa zagotavlja tudi možnost njihove avtomatske analize.

V SDT je trenutno zajet prvi del romana, ki vsebuje tretjino besedila, tj. okoli 30.000 besed oz. 2.000 povedi. Korpus glede na kvaliteto in obseg že vsebovanih oznak sicer nudi dobro osnovo, ima pa takšen izbor tudi nekaj očitnih pomanjkljivosti: korpus sestavlja eno samo prevodno umetnostno besedilo, ki vsebuje tudi izmišljen jezik (novorek), poleg tega pa so za tekst značilni dolgi stavki in premi govor, roman pa je na nekaterih mestih tudi nekoliko slabše preveden in zlektoriran, kar njegovo označevanje zelo otežuje.

3.2. Korpus SDT 0.4

Verzija 0.4 korpusa SDT[9] obsega 1998 povedi (29.991 besed in 6.563 ločil), ki so bile ročno površinskoskladenjsko označene, že izvorni MULTEXT-

funkcijskoskladenjskem smislu še ni bila opisana), zato jih bomo v priročnik za označevanje dodajali vedno znova.

[6] Upoštevati pa moramo tudi dejstvo, da se označevalna sistema PDT in SDT razlikujeta že na oblikoskladenjski ravni, saj prvi predvideva približno 4700 oznak, drugi pa le okrog 2100 oznak.

[7] Kolikor je avtorjema članka znano, za noben jezik skladenjsko označeni govorni korpus še ne obstaja, obstajajo le načrti zanj (npr. pri projektu PDT).

[8] Takšna razdelitev teksta je pomembna zlasti iz psiholoških razlogov, saj lahko označevalec na dan označi le približno 50 povedi. Število označenih datotek je za označevalca pomembno merilo napredka dela.

[9] Kolofon SDT 0.4 je dosegljiv na naslovu http://nl.ijs.si/sdt/sdtHeader-2006-05-17.html

East pa ročno lematiziran in oblikoskladenjsko označen. Dostopen je v nekaj različnih formatih, kanoničen format je format korpusa MULTEXT-East TEI P4 z dodanimi atributi pojavnic, v katerih je kodirana kazalka na neposredno nadrejeno pojavnico (parent node, atribut dep) in tip skladenjske odvisnosti med starševsko in hčerinsko pojavnico (vloga pojavnice na površinskoskladenjski ravni, atribut afun in, za koordinacije, parallel). Primer z začetka korpusa je podan na Sliki 2.

```
<text id="Osl." lang="sl">
 <body>
  <div type="part" id="Osl.1">
   <div type="chapter" id="Osl.1.2">
    <p id="Osl.1.2.2">
     <s id="Osl.1.2.2.1">
      <w id="s1t1" dep="s1t8"
          afun="Pred" parallel="Co"
          ana="Vcps-sma"
          lemma="biti">Bil</w>
      <w id="s1t2" dep="s1t1"
          afun="AuxV"
          ana="Vcip3s--n"
          lemma="biti">je</w>
      <w id="s1t3" dep="s1t4"
           afun="Atr" parallel="Co"
           ana="Afpmsnn"
           lemma="jasen">jasen</w>
      <c id="s1t4" dep="s1t7"
           afun="Coord">,</c>
```

Slika 2: SDT v kanoničnem formatu TEI; začetek korpusa "Bil je jasen, mrzel aprilski dan …".

Korpus SDT je sestavljen iz kolofona TEI, ki korpus dokumentira, in treh dokumentov TEI. Prvi vsebuje formalne oblikoskladenjske specifikacije korpusa MULTEXT-East, ki definirajo nabor oznak za oblikoskladenjske kategorije, uporabljene pri označevanju korpusa (torej vrednosti atributa ana). Drugi dokument prinaša seznam možnih analitičnih oznak (afun in parallel). Tretji dokument obsega zaenkrat edino (dokončno urejeno) komponento skladenjsko označenega korpusa, kot rečeno, 1. tretjino slovenskega prevoda romana *1984*.

3.3. Razširitev korpusa

V nadaljnjih fazah projekta se bomo osredotočili predvsem na dva segmenta dela. Prioriteta bo še naprej prilagajanje priročnika za površinskoskladenjsko označevanje (v prvi fazi smo pozornost posvečali predvsem prilagajanju sistema označevanja struktur, ki jim v slovenskem jezikoslovju navadno pripisujemo vlogo povedka), k delu bo zato treba pritegniti tudi več novih označevalcev, poleg tega pa bomo pripravili nov tekst za označevanje, s katerim bomo korpus SDT razširili. Ob tem se seveda pojavlja dilema, kateri tekst naj kot novo komponento korpusa izberemo.

Kot pomemben dejavnik je treba upoštevati Penn Treebank (Marcus et al., 1993), enega najrelevantnejših skladenjsko označenih korpusov. Oblikovanje korpusa, ki bi bil glede dokumentiranosti in označevanja primerljiv s korpusom Penn Treebank, bi omogočilo komparativne analize in poenostavilo druge raziskave. Poleg tega se je z objavo korpusa Prague Czech-English Dependency Treebank, ki vsebuje prevod dela korpusa Penn Treebank v češčino, oba dela vzporednega korpusa pa sta označena z analitičnimi oznakami, pojavila nova priložnost za raziskave. Prevod istega dela korpusa Penn Treebank v slovenščino in njegova označitev bi pomenila nastanek trijezičnega vzporednega korpusa, tak jezikovni vir pa bi bil odličen za medjezikovne raziskave in raziskave strojnega prevajanja. Zagotovljena bi bila tudi možnost za raziskave učenja avtomatskega skladenjskega označevanja s pomočjo prenosa pravil med jeziki (Kuhn, 2004), pri čemer bi se skušali naučiti označevanja slovenščine prek skladenjskega razčlenjevanja češčine.

Slovensko odvisnostno drevesnico bi radi uporabili čim prej – za to je pomembno, da podatke za razvoj računalniških aplikacij za specifično rabo pridobimo iz tekstov, povezanih z istim specifičnim področjem. Zato bomo v naslednji fazi označili vzorec povedi iz dveh korpusov, SVEZ-IJS (Erjavec, 2006) in korpusa časopisnih člankov. Vzorec iz vzporednega angleško-slovenskega korpusa SVEZ-IJS, ki obsega približno 800 povedi in 15.000 besed, je površinskoskladenjsko že označen, vendar pa je zaenkrat dostopen le v formatu fs. Za njegovo označevanje smo se odločili, ker so bile oblikoskladenjske oznake vzorca ročno popravljene in ker je bil korpus dostopen, hkrati pa je aplikativno zanimiv – z nastankom velikega vzporednega skladenjsko označenega korpusa bi bile mogoče raziskave strojnega prevajanja in druge medjezikovne raziskave. Z vidika reprezentativnosti je izbira korpusa seveda manj ustrezna, saj ga sestavljajo prevodni teksti, zato je tipologija dobljenih stavčnih vzorcev za slovenščino (lahko) vprašljiva, poleg tega pa so upravno-pravni teksti v skladenjskem smislu specifični (obsežne naštevalne enote ter "tabelarnost" in siceršnja skrajna formaliziranost določenih delov teksta npr. povzročajo, da je precejšen del povedi interpretiran kot niz elips, poleg tega pa velik del vzorca sestavljajo zelo obsežne in kompleksne samostalniške zveze). Status normativne reference, rečeno pogojno, bi lažje pripisali korpusu slovenskih časopisnih besedil, ki ga bomo označevali v naslednji fazi dela, vzorec reprezentativnejših besedilnih tipov pa bo (verjetno) vzet iz korpusov Fida ali FidaPlus.

4. Šolanje in evalvacija razčlenjevalnikov na korpusu SDT

Drevesnice so po eni strani uporabne za jezikoslovne raziskave jezikov, po drugi pa za razvoj jezikovnih tehnologij, saj avtomatsko skladenjsko razčlenjevanje besedil omogoča bistveno boljšo osnovo za nadaljnje obdelave, npr. strojno prevajanje, iskanje informacij, avtomatsko sumarizacijo itd.

Tradicionalni skladenjski razčlenjevalniki iz 70. in 80. let so temeljili na ročno napisanih pravilih in leksikonu, vendar pa je bilo zanje tipično majhno pokritje pravil, poleg tega niso bili odporni na napake v besedilih, na neznane besede in konstrukcije, niso pa tudi semantično oz. kontekstno razdvoumili besedila, kar pomeni, da je dobila ena poved veliko število različnih analiz. Samo za nekaj največjih jezikov so bili razčlenjevalniki (pravila,

leksikon) izdelani do te mere, da so postali uporabni za analize odprtega besedila, saj zahteva izdelava potrebne infrastrukture ogromno dela in sredstev.

V zadnjih letih se je izredno okrepilo zanimanje za metode obravnave jezika, ki temeljijo na pristopih strojnega oz. statističnega učenja. Skupno jim je to, da se orodja, ki te metode uporabljajo, določenega modela jezika induktivno naučijo iz vnaprej pripravljenih podatkov, v našem primeru skladenjsko označenega korpusa. Ti pristopi so robustni in (seveda ob izdelanem označenem korpusu) ceneni, vendar pa dostikrat delajo "neumne" napake, naučeni modeli pa so netransparentni.

Aktualnost večjezičnega induktivnega skladenjskega razčlenjevanja se je pokazala v precejšnjem zanimanju za uporabo korpusa SDT, kljub njegovemu majhnemu obsegu in dejstvu, da je sredi razvoja. Že prototipna inačica SDT 0.1 je bila uporabljena v raziskavi o stopnji natančnosti skladenjskega razčlenjevanja (Chanev, 2005), kasneje pa tudi v drugih raziskavah z razčlenjevalnikom MALT (Nivre in Hall, 2005).

SDT 0.3 je bil nato vključen v dosti širšo evalvacijo, ki se je dogajala v sklopu konference CoNLL-X "10th Conference on Computational Natural Language Learning"[10] (CoNLL, 2006). CoNLL vsako leto organizira, po vzoru sedaj že številnih drugih konferenc, odprto tekmovanje (shared task) iz nekega področja strojnega učenja jezikovnih podatkov, pri čemer je bilo tekmovanje v letu 2006 posvečeno učenju odvisnostnih slovnic.[11] Naloga je zajemala testiranje na korpusih več jezikov, saj je moral vsak tekmovalec svoj razčlenjevalnik preizkusiti na vseh ročno označenih drevesnicah jezikov, ki so bile v tekmovanje vključene. Kot baze podatkov so bili testirani korpusi[12] 13 jezikov, od daljno- in bližnjevzhodnih do obilice evropskih, tudi češčina in slovenščina. Češki korpus je bil korpus PDT, z milijon besedami je bil eden večjih, slovenski SDT pa je bil najmanjši korpus, ki je na tekmovanju sodeloval.

Na tekmovanje je bilo prijavljenih 20 sistemov, potekalo pa je tako, da so tekmovalci dobili večji del korpusa za učenje svojega razčlenjevalnika, delovanje naučenega sistema pa je bilo potem preizkušeno na skritem delu korpusa. Tako učni kot skriti korpus sta vsebovala tudi oblikoskladenjske oznake in leme, kar je sistemom razčlenjevanje lahko olajšalo. Ocenjevanje je potekalo z enotnim programom, ki je meril rezultat (v odstotokih) za:

1. označeno povezanost (OP): pravilne so tako odvisnostne povezave kot tudi oznake (labeled attachment score);
2. neoznačeno povezanost (NP): pravilne so odvisnostne povezave, pravilnost oznak ni relevantna (unlabeled attachment score);
3. oznake (OZ): pravilne so oznake, pravilnost odvisnostnih povezav ni relevantna (label accuracy).

[10] New York, 8–9 junij 2006, http://www.cnts.ua.ac.be/conll/

[11] Opis vseh sistemov (zbornik) in rezultati so dostopni na http://nextens.uvt.nl/~conll/

[12] Tekmuje se v natančnosti, ki jo razčlenjevalniki pri analizi veliko različnih drevesnic dosežejo, vendar pa tovrstno tekmovanje daje posredno tudi informacije o natančnosti označevanja korpusov samih in o tipu razčlenjevalnikov, ki pri določeni predstavitvi jeziko(slo)vnih podatkov dosegajo najboljše rezultate.

Če je npr. zveza "bela hiša" v neosebkovi vlogi označena tako, da obstaja povezava med podrejeno pojavnico "bela" in nadrejeno pojavnico "hiša" in je ta povezava označena kot »Subj«, je neoznačena povezanost pravilna, označena povezanost in oznaka pa ne.

Rezultati posameznih sistemov se zelo razlikujejo, tako med seboj kot glede na jezik. Najboljše rezultate, tako v povprečju za vse jezike kot tudi za češčino in slovenščino, sta imela dva sistema:

1. dvostopenjski razlikovalni razčlenjevalnik (two-stage discriminative parser) (McDonald et al., 2006);
2. označeno psevdoprojektivno odvisnostno razčlenjevanje z uporabo SVM (labeled pseudo-projective dependency parsing with support vector machines) (Nivre et al., 2006).

Natančnost obeh sistemov se zelo razlikuje glede na obravnavane jezike. McDonald et al. (2006) imajo npr. najboljši rezultat za japonščino in bolgarščino, najslabšega pa za češčino, danščino, slovenščino, turščino in, končno, arabščino, pri čemer so razlike odraz ne samo različnosti jezikov, temveč tudi velikosti in raznovrstnosti korpusov, teoretičnih izhodišč in doslednosti označevanja.

V Tabeli 1 vidimo dosežene stopnje natančnosti za SDT in PDT, in sicer posebej za oba razčlenjevalnika in v povprečju za vseh 20 prijavljenih sistemov. Tabela kaže, da rezultati za slovenski jezik zaostajajo za rezultati za češčino, vendar je to delno pričakovano zaradi ogromne razlike v velikosti korpusov. Češki je za sodobne sisteme včasih celo prevelik, saj nekateri sistemi zaradi časovno zahtevnega učenja niso mogli uporabiti celega korpusa. Vendar pa je češki korpus tudi bolj raznolik, saj vsebuje besedila iz mnogih virov, zato je njegovo označevanje v primerjavi s SDT, ki vsebuje samo en roman, v splošnem verjetno težje.

	SDT	PDT
OP McDonald et al.	73.44	80.18
OP Nivre et al.	70.30	78.42
OP povprečno	65.16	67.17
NP McDonald et al.	83.17	87.30
NP Nivre et al.	78.72	84.80
NP povprečno	76.53	77.01
OZ McDonald et al.	82.51	86.72
OZ Nivre et al.	80.54	85.40
OZ povprečno	76.31	76.59

Tabela 1: Rezultati CoNLL-X za slovenski in češki jezik. OP = označena povezanost, NP = neoznačena povezanost, O = oznake

Vseeno so rezultati za slovenski jezik spodbudni, saj je bilo z avtomatskimi orodji, ki so se učila na zelo majhnem vzorcu jezika, mogoče pravilno označiti skoraj tri četrtine povezav skupaj z njihovimi oznakami. Seveda pa se je v zvezi s temi rezultati treba zavedati, da so oblikoskladenjske oznake v SDT ročno pregledane. V realnih sistemih se te oznake določajo strojno, pri čemer pride tudi do napak. Natančnost razčlenjevalnika, ki bi potem take oznake uporabljal, bi bila brez dvoma bistveno manjša.

5. Zaključek

Prispevek prikazuje rezultate prve faze gradnje korpusa Slovene Dependency Treebank, ki je oblikovan po korpusu Prague Dependency Treebank. Čeprav korpus ni obsežen, je že bil uporabljen v nekaj raziskavah. Da bi bil maksimalno uporaben, smo ga pretvorili v tri formate, format TEI P4, format urejevalnika dreves TrEd, tj. format fs, in v tabularno datoteko (tabular file), format, uporabljen v raziskavah v okviru CoNLL-X. SDT je opisan na domači strani projekta http://nl.ijs.si/sdt/ in je prosto dostopen za raziskovalne namene.

Predstavili smo tudi načrte za nadaljnje delo, ki pa so odvisni tudi od možnosti financiranja projekta. Še naprej se bomo ukvarjali s prilagajanjem priročnika za označevanje, razširitvijo korpusa z novimi teksti, poleg tega pa se bomo začeli posvečati tudi raziskavam indukcije pravil z avtomatskim skladenjskim razčlenjevalnikom.

Literatura

Bémová, A., Buráňová, E., Hajič, J., Panevová, J., Urešová Z. (1999). Annotations at Analytical Level: Instructions for Annotators. Praga, UK MFF UFAL.

Boguslavsky, I., Grigorieva, S., Grigoriev, N., Kreidlin, L., & Frid, N. (2000). Treebank for Russian: Concept, tools, types of information. COLING-2000.

Chanev, A. (2005). Portability of Dependency Parsing Algorithms - an Application for Italian. V: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT'05). Barcelona.

CoNLL (2006). CoNNL-X "10th Conference on Computational Natural Language Learning", New York, 8–9 junij 2006.

Čmejrek, M., Curín, J., Havelka, J., Hajič, J., Kuboň, V. (2004). Prague Czech-English Dependency Treebank: Syntactically Annotated Resources for Machine Translation. LREC'04.

Džeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Žabokrtsky, Z., Žele, A. (2006). Towards a Slovene Dependency Treebank. V: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'2006). Pariz, ELRA.

Erjavec, T., Gorjanc, V., Stabej, M. (1998). Korpus FIDA. Konferenca Jezikovne tehnologije za slovenski jezik. Ljubljana, Institut Jožef Stefan.

Erjavec, T. (2006). The English-Slovene ACQUIS corpus. V: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'2006). Pariz, ELRA.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. V: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004). Pariz, ELRA.

Hajič, J., Pajas, P. in Vidová Hladká, B. (2001). The Prague Dependency Treebank: Annotation Structure and Support. IRCS Workshop on Linguistic databases, 2001 (pp. 105--114).

Hajič, J., Panevová, J., Buráňová, E., Urešová, Z., Bémová, A. (1997). A Manual for Analytic Layer Tagging of the Prague Dependency Treebank. UFAL Technical Report TR-1997-03, Karlova univerza, Češka republika.

Halliday, M. A. K. (1989). Spoken and written language. Oxford, University Press.

Jakopin, P., Bizjak, A. (1997). O strojno podprtem oblikoslovnem označevanju slovenskega besedila. Slavistična revija, 45(3–4), 513--532.

Kuhn, J. (2004). Experiments in Parallel-Text Based Grammar Induction. V: Proceeding of the ACL'04.

Ledinek, N. (2005) Površinskoskladenjsko označevanje korpusa Slovene Dependency Treebank (s poudarkom na predikatu). Diplomska naloga. Univerza v Ljubljani.

Ledinek, N., Žele, A. (2005). Building of the Slovene Dependency Treebank According to the Prague Dependency Treebank. V: Zbornik konference Gramatika & korpus. Praga, Ústav pro jazyk český. [V tisku].

Linguistic Data Consortium, (2001). Prague Dependency Treebank 1. LDC2001T10.

Linguistic Data Consortium, (2004). Prague Czech-English Dependency Treebank Version 1.0, LDC2004T25.

Lönneker, B. (2005). Strojno oblikoslovno označevanje slovenskih besedil: Kako daleč smo, Slavistična revija, 53(2), 193--210.

Marcus, M., Beatrice, P. S. & Markiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, 19/2.

McDonald, R., Lerman, K., Pereira, F. (2006). Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. CoNLL-X Shared Task: Multi-lingual Dependency Parsing. New York, 8–9 junij 2006. http://nextens.uvt.nl/~conll/slides/McDonald.pdf

Nivre, J., Hall, J. (2005). MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. V: Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT'05). Barcelona.

Nivre, J., Hall, J., Nilsson, J., Eryiğit, G., Marinov, S. (2006). Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines. CoNLL-X Shared Task: Multi-lingual Dependency Parsing. New York, 8–9 junij 2006. http://nextens.uvt.nl/~conll/slides/Nivre.pdf

Simov, K., Osenova, P., Slavcheva, M., Kolkovska, S., Balabanova, E., Doikoff, D., Ivanova, K., Simov, A., & Kouylekov, M. (2002). Building a linguistically interpreted corpus of Bulgarian: the BulTreeBank. LREC'02.

Sperberg-McQueen, C. M. in Burnard, L. (ur.) (2002). Guidelines for Electronic Text Encoding and Interchange, the XML Version of the TEI Guidelines. The TEI Consortium.

Toporišič, J. (1982). Nova slovenska skladnja. Ljubljana, DZS.

Toporišič, J. (1984). Slovenska slovnica. Maribor, Obzorja.

Žele, A. (2001). Vezljivost v slovenskem jeziku (s poudarkom na glagolu). Ljubljana, Založba ZRC, ZRC SAZU.

Žele, A. (2003). Glagolska vezljivost: iz teorije v slovar. Ljubljana, Založba ZRC, ZRC SAZU.

Oblikoslovno označevanje slovenskega jezika: primer korpusa SVEZ-IJS

Tomaž Erjavec†, Bence Sárossy‡

†Odsek za tehnologije znanja, Institut »Jožef Stefan«
Jamova 39, 1000 Ljubljana
tomaz.erjavec@ijs.si
‡Budimpešta, steksz@freemail.hu

Povzetek

Avtomatsko oblikoslovno označevanje (*part-of-speech tagging* oz. *word-class syntactic tagging*) je postopek, pri katerem se vsaki besedi, ki se v besedilu pojavi, pripiše oblikoslovna oznaka. Za slovenski jezik so raziskave te, za jezikovne tehnologije izredno pomembne, komponente še v povojih. V prispevku predstavljamo evalvacijo označevalnika TnT, ki je vgrajen v program totale in naučen modela označevanja na jezikovnih virih MULTEXT-East. Testna domena je prosto dostopen vzporedni angleško-slovenski korpus SVEZ-IJS, ki vsebuje pravni red Evropske unije. V prispevku opisujemo ročno popravljen vzorec iz korpusa in podrobno analiziramo napake pri označevanju. Predstavljamo tudi enostaven pretvorbeni program, ki popravi nekaj najbolj pogostih napak, in podajamo zaključke in smernice za nadaljnje delo.

Word-class Syntactic Tagging of Slovene: the case of the SVEZ-IJS corpus

Part-of-speech tagging or, more accurately, word-class syntactic tagging is a procedure that assigns to each word token appearing in a text its morphosyntactic description. Research on this important component of many language technology applications is, for the Slovene language, still at a preliminary stage. In the paper we evaluate the accuracy of the TnT tagger, as a part of the totale annotation tool, which had been trained on the MULTEXT-East language resources for Slovene. The test data come from the freely available parallel English-Slovene corpus SVEZ-IJS, which contains legal acts of the EU. Presented are the details of the manually corrected sample from the corpus and an analysis of the tagging errors. The paper also discusses a simple transformation-based program that fixes some of the more common errors, and concludes with some directions for future work.

1. Uvod

Oblikoslovno označevanje (*part-of-speech tagging* oz. *word-class syntactic tagging*) (van Halteren, 1999) je postopek, s katerim vsaki besedi, ki se v besedilu pojavi, pripišemo enolično oblikoslovno oznako. Programi za oblikoslovno označevanje morajo poljubnim besednim oblikam določiti možne oznake, nato pa izmed teh oznak izbrati pravo glede na kontekst, v katerem se besedna oblika pojavi. Tako ima npr. besedna oblika *hotel* tri možne oznake: dve samostalniški (ednina imenovalnika in tožilnika) in eno glagolsko (deležnik na -l moškega spola), v stavku *Šel je v hotel* pa mora biti ta pojavnica označena kot samostalnik v tožilniku.

Oblikoslovno označevanje je bilo najprej razvito za angleški jezik, v katerem so nabori oblikoslovnih oznak razmeroma majhni (~50, odvisno od specifike posameznih naborov oznak), problemi pa predvsem v dvoumnosti glede besedne vrste. Dosti kasneje so bili označevalniki razviti tudi za oblikoslovno bistveno bogatejše jezike, predvsem češčino (Hajič in Hladka, 1998). Pri pregibno bogatih jezikih, kot sta češčina ali slovenščina, je različnih oznak tudi po več tisoč, največji problem pa je, vsaj na prvi pogled, predvsem v razdvoumljanju sinkretičnih pregibnih oblik, torej v okviru besedne vrste.

Večina sodobnih oblikoslovnih označevalnikov se modela določenega jezika nauči, in to iz ročno označenih korpusov, po možnosti podprtih z oblikoslovnimi leksikoni. Takšni programi so sicer robustni (analizirajo tudi njim neznane besede oz. besedne zveze in so odporni na napake v besedilu), vendar pa delajo napake. Natančnost označevanja je odvisna od velikosti učne množice, konkretnega nabora oznak, besedilnozvrstne podobnosti testne in učne množice in seveda od izbranega algoritma za učenje in označevanje.

Tekst v slovenskem jeziku so verjetno prvi skušali oblikoslovno označiti na ZRC SAZU (Jakopin in Bizjak, 1997), in sicer z v urejevalnik EVA vgrajenim označevalnikom, ki deluje na osnovi pravil in velikega podpornega leksikona. Označevalnik je polavtomatski, saj se pri neznanih besedah in besedah, ki mu jih ni uspelo razdvoumiti, zanaša na intervencijo človeka. Zato je koristen predvsem za izdelavo ročno označenih korpusov, za kar je bil do sedaj tudi uporabljan.

Na tem mestu velja omeniti še avtomatsko označevanje korpusa FidaPLUS (http://www.fidaplus.net/), ki ga je razvilo podjetje Amebis, d.o.o. Vendar pa je ta označeni korpus še v nastajanju, o označevanju pa, kolikor nam je znano, še ni bilo nobene publikacije

Na IJS so poskusi z avtomatskim označevanjem vezani na nabor oblikoslovnih oznak, rabljenih pri projektu MULTEXT-East (Erjavec, 2004, http://nl.ijs.si/ME/), in na ročno označen korpus tega projekta, tj. Orwellov roman »1984«. Prvi poskusi (Erjavec et al., 2000) so pokazali, da ima od štirih preizkušenih javno dostopnih označevalnikov najboljše lastnosti statistični označevalnik TnT (Brants, 2000), ki je dosegel stopnjo natančnosti 89,2%, pri čemer je bila testna množica ravno tako iz romana »1984«. V članku smo analizirali tudi natančnost označevanja po besednih vrstah, ki je bila za TnT 96,6%. V nadaljevanju dela z avtomatskimi označevalniki na IJS (Erjavec in Džeroski, 2004) smo označevanje rahlo izboljšali in implementirali ter evalvirali tudi lematizacijo neznanih besed, tj. pripisa osnovne oblike vsaki besedi v besedilu, npr. *hotela* → *hotel* oz. *hoteti*, odvisno od konteksta. Problema sta povezana, saj je naša

implementacija lematizacije izrazito odvisna od oblikoslovnega označevanja.

Kolikor nam je znano, je edina druga raziskava, ki je imela namen evalvirati avtomatsko oblikoslovno označevanje slovenskega jezika, Lönneker (2005), ki opisuje uporabo označevalnika TreeTagger (Schmid, 1994) na ročno označenem korpusu ZRC SAZU, podaja pa tudi primerjavo z rezultati iz raziskave Erjavec et al. (2000). K primerjavi rezultatov raziskave Lönneker (2005) z našimi rezultati se bomo vrnili v razdelku 5.

V pričujočem članku ponavljamo evalvacijo označevanja iz raziskave Erjavec et al. (2000) v razširjeni obliki na novem korpusu, in sicer na vzorcu iz slovenskega dela korpusa SVEZ-IJS (Erjavec, 2006). Zanimalo nas je, kakšna je natančnost označevanja na korpusu, ki je po zvrstnosti zelo drugačen od učnega, katere napake so najbolj pogoste in škodljive in ali jih je mogoče, in do katere mere, na enostaven način odpraviti.

V nadaljevanju naprej predstavljamo program totale (Erjavec et al., 2005), ki ga uporabljamo za avtomatsko označevanje, nato korpus SVEZ-IJS in testni vzorec iz tega korpusa, ki smo ga ročno popravili. Temu sledi analiza napak pri avtomatskem označevanju, opis pretvorbenega programa, ki popravi najbolj pogoste napake, primerjava stopenj natančnosti, doseženih pri posameznih eksperimentih, ter nekaj zaključkov.

2. Testni podatki

Podatki, ki smo jih uporabili za evalvacijo, so sestavljeni iz vzorca korpusa SVEZ-IJS. Vzorec je bil avtomatsko označen s programom totale in nato ročno pregledan in popravljen.

2.1. Označevanje s totale

Za jezikoslovno označevanje uporabljamo program totale (tokenisation, tagging, and lemmatisation) (Erjavec et al., 2005), ki:

1. Besedilo tokenizira, torej razdeli na besede, ločila in povedi (z v totale vgrajenim programom mlToken).
2. Besede oblikoslovno označi (z v totale vgrajenim programom TnT) (Brants, 2000).
3. Besedilo lematizira (z v totale vgrajenim programom CLOG, upoštevajoč pripisane oblikoslovne oznake).

Takó oblikoslovno označevanje kot lematizacija se izvajata s programi, ki se modela jezika naučijo iz vnaprej pripravljenih podatkov, torej iz ročno označenega korpusa in oblikoslovnega leksikona. Naš model oblikoslovnega označevanja za slovenski jezik je naučen iz korpusa MULTEXT-East, tj. romana »1984« G. Orwella (100.000 pojavnic), ter majhnega vzorca iz korpusa IJS-ELAN (5.000 pojavnic), lematizator pa je naučen na oblikoslovnem slovarju MULTEXT-East (polne paradigme 15.000 lem), ki smo ga uporabili tudi za boljše delovanje označevalnika (Erjavec in Džeroski, 2004).

Korpus in leksikon vsebujeta oblikoslovne oznake MULTEXT-East za slovenski jezik, in sicer se v korpusu pojavi 1.023 različnih oznak, v leksikonu pa je predvidenih kar 2.083 oznak. Oznake so izvorno angleške[1] (tj. sestavljajo jih prve črke angleških besed) in so sestavljene iz niza črk, pri čemer prva črka označuje besedno vrsto, ostale pa, glede na besedno vrsto, vrednosti njenih atributov. Tako npr. oznaka *Ncmsn* pomeni: *PoS = noun*, *type = common*, *gender = masculine*, *number = singular*, *case = nominative*, ekvivalentna slovenska oznaka *Somei* pa: *besedna vrsta = samostalnik*, *vrsta = občno ime*, *spol = moški*, *število = ednina*, *sklon = imenovalnik*.

2.2. Korpus SVEZ-IJS

Vzporedni angleško-slovenski korpus SVEZ-IJS, http://nl.ijs.si/svez/ (Erjavec, 2006), zajema pravni red EU, t. i. Acquis Communautaire. Inačica 1.0 tega korpusa vsebuje 2 x 5 milijonov besed, nastala pa je leta 2004 na osnovi tedanjega pomnilnika prevodov prevajalske skupine pri SVEZ, Službi vlade RS za evropske zadeve (Erbič et al., 2005). Korpus je sestavljen iz poravnanih segmentov v angleškem in slovenskem jeziku, ki tipično vsebujejo eno poved ali del povedi. Korpus je zanimiv iz več razlogov:

- Je velik in vsebuje kvalitetno poravnane povedi v za nas najbolj aktualnem jezikovnem paru.
- Vsebuje (skoraj) identična besedila kot JRC-Acquis (Steinberger, 2006), v katerega pa je dodatno vključenih še 19 drugih jezikov, vendar pa ima korpus slabšo poravnavo.
- Je prosto dostopen za raziskovalne namene.

Ker je SVEZ-IJS torej izredno koristna podatkovna množica za raziskave in razvoj jezikovnih tehnologij za slovenski jezik, se nam je zdelo koristno preučiti napake v njegovih oblikoslovnih oznakah in poskusiti zmanjšati njihovo število.

2.3. Vzorec

Za evalvacijo označevanja smo iz avtomatsko označenega korpusa najprej izdvojili vzorec, v katerega smo vključili po 3 zaporedne slovenske segmente na vsakih 1.000 segmentov, s čimer smo zajeli približno 3 ‰ slovenskega besedila v korpusu. Ta vzorec je bil potem izvožen v Excelovo tabelo in tam ročno popravljen, pri čemer smo obdržali prvotne, avtomatsko pripisane oznake. S tem smo dobili testno množico, iz katere izvirajo vse statistike v pričujočem članku.

Enota	n	Razmerje	
Znakov	513.650		A
Segmentov	821	625 A/B	B
Vseh pojavnic	15.765	19 C/B	C
Ločil (pojavnic)	2.346	15% C	Č
Besed (pojavnic)	13.419	85% C	D
Besed (različnic)	5.189	2,59 D/E	E
Lem (različnic)	3.062	4,38 E/F	F
Oblik. oznak (različnic)	452	29,69 D/G	G

Tabela 1: Testni podatki, osnovna statistika.

Podrobna analiza velikosti vzorca je podana v Tabeli 1, kjer v stolpcu *n* npr. vidimo, da vsebuje vzorec približno pol milijona znakov in nekaj več kot 15.000 pojavnic, od katerih je skoraj šestina ločil, ter da je

[1] Oznake je možno avtomatsko preslikati oz. prevesti v slovenski jezik, in take se uporabljajo v korpusu FIDA.

različnih besednih oblik okoli 5.200, lem pa 3.000. Pri tem se za različne besedne oblike štejejo ortografsko različne oblike, tako da sta npr. *tudi* in *Tudi* dve različnici, vse leme pa so zapisane v malih črkah, zato imata besedni obliki *koren* in *Koren* enako lemo. Končno nas vrstica G opozarja na (presenetljivo majhno) število oblikoslovnih oznak v vzorcu, okoli 450.

Stolpec *Razmerja* podaja ulomke različnih (v stolpcu *n* prikazanih) števil in nas s tem seznanja npr. s povprečno dolžino segmenta v znakih (625) oz. pojavnicah (19) ali pa prikazuje, koliko različnih besednih oblik v povprečju pokriva ena lema (4.4).

	n	% vseh	% besed
Besed	13.419	85,1%	100%
(Znanih)	10.996	69,7%	81,9%
(Neznanih)	2.423	15,4%	18,1%
Samostalnik (N)	4.928	31,3%	36,7%
Glagol (V)	1.287	8,2%	9,6%
Pridevnik (A)	1.694	10,7%	12,6%
Prislov (R)	373	2,4%	2,8%
Številnik (M)	795	5,0%	5,9%
Zaimek (P)	743	4,7%	5,5%
Veznik (C)	1.102	7,0%	8,2%
Predlog (S)	1.787	11,3%	13,3%
Členek (Q)	107	0,7%	0,8%
Okrajšava (Y)	474	3,0%	3,5%
Neuvrščene (X)	128	0,8%	1,0%

Tabela 2: Testni podatki, znane/neznane besede in statistika po besednih vrstah.

V Tabeli 2 je bolj podrobno predstavljeno razmerje med besedami. Stopnja napake pri avtomatskem označevanju je seveda zelo odvisna od tega, ali se je neka besedna oblika v učni množici oz. leksikonu pojavila ali pa je sistemu povsem neznana. V tabeli vidimo, da neznane besede obsegajo okoli 15% vseh pojavnic in več kot 18% vseh besed, kar lepo kaže na leksikalno različnost korpusa SVEZ-IJS od korpusa MULTEXT-East oz. na pomanjkljivo in premajhno učno množico. Tabela podaja še statistiko po (ročno določenih, torej pravilnih) besednih vrstah, ki kaže, da je v besedilu največ samostalnikov, pridevnikov in predlogov, ki skupaj tvorijo kar polovico pojavnic. To pomeni, da je za visoko stopnjo natančnosti pomembna predvsem pravilna interpretacija teh treh besednih vrst, predvsem samostalnikov.

Posebno pozornost si zaslužita zadnji dve kategoriji v razpredelnici, in to iz dveh razlogov. Neuvrščenih besed (X) in okrajšav (Y) namreč ne kategorizirata po besednih vrstah in sta zato nekakšni metakategoriji. Tako so okrajšave z oblikoskladenjskega vidika tipično samostalniki, poleg tega pa kategorija Y zajema tudi kratice, ki lahko skladenjsko delujejo kot celotne fraze, kot npr. *itd.* Z X (tj. kot neuvrščene besede) označujemo tujejezične besede, ki jih je običajno po več skupaj, s skladenjskega vidika pa delujejo kot samostalniške fraze, npr. *Carte de séjour de résident privilégié de Monaco*. Druga posebnost kategorij, označenih z X in Y, pa je število pojavnic, ki jih zajemata, saj take pojavnice v korpusu SVEZ-IJS tvorijo razmeroma velik del besedila, 4,5%. Kot bomo videli, sta ti dve kategoriji odgovorni za velik delež napak pri avtomatskem označevanju.

3. Analiza avtomatskega označevanja

Na osnovi ročno popravljenega vzorca smo nato evalvirali natančnost avtomatskega označevanja s totale/TnT. Ob tem je pomembno ločiti dve vrsti napak: pri prvi vrsti je oblikoslovna oznaka sicer napačna, vendar je pravilno določena vsaj besedna vrsta, pri drugi vrsti pa je napačna tudi oznaka besedne vrste. Program pri prvi vrsti napak npr. napačno določi sklon ali število (ali oboje) ali pa (pod)vrsto besedne vrste, npr. lastna imena proti občnim imenom, kakovostni pridevniki proti vrstnim pridevnikom, pri drugi vrsti napak pa npr. proglasi samostalnik za glagol ali pa neuvrščeno besedo za samostalnik. Razlikovanje med temi napakami je koristno zato, ker je za marsikatero potencialno uporabo oz. uporabnika oblikoslovnih oznak pomembna samo besedna vrsta, ne pa npr. pregibne lastnosti. Zanimivo je ločiti tudi natančnost označevanja za znane in neznane besede, torej tiste, ki jih sistem v učni množici ni srečal. V Tabeli 3 predstavljamo absolutno število napak v oznaki in besedni vrsti, podajamo pa tudi natančnost lematizacije. Za vsako kategorijo (oz. število) prikazujemo ustrezno natančnost glede na vse pojavnice in glede na besedne pojavnice v vzorcu.

	n	Točnost pojavnice	Točnost besede
Napačna oznaka	1.799	88,6%	86,6%
Za znane besede	950	92,9%	91,4%
Za neznane besede	849	65,0%	65,0%
Napačna bes. vrsta	748	95,3%	94,4%
Za znane besede	155	98,8%	98,6%
Za neznane besede	593	75,5%	75,5%
Napačna lema	220	98,6%	98,4%
Za znane besede	88	99,3%	99,2%
Za neznane besede	132	94,6%	94,6%
Za napačno oznako	217	87,9%	87,9%
Za pravilno oznako	3	99,8%	99,8%

Tabela 3: Točnost avtomatskega označevanja.

Po najbolj strogi oceni ima sistem natančnost 86,6%. Tako nizka je predvsem zaradi precejšnjega števila neznanih besed. Natančnost glede na besedno vrsto je precej večja, 94,4%, je pa tudi tu napaka pri neznanih besedah bistveno večja kot pri znanih. Lematizacija ima celo večjo natančnost, 98,4%, kot označevanje besedne vrste, saj veliko napak označevalnika na pravilno lematizacijo ne vpliva, je pa natančnost za neznane besede zopet precej manjša. V zadnjih dveh vrsticah vidimo, da so za napačno delovanje lematizatorja skoraj v celoti krive napačne oblikoslovne oznake: samo v treh primerih se zgodi, da je oblikoslovna oznaka pravilna, lema pa ne.

3.1. Napake med besednimi vrstami

Zaradi pomembnosti napak pri označevanju besedne vrste bomo analizo le-teh ločili od tistih napak, ki nastanejo pri označevanju v okviru besedne vrste.

V Tabeli 4 podamo matriko števila napak glede na dejansko besedno vrsto (vodoravno) in besedno vrsto, ki jo je pojavnici pripisal označevalnik (navpično). Števila v diagonali tako predstavljajo napake, ki se zgodijo pri označevanju v okviru besedne vrste, ostale pa kažejo, katere zamenjave so najbolj pogoste, na primer, da so bili samostalniki petindevetdesetkrat označeni kot glagoli.

Tabela kaže, da je označevanje (glede na nabor) odprtih besednih vrst (zapisane so v krepkem tisku) bistveno manj uspešno kot označevanje funkcijskih besed, kar je razumljivo, saj je večina slednjih označevalniku poznana. Do neke mere je izjema zaimek, pa še ta samo glede stopnje napak pri označevanju v okviru besedne vrste. Razlog za majhno absolutno natančnost oznak za zaimek je, da imajo oblikoslovne oznake te besedne vrste izrazito razvejano strukturo, tako da zajemajo zaimki skoraj polovico (čez tisoč) vseh oblikoslovnih oznak, in da so besedne oblike vseh zaimkov sicer vsebovane v leksikonu, ne pa tudi v učnem korpusu, zato so napake pri sinkretičnih pregibanjih pogoste.

	N	***V***	***A***	***R***	*M*	*P*	*C*	*S*	*Q*	***I***	***X***	***Y***	*
N	**609**	6	9	4	47	0	1	1	0	0	69	241	987
V	95	**18**	2	1	28	2	2	0	0	0	35	17	200
A	28	1	**275**	12	8	3	0	0	0	0	14	9	350
R	14	1	4	**15**	0	1	1	1	0	0	6	11	54
M	0	0	1	0	**11**	6	0	0	0	0	0	18	36
P	1	0	1	0	2	**105**	0	0	0	0	0	1	110
C	0	1	0	3	0	0	**0**	0	10	0	0	11	25
S	1	0	0	0	1	0	0	**18**	0	0	1	6	27
Q	0	0	0	3	0	0	2	0	**0**	0	0	0	5
I	1	0	0	0	0	0	0	0	0	**0**	1	0	2
X	0	0	0	0	0	0	0	0	0	0	**0**	0	0
Y	0	0	0	0	1	0	0	0	0	0	2	**0**	3
*	749	27	292	38	98	117	6	20	10	0	128	314	1799

Tabela 4: Pregled napak pri določitvi besednih vrst.
N = samostalnik, V = glagol, A = pridevnik, R = prislov, P = zaimek, S = predlog, C = veznik, Q = členek, I = medmet, M = številnik, Y = okrajšava, X = neuvrščeno

Največkrat je napačna besedna vrsta pripisana samostalnikom, številnikom, neuvrščenim besedam in okrajšavam. Pri samostalnikih je relativna napaka sicer majhna, vendar je zaradi velikega števila samostalnikov za celotno natančnost vseeno pomembna. Posebej je opazna zamenjava samostalnika z glagolom, ki deloma verjetno izhaja tudi iz narave učne množice.

Pri ostalih treh besednih vrstah, namreč številnikih, neuvrščenih besedah in okrajšavah, so razlogi za veliko število napak podobni. Besede so po eni strani skoraj vedno neznane (še posebej zato, ker jih niti učni korpus niti leksikon tako rekoč ne vsebujeta, z izjemo omejenega števila številnikov), po drugi strani pa so oblikoskladenjsko slabo definirane, kar še posebej velja za neuvrščene besede in okrajšave. K temu problemu se bomo vrnili v razdelkih 4 in 5.

3.2. Napake v okviru besednih vrst

V tem razdelku bomo bolj podrobno pogledali najbolj pogoste napake pri označevanju v okviru besedne vrste. Kot kaže Tabela 4, je največ napak pri samostalnikih, ki imajo v oznakah MULTEXT-East pet atributov: vrsta, spol, število, sklon in živost. Kar 85% napak je vezanih na sklon, čeprav tudi v kombinaciji s številom, podobno pa je tudi pri pridevnikih in zaimkih, čeprav je pri teh večkrat napačen tudi spol. Bolj podrobna analiza teh napak je pokazala, da je v velikem številu primerov pravilno oznako pravzaprav nemogoče ugotoviti samo na osnovi lokalnega konteksta, torej takega, ki ga uporablja TnT. Zato bi bil za odpravljanje takšnih napak potreben drugačen pristop.

Ostale besedne vrste so za označevanje manj težavne. Pri glagolih sta najbolj problematični lastnosti spol in število.

4. Pretvorbeno označevanje s pravili

Ključno vprašanje seveda je, kako izboljšati natančnost označevanja. Kot preizkus v to smer smo implementirali program, ki popravi nekatere napake označevalca. V tem razdelku opisujemo delovanje programa in evalviramo rezultate.

Program, napisan v programskem jeziku Perl, vzame kot vhodne podatke že označeno besedilo, s čimer ima dostop do podatkov o obliki in tipu (ločilo, beseda) vsake pojavnice, za besede pa lahko preveri še pripisano (mogoče napačno) oblikoslovno oznako in frekvenco besedne oblike v učnih podatkih (0 = neznana beseda). Za označevanje vsake besedne pojavnice program nato sproži kaskado ročno napisanih pravil. Vsako pravilo je oblike »če *pogoj*, potem *pripiši oblikoslovno oznako*, drugače *naslednje pravilo*«. V pogojih uporabljamo funkcijo *feature*, ki vzame kot prvi argument lastnost, kot drugi pa pojavnico ter vrne vrednost te lastnosti za pojavnico. Kot primer podajamo prvi dve pravili:

```
        ...
①       elsif ($freq == 0  and feature("idwrd",$sent[$focus])=~/^[IVX]+$/)
          {$outmsd="Mc---r" }
②       elsif ($freq == 0 and
                feature("case",$sent[$focus]) eq 'uc' and
                not (feature("case",$sent[$focus-1])  eq  'uc' or
                     feature("case",$sent[$focus+1]) eq  'uc'))
          {$outmsd="Y"}
        ...
```

Prvo pravilo oblikoslovno oznako popravi v oznako `Mc---r`, ki označuje rimsko številko, saj je bilo napačno prepoznavanje le-teh eden večjih problemov pri označevanju številnikov. Pogoj določa, da mora biti beseda neznana (`$freq == 0`), obenem pa mora biti besedna oblika (lastnost `idwrd`) pojavnice, katere oznako popravljamo (`$sent[$focus]`), sestavljena samo iz znakov `I`, `V` in `X` (regularni izraz `/^[IVX]+$/`). Drugo pravilo popravi oznako besede tako, da jo uvrsti v kategorijo `Y`, tj. označi jo kot okrajšavo, in sicer če je beseda neznana, sestavljena iz samih velikih črk, pred in za njo pa nista kapitalizirani besedi. Na ta način npr. pravilno označimo zglede tipa *Čist dobiček ECB se prenese ...*, ne pride pa do napačne

označitve neznane besede v zgledih tipa *profesor dr. Walter HALLSTEIN , državni sekretar …*

Zaenkrat smo implementirali pet pravil, ki izhajajo iz analize nekaj najbolj pogostih, pa obenem najbolj »popravljivih« napak. Prvi dve pravili sta že bili opisani, tretje spremeni oznako v okrajšavo, če je neznana beseda sestavljena iz številk in največ treh črk (npr. *2002/917/ES*), četrto spremeni oznake vseh namenilnikov v oznake za samostalnike moškega spola v osnovni obliki, peto pa spremeni oznako *a*, če mu sledi ločilo, tako, da mu namesto vezniške vloge podeli vlogo okrajšave.

Tabela 5 podaja rezultat popravkov pri označevanju s temi petimi pravili. V prvem stolpcu so števila za popravek besedne vrste, v drugem pa za popravek celotne oblikoslovne oznake. Prva vrstica prikazuje število pravilno popravljenih pojavnic, druga vrstica pa število pojavnic, ki so bile po prvotnem označevanju pravilne, vendar so bile popravljene v napačno besedno vrsto oz. oblikoslovno oznako. V tretji vrstici je število oznak, ki so bile napačne po prvotnem označevanju in popravljene v ravno tako napačno oznako, zadnja vrstica pa opozarja na število oznak, ki jih je neko pravilo sicer popravilo, vendar so popravljene oznake identične izvornim. Vrednosti, prikazane v zadnjih dveh vrsticah, na natančnost označevanja ne vplivajo, je pa vseeno zaželeno, da so vrednosti v predzadnji vrstici čim manjše, saj so nove napake bolj kompleksne vrste kot pa izvorne napake. Absolutno natančnost dobimo tako, da odštejemo drugo vrstico od prve. Rezultat je podan v zadnji vrstici.

	Besedna vrsta	Oblikoslovna oznaka
Popravljeno	291	289
Pokvarjeno	4	4
Zamešano	14	16
Enako	2	2
Izboljšanje	287	285

Tabela 5: Rezultat pri avtomatskem popravljanju napak.

Napake v oblikoslovnih oznakah se z uporabo programa za popravke torej zmanjšajo za 287 pojavnic. S tem napako glede na besedne pojavnice zmanjšamo za 16% oz. dvignemo natančnost s 86,6% na 88,9%. Razlika ni ogromna, vendar pa maksimiziranje te mere natančnosti niti ni bil naš cilj, saj vsa pravila popravljajo oznako besedne vrste. Izboljšanje natančnosti za besedne vrste pa je dosti bolj opazno: z uporabo petih pretvorbenih pravil se ta izboljša za 38,4% oz. se z 94,4% poveča na 96,6% absolutne natančnosti.

5. Primerjave natančnosti označevanja

V Tabeli 6 podajamo pregled in primerjavo natančnosti označevanja po oblikoslovnih oznakah in besednih vrstah. Prva vrstica predstavlja rezultate iz raziskave Erjavec et al. (2000), v kateri je bil tako za učno kot za testno množico uporabljen korpus MULTEXT-East, tj. roman »1984«. Druga vrstica prikazuje glavno evalvacijo označevanja s programom TnT/totale na vzorcu iz korpusa SVEZ-IJS. Natančnost je sicer nižja, kar glede na precejšnje razlike med obema korpusoma ni presenetljivo, vendar ne v taki meri, kot bi mogoče pričakovali. Naslednja vrstica podaja rezultate, dobljene po izvajanju pretvorbenega programa, ki vsebuje 5 ročno napisanih pravil, dobljenih s pomočjo analize najbolj pogostih napak. Treba je opozoriti, da so tri pravila od petih (in to tista, ki pokrijejo največ primerov) popravljala oznake okrajšav oz. rimskih številk.

Zdi se, da so te napake in napake pri nerazporejenih besedah v večji meri problem tokenizacije kot pa samega oblikoslovnega označevanja. Robustna rešitev problema označevanja okrajšav in tujejezičnih citatov bi bila zato prej v izdelavi dodatnega modula, ki označevalniku na osnovi tipografskih značilnosti pojavnic dopolni leksikon za konkreten dokument. Zato je zanimivo pogledati še, kakšna bi bila natančnost označevanja, če bi pojavnice kategorij X in Y iz evalvacije izpustili. Natančnost se v tem primeru znatno poveča in doseže že 894% za oblikoslovne oznake oziroma 97,6% za besedne vrste.

V tabeli prikazujemo za primerjavo z rezultati našega sistema označevanja tudi natančnost najboljšega označevanja, predstavljenega v raziskavi Lönneker (2005), ki je glede na oblikoslovno oznako 83,6%, glede na besedno vrsto pa ni podana. Poskusa se v marsičem razlikujeta (v naboru oblikoslovnih oznak, velikosti učnega korpusa in sestavi testnega korpusa), zato je natančnosti težko neposredno primerjati, vseeno pa razlika v rezultatih preseneča, posebej glede na to, da je učni korpus pri poskusu ZRC/TreeTagger vseboval preko milijon besed in je bil torej desetkrat večji kot naš. Lönneker (2005) postavi nekaj hipotez, zakaj je natančnost pri njenih poskusih manjša, mdr. izpostavi bolj podrobne oznake, ki npr. ločijo različne vrste imen (osebna, krajevna, mitološka), in manjšo konsistentnost pri samem označevanju učnega in testnega korpusa. Zdi se nam verjetno, da je dodaten razlog tudi to, da je označevalnik TnT boljši kot TreeTagger, predvsem pri označevanju neznanih besed. Avtorica žal ne podaja posebej natančnosti za znane in neznane besede, tako da te hipoteze ne moremo preveriti.

	Oblikoslovna oznaka	Besedna vrsta
1984: TnT	89,2%	96,6%
SVEZ-IJS: TnT	86,6%	94,4%
SVEZ-IJS: TnT + Trans	**88,9%**	**96,6%**
SVEZ-IJS – X,Y: TnT	89,4%	97,6%
ZRC ISJ: TreeTagger	83,6%	?

Tabela 6: Ocene napak.

6. Zaključki

V članku smo analizirali natančnost avtomatskega oblikoslovnega označevanja z označevalnikom TnT, ki je vgrajen v program totale in izšolan na oblikoslovnih virih MULTEXT-East za slovenski jezik. Evalvacija je potekala na ročno popravljenem vzorcu iz slovenskega dela korpusa SVEZ-IJS, velikem približno 15.000 pojavnic, ki vsebuje okoli 15% sistemu neznanih besed. Evalvacija je pokazala, da je absolutna natančnost glede na besedne pojavnice v vzorcu 86,6% za polno označevanje oz. 94,4%, če opazujemo samo napake v besedni vrsti. V

primeru, da označevanje izboljšamo s pretvorbenim programom, ki odpravi nekaj najbolj pogostih in obenem enostavnih napak, se natančnost poveča na 88,9% za oznake oz. na 96,6% za besedne vrste.

Dodaten način, kako izboljšati natančnost, smo že omenili, in sicer s predprocesiranjem, ki bi identificiralo okrajšave in nerazporejene besede (tujejezične citate). Očiten korak k doseganju večje natančnosti bi bil tudi povečanje učnega korpusa in raznovrstnosti besedil v njem, pri čemer je glavna težava zamudnost (cena) takšnega označevanja. Žal pa bo to delo treba podvajati, saj veliki ročno označeni korpus ZRC (za razliko od npr. virov MULTEXT-East in korpusa SVEZ-IJS) izven matične institucije ni na voljo.

V literaturi najdemo tudi številne druge napotke, kako je možno označevanje izboljšati. Zanimiv pristop, in javno dostopen program, je opisan v raziskavi Brill (1992) in je bil tudi izhodišče za naš pretvorbeni program. Bistvena razlika pa je, da smo mi pravila napisali ročno, program, opisan v omenjeni raziskavi, pa se pretvorbenih pravil nauči na osnovi učne množice, pri čemer lahko sami definiramo lastnosti, glede na katere naj se program uči. Drugačen pristop, ki se je tudi že uporabljal za jezike z bogatim naborom oznak in majhno učno množico, je opisan v Tufiş (2006). Pristop predvideva redukcijo števila oblikoslovnih oznak, vendar tako, da je polne oznake še vedno možno rekonstruirati iz leksikona.

Zahvala

Avtorja se zahvaljujeta anonimnima recenzentoma za koristne pripombe. Raziskavo, opisano v pričujočem prispevku je podprl raziskovalni program ARRS »Tehnologije znanja« (prvi avtor), in štipendija CMEPIUS RS (drugi avtor).

Literatura

Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. V: Proceedings of the Third Conference on Applied Natural Language Processing, ACL. Trento, Italija.

Brants, T. (2000). TnT - A Statistical Part-of-Speech Tagger. V: Proceedings of the Sixth Applied Natural Language Processing Conference, ANLP-2000. Seattle, WA. 224--231.

Erjavec, T., Džeroski, S., Zavrel, J. (2000). Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. V: Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'2000). ELRA, Pariz.

Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. V: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'2004). ELRA, Pariz.

Erjavec, T., Džeroski, S. (2004). Machine Learning of Language Structure: Lemmatising Unknown Slovene Words. Applied Artificial Intelligence, 18, 17--41.

Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R. (2005). Massive multilingual corpus compilation: ACQUIS Communautaire and totale. V: Proceedings of the Second Language Technology Conference. April 2004, Poznan.

Erjavec, T. (2006). The English-Slovene ACQUIS corpus. V: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'2006). ELRA, Pariz.

Erbič, D., Krstič Sedej, A., Belc J., Zaviršek-Žorž, N., Gajšek, N., Željko, M. (2005). Slovenščina na spletu v dokumentih slovenske različice pravnega reda Evropske unije, terminološki zbirki in korpusu. V: Zbornik Simpozija Obdobja 24: Razvoj slovenskega strokovnega jezika. Ljubljana.

Hajič, J., Hladka, B. (1998). Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. COLING-ACL'98. ACL.

van Halteren, H. (ur.) (1999). Syntactic Wordclass Tagging. Kluwer.

Jakopin, P., Bizjak, A. (1997). O strojno podprtem oblikoslovnem označevanju slovenskega besedila. Slavistična Revija. 45/3-4. 513--532.

Lönneker, B. (2005). Strojno oblikoslovno označevanje slovenskih besedil: Kako daleč smo. Slavistična revija 53/2. 193--210.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. V: Proceedings of International Conference on New Methods in Language Processing. Manchester. 44--49.

Tufiş, D. (2006). Tagset Design for High Accuracy POS Tagging and Automatically Building Mapping between Arbitrary Tagsets. Workshop on Annotation Science: State of the Art in Enhancing Automatic Linguistic Annotation (LREC'06). ELRA, Paris.

SPIN: A Semantic Parser for Spoken Dialog Systems

Ralf Engel

DFKI GmbH
Stuhlsatzenhausweg 3, Saarbrücken, Germany
ralf.engel@dfki.de

Abstract

This paper presents SPIN, a semantic parser developed for spoken dialog systems. The parser provides a powerful rule language for an easy and efficient creation of the rule set. Important features of the rule language include order-independent matching, built-in support for referring expressions, rule ordering, constraints and action functions. On the basis of an example utterance the advantages of the introduced features are shown. The increased processing complexity caused by the powerful rule language is handled by a new parsing approach that delivers sufficient performance for rule sets that are typical for dialog systems. We also show how the parser can be used for text generation. The paper closes with an evaluation of the parser performance showing that the approach is well suited for dialog systems.

SPIN: Pomenski parser za sisteme govorjenega dialoga

V članku je predstavljen SPIN, semantični razčlenjevalnik, ki je bil razvit za sisteme govorjenega dialoga. Razčlenjevalnik ima zmogljiv jezik za tvorjenje pravil, ki enostavno in učinkovito tvori nabor pravil. Pomembne značilnosti jezika za tvorjenje pravil so ujemanje ne glede na besedni red, vgrajena podpora referenčnim izrazom, razvrstitev pravil, omejitve in opravilne funkcije. Na podlagi primera izjave so prikazane prednosti vpeljanih lastnosti. Povečana kompleksnost procesiranja zaradi zmogljivega jezika za tvorjenje pravil obvladujemo z novim pristop k skladenjski analizi, ki ima zadosten učinek pri naboru pravil, značilnih za sisteme dialoga. Prikažemo tudi, kako je lahko parser uporabljen za tvorjenje besedila. Članek zaključimo z vrednotenjem delovanja parserja, ki pokaže, da je pristop primeren za sisteme dialoga.

1. Introduction

This paper presents SPIN, a semantic parser which is especially designed for spoken dialog systems. The parser operates directly on typed feature structures whereby the available types and features are taken from the system-wide ontology. A syntactic analysis of the input utterance is not performed, but the ontology instances are created directly from word level. The typical advantages of such an approach are that processing is faster and more robust against speech recognition errors and disfluencies produced by the user, and the rules are easier to write and maintain. Also, multilingual dialog systems are easier to realize as a syntactic analysis is not required for each supported language. A disadvantage is that the complexity of the possible utterances is somewhat limited, but this is acceptable for most dialog systems.

Most semantic parsers use as underlying formalisms context free grammars (CFGs), e.g., (Gavaldà, 2000) or finite state transducers (FSTs), e.g., (Potamianos and Kuo, 2000) or variants of them, e.g., (Ward, 1991; Kaiser et al., 1999). The SPIN parser uses a more powerful rule language to simplify writing of rules and to reduce the amount of required rules.

Properties of the rule language include:

- Direct handling of nested typed feature structure is available, which is important for processing more complex utterances.
- Order-independent matching is supported, i.e., the order of matched input elements is not important. This feature helps processing of utterances in free word order languages, like German, Turkish, Japanese, Russian or Hindi, and simplifies writing of rules that are robust against speech recognition errors and disfluencies produced by the user.
- Built-in support for referring expressions is available.
- Regular expressions are available. Formulating the rules in a more elegant way is supported by this feature whereby the amount of required rules is reduced. Furthermore, writing of robust rules is simplified.
- Constraints over variables and action functions are supported providing enough flexibility for real-world dialog system. Especially, if the ontology is developed without the parsing module in mind, flexibility is highly demanded.

SPIN's powerful rule language requires an optimizing parser, otherwise processing times would not be acceptable. Principally, the power of the rule language avoids the development of a parser which delivers sufficient performance for an arbitrary rule set. Therefore, the parser is tuned for rule sets that are typical for dialog systems. A key feature to achieve fast processing is pruning of results that can be regarded as irrelevant for further processing within the dialog system.

Currently, the parser is used in the SMARTWEB project[1] (Wahlster, 2004). Earlier versions were successfully used in the MIAMM project (Reithinger et al., 2005) and the SmartKom project (Reithinger et al., 2003). SMARTWEB is a multimodal dialog system whose purpose is to provide a mobile and unified access to semantic databases, web services and internet search. The semantic databases include a database containing information about current and

[1] http://www.smartweb-project.org

previous football World Cups, the web services include, among others, information about POIs (point of interests), weather forecast, route planning and traffic information. Supported languages are German and English (with a reduced functionality). SMARTWEB is a joint project of several industrial and academic partners mainly located in Germany. A client-server architecture is used whereby the clients are ordinary smartphones or special onboard-units built into motor-bikes or cars. The clients are connected to the server via UMTS or WLAN. The multimodal recognizers, the dialog system, and the access subsystems are located on the server. All modules communicate using either XML messages based on the EMMA standard[2] or RDF messages based on the system-wide used ontology SWIntO (SmartWeb Integrated Ontology)[3] (Cimiano et al., 2004). SWIntO combines the DOLCE ontology (Gangemi et al., 2002) and the SUMO ontology (Niles and Pease, 2001) and contains also domain specific classes, properties and instances.

The paper is structured in the following way: Section 2 presents the rule language, section 3 contains a processing example, section 4 describes the parsing approach, and section 5 discusses how the parser can also be used for text generation. Section 6 reports on an evaluation of the parsing performance.

2. Rule language

2.1. Working memory

The rules operate on a working memory (WM) which consists of typed feature structures. The allowed types and features are extracted from the system-wide ontology, e.g., SWIntO in the SMARTWEB system, plus an additional type `Word` for representing words.

The top types are automatically extended with internal features which are defined in a reserved namespace. A list of the available internal features is shown in table 1.

The WM is initially filled with instances of the type `Word` representing the recognized words. The type `Word` has the predefined features `orth` (for the orthography of the word), `stem` and `pos` (part of speech). The features are filled by a lexicon lookup. If a feature is not provided in the lexicon, it remains unspecified.

2.2. Rule format

Like in classic rewriting systems, each rule consists at least of a set of conditions matching elements in the WM and a set of actions replacing the matched elements. Furthermore, constraints over the content bound to variables and processing options can be specified.

2.2.1. Conditional part

The conditional part consists of one or more conditions. Default mode is order-independent matching, i.e., the order within the WM and the features `leftMargin` and `rightMargin` are ignored. Order-independent matching simplifies the writing of rules for free-word order languages and the writing of rules that are robust against speech recognition errors and disfluencies produced by the user. Order-dependent matching can be activated by using square brackets. In this case, the values of `leftMargin` and `rightMargin` are considered.

Feature	Description
leftMargin	the index of the leftmost word
rightMargin	the index of the rightmost word
words	contains the input words used to create this instance
syn	contains syntactic information, like gender, number and case
scoreClass	contains the class used for scoring
score	contains the score (used in combination with scoreClass)

Table 1: List of internal features which are added automatically to each top type.

A single condition checks if an instance within the WM is of a certain type and if the specified features are also set in the tested instance. The type test considers the type hierarchy specified in the system-wide ontology. The tested type of the instance in the WM may be a subtype of the type in the condition, but also a supertype. The latter supports processing of referring expressions. For example, a pronoun can be mapped to a general type representing objects, like `PhysicalObject` in the SMARTWEB project. If the matched instance is inserted again in the WM, the type is refined to the type of the condition. The refined type can support reference resolution if several candidates are available in the dialog history.

Substructures within the conditions can be associated with variables. The variables can be reused in the constraints and in the action part. Within the conditions, disjunctions and negations are possible.

A test on an instance representing a word can be abbreviated with the orthography of the word. This test is replaced internally with a test on the stem. This avoids that individual rules must consider inflectional variants, a special advantage for languages with a rich usage of inflections like German.

An example for a condition that tests on the country with the name Brazil and assigns the name to the variable *N* is

```
Country(name:$N=Brazil)
```

2.2.2. Constraints

Constraints enable additional tests on the content bound to variables. Some built-in constraints are already available, but it is also possible to add user defined constraints.[4] Built-in constraints include a constraint that checks if the content is exactly of the specified type, ignoring the type hierar-

[2] `http://www.w3.org/TR/EMMA`

[3] `http://www.smartweb-project.org/ontology_en.html`

[4] User defined constraints have to be written as Java classes which have to be specified in the configuration options of the parser.

chy (`!isTypeOf`)[5], a constraint that checks if the words responsible for the content satisfy the specified syntactic property (`!syn`) and a constraint that checks if the content contains a specified substructure (`!contains`).

An example for a constraint that checks if the content bound to the variable *$V* contains an instance of the type `Country` is

```
!contains($V,Country())
```

2.2.3. Action part

The action part specifies the elements which replace the matched elements in the WM. Possible elements in the action part are typed feature structures, variables and action functions. Action functions allow to post-process the content bound to variables.

Available built-in action functions include a function that acts differently in cases where a specified variable is bound or not (`@if`)[6], a function that insert a specified syntactic property (`@syn`) and functions that provide string operations (`@concat`, `@toUpperCase`).

An example for an action inserting an instance of the type `Country` with the feature `name` set to the value of the variable *$N* in upper case is

```
Country(name:@toUpperCase($V))
```

2.2.4. Processing options

Processing options include an option that the test is not performed only on top level, but also within embedded instances (∼`deepMatch`), an option that a rule is always applied optional (∼`opt`), and the possibility to specify an ordering label. The ordering label can be used to force that a rule is applied before or after other rules. This allows, e.g., to write clean-up rules that are performed when parsing is finished.

3. Processing example

In this section, we will demonstrate how the SPIN parser can be used to process the utterance *Wie spielte diese Mannschaft gegen Brasilien? (How did this team play against Brazil?)*.[7]

The country *Brasilien (Brazil)* is handled by the following rule[8]:

```
(R1) Brasilien
     → Country(name:BRAZIL)
```

The queried database is language independent and uses English identifiers in uppercase. Therefore, the country name has to be set to *BRAZIL*.

In our domain, a country name can stand for a national football team stemming from that country. A rule performs this transformation:

```
(R2) ∼opt $C=Country()
     → FootballNationalTeam(origin:$C)
```

As the country instance is used also in its original meaning, the rule is marked as optional (∼`opt`). Otherwise, the parser optimizations may cause that the solution without the rule being applied is not produced.

The word *Mannschaft (team)* is simply mapped to an empty instance of the type `Team`.

```
(R3) Mannschaft → Team()
```

The next rule handles the determiner *dieser (this)*.

```
(R4) [ dieser $O=PhysicalObject() ]
     → $O(lingInfo:RefProp(type:def,
            gender:@syn($O,gender),
            number:@syn($O,number)))
```

In this case, the order of the matched elements is relevant, so order-dependent matching is activated, indicated by the square brackets. This rule exploits the hierarchy of the system-wide used ontology as all objects that can be referred to inherit from the type `PhysicalObject`. In the SMARTWEB system, the reference resolution module uses gender and number as a criterion to find a suitable referent. The action function `@syn` examines the words that have been used to create the instances bound to the specified variables and computes the specified features `gender` and `number`. The corresponding entry in the lexicon is

```
Mannschaft,syn:female-singular
```

Although not required for processing of the example utterance, we present a rule processing *sie* (*it* in this case) to show how pronouns are processed.

```
(R5) sie
     → PhysicalObject(lingInfo:RefProp(
           type:det, gender:female,
           number:singular))
```

Questioned instances are marked in the SMARTWEB query language with a variable that contains the requested media type; it is also possible to asked explicitly for images or videos. The rule processing *welches (which)* is

```
(R6) [ welches $PO=PhysicalObject() ]
     → $PO(var:Variable(
           focus:Text()))
```

A corresponding rule for *wann (when)* is

```
(R7) wann
     → TimePoint(var:
         Variable(focus:Text()))
```

The verb phrase *wie spielte <Team1> gegen <Team2> (how did <Team1> play against <Team2>)* is handled by the rule

```
(R8) %wie spielte $T1=Team()
     %[ gegen $T2=Team() ]
     %[ %bei $T=Tournament() ]
     %$TP=TimePoint()
     %[ %in $R=TournamentRoundStage() ]
     → Match(team:T1, team:T2,
           tournament:$T,
           inRound:$R,
           @if($TP,happensAt:
              TimeInterval(begins:$TP)))
```

[5] All constraints are prefixed with `!`.

[6] Action functions are prefixed with `@`.

[7] As processing of free-word order phenomena should be shown, the example utterances are in German.

[8] The expression `(RX)` is not part of the rule and is only used for referring purposes.

This rule is able to integrate further information, like a specified tournament, a time point or a round stage like final. The additional information is matched by optional conditions, indicated by the prefix `%`.

Besides our example utterance, this single rule (together with other preprocessing rules for tournaments, rounds, etc.) can process a lot of other utterances including

Wie spielte Brasilien im Finale 1990?
(How did Brazil play in the 1990 final?)

Wie spielte Brasilien bei der WM in Spanien?
(How did Brazil play at the World Cup in Spain?)

Gegen welche Mannschaften spielte Brasilien bei der WM 1974?
(Which teams did Brazil play against at the World Cup 1974?)

Wann spielte Brasilien gegen Frankreich?
(When did Brazil play versus France?)

Covering such a variety of utterances with a single rule is only possible because the rule language supports optional conditions and mixing of order-dependent and order-independent matching.

In addition, order-independent matching makes processing more robust against speech recognition errors, as misrecognized words can be simply skipped in many cases. If the recognition errors affect only words which are not essential for the understanding of the utterance, the utterance can be analyzed at least partially. Examples are:

spielte diese Mannschaft gegen Frankreich?
(did this team play against France?)
(*Wie (How)* was not recognized)

Wie spielte Brasilien Frankfurt 1990?
(*How did Brazil play Frankfurt 1990?*)
(*im Finale (in the final)* was misrecognized as *Frankfurt*)

As the query module for the semantic database of the World Cup data expects that at least one instance is marked as questioned, a cleanup rule checks whether an embedded instance is marked as questioned and adds the dialog act *Question*. If this is not the case the matched instance itself is marked as questioned (second rule):

```
(R9) cleanup1:  $M=Match()
     !contains($M,Variable())
     → Question(content:$M())

(R10) cleanup2:  $M=Match()
      → Question(content:$M(var:
            Variable(focus:Text())))
```

In the configuration options of the parser, it is specified that rules marked with the ordering label `cleanup1` are applied before rules marked with the ordering label `cleanup2`.

A rule that handles general utterances like *bitte (please)* is

```
(R11) cleanup3:  bitte $D=DialogAct()
      → $D(mode:polite)
```

Due to order-independent matching this works also if *bitte* is placed in the middle of the utterance, e.g.,

wie spielte bitte diese Mannschaft gegen Brasilien
(Please, how did this team play against Brazil?)

The generated result structure for the utterance *wie spielte diese Mannschaft gegen Brasilien* is finally

```
Question(content:Match(
     var:Variable(focus:Text()),
     team:Team(lingInfo:RefProp(
        gender:female, number:singular))
     team:Team(origin:Country(
        name:BRAZIL))))
```

4. Parsing algorithm

In this section, the main ideas of the parsing algorithm are presented, a more detailed description can be found in (Engel, 2005).

4.1. Parsing challenge

Fast CFG parsing approaches like *Earley parsing* or *Tomita's parsing approach* cannot be used because SPIN's rule language allows order-independent matching. (Huynh, 1983) has shown that parsing of rule languages that support order-independent matching is NP-complete.

Typically, parsing algorithms are optimized to avoid the generation of multiple identical (intermediate) results and intermediate results that cannot be further processed. The first issue can be addressed using a chart, the second one using top-down predictions.

But the main problem in parsing rule languages which support order-independent matching is that most of the generated WMs are irrelevant for further processing in other modules within the dialog system as they contain unprocessed elements. The basic idea of the presented approach is to avoid the generation of as many irrelevant results as possible. Two starting points have been discovered:

(1) For many rules it is not appropriate to be applied before some other rules, as in this case irrelevant results are generated. An example is the application of rule (R8) before rule (R4) is applied. The problem is that *dieses (this)* is not integrated and, even worse, the word cannot be integrated later on, as the instance `FootballNationalTeam` is embedded after the application of rule (R8) and therefore unreachable for rule (R4). To overcome this problem, the idea is to order the rules offline, so that rule (R4) is applied before rule (R8).

(2) In many cases, the original WM can be deleted after the application of a rule. In a standard bottom-up parser, the result of a rule application is always added to the already existing set of alternative WMs. This is necessary as otherwise relevant results may not be generated. But if alternative rules do not exist, maintaining the original WM is not necessary. So the idea is to detect offline which rules can match the same input and to maintain only the original WM in these cases. All other rules are marked as destructive, i.e., the original WM is deleted after the application of that rule. As the presented example rules are not ambiguous, all rules are marked as non-destructive with the exception of rule (R2) which is marked explicitly as optional.

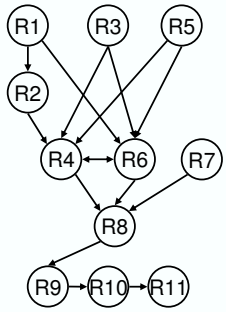


Figure 1: The generated dependency graph for the rules (R1) to (R11). One cycle exists containing rules (R4) and (R5). Transitive transitions are omitted.

4.2. Realization

First, the rules are ordered using a dependency graph. Therefore, each rule is compared with all other rules. If rule A creates instances that can be processed by rule B, a dependency transition is inserted between rule A and rule B. After all rules have been processed, the rules are linearized by walking through the graph and assigning each rule an application number.

Before the dependency graph is linearized, the graph is checked for cycles. A cycle means that a rule A can process the result generated by a rule B, but rule B can also process the result generated by rule A. If a cycle is detected, all rules of that cycle get the same application number. Rules with the same number are applied in a loop until none of the rules can be applied anymore.

After the rules are ordered, each rule is examined if it is in "competition" with at least one other rule applied afterwards. If this is the case, the rule is marked as non-destructive, i.e., the original WM is kept in the set of alternative WMs, otherwise the rule is marked as destructive, i.e., the original WM is deleted from the set of alternative WMs.

If a rule A is in competition with a rule B depends on the following: The application number of rule A has to be greater or equal than the application number of rule B, and a WM must exist so that rule A and rule B can be applied to that WM, and at least one element of the WM is matched by both rules. The algorithm to detect if two rules can match partially the same input is quite complex and is not described in this paper.

Figure 1 shows the constructed dependency graph for the rules used in section 3.

5. Text generation

When the development of the parser was started, using the parser as part of a text generation module was not intended. But the parser has proven as flexible enough to support this task. An early version of the text generation module called NipsGen[9] is already used in the SMARTWEB project. The SPIN parser is used in combination with a TAG (tree adjoining grammar) module (Becker, 2006). The TAG grammar of this module is derived from the XTAG grammar for English developed at the University of Pennsylvania[10].

The input of the generation module is the result of a query and is represented as an instance of SWIntO. The input is transformed to a text string in three steps:

1. A derivation tree for the TAG-grammar is created using SPIN rules which are applied on the semantic input structure.

2. The actual syntax tree is constructed using the derivation tree. After the tree has been built up, the features of the tree nodes are unified.

3. The correct inflections for all lexical leafs are looked up in a lexicon. Traversing the lexical leafs from left to right generates the text string.

The focus of the further description is put on the first step, the creation of the derivation tree.

A direct generation of the TAG tree description would lead to too complicated and unintuitive rules. Instead, the generation process is split into two phases. First, an intermediate representation is built up on a phrase level. This phase is domain dependent. In a second step, the intermediate description is transformed to a derivation tree. The intermediate layer is domain independent and therefore the transformation rules for the second part are also domain independent.

The generation of a text string is illustrated by the input structure

```
VP(o:Match(
   team1:FootballNationalTeam(origin:
      Country(name:GERMANY))
   team2:  FootballNationalTeam(origin:
      Country(name:BRAZIL))
   result:  "1:0" ))
```

which should be verbalized as

Deutschland spielte gegen Brasilien 1:0
(Germany played against Brazil 1:0)

Two examplary rules of the first phase are presented. The first rule produces the verb phrase (VP) with *spielen (play)*, the second one verbalizes the teams.

```
$VP=VP(o:Match(
      team1:$T1,team2:$T2,
      result:$R,not(lex:))
→ $VP(lex:spielen,
      sub:NP(o:$T1),
      pp:PP(lex:gegen,np:NP(o:$T2)),
      adv:AdvP(lex:$R))

$NP=NP(o:FootballNationalTeam(origin:
      Country(name:Brazil)),not(lex:))
→ $NP(lex:Brasilien)
```

In the second phase, the phrase structure is converted to a derivation tree for the TAG grammar. Each tree in the TAG grammar has a corresponding type in the ontology.

[9] Nips in the name NipsGen stands for the reverse usage of SPIN parser, Gen stands for generation.

[10] http://www.cis.upenn.edu/∼xtag/

The features of a TAG tree type represent the type of operation (adjunction (a), substitution (s), lexical replacement (l)) and the position in the tree, e.g., 211.

An example for a rule transforming a verb phrase to the intermediate representation to the TAG tree anCnx0VADJ is

```
VP(lex:$L,sub:$S,adv:$A,%pp:$PP,%fvp:$F)
→ anCnx0V(
      l_211:$L,
      s_1:$S(fvp:Fvp(case:nom)),
      a_221:$A,
      a_222:@if($PP,$PP(mode:vpAdj)),
      fvp:$F)
```

For text generation, the parser is driven in a slightly different mode: The automatic ordering of rules is switched off, instead the order in which the rules are applied is taken from the file containg the rules. Regions that have to be applied in a loop and rules that have to be applied optionally are marked explicitly. In the current system, two loops exists, one for each phase. In cases where multiple solutions should be produced, the alternative rules have to be marked as optional.[11]

Currently, the generation module contains 179 rules for the first phase and 38 rules for the second phase.

6. Evaluation of parsing performance

The rule set used for the SMARTWEB project consists of 1069 rules where 363 rules are created manually, and 706 are generated automatically from the linguistic information stored in SWIntO, e.g., country names. The lexicon contains 2250 entries.

In the offline rule ordering 12 loops are generated with an average size of 4.2 rules, the largest loop contains 20 rules. 242 rules are marked as non-destructive.

The parser is written in Java 1.5. We tested the performance on a Pentium IV 3.2GHz computer with a test corpus of 175 utterances with an average length of 6.5 words and a maximal length of 13 words. The average processing time was 45.9 ms, the largest one 183.4 ms.

7. Conclusion and outlook

In this paper we presented SPIN, a semantic parser providing a powerful rule language. After a short description of the rule language, a processing example was provided showing some of the advantages of the powerful rule language. A parsing approach which provides fast processing with rule sets that are typical for dialog systems was outlined, and the inclusion of the SPIN parser in the text generation module was presented. The evaluation of the parsing performance shows that the parser provides sufficient performance for real-world dialog systems.

The current research focus is on the development of tools for efficient rule writing and maintaining, and on further optimizations of the parser, like pruning of irrelevant results caused by optional conditions.

[11] A separate component selects one of the generated solutions.

8. Acknowledgments

This research was funded by the German Federal Ministry for Education and Research under grant number 01IMD01A. The views expressed are the responsibility of the authors. Points of view or opinions do not, therefore, necessarily represent official Ministry for Education and Research position or policy.

9. References

Becker, Tilman, 2006. Natural language generation with fully specified templates. In Wolfgang Wahlster (ed.), *SmartKom: Foundations of Multi-modal Dialogue Systems*. Heidelberg: Springer, pages 401–410.

Cimiano, Philipp, Andreas Eberhart, Pascal Hitzler, Daniel Oberle, Steffen Staab, and Rudi Studer, 2004. The smartweb foundational ontology. Technical report, Institute for Applied Informatics and Formal Description Methods (AIFB) University of Karlsruhe, Karlsruhe, Germany. SmartWeb Project.

Engel, Ralf, 2005. Robust and efficient semantic parsing of free word order languages in spoken dialogue systems. In *Proc. of Interspeech-2005*. Lisboa.

Gangemi, Aldo, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schcneider, 2002. Sweetening Ontologies with DOLCE. In *Proc. of EKAW02*, volume 2473 of Lecture Notes in Computer Science. Sigünza, Spain.

Gavaldà, Marsal, 2000. SOUP: A parser for real-world spontaneous speech. In *Proc. of 6th IWPT*. Trento, Italy.

Huynh, Dung T., 1983. Communicative grammars: The complexity of uniform word problems. *Information and Control*, 57(1):21–39.

Kaiser, Edward C., Michael Johnston, and Peter A. Heeman, 1999. PROFER: Predictive, robust finite-state parsing for spoken language. In *Proc. of ICASSP-99*, volume 2. Phoenix, Arizona.

Niles, Ian and Adam Pease, 2001. Towards a Standard Upper Ontology. In Chris Welty and Barry Smith (eds.), *Proc. of FOIS-2001*. Ogunquit, Maine.

Potamianos, Alexandros and Hong-Kwang Kuo, 2000. Statistical recursive finite state machine parsing for speech understanding. In *Proc. of 6th ICSLP*. Beijing, China.

Reithinger, Norbert, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löckelt, Jochen Müller, Norbert Pfleger, Peter Poller, Michael Streit, and Valentin Tschernomas, 2003. SmartKom - adaptive and flexible multimodal access to multiple applications. In *Proc. of ICMI 2003*. Vancouver, B.C.

Reithinger, Norbert, Dirk Fedeler, Ashwani Kumar, Christoph Lauer, Elsa Pecourt, and Laurent Romary, 2005. Miamm: A multi-modal dialogue system using haptics. In L. Dybkjaer and J. van Kuppevelt (eds.), *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*. Kluwer.

Wahlster, Wolfgang, 2004. SmartWeb: Mobile Applications of the Semantic Web. In Peter Dadam and Manfred Reichert (eds.), *GI Jahrestagung 2004*. Springer.

Ward, Wayne, 1991. Understanding spontaneous speech: the Phoenix system. In *Proc. of ICASSP-91*.

Increasing the coverage of answer extraction by applying anaphora resolution

Jori Mur

Humanities Computing
Faculty of Arts
University of Groningen
Oude Kijk in 't Jatstraat 26, 9712 EK Groningen
j.mur@rug.nl

Abstract

Off-line answer extraction using patterns is a technique for corpus based Question Answering (QA) that has proven to be very effective. The results typically show a high precision score. However, the main problem with this technique is the lack of coverage of the extracted answers. One way to increase the coverage is to apply anaphora resolution. Using anaphora resolution we can turn information-poor extractions (e.g. she is the queen of Holland) into potential answers (e.g. Beatrix is the queen of Holland). In this paper we show that by applying a simple anaphora resolution technique the number of facts extracted from a Dutch newspaper corpus increased with more than 50% and although precision went down the increase in recall had a positive effect on the performance of a state-of-the art QA system.

Povečanje pokritja luščenja odgovorov z uporabo razrešitve anafore

Off-line luščenje odgovorov z uporabo vzorcev je metoda na korpusu temelječega sistema za odgovarjanje na vprašanja, ki se je izkazala za zelo učinkovito. Rezultati običajno dosegajo visoko točnost. Vendar pa je glavni problem metode pomanjkljivo pokritje izluščenih odgovorov. Ena od možnosti za povečanje obsega je razrešitev anafore. Z uporabo razrešitve anafore lahko pretvorimo slabo informativne izluščene izraze (npr. ona je nizozemska kraljica) v možne odgovore (npr. Beatrix je nizozemska kraljica). V članku prikažemo, da se z uporabo preproste tehnike razrešitve anafore število izluščenih dejstev iz nizozemskega časopisnega korpusa poveča za več kot 50%. Čeprav se točnost zmanjša, pa ima povečanje pokritja pozitivne učinke na delovanje sodobnega sistema za odgovarjanje na vprašanja.

1. Introduction

There is a need for tools which help users to find the information they are looking for. Search engines such as Google overcome this need by presenting on a user's query a ranked list of links to relevant documents. However, sometimes a user simply has a question and what he wants is an answer. He needs a system which analyses the relevant documents for him and which only returns the answer, rather than a list of documents, thus saving him a lot of time. Question Answering (QA) systems respond to this need. The task of a question answering system is to retrieve answers to questions posed in natural language, given a collection of documents.

A typical approach to question answering is the following: a question is analysed by a question classifier module that assigns a certain type to the question. Example types are 'function' (Who is the president of the United States?) or 'location of birth' (Where was Vincent van Gogh born?). Often this module also determines the expected answer type. For example, 'person name' or 'location name'. Keywords are selected from the question and they are fed into a document retrieval module. This module identifies relevant articles or paragraphs that are likely to contain the answer. Then terms of the same type as the type of the expected answer are extracted. The system uses further clues to rank the candidate answers. The answer ranked first is returned to the user.

A second approach, which some QA research teams have added as a module to their basic system is the technique of off-line answer extraction. Off-line methods have proven to be very effective in QA (Fleischman et al., 2003; Jijkoun et al., 2004; Bouma et al., 2005a). Before actual questions are known, a corpus is exhaustively searched for potential answers to questions of a specific question type (capital, abbreviation, inhabitants, year of birth, ...). Highly precise relations are extracted from the corpus off-line and stored as an answer repository for quick and easy access. This method typically results in a very high precision score.

However, one major drawback is the lack of coverage of the extracted answers. If the module finds an answer it is usually the correct one, but the module finds too few answers in general. Jijkoun et al. (2004) have shown for English that extraction patterns defined in terms of dependency relations can lead to significant improvements in recall over systems based on regular expression pattern matching. Yet, lack of coverage remains an issue (Tjong Kim Sang, 2005).

The aim of this paper is to address the low coverage problem of answer extraction by incorporating an anaphora resolution module into an answer extraction system. The task of anaphora resolution is to determine for a reference in the text the entity to which it refers, the antecedent. Patterns are often defined in such a way, that answers are only extracted when they appear within the matching sentence, whereas applying anaphora resolution will result in the extraction of information that is referenced anaphorically as well. The following example illustrates this:

(1) **Question:** Who is the Queen of Holland?
Text: Beatrix was invited to speak before the European Parliament. The Queen of Holland emphasised in her speech the equality of everyone

who lives in Europe.
Answer: Beatrix

If we know that *The queen of Holland* in the second sentence refers back to *Beatrix* in the first sentence, we can answer the question.

For our experiments we used a Dutch answer extraction module, which is part of our Dutch QA system. We developed an anaphora resolution system based on this module. The results indicate that anaphora resolution can be used effectively to increase the coverage of off-line answer extraction. Many more facts were extracted and in spite of a low precision score for the anaphora resolution system and the limited number of question types to which it was applied, the performance of the QA system overall increased as well.

The remainder of this paper has been organised in the following way. In the next section we provide details about the extraction methods. The anaphora resolution system is described in section 3. Section 4 contains a description of our experiments and section 5 shows the results. Finally, we discuss research of others related to our work in section 6 and we conclude in section 7.

2. Off-line answer extraction

In this section we describe the extraction method we used to create fact bases.

Since we want to define patterns in terms of dependency relations, we parsed the whole corpus. For this task we used the Alpino parser, a wide-coverage dependency parser for Dutch (Malouf and van Noord, 2004). Malouf and van Noord (2004) show that the accuracy of the system, when evaluated on a test-set of 500 newspaper sentences, is over 88%, which is in line with state-of-the-art systems for English. The Alpino system also incorporates a Named-Entity classifier to recognise and classify proper names.

The dependency analysis of a sentence gives rise to a set of dependency relations of the form ⟨`Head, Rel, Dep`⟩, where `Head` is the root form of the head of the relation, and `Dep` is the head of the constituent that is the dependent. `Rel` is the name of the dependency relation. For instance, the dependency analysis of sentence (2-a) is (2-b).

(2) a. Amsterdam telt 800.000 inwoners (*Amsterdam counts 800,000 inhabitants*)

b. $\left\{ \begin{array}{l} \langle \texttt{tel, subj, amsterdam} \rangle, \\ \langle \texttt{tel, obj, inwoners} \rangle, \\ \langle \texttt{inwoners, det, 800.000} \rangle \end{array} \right\}$

The module also accounts for syntactic variation. For instance, the subject of an active sentence may be expressed as a PP-modifier headed by *door* (*by*) in the passive. In (Bouma et al., 2005b) this module is described in more detail.

A dependency pattern is a set of (partially underspecified) dependency relations as in (3).

(3) $\left\{ \begin{array}{l} \langle \texttt{tel, subj, } \textbf{<LOCATION>} \rangle, \\ \langle \texttt{tel, obj, inwoners} \rangle, \\ \langle \texttt{inwoners, det, } \textbf{<NUMBER>} \rangle \end{array} \right\}$

To get a clear picture of the impact of anaphora resolution on the off-line construction of knowledge bases, we selected 12 question types that we expect to benefit from anaphora resolution.[1]

They are shown in table 1. We extracted for each type the number of facts listed in the second column of table 1 using the basic patterns, i.e. without applying anaphora resolution.

For our experiments we adjusted the patterns. We replaced the slot for the named entity with a slot for a pronoun. For instance, the pattern from the example above is changed into the following:

(4) $\left\{ \begin{array}{l} \langle \texttt{tel, subj, } \textbf{<Pronoun>} \rangle, \\ \langle \texttt{tel, obj, inwoners} \rangle, \\ \langle \texttt{inwoners, det, } \textbf{<NUMBER>} \rangle \end{array} \right\}$

It will match the parse of a sentence such as (5):

(5) Het telt slechts 1.000 inwoners.
(*It counts only 1,000 inhabitants*)

Similarly, we adjusted the patterns to match sentences with a definite noun. We considered noun phrases preceded by a definite determiner as definite noun phrases.

In the next section we describe the anaphora resolution technique we used to resolve the pronouns and definite NPs that we found with the patterns.

For the question types *capital* and *function* the anaphor could be part of the fact we want to extract. An example for the function type is given in example (1) on page 1. We extract both the antecedent *Beatrix* and the anaphor *The queen of Holland*.

For the other types and for the pronoun patterns we extract the antecedent and the answer terms, but not the anaphor. For example, if we should encounter the following text: *Beatrix was Juliana's first child. The Queen of Holland was born in 1938.* then we extract the antecedent *Beatrix* and the answer term *1938*.

3. Anaphora resolution

The pronouns and definite NPs we found during the pattern matching process have to be linked to the correct antecedent in order to extract complete and meaningful facts for the tables. We developed for both types of anaphoric NPs separate but similar techniques as described in the following two sections.

3.1. Resolving definite NPs

Our strategy for resolving definite NPs is based on knowledge about the categories of named entities, so-called instances (or categorised named entities). Examples are *Van Gogh* IS-A *painter*, *Seles* IS-A *tennis player*. Since the whole corpus was parsed we were able to acquire instances by scanning the corpus for apposition relations and predicate complement relations[2]. The apposition relation holds between *Van Gogh* and *painter* in:

[1] In total we defined 22 question types. The remaining types are: abbreviation, currency, date, firstname, location, measure, result, definition, which, what

[2] We limited our search to the predicate complement relation between named entities and a noun and excluded examples with

Question Type	# of facts	Clarification
Age	21669	Who is how old
Location of Birth	776	Who was born where
Date of Birth	2358	Who was born when
Capital	2220	Which city is the capital of which country
Age of Death	1160	Who died at what age
Date of Death	1002	Who died when
Cause of Death	3204	Who died how
Location of Death	585	Who died where
Founder	741	Who founded what when
Function	58625	Who full fills what function in life
Inhabitants	823	Which location contains how many inhabitants
Winner	334	Who won which Nobel prize when

Table 1: Question types for which we defined patterns together with the number of facts we extracted for each type

(6) Van Gogh, the famous Dutch painter.

We only consider the head of the definite NP for the instance list. And here is an example of a predicate complement relation:

(7) Van Gogh is a famous Dutch painter.

We extracted around 1 million appositions (tokens) and 0.3 million predicate complements (tokens) resulting in 1 million types overall.

Our strategy is as follows: We scan the left context of the definite NP for named entities from right to left (i.e. the closest named entity is selected first). For each named entity we encounter, we check whether it occurs together with the head of the definite NP as a pair on the instance list. If so, the named entity is selected as the antecedent of the NP. As long as no suitable named entity is found we select the next named entity and so on until we reach the beginning of the document. In a previous study (Mur and van der Plas, to appear) it was shown that this strategy leads to high precision, but low recall. So we decided to implement a fall back mechanism: if no suitable named entity is found, i.e., no named entity is found that forms an instance pair with the head of the definite NP, we select simply the first preceding named entity.

In order to explain our strategy for resolving definite NPs we will apply it to the next example:

He was the opponent of the quiet Ivanisevic in December 1995. Todd Martin who defeated the local hero Boris Becker a day earlier, was beaten by the 26-year old Croatian during the finals of the Grand Slam Cup in 1995 [...].

In the example above, the left context of the NP *the 26-year old Croatian* is scanned from right to left. The named entities *Boris Becker* and *Todd Martin* are each selected before the correct antecedent *Ivanisevic*. Neither *Boris Becker* nor *Todd Martin* is found in an instance relation with *Croatian*, so they are put aside as unsuitable candidates. Then *Ivanisevic* is selected and this candidate *is* found to be on the instance list with *Croatian*, so *Ivanisevic* is taken as the antecedent of *Croatian*. The fact `Ivanisevic, 26-year old` is added to the Age table.

3.2. Resolving Pronouns

We applied a similar technique for resolving pronouns. The pronouns we tried to resolve were the nominative forms of the singular pronouns *hij* (he), *zij/ze* (she), *het* (it) and the plural pronoun *zij/ze* (they). We chose to resolve only the nominative case, as in almost all patterns the slot for the name was the slot in subject position. The number of both the anaphor and the antecedent was determined by the number of the main verb.

Since we find the anaphors by matching patterns, we knew the named entity (NE) tag of the antecedent. For example, if we match a pattern defined for the location-of-birth type, we are looking for a person, if we match a pattern defined for the capital type, we are looking for a location and so on.

Again we scan the left context of the anaphor (now a pronoun) for named entities from right to left. We implemented a preference for proper nouns in the subject position. They were analysed before the other proper nouns in the same sentence. For each named entity we encounter, we check whether it has the correct NE-tag and whether its number corresponds to the number of the pronoun. If so and if it concerns a non-person NE-tag, the named entity is selected as the antecedent. If we are looking for a person name as the antecedent, we have to do another check to see if the gender of the name corresponds to the gender of the pronoun. To determine the gender of the selected name we created a list of boy's names and girl's names by downloading such lists from the Internet[3]. The female list contained 12,691 names and the male list 11,854 names. To be accepted as the correct antecedent, the proper name should not occur on the name list of the opposite sex of the pronoun.

After having resolved the anaphor, the fact was added to the appropriate table.

negation

[3] http://www.namen.info, http://www.voornamenboek.nl, http://www.babynames.com and http://prenoms.free.fr

baseline	pronouns	definite nouns	total anaphora
93,497 (86%)	+3,915 (40%)	+47,794 (33%)	+51,644 (34%)

Table 2: Number of added fact tokens (precision)

4. Experiment

The aim of the experiment is firstly to determine whether anaphora resolution on definite NPs and pronouns helps to acquire more facts and secondly to investigate if it improves the performance of a state-of-the-art QA system. To this end, we first create tables using the patterns based on anaphora resolution techniques and we compare these tables to the tables created by using the baseline patterns (i.e. the patterns that extract facts in a straightforward way). For both extraction modules we randomly selected a sample of around 200 extracted facts and we manually evaluated these facts on the following criteria:

- correctness of the fact;
- and in the case of anaphora resolution, correctness of the selected antecedent.

Secondly, we evaluated both extraction modules as part of a QA system. We measured the performance by counting how many questions were answered correctly.

4.1. Corpus and parser

We apply our answer extracting techniques to the Dutch CLEF corpus. This corpus is used in the annually organised CLEF evaluation track for Dutch question answering systems. It consists of newspaper articles from 1994 and 1995, taken from the Dutch daily newspapers *Algemeen Dagblad* and *NRC Handelsblad*. The corpus contains about 78 million words. The whole collection was parsed automatically using the Alpino parser described in section 2.

4.2. QA system and questions

For the experiments we used an open-domain corpus-based Dutch QA system, Joost (Bouma et al., 2005a). It achieved a score of 49,5% on the question answering track of CLEF-2005, the best result for the Dutch track. The system implements an IR-based approach as well as a table-lookup strategy. An incoming question is analysed and assigned to one of the predefined question types. If the type is one of the twelve types listed in table 1, the table lookup mechanism identifies knowledge bases where answers to the question can potentially be found. It uses keywords from the question to identify relevant entries in the selected knowledge bases and extracts candidate answers. Finally, the QA system re-ranks and sanity checks the candidates and selects the final answer. If the question was of a different type than the twelve listed above, an answer is found by the IR-based technique. We performed our experiments with the 200 Dutch questions of the CLEF-2005 data set.

5. Results

The numbers of extracted facts for each method are given in table 2. Between brackets you see the precision score of the extracted facts. 65 facts were extracted by applying both pronoun resolution and definite noun resolution. These facts are typical founders facts: *He founded the organisation*, where both *He* and *organisation* are anaphoric. They are included in the number of facts found by the pronoun patterns as well as by the number of facts found by the definite noun patterns.

The number of facts we extracted by the pronoun patterns is quite low. We did a corpus investigation on a subset of the corpus which consisted of sentences containing terms relevant to the 12 selected question types[4]. In only 10% of the sentences one or more pronouns appeared. This result indicates that is it not surprising that we only extracted 4% more compared to the baseline.

The precision of the new facts was a bit disappointing. In (Mur and van der Plas, to appear) we reported a very high precision for the facts added by applying definite noun resolution. There was a difference in precision between the original and the expanded tables of only 1%. The difference with the method used in the current experiment lies in the fact that we did not use a fall back method in the previous study. We resolved the anaphor if and only if an instance relation was found between the anaphoric NP and the candidate antecedent. There is also a difference in evaluation. In (Mur and van der Plas, to appear) we evaluated the tables containing all the facts together. In this experiment we evaluated them separately.

Nevertheless, if we assume this estimation of the precision to be correct we have added 17,559 valid facts to the original tables.

Since we were interested in the increase of coverage we also calculated the number of additional fact *types* we found with the new patterns, listed in table 3. If we had only used the pronoun patterns we would have found 3,627 new facts. On the other hand, if we had only used the definite noun patterns we would have found 35,687 new facts. Using both we extracted 39,208 additional facts.

baseline	pronouns	definite nouns	both anaphora
64,627	+3,627	+35,687	+39,208

Table 3: Number of added fact types

We also wanted to know what the effect of the extended tables would be on the performance of a state-of-the-art QA system. The results are shown in row (1) of table 4. Clearly, the low precision score of the added facts did not hurt performance. We believe that this effect is due to frequency counts. Incorrect answers are typically outnumbered by correct ones. In total two more questions are answered correctly. In fact, three more questions were answered cor-

[4]terms such as "geboren" (*born*), "stierf" (*died*), "hoofdstad" (*capital*) etc.

rectly, but for another question now an incorrect answer was found. This was due to an incorrectly chosen antecedent.

Further investigation showed that only 40 questions were assigned one of the twelve question types selected for anaphora resolution. Improvement was therefore only possible for those questions. Looking at this subset of questions we see an increase of 5% in performance.

	baseline	anaphora patterns
(1) Total	103/200 (51.5%)	105/200 (52.5%)
(2) 12 types	26/40 (65.0%)	28/40 (70.0%)

Table 4: Number of questions answered correctly

However, besides the low precision score for the anaphora resolution mechanism and the limited number of questions that fell into one of the twelve selected question types there was another issue that possibly caused the small improvement for QA. Question 107 in the question set was as follows: `Wie was piloot van de missie die de astronomische satelliet, de Hubble Space Telescope, repareerde?` (*Who was the pilot of the mission that repaired the astronomic satellite, the Hubble Space Telescope?*). The answer we found was extracted from a sentence in the Algemeen Dagblad of September 19th, 1994 which was formulated as follows: `Bowersox was piloot van de missie die de astronomische satelliet, de Hubble Space Telescope, repareerde.` (*Bowersox was the pilot of the mission that repaired the astronomic satellite, the Hubble Space Telescope.*).

In (Magnini et al., 2004) the authors claim that they created the questions independently from the document collection, thus avoiding any influence in the contents and in the formulation of the queries. However, the example above suggests otherwise. If questions are re-formulations of sentences in the newspaper corpus such as question 107, then it is not surprising that anaphora resolution has little effect.

6. Related work

In last QA tracks of TREC and CLEF (2005) 30 and 24 systems were evaluated respectively. After reviewing these systems, we can notice that only few systems model some co-reference relations between entities in the documents (Schone et al., 2005; Jiangping et al., 2005; Hartrumpf, 2005; Neumann and Sacaleanu, 2005; Laurent et al., 2005).

Schone et al. (2005) apply a symbolic method which tries to resolve pronouns and draw associations between definite NPs. This has a small positive effect on the performance of their QA system. Hartrumpf (2005)'s error analysis of his results for the QA track of CLEF 2004 indicated that the lack of co-reference resolution was a major source of errors. Therefore he incorporated a co-reference resolution system. This system, called CORUDIS, combines syntactico-semantic rules with statistics derived from an annotated corpus. Its results show an F-score of 66% for handling coreference relations between all kinds of NPs (e.g. pronouns, common nouns and proper nouns). The improvements for the QA system obtained by incorporating CORUDIS were unfortunately not significant due to the limited recall value of the co-reference resolution system. In other cases benefits of applying these reference techniques have not been analysed and measured separately.

There are also some systems from earlier years that have evaluated the contribution of reference resolution to the performance of their QA systems. (Watson et al., 2003; Stuckardt, 2003; Mollá et al., 2003). The earliest approaches that evaluate the contribution of reference resolution to QA are by Morton (2000) and Vicedo and Ferrández (2000a).

Morton (2000)'s approach models identity, definite NPs and non-possessive third person pronouns. For pronoun resolution and common noun resolution, he uses a set of features and a collection of annotated data to train a statistical model. For the resolution of coreferent proper nouns simple string-matching techniques were applied.He reports a small improvement, but his results do not quantify the effect of co-reference resolution effectively, since his baseline system includes terms from surrounding sentences.

Vicedo and Ferrández (2000a) analysed the effects of applying pronominal anaphora resolution to QA systems. They apply a knowledge-based approach, dividing the different kinds of knowledge (e.g. pos-tags,syntactic knowledge and morphological knowledge) into preferences and restrictions. Both the restrictions and the preferences are used to discard candidate antecedents. Contrary to our results their outcomes show a great improvement in QA performance. This difference in results can be explained by several aspects of both our experiments.

First, their anaphora resolution system achieved a higher success rate, 87% for Spanish and 84% for English. Our results could be improved. Hoste and Daelemans (2005), for example, reached a precision score for pronouns around 65% and for common nouns around 48% using a machine learning approach for Dutch co-reference resolution.

Second, they also applied anaphora resolution for query terms that are referenced pronominally in the target sentence (sentence containing the correct answer). We only looked at possible answers that were realised anaphorical.

Third, the authors consider an answer to a question to be correct if it appeared into the ten most relevant sentences returned by the system for each question, while we only evaluated the answer ranked first.

And last but not least, their experiment differs from our experiment in that they created their own question set. These questions were known to have an answer in the document collection. Moreover, for more than 50% of these questions the answer or a term in the query was referenced pronominally in the target sentence. This percentage depends on the corpus and the question set and it was probably lower for our data set. The authors define a well-balanced question set as a set that would have a percentage of target sentences that contain pronouns similar to the pronominal reference ratio of the text collection that is being queried. However, most question sets made available by the well-known evaluation fora TREC and CLEF seem to be not that well-balanced according to the definition of Vicedo and Ferrández (2000a). On the other hand, does such a well balanced question set represent a typical set of

question set asked by users? It needs further investigation to decide what makes a good question set for the evaluation of the contribution of anaphora resolution to QA.

Vicedo and Ferrández (2000b) participated in the QA track of TREC 2000. The results achieved there were more similar to ours. Application of pronominal anaphora resolution produced only a small benefit, around a 1%. The authors argue there are two main reasons for this result. First, they noticed that the number of relevant sentences involving pronouns is very low. That is in line with our findings. Second, the authors observed that there were a lot of documents related to the same information: sentences in a document that contain the right answer referenced by a pronoun, can also appear in another document without pronominal anaphora. This observation affirms that further investigation to the question set and corpus is needed.

7. Conclusions and Future work

We can conclude from our results that applying anaphora resolution is a way to improve the coverage of answer extraction. In our experiments it resulted in an improvement of the performance of a state-of-the-art QA system for Dutch, in spite of three impediments. Firstly, the precision score for the anaphora resolution was quite low. Secondly, only a limited number of questions was assigned one of the twelve question types for which we applied anaphora resolution. Thirdly, it seemed that questions could be re-formulations of sentences in the corpus.

In the future we should investigate what happens if we improve the anaphora resolution technique and if the domain of question types on which anaphora resolution is applied is broadened. In addition, we need to examine the impact of certain corpora and question sets on the evaluation of the contribution of anaphora resolution to QA.

8. References

G. Bouma, J. Mur, G. van Noord, in L. van der Plas. 2005a. Question answering for dutch using dependency relations. V: *Proceedings of the CLEF 2005 Workshop*.

G. Bouma, J. Mur, in G. van Noord. 2005b. Reasoning over dependency relations for qa. V: *Proceedings IJCAI Workshop on Knowledge and Reasoning for Answering Questions*.

M. Fleischman, E. Hovy, in A. Echihabi. 2003. Offline strategies for online question answering: Answering questions before they are asked. V: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

S. Hartrumpf. 2005. University of hagen at qa@clef 2005: Extending knowledge and deepening linguistic processing for question answering. V: *Proceedings of CLEF 2005 Workshop*.

V. Hoste in W. Daelemans. 2005. Learning dutch coreference resolution. V: T. van der Wouden, Michaela Poß, Hilke Reckman, in Crit Cremers, ur., *Computational Linguistics in the Netherlands 2004*, str. 133–148, Utrecht. LOT.

C. Jiangping, Y. Ping, in G. He. 2005. Unt 2005 trec qa participation: Using lemur as ir search engine. V: *Proceedings of TREC 2005*.

V. Jijkoun, J. Mur, in M. de Rijke. 2004. Information extraction for question answering: Improving recall through syntactic patterns. V: *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*.

D. Laurent, P. Séguéla, in S. Nègre. 2005. Cross lingual question answering using qristal for clef 2005. V: *Proceedings of the CLEF 2005 Workshop*.

B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Pe nas, V. Peinado, F. Verdejo, in M. de Rijke. 2004. The multiple language question answering track at clef. V: C. Peters, M. Braschler, J. Gonzalo, in M. Kluck, ur., *Results of the CLEF 2003 Evaluation Campaign. Lecture Notes in Computer Science*, Berlin Heidelberg New York. Springer-Verlag.

R. Malouf in G. van Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. V: *IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*.

D. Mollá, R. Schwitter, F. Rinaldi, J. Dowdall, in M. Hess. 2003. Anaphora resolution in extrans. V: *International Symposium on Reference Resolution and its Application to Question-Answering and Summarisation (ARQAS)*.

T.S. Morton. 2000. Coreference for nlp applications. V: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*.

J. Mur in M.L.E. van der Plas. to appear. Anaphora resolution for off-line answer extraction using instances. V: *Workshop on Anaphora Resolution*.

G. Neumann in B. Sacaleanu. 2005. Dfki's lt-lab at the clef 2005 multiple language question answering track. V: *Proceedings of the CLEF 2005 Workshop*.

P. Schone, G. Ciany, R. Cutts, P. McNamee, J. Mayfield, in Tom Smith. 2005. Qactis-based question answering at trec-2005. V: *Proceedings of TREC 2005*.

R. Stuckardt. 2003. Coreference-based summarization and question answering: a case for high precision anaphor resolution. V: *International Symposium on Reference Resolution and its Application to Question-Answering and Summarisation (ARQAS)*.

E. Tjong Kim Sang. 2005. Developing offline strategies for answering medical questions. V: *Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains*.

J.L. Vicedo in A Ferrández. 2000a. Importance of pronominal anaphora resolution in question answering systems. V: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*.

J.L. Vicedo in A Ferrández. 2000b. A semantic approach to question answering systems. V: *Proceedings of TREC-9*.

R. Watson, J. Preiss, in E.J. Briscoe. 2003. The contribution of domain-independent robust pronominal anaphora resolution to open-domain question-answering. V: *International Symposium on Reference Resolution and its Application to Question-Answering and Summarisation (ARQAS)*.

Fast extraction of discontiguous sequences in text: a new approach based on maximal frequent sequences.

Antoine Doucet, Helena Ahonen-Myka

Department of Computer Science
PO Box 68 (Gustaf Hällströminkatu 2B)
FI-00014 University of Helsinki

Abstract

In this paper, we present a new technique for the extraction of discontiguous sequential descriptors from text. We are able to form word sequences without any restriction on their size or on the distance between their components. Based on the concept of a maximal frequent sequence (MFS), our approach allows for the extraction of compact text descriptors of quality in a more efficient manner than other previously known techniques. It further scales up to document collections of virtually any size, when other approaches normally fail for collections large enough. After a review of the related work and the presentation of our approach, $MFS\_MineSweep$, we introduce measures of the quality and quantity of information in a set of sequential descriptors representing a document collection. We finally present experiments whose results demonstrate the real-life applicability and superiority of the proposed method.

Hitra ekstrakcija nepovezanih zaporedij v besedilu: nov pristop na podlagi najpogostejših zaporedij

V prispevku predstavljamo novo tehniko za ekstrakcijo deskriptorjev za nepovezana zaporedja v besedilu. Tvorimo lahko zaporedja besed brez omejitev v velikosti ali razdalji med posameznimi komponentami. Na podlagi koncepta najpogostejšega zaporedja omogoča ta pristop učinkovitejšo ekstrakcijo kakovostnih deskriptorjev zgoščenega besedila v primerjavi z doslej uporabljanimi tehnikami. Poleg tega je z njim mogoče obdelovati zbirke dokumentov vseh velikosti, medtem ko drugi pristopi navadno odpovejo pri večjih zbirkah. Po pregledu drugih del s tega področja in predstavitvi našega pristopa $MFS\_MineSweep$ predstavimo serijo meritev kvalitete in kvantitete informacij v množici zaporednih deskriptorjev, ki predstavljajo zbirko dokumentov. Na koncu predstavimo eksperimente, katerih rezultati kažejo prednost predstavljene metode in njeno uporabnost v resničnem življenju.

1. Introduction

Most document models rely on single word terms. A trend to improve this fact is to extract and use multi-word units or phrases. The problem of detecting such cohesive lexical units is a very difficult one as the number of possible word compounds in text is enormous and a vast majority of them do not constitute true multi-word units.

Exhaustive approaches are clearly exponential and researchers have therefore always had to place several restrictions on the search space, such as a maximal phrase length (Church and Hanks, 1990), fixed relative positions (Dias, 2003), or linguistic filtering (Mitra et al., 1987). Maximal frequent sequences (MFS) (Ahonen-Myka and Doucet, 2005) are a type of phrases that presents the advantage to remove most of these constraints. They can be formed of words separated by any distance and they are not restricted in length. MFSs as content descriptors present two major strengths. Firstly, they offer a very compact description: with a maximal phrase length of 8 words, it takes thousands of 8-sequences to replace a single phrase of length 20 (precisely, $\binom{20}{8} = 125\,970$ such sequences). Secondly, they do not require any knowledge about the data at hand. They can therefore be applied to documents in any domain and written in any language. We believe this is a very strong point when billions of heterogeneous documents coexist in real-world document collections such as the World Wide Web.

This paper presents two contributions. The first one is $MFS\_MineSweep$, a technique relying on MFSs to extend the extraction of compact sequential descriptors to very large document collections. The resulting phrasal document descriptions are far more exhaustive and have a higher discriminative power. The second contribution is the introduction of metrics to measure the quality and density of a sequence-based representation of a document collection.

In the next section, we will formally define the concept of a Maximal Frequent Sequence (MFS) and review the current state of the art of work addressing the problem of their extraction We will then present $MineMFS$, the current best-performing algorithm for the extraction of MFSs in text and expose some of its limitations. In Section 3, we will introduce our contribution, $MFS\_MineSweep$, a technique that relies upon $MineMFS$ to extract relevant document descriptors from document collections of virtually any size. We will then introduce a set of metrics for the evaluation of a sequence-based document description (Section 4). Before concluding the paper, we will present and discuss our experiments in Section 5.

2. Maximal Frequent Sequence (MFS)

In this section, we will introduce the concept of a Maximal Frequent Sequence in further detail (Ahonen-Myka and Doucet, 2005). We will then overview the data mining techniques that aim at the extraction of sequential patterns, and particularly those that permit to extract MFSs.

2.1. Definitions

Definition 1 *A sequence* $p = a_1 \cdots a_k$ *is a* subsequence *of a sequence* q *if all the items* $a_i, 1 \leq i \leq k$, *occur in* q

1. The **Congress** subcommittee backed away from mandating specific **retaliation against foreign** countries for **unfair** foreign **trade practices**.

2. He urged **Congress** to reject provisions that would mandate U.S. **retaliation against foreign unfair trade practices**.

3. Washington charged France, West Germany, the U.K., Spain and the EC Commission with **unfair practices** on behalf of Airbus.

Figure 1: A set of sentences from the Reuters-21578 collection (1987).

and they occur in the same order as in p. If p is a subsequence of q, we also say that p occurs *in q and that q is a* supersequence *of p.*

For instance, the sequence *"unfair practices"* can be found in all of the three sentences in Figure 1.

Definition 2 *A sequence p is* frequent *in a set of fragments S if p is a subsequence of at least σ fragments of S, where σ is a given frequency threshold.*

If we assume that the frequency threshold is 2, we can find the following frequent sequences in our sample set of sentences: *"congress retaliation against foreign unfair trade practices"* and *"unfair practices"* (Fig. 1).

Definition 3 *A sequence p is a* maximal frequent (sub)sequence *in a set of fragments S if there does not exist any sequence p' in S such that p is a subsequence of p' and p' is frequent in S.*

In our example, the sequence *"unfair practices"* is not maximal, since it is a subsequence of the frequent sequence *"congress retaliation against foreign unfair trade practices"*. This latter sequence is maximal.

With this simple example, we already get a glimpse of the compact descriptive power of MFSs. Should we be restricted to word pairs, the 7-gram *"congress retaliation against foreign unfair trade practices"* would need to be replaced by 21 bigrams. With MFSs, we can obtain a very compact representation of the regularities of text. The rest of this section will focus on the problem of their efficient extraction in a document collection.

2.2. Related Work

Given a document collection and a minimal frequency threshold, the naïve approach is to go through the document collection, collect each frequent word, and use the set of all frequent words to produce candidate word pairs (bigrams) and retain only the frequent ones. The process of forming and counting the frequency of (n+1)-gram candidates from the set of all frequent n-grams can be repeated iteratively as long as frequent (n+1)-grams are found. To obtain the set of all MFSs, it remains to remove every frequent sequence that is a subsequence of another frequent sequence. But this approach is clearly computationally inefficient.

2.2.1. Sequential Pattern Mining

Agrawal and Srikant (1995) introduced the problem of *mining sequential patterns* as an advanced subtask of data mining, where typical data consists of customer transactions, that is, database entries keyed on a *transaction id* and each consisting of a *customer id* associated to the list of items that she bought in this very transaction. The problem of mining sequential patterns is an advanced version of that of the extraction of interesting *item sets*. But in sequential pattern mining, we also aim to exploit the fact that the transaction entries of the databases include a time field that permits to sort the transactions in chronological order and even know the time interval (or distance) that separates them. A motivating example of a sequential pattern, from (Agrawal and Srikant, 1995), would be that customers typically rent the movie "Star Wars", then "The Empire Strikes Back", and finally "The Return of the Jedi".

Agrawal and Srikant (1995) presented an improvement of the naïve approach that benefits of an intermediary pruning step to remove all (n+1)-gram candidates that contain at least one non-frequent n-gram. This permits to avoid a number of useless frequency counts. Most approaches are fueled by the same idea of pruning a number of "candidate frequent sequences", to avoid costly frequency counts.

Zaki (2001) presented $SPADE$, an advanced technique for the discovery of sequential patterns. Its architecture relies on a vertical database that fastens frequency counts and a lattice-theoretic approach permits to reduce the search space. Unfortunately, the main weakness of $SPADE$ is that it still enumerates all the candidate sequences by forming candidate (n+1)-sequences through the combination of each two n-sequences. DFS_Mine (Tsoukatos and Gunopulos, 2001) was subsequently designed to try to discover n-sequences without enumerating all the frequent sequences of length (n-1). This is done by storing two lists, containing "minimal non-frequent sequences" (because their supersequences are necessarily infrequent) and "maximal frequent sequences" (because their subsequences are necessarily frequent). A significant number of frequency counts can then be avoided. The problem with DFS_Mine is that the candidate (n+1)-sequences are formed by combining an n-sequence with the items of the database. While this may function with spatiotemporal data, the presented application of DFS_Mine, where the number of items is low, this is not reasonable for text, where the number of items (words) can be enormous.

2.2.2. Sequential Patterns and Text

The key particularity of text as a sequential data type is the number of items. For instance, the vocabulary of the widely known *Brown corpus* contains $50,406$ distinct words, whereas, e.g., biosequences have a very limited vocabulary: there are only 20 amino acids, and only 4 molecules containing nitrogen in DNA and RNA (A, C, G, and T). Another particularity of text is that the distribution of words is skewed. There is a small number of words that are very frequent, whereas the majority of words are infrequent. The words with moderate frequency are usually considered the most interesting and most informative.

These special characteristics of textual data have a strong influence on the discovery of interesting sequences in text. All the breadth-first, bottom-up approaches are failing quickly for a number of reasons. They permit pruning but require to keep in memory all the subsequences of two distinct lengths. They further generate a large number of candidates whose frequency is slow to count. Depth-first search takes less memory, but the number of items (words) to be intersected with a given sequence is prohibitive.

2.3. Sequential Pattern Mining in Text: $MineMFS$

MineMFS (Ahonen-Myka and Doucet, 2005) is a method combining breadth-first and depth-first search that is particularly well-suited for text. It extracts MFSs of any length, i.e., also very long sequences, and it allows an unrestricted gap between words of the sequence. In practice, however, text is usually divided into sentences or paragraphs, which indirectly restricts the length of sequences, as well as the maximal distance between two words of a sequence. The constraints used in the method are minimum and maximum frequency. Hence, words that are less (respectively, more) frequent than a minimum (respectively, maximum) frequency threshold are removed.

Algorithm. As for DFS_Mine, an important idea in $MineMFS$ is to compute frequent (n+1)-sequences without enumerating all the frequent n-sequences. It relies on a set of "n-gram seeds", initialized with the set of all frequent bigrams. The main idea is to pick an n-gram seed and try to combine it with other grams in a greedy manner, i.e., as soon as the n-gram seed is successfully expanded to a longer frequent sequence, other expansion alternatives are not checked, but only that longer frequent sequence is tentatively expanded again. This expansion procedure is repeated until the longer frequent sequence at hand can only be expanded to infrequent sequences. This sequence is maximal. When all the n-gram seeds have been processed, those that cannot be used to form a new maximal frequent sequence of size more than n are pruned. The remaining ones are joined to produce candidate (n+1)-gram seeds that will be used in a new iteration of the process. This process is repeated until no new maximal frequent sequence can be discovered.

Strengths. A main strength of $MineMFS$ versus DFS_Mine is the fact that the choice of items that may be inserted to expand an n-gram is restricted to the other non-pruned frequent n-grams. Whereas in DFS_Mine, an n-gram is expanded by trying to insert every (or most) frequent word, which is too costly for textual data. Further sophisticated pruning techniques permit restricting the depth-first search, which means only a few alternatives need to be checked to try to expand a sequence, despite the large vocabulary size.

Limitations. Even though the use of minimal and maximal frequency thresholds permits to reduce the burstiness of word distribution, it also causes the miss of a number of truly relevant word associations. For large enough collections, the $MineMFS$ process fails to produce results, unless excessive minimal and maximal frequencies are decided upon, in which case the set of MFSs produced is small and contains mostly non-interesting descriptors. One reason may be the pruning step, which runs through the set of n-grams and compares each two of them that may form an (n+1)-gram, by checking if a new item can be added between every two adjacent words. The number of possible positions of insertion shall be problematic.

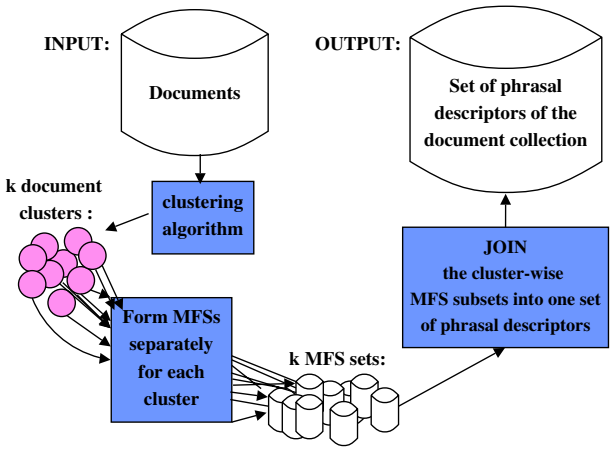


Figure 2: The different phases of $MFS\_MineSweep$.

3. Partitioning the Collection to Approximate the MFS set efficiently

We have seen that $MineMFS$ fails to extract the MFS set of a sufficiently large document collection. In this section, we will introduce *MFS_MineSweep*, a technique to decompose a collection of documents into several disjointed subcollections, small enough so that the MFS set of each subcollection can be extracted efficiently. Joining all the sets of MFSs, we obtain an approximate of the maximal frequent sequence set for the full collection. $MFS\_MineSweep$ permits extracting more and sharper descriptors from document collections of virtually any size. Its main drawback is the loss of the maximality property, producing a less compact set of content descriptors.

3.1. Description and Claims

Our approach relies on the idea to partition the document collection into a set of homogeneous subcollections. The initial motivation to do this is that $MineMFS$ does not produce any result at all for sufficiently large document collections. Figure 2 describes the steps of $MFS\_MineSweep$. In the first phase, we apply $MineMFS$ on a number of disjoint subcollections, so as to obtain an MFS set corresponding to each subcollection. The second step is to gather the MFS sets of each subcollection to form a set of content descriptors for the whole collection. This gathering operation mainly consists in appending the sets of MFSs, as there is no clear way to join a sequence (maximal frequent in a subcollection) to its subsequence (maximal frequent in another). Only identical sequences can be merged. Thus, the maximality property is lost, and therefore, the content description of our pre-partitioning technique is always less or equally compact to that of the MFSs of the whole collection.

With this technique, we make two main claims that we will try to confirm or disprove in the evaluation. The main

- d_1: Mary had a little lamb whose fleece was white as snow.
- d_2: A radio station called Sputnik broadcasts Russian programs in Saint-Petersburg and Helsinki. It was named after the first satellite ever launched.
- d_3: History changed on October 4, 1957, when the Soviet Union successfully launched Sputnik I. The world's first artificial satellite was about the size of a basketball, weighed only 183 pounds, and revolved around the Earth in about 98 minutes.
- d_4: Everywhere that Mary went, her lamb was sure to go.

Figure 3: A collection of four documents.

motivation for developing $MFS\_MineSweep$ is to efficiently obtain a more detailed description of the document collection (*Hypothesis H1*), as we can use looser frequency thresholds. This is easily understood by thinking of an extreme case; if a collection of $|D|$ documents is split into $|D|$ subcollections of size 1 and the minimal frequency is 1, we can obtain the corresponding sets of MFS instantly: each MFS set contains only one sequence of frequency 1, the unique document in the corresponding subcollection. No information is lost, but the content description is probably too large.

Our second main claim is about the optimal way to form the disjointed subcollections. We conjecture that more consistent subcollections permit to obtain better descriptors (*Hypothesis H2*). The main reason of this train of thought relies on the fact that a collection made of similar documents will contain more interesting MFSs than a collection made of dissimilar documents. Again, thinking of extreme cases makes this point easier to see, as a collection where no two documents have a word in common will not contain any frequent sequences, except for the documents themselves (if the frequency threshold is 1).

For example, let us assume that we want to partition the collection of four documents presented in Figure 3 into 2 subcollections of 2 documents each, and use a minimal frequency of 2 for extracting MFSs from the subcollections. Only by clustering together the similar documents (d_1, d_4) and (d_2, d_3), will we obtain sequences of words, that is, *phrasal descriptors*. Those descriptors are: "Mary lamb was" for d_1 and d_4, and "Sputnik first satellite launched" for d_2 and d_3. Any other way to partition the collection produces an empty *phrasal description*.

4. Evaluating a Phrasal Text Description

To confirm or disprove the hypotheses we just made, we need measures to compare different sets of phrasal descriptors. Ideal metrics upon which to compare sets of descriptors should be able to evaluate two things: 1) the size of the phrasal text representation, and 2) the amount (and density) of information it contains.

In general, the problem of comparing two sets is not an easy one. A large quantity of work in the domains of document clustering and textual classification has proposed measures to compare different ways to partition document sets (Sebastiani, 2002). Unfortunately, we cannot exploit this work to solve our problem, because such techniques rely on the comparison of a given clustering (or classification) to a gold standard. In the general case of textual representation, without aiming at a specific application, there is no clear way to define a gold standard of the phrasal description of a document collection.

Fortunately, the problem we are facing here is a subproblem of the above. The sets we need to compare are indeed similar in nature. For example, a major difficulty in comparing general sequences would be the comparison of long grams to their subgrams. However, in the specific case where all the descriptors are MFS (either of the whole collection or of one of its subcollections), we can simplify the problem by normalizing each descriptor to a set of all its subpairs. This is because the unlimited distance allowed between any two words of an MFS ensures that the assertion "$ABCD$ is an MFS" implies "AB, AC, AD, BC, BD, and CD are frequent bigrams".

We can thus transform each set of phrasal descriptors into a set of comparable items, the frequent bigrams it contains. Let R_D be the phrasal description of a document collection D, and R_d be the corresponding set of phrases describing a document $d \in D$. We can write the corresponding set of word pairs as $bigrams(R_d)$. For $b \in bigrams(R_d)$, we also define df_b as the document frequency of the bigram b. Finally, we define the random variable X over the set $bigrams(R_d)$. For all $b \in bigrams(R_d)$:

$$p(X = b) = \frac{df_b}{\sum_{y \in \{ \bigcup_{d \in D} bigrams(R_d) \}} df_y},$$

where $\sum_{y \in \{\bigcup_{d \in D} bigrams(R_d)\}} df_y$ is the total number of bigram occurrences resulting from the phrasal description R_D. It can be thought of as the sample size.

Size of the representation of a document collection. The phrasal representation of a document collection can be seen as a set of associations between descriptive n-grams and documents. We define $|R_D|$ as the size of the phrasal representation R_D in a very intuitive way:

$$|R_D| = \sum_{d \in D} |R_d|.$$

Hence, $|R_D|$ is the number of document-phrase associations in the collection representation R_D.

Implied quantity of frequent bigrams in the representation. Several phrases may contain identical bigrams that represent the same document. To count the number of implied document-bigram associations permits to ignore redundant information stemming from the long descriptors. We shall therefore measure the quantity of information in the description with the number of document-bigram associations that correspond to the description R_D. This value is $bigram\_size(R_D)$, defined as follows:

$$bigram\_size(R_D) = \sum_{d \in D} |bigrams(R_d)|.$$

Hence, $bigram\_size(R_D)$ is the number of document-bigram associations stemming from the collection representation R_D.

Density of the description. To measure whether the description is loose or dense, we can use the two preceding metrics in a very simple way. By computing the ratio between the number of document-bigram associations in a document representation and its size, we obtain a relative measure of the number of document-bigram associations that can be avoided with longer n-grams:

$$Density(R_D) = \frac{bigram\_size(R_D)}{|R_D|}.$$

For example, a density value of 1.1 means that the bigram representation of R_D contains 10% more associations than the equivalent representation R_D. The higher $Density(R_D)$, the more storage space we save by using R_D instead of frequent pairs only.

5. Experiments and Results

5.1. Experiments and Results

In this section, we will detail and implement a set of experiments that permit to test our initial hypotheses. It is important to observe that the extraction of the set of MFSs is an independent process for each distinct subcollection. A profitable alternative is to run the extraction of the MFS sets in parallel, on distinct computers. The total running time is then the time of the slowest MFS set extraction, plus the time for splitting the document collection. The experiments are based on a set of desktops with a 2.80 Ghz processor and 1024Mb of RAM.

5.1.1. $MFS\_MineSweep$ extracts better, but less compact descriptors *(Hypothesis H1)*.

The claim of hypothesis *H1* is that we can extract more information using $MFS\_MineSweep$, although we then lose the maximality property, subsequently leading to a less compact description. To verify this, we experiment with the 16Mb Reuters-21578 newswire collection (Reuters-21578, 1987), which originally contains about $19,000$ non-empty documents. To place both techniques on equal grounds, we find a frequency range for every subcollection individually, such that the corresponding MFS extraction time is always between 4 and 5 minutes. This was achieved with a fairly simple heuristic, interrupting the process and decreasing the frequency range when the extraction was too slow, and increasing the frequency range after too fast an extraction. We then compare the resulting sizes, amounts and densities of information in Table 1. Note that every value resulting from a random partition into n subcollections is actually the average outcome of 10 distinct iterations of the random partitioning and evaluation process. Experiments have shown the variance is very small.

MFS_MineSweep outperforms MineMFS. Our first observation is that both the number of descriptors and the number of equivalent bigrams are always much higher for $MFS\_MineSweep$ than for $MineMFS$. These numbers increase with the number of partitions.

Partitions (min,max)	Bigrams	Descriptors	Density
1 [MineMFS] (85,900)	147,000	126,000	1.17
2 (60-70, 900-1000)	841,000	819,000	1.03
3 (40, 650-715)	1,223,000	1,197,000	1.02
5 (25-30, 400-600)	1,605,000	1,574,000	1.02
10 (5-28, 72-350)	1,453,000	1,466,000	0.99
20 (10-28, 162-385)	1,643,000	2,555,000	0.64
50 (4-20, 60-208)	2,927,000	7,448,000	0.39
100 (3-45, 27-630)	3,570,000	11,038,000	0.32

Table 1: Reuters. Corresponding frequency ranges when every subcollection is computed within 4 and 5 minutes using $MineMFS$ directly and $MFS\_MineSweep$ on random partitions of size 2, 3, 5, 10, 20, 50 and 100.

The description is less compact. Consequently, the density of the phrasal representations is decreasing with the number of subcollections. What we did not expect is that the density ratio goes down to values below 1, meaning that the number of equivalent bigrams is less than the number of phrasal descriptors. This steep density decrease expresses more than the loss of the maximality property. A lower density means that the number of descriptors is growing faster than the number of bigrams. When we split the collection into more disjoint subcollections, this means that more and more of the new descriptors we find are only new combinations of bigrams that we already found when we split the collection in less partitions. This sharp decrease in density is in fact an indication that the discriminative power of the phrasal description is peaking, and that further augmentations of the number of partitions will be comparatively less and less worthwhile.

The hypothesis *H1* is verified, an increase in the number of subcollections is followed by a more exhaustive, but less compact document description. We shall suspect that with homogeneous partitioning, a rise in the number of subcollections will increase their internal similarity and facilitate the discovery of new descriptors, with a strong discriminating power. This is to be verified in the following subsection.

5.1.2. The more homogeneous the subcollections, the better the descriptors *(Hypothesis H2)*.

To support *H2*, we use the same newswire collection and compare the size, amount and density of information obtained when splitting the collection into random and homogeneous subcollections. In the experiments, we formed homogeneous subcollections with the well-known k-means clustering algorithm. We used the publicly available clustering tool implemented by George Karypis at the University of Minnesota[1]. The phrasal descriptors resulting of homogeneous subcollections are evaluated in Table 2.

MFS_MineSweep outperforms MineMFS. What we had observed with random partitions is confirmed with homogeneous collections. We get a more exhaustive description of the document collection if we use $MFS\_MineSweep$ than if we use $MineMFS$ alone.

[1]CLUTO, http://www-users.cs.umn.edu/∼karypis/cluto/

Clusters (min,max)	Bigrams	Descriptors	Density
1 [MineMFS] (85,900)	147,000	126,000	1.17
2 (40-130, 660-1569)	554,000	568,000	0.97
3 (7-129, 180-1470)	449,000	498,000	0.90
5 (3-55, 47-1224)	995,000	993,000	1.00
10 (5-22, 58-671)	1,255,000	1,280,000	0.98
20 (3-14, 11-682)	1,767,000	1,904,000	0.93
50 (2-37, 5-289)	2,201,000	2,748,000	0.80
100 (2-28, 7-220)	2,932,000	4,597,000	0.64

Table 2: Reuters. Corresponding frequency ranges when every subcollection is computed within 4 and 5 minutes using $MineMFS$ directly and $MFS\_MineSweep$ on homogeneous partitions of size 2, 3, 5, 10, 20, 50 and 100.

Partitions	Random	Homogeneous
2	841,000 (1.03)	554,000 (0.97)
3	1,223,000 (1.02)	449,000 (0.90)
5	1,605,000 (1.02)	995,000 (1.00)
10	1,453,000 (0.99)	1,255,000 (0.98)
20	1,643,000 (0.64)	1,767,000 (0.93)
50	2,927,000 (0.39)	2,201,000 (0.80)
100	3,570,000 (0.32)	2,932,000 (0.64)

Table 3: Reuters. Quantities and densities of information when every subcollection is computed within 4 and 5 minutes using $MFS\_MineSweep$ on random and homogeneous partitions of size 2, 3, 5, 10, 20, 50 and 100.

To permit an easier direct comparison, the quantities and densities of information obtained with random and homogeneous partitions are presented in Table 3.

Homogeneity provides better discrimination. We can observe that when the number of partitions rises, the density of the description resulting from homogeneous subcollections decreases slowly, whereas the steep is much sharper for random partitions. The fact that the description densities resulting from homogeneous collections remain nearly stable shows that there is room to improve the discriminative power of phrasal descriptions if we partition the document collection in even more clusters. The reason is simple. The descriptors extracted from random subcollections are ones that are present all over the collection. Splitting the collection into more subsets permits finding more of those frequent n-grams, formed by the same frequent words, but we reach a point where we only find combinations of the same frequent words originating from different subcollections. On the other hand, homogeneous subcollections permit gathering similar documents together, excluding non-similar documents. Hence, the frequency range can be adapted to extract the specifics of each subcollection. In the homogeneous case, increasing the number of subcollections permits embracing more specificities of the document collections, whereas in the random case, it only permits catching more descriptors of the same kind.

Clustering is safer. As opposed to random partitioning, clustering provides $guarantees$. It is more reliable, because it ensures result. The strength of random partitioning is it gives good results and permits MFS extraction in predictable times. But these facts are only true *on average*. The problem if we use random partitioning is that we should, in fact, run several iterations to protect ourselves from an "unlucky" draw. We mentioned earlier that running several random iterations increases the exposure to factors of difficult extraction. Another issue with averaging numerous iterations is practical. Assume document d was represented 3 times by $gram_A$, and 1 time by $gram_B$ and $gram_C$, what should be the average document description of d? Because the extraction of MFS sets from homogeneous subcollections is unique and needs to be done only once, it is generally less costly in the end.

6. Conclusion

In this paper, we introduced $MFS\_MineSweep$, a new solution for the extraction of compact phrasal descriptors from sequential data. We further defined metrics for the evaluation of such descriptions. We presented experiments on textual data that showed the capacity of $MFS\_MineSweep$ to extract a better description efficiently, by applying an MFS extraction algorithm on partitions of the document collection. Our approach permits to obtain a more exhaustive description faster. This improvement is strengthened by the possibility to run the costliest computations in parallel. We further established that the use of homogeneous partitions improves the quality of the description.

7. References

Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining sequential patterns. In Yu and Chen, editors, *11th International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan. IEEE Computer Society Press.

Helena Ahonen-Myka and Antoine Doucet. 2005. Data mining meets collocations discovery. In *Inquiries into Words, Constraints and Contexts, Festschrift for Kimmo Koskenniemi*, pages 194–203. CSLI Publications, University of Stanford.

Kenneth W. Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Gaël Dias. 2003. Multiword unit hybrid extraction. In *Workshop on Multiword Expressions of the 41st ACL meeting. Sapporo. Japan.*

Mandar Mitra, Chris Buckley, Amit Singhal, and Claire Cardie. 1987. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO97, Computer-Assisted Information Searching on the Internet*, pages 200–214.

Reuters-21578. 1987. Text categorization test collection.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Survey*, 34(1):1–47.

Ilias Tsoukatos and Dimitrios Gunopulos. 2001. Efficient mining of spatiotemporal patterns. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD)*, pages 425–442.

Mohammed J. Zaki. 2001. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60.

Finite State Transducers for Recognition and Generation of Compound Words

Cvetana Krstev*, Duško Vitas†

*Faculty of Philology, University of Belgrade
Studentski trg 3, 11000 Belgrade, Serbia
cvetana@matf.bg.ac.yu

†Faculty of Mathematics, University of Belgrade
Studentski trg 16, 11000 Belgrade, Serbia
vitas@matf.bg.ac.yu

Abstract

In this paper we present how finite state transducers can be effectively used for compound treatment in text analysis. The approach that we use is particularly well suited for text processing based on the usage of morphological electronic dictionaries and finite state technology. The results that we present do not aim to be comprehensive but rather illustrative of the power of possibilities, one of which is that compounds processed in the suggested way can be used in much the same way as simple words.

Končni transduktorji za razpoznavanje in generiranje tvorjenk

V prispevku pokažemo, kako lahko končne transduktorje učinkovito uporabljamo za obravnavanje zloženk pri analizi besedila. Pristop, ki ga uporabljamo, je posebej primeren za obdelovanje besedila na podlagi uporabe morfoloških elektronskih slovarjev in tehnologije končnih avtomatov. Predstavljeni rezultati niso izčrpni; njihov namen je namreč ponazoritev možnosti. Ena od teh možnosti je, da tvorjenke, ki so obdelane na predlagani način, lahko uporabljamo zelo podobno kot netvorjene besede.

1. Introduction

One method of text processing and tagging is based on the use of electronic dictionaries. Generally speaking, this method applies electronic dictionaries to text trying to match every simple word form from the text with some simple word lexical entry from the dictionary. When match is found, all or some information found in the dictionary is attached to the simple word form.

The method is based on a formal assumption that some characters are alphabetic and that only these characters are used to form the simple word forms. The other characters are treated as tokens that separate the simple word forms. However, this approach is too formal and can lead to the erroneous or misleading tokenization in many cases. For instance, one can easily find in a Serbian text a sequence of tokens *92 miliona i 850 hiljada* '92 millions and 850 thousands' that would be treated as at least five different tokens, two numbers and three simple words. It can be argued that the sequence represents one unit, the compound numeral. In other cases, separators divide units into several simple words that cannot be tagged correctly. For instance, three different tokens are found in the sequence *pop-kultura* 'pop culture', two of them being the simple word forms: *pop* and *kultura*. The first of them is used only in compounds, *pop-zvezda* 'pop-star', *pop-koncert* 'pop-concert', *pop-pevačica* 'pop-singer', etc, so it is difficult to tag it properly. Moreover, in this case *pop* would not even be among the unrecognized (unknown) words since the dictionary would contain the homographic form *pop* 'priest'.

Yet another problem is the attachment of semantic markers to the simple word forms. The semantic markers are a way to encode certain kind of information in dictionaries. The marker that applies to the simple word form is not necessarily correct for the compound. For instance, the semantic marker attached to the simple word form *mikser* found in the compound *video-mikser* 'video-mixer' would be +Art, suggesting that it is an artifact, like a kitchen utensil. The same marker would be incorrect for the compound, for which +Hum should apply since it represents a profession.

The text processing of Serbian that we used is close to the approach described in (Laporte, 2003). In this approach the finite state automata methodology is used for text representation, as well as dictionary and query representation (Maurel and Guenthner, 2005). This approach relies on various lexical resources, the most important being the morphological e-dictionaries of simple and compound words in so called LADL format (Courtois and Silberztein, 1990) and FSTs for the inflection of simple words in Intex/Unitex format[1]. The lexical resources in this format for Serbian are presented in more detail in (Vitas et al. 2003).

In this paper we will represent how finite state transducers (FST) can be used in two different ways to correctly recognize and tag various compounds. In section 2 we will present how FSTs can be used to correctly recognize in text certain types of compounds, and in section 3 we will show how FSTs can be used to generate compound lexical entries for the dictionaries. In section 4 we will give some examples of the usage of text in which compound words were tagged using the described methods. The FSTs presented are in Intex (Silberztein 2004) or Unitex (Paumier 2002) format.

2. FSTs for recognition of compounds

FSTs are mainly used during the text analysis in order to re-join the compound components that would otherwise be separated due to the formal approach to the word characters. In the following subsections we will demonstrate the usage of FSTs in recognition and appropriate tagging of acronyms, numerals and compound nouns and adjectives usingnumerals in digit form.

[1] Intex homepage: http://msh.univ-fcomte.fr/intex/
Unitex homepage: http://www-igm.univ-mlv.fr/~unitex/

2.1. Recognition and tagging of acronyms

The acronyms usually represent the large part of unrecognized (or unknown) words in a text, especially newspaper texts. In a language there are well established acronyms, like *UN* or *MUP* in Serbian (from *Ujedinjene nacije* 'United Nations' and *Ministarstvo unutrašnjih poslova* 'Ministry of Internal Affairs' respectively), and those that appear as occasional elements in text, like *SNAP* or *ASEED* for which it is difficult to decide from which name have they been derived. For that reason, it is difficult to produce a comprehensive e-dictionary of acronyms. In many languages, including Serbian, acronyms are written using capital letters only, and that can be used to approximate their recognition. Namely, all simple word forms written in capital letters only that could not have been matched with any lexical word form from e-dictionary can be treated as acronyms and tagged accordingly. For instance, the acronym *UN* could be tagged as UN,ABB+Acr.

The additional peculiarity of Serbian is that acronyms inflect in similar way as nouns. The inflectional endings of an acronym are not written with capital letters and are separated from it with the hyphen. For instance, the inflected forms of *UN* are *UN-a* (genitive), *UN-u* (dative or locative), and *UN-om* (instrumental). Moreover, possessive adjectives can be derived as well, and the possessive adjective suffix is added to the acronym in the same manner as the inflectional endings: e.g. *UN-ov* 'belonging to UN'. Finally, possessive adjectives derived in this manner also inflect: *UN-ovog* (genitive, masculine, singular), *UN-ovom* (dative or locative, masculine, singular), etc.

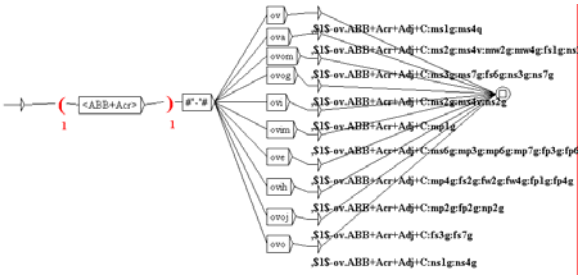


Figure 1. The FST that recognizes the inflected forms of possessive adjectives derived from acronyms

The dictionary FSTs have been derived that recognize and tag acronyms, their inflected forms, as well as possessive adjectives derived from acronyms and their inflected forms (see Figure 1). Te use of these FSTs enables the usage of lexical patterns in queries in usual way. For instance, the pattern <OPEK.ABB> retrieves all the forms of the acronym *OPEK* 'OPEC'. Some concordance lines produced by this query on a sample text are given in Figure 2. The acronyms can also be used in familiar way in syntactic queries. For instance, the query

<PREP+p2> (<ABB+Acr+Adj:ms2>+<A+Pos:ms2>) <N:ms2>

recognizes all the occurrences of the syntactic construction: preposition requiring the genitive case, followed by a possessive adjective and a masculine noun in singular in the genitive case. This query would recognize both *ispred UMNIK-ovog zatvora* 'in front of the UMNIK's prison' and *bez čovekovog upliva* 'without human's influnce' as syntactically equivalent.

petrolejskog kartela OPEK da poveća dnevnu produkciju z
za naftu članica OPEK-a su, međutim, zatražili da zemlje
ćanja proizvodnje u OPEK-u došlo je zbog opšte uzbune u

Figure 2. A few results obtained by the pattern <OPEK.ABB>

2.2. Recognition and tagging of numerals

In Serbian, the components of compound numerals are separated with blanks, for instance *trideset sedam* 'thirty seven'. The parts of compound numerals can be written using digits, like in *osam miliona i 800 hiljada* 'eight millions and 800 thousands'. The dictionary FSTs have been derived that recognize and tag numerals that represent tens, hundreds, thousands, millions and billions (see Figure 3).

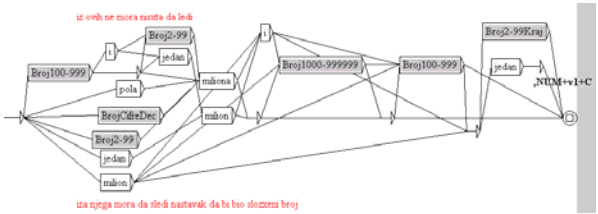


Figure 3. Recognition and tagging of numerals representing million values

Special attention in tagging numerals is to give them the appropriate syntactic tags +v1, +v2, +v3, +v4, and +v5 that will govern the agreement in number with inflected word forms. Namely, numerals with marker +v1 can agree with singular, numerals with markers +v2, +v3, +v4 agree with paukal, while numerals with marker +v5 agree with plural. Paukal is the value of number category that is used in Serbian with small numerals, two, three and four. However, number is a grammatical category, and thus paukal is used for large numbers too as long as they end with one of those small numbers. The output of the FSTs takes care about this, as can be seen in Figure 3. The numerals ending with *jedan* 'one' obtain the marker +v1, while for the markers ending with other simple numerals is responsible the sub-graph Broj2-99Kraj. The concordance lines that illustrate this phenomenon are given in Figure 4.

ku književnost, i dvadeset jedan čas za makedonske pisce.
jednog utorka, dvadeset i četiri časa posle one Dragišine
oko dvadeset i pet časova posle dolaska putnika u London

Figure 4. The compound numerals illustrating different agreements that depend on the last numeral constituent

2.3. Recognition and tagging of word forms prefixed with numerals in digit form

Quite a number of nouns and adjectives are obtained by concatenation of a numeral and some simple word form. When a numeral is in a letter form then as a result a new simple word form is obtained, for instance *devetomesečni* 'lasting nine months'. The recognition and tagging of such forms can be done by so called morphological FSTs. These graphs were implemented invented by Max

Silberztein and they are described in (Silberztein, 2004). Their application for Serbian is presented in (Vitas, 2005). In some cases, especially in newspaper texts, the numeral in the derived from is in digit form, and in these cases the new form is a compound, e.g. *9-mesečni* instead of *devetomesečni*.

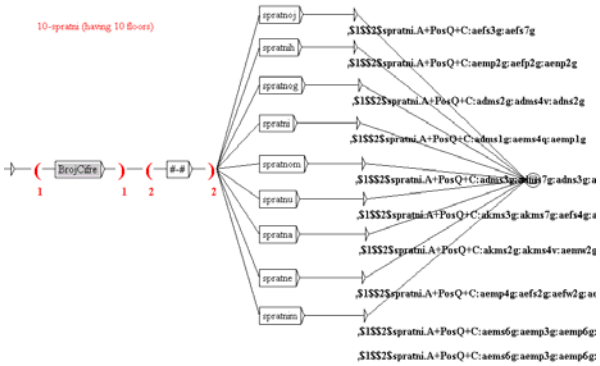


Figure 5. The FST that recognizes and tags the inflected forms of the adjectives of the form <NUM+Dig>-*spratni*

In order to recognize and tag correctly such cases a number of dictionary FSTs have been developed, one of which is given in Figure 5. Some of those FSTs recognize compound nouns, like *10-godišnjica* '10th anniversary', the others compound adjectives, like *10-godišnji* 'lasting 10 years'. The graphs were produced only for the frequently used compounds, in which practically any number can occur, e.g. *2000-godišnjica* '2000th anniversary'. The cases that occur occasionally in texts due to the extraordinary inventiveness or laziness of mostly sports journalists were not taken in the account, like *11-terac* '11 meters penalty spot', since only a few numbers can be used.

3. FSTs for Generation of Compound Lexical Entries

In the text processing that is based on lexical recognition the largest part of the compounds will be recognized by appropriate morphological e-dictionaries in the same way as the simple words are recognized. It means that in order to produce such a dictionary the following steps have to be performed:

1. The compound lemmas have to be collected;
2. Each lemma's inflectional properties have to be established and adequately formalized;
3. The inflected forms of all collected lemmas are automatically generated.

These steps are basically the same as those undertaken for the generation of the morphological e-dictionaries of simple words. The main differences are in step 2, since inflectional properties of compounds are more difficult to establish and formalize for an effective use. Namely, when considering the inflectional properties of compounds one has to take into consideration three main points: (a) how the compound components as simple words inflect; (b) under what constraints they inflect in each particular compound; and (c) how the inflection of compound components agree with each other.

Several methods were suggested for the formalization of this process. The more detailed description of these approaches is given in (Krstev, 2006). For the procession of Serbian compounds we have adopted the approach suggested in (Savary, 2005). One of the reasons that this approach has been chosen is that it relies on the same resources that are already used for the text processing. The other reason is that it is well suited for the highly inflected languages.

The approach that we have chosen is based on a new type of FSTs that rely on FSTs for simple word inflection, but are independent of them. This means that the compound FSTs deal only with the problems of compound inflections and leave all the peculiarities of simple word inflections to the standard FSTs.

The features of the new graphs can best be introduced wih one simple example. The FST in Figure 6. describes the inflectional properties of the compound *zvezda vodilja* 'guiding star', which consists of two nouns whose inflection agree in number and case. From this FST we see that it inflects compounds that consist of three constituents, and their names are $1, $2, and $3, respectively. In our example these three components would be: *zvezda*, ' ' (blank), *vodilja*. The second component is a separator that does not inflect; it is used as it is in all the inflected forms. The first and the third component are nouns that have four morphological categories: number (name Nb is given to that category), case (name Case), animatness (name Anim), and gender (name Gen). These two constituents inflect in the first two categories, and that is expressed by the usage of one equal sign after the name of the category. Following the equal sign is the name of the variable that receives subsequently all the possible values for the respective morphological category. For instance, for the category Nb the variable $n will receive values s (for singular), p (for plural), and w (for paukal). The first two constituents also agree in the first two categories, and that is expressed by the usage of the same variables for the same categories, those are $n and $c respectively. On the other hand, neither third not forth category inflect, and that is expressed by the usage of the double equal sign after the name of the category. They don't agree either, and that is why they use different variables, $a and $a1, $g and $g1, respectively. Since for these categories constituents does not inflect, these variables receives the values that respective categories have for corresponding constituents in the compound lemma.

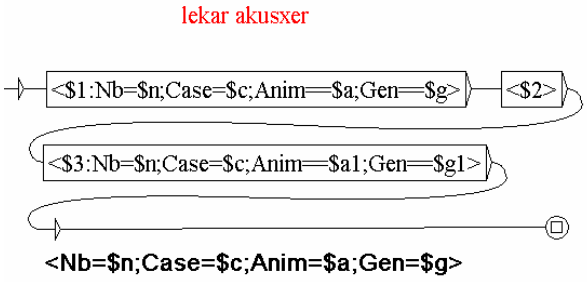


Figure 6. FST for the inflection of the compounds that consist of two nouns agreeing in two categories

This kind of FTS will produce as many DELACF dictionary entries as there are combinations of values of morphological categories listed in the FST's output. For the example from figure 6, variables $a and $g have fixed values, while variables $n and $c take all possible values for that categories, that is 7 and 3. The number of

produced entries, however, is not 21 but 16, since number paukal exist only or two cases (see Table 1). The output also shows that the values of the categories animatness and gender of the compound noun are inherited from the first noun constituent (the usage of the variables `$a` and `$g`, not `$a1` or `$g1` in the FST's output).

zvezda(zvezda.N600:fs1q) vodilja(vodilja.N600:fs1q)
zvezda vodilja:fs1q:fp2q
zvezde vodilje:fs2q:f2wq:f4wq:fp1q:fp4q:fp5q
zvezdi vodilji:fs3q:fs7q
zvezdu vodilju:fs4q
zvezdo vodiljo:fs5q
zvezdom vodiljom:fs6q
zvezdama vodiljama:fp3q:fp6q:fp7q

Table 1. Entry in the dictionary Delac for the lemma *zvezda vodilja*, and automatically produced inflected forms in Delacf format

The independence of the inflection of simple words and compounds can be illustrated by this same example. The same FST from Figure 6 can be used for *lekar akušer* 'obstetrician', although its constituents differ both in the values of unchangeable morphological categories and in the way the other categories inflect. This becomes obvious when looking at the Delac entry for *lekar akušer*:

lekar(lekar.N2:ms1v) akušer(akušer.N2:ms1v),N+Hum

The FST from the Figure 6 illustrates the basic features of this new type of FSTS. However, they can be used in much more versatile way, which will be illustrated by some examples in the following subsections.

3.1. Orthographic variances

Many compounds can have a few orthographic variants, especially concerning the usage of hyphen and/or blank. This is especially the case for Serbian which is not a strictly normalized language.

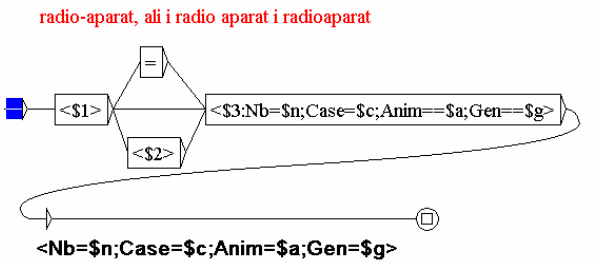


Figure 7. FST for the inflection of compounds with optional blank and hyphen

FST from Figure 7 inflects the compounds consisting of three constituents; the first two of them does not inflect, while the third inflect in number and case, and has the fixed values for the categories animatness and gender, inherited from the compound lemma. As this FST suggests, the second constituent can be copied as such in all the inflected forms, can be omitted, or replaced by blank. Two compounds inflected by this FST are *radio-aparat* 'radio set' and *akten-tašna* 'brief-case' whose entries in Delac dictionary would be:

radio-aparat(aparat.N1:ms1q),N+Art
akten-tašna(tašna.N660:fs1q),N+Art

As a result, all inflected forms in Delacf dictionary will have the same lemma, the one with a hyphen:

radio-aparatu,radio-aparatu.N+C+Art:ms3q
radioaparatu,radio-aparatu.N+C+Art:ms3q
radio aparatu,radio-aparatu.N+C+Art:ms3q

3.2. The omission of constituents

Some compound constituents are optional, that is they are not obligatory. Such is the case with *professor ruskog jezika* 'professor of Russian language' that is often used in a shortened version *professor ruskog*. Its Delac entry is:

profesor(profesor.N2:ms1v) ruskog jezika,N+C+Hum

As before, this would be lemma for both full and shortened form as illustrated by these few Delacf entries automatically produced using the FST from Figure 8:

profesoru ruskog jezika,profesor ruskog jezika.N:ms3q
profesoru ruskog, profesor ruskog jezika.N:ms3q

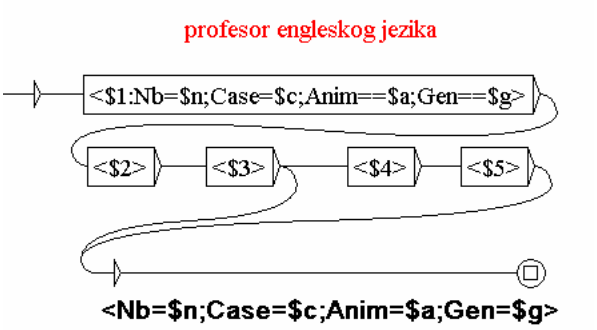


Figure 8. FST for the inflection of the compounds begin with a noun followed by four constituents that does not change, the last two of which can be omitted.

3.3. The order of constituents

In some cases the order of constituents in a compound can change, for instance *muva ce-ce* or *ce-ce muva* 'tsetse fly'. More interesting is the example of compound adjectives that are composed of two adjectives connected by a hyphen in which the adjective components can be reversed, like in *ekonomsko-finansijski* and *finansijsko-ekonomski* 'economic and financial'. In the adjectives of this type the last constituent inflects, while the first constituent is fixed in the neuter gender singular number form. The FST in Figure 9 inflects this type of compounds. The upper path is straightforward: it states that the first two constituents does not inflect while the third constituent inflects in number, case, gender, animatness, and definiteness. The only category for which it does not inflect is comparison, as compound adjectives of this type do not have comparative and superlative form.

In the lower path of the FST in Figure 9 we see that the first and the third constituent have changed order, while the last constituent inflects in the same way as in the upper path. The first constituent in the lower path, however, is not in the form it should be as in lemma it is usually in the masculine gender singular number. Therefore, the values of morphological categories have to

be assigned as needed. For instance, the entry in Delac for *ekonomsko-finansijski* is:

ekonomsko(ekonomski.A2:aens1g)-
finansijski(finansijski.A2:adms1g),A+C+PosQ

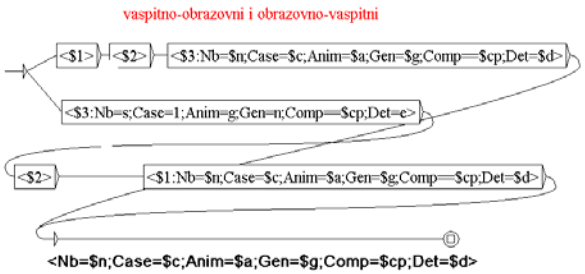


Figure 9. FST for the inflection of the compounds that consist of two nouns agreeing in two categories

Two Delacf entries automatically produced from this Delac entry using the FST from Figure 9 would be:

ekonomsko-finansijskoj,ekonomsko-
finansijski.A+C+PosQ:s3gfae
finansijsko-ekonomskoj,ekonomsko-finansijski.A+
+PosQ:s3gfae

3.4. Conditional paths

Many compounds in Serbian have the form <A> <N> and it is usually said that in this case adjective and noun agree in number, gender, and case. In one particular case they also agree in animatness – namely, the form of the adjective depends on the animatness of the masculine gender nouns when in the accusative case singular. So the category animatness for adjectives also inflect and can take three values: v for animated, q for non-animated, and g for don't care. This last category has been introduced since the animatness of the nouns is for most of the cases of no consequence for the inflection of the adjective. In the FST in Figure 10 the lower path is taken for the masculine gender noun in accusative case singular, and in that case the animatness is inherited form the noun while adjectives has to agree with it (the use of the same variable $a). For the generation of all other inflective forms the upper path is taken, the animatness is again inherited form the noun, but adjectives do not to agree with it (the value of category Anim for the adjective is g and it cannot agree with the value of the variable $a, since for nouns this value is either v or q).

For instance, the entries in Delac for *redovni professor* 'full-time professor' and *prljav veš* 'dirty laundry' are:

redovni(redovni.A2:adms1g)
profesor(profesor.N2:ms1v),N +Hum
prljav(prljav.A17:akms1g) veš(veš.N1001:ms1q),N

They can be inflected using the same FST from Figure 10, and for accusative case singular the following entries would be generated (it can be seen that the form of the adjectives are different due to the different animatness of the nouns):

redovnog profesora,redovni profesor.N+Hum:ms4v
prljavi veš,prljav veš.N:ms4q

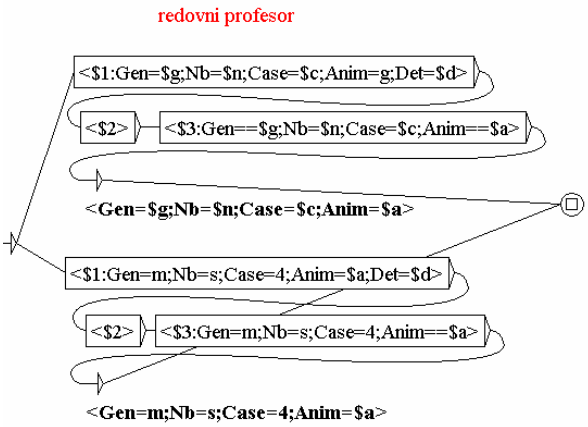


Figure 10. FST for the inflection of the compounds that consist of an adjective followed by a noun

3.5. Multiple outputs

In Serbian there are simple words for which multiple values can be assigned for various categories. For instance, *gar* 'carbon black' can be masculine and feminine gender. This is even more the case for the compounds. Here we will give one particularly complex example: *Trinidad i Tobago* 'Trinidad and Tobago'. When considering the inflectional properties of this name one has to establish (a) its gender, (b) its number, (c) which constituents inflect; and (d) do the constituents agree and how. Since this information is not to be found in any grammar book a small "Trinidad and Tobago" corpus was assembled. Fortunately, Trinidad and Tobago has participated in the Football World Cup finals in 2006 so this small country has been mentioned quite frequently on the Serbian web sites. The analysis of the corpus occurrences shows that the gender is always masculine (both *Trinidad* and *Tobago* are masculine). The number is more often singular, but in a few cases also plural. Usually both Trinidad and Tobago inflect, but sometimes Trinidad does not. The examples for this latter case are rare; however, there are evidences for all cases (see Figure 11).

a) do sada **je** i Trinidad i Tobago **igrao** ofanzivnije od nas Trinidad i Tobago **su postali** nezavisna država u okviru Britanskog Komonvelta...
b) Selektor Trinidad**a** i Tobaga (je) srećan ...meč B grupe između Engleske i Trinidad i Tobaga...
c) ... već je poslednji put viđen u Trinidad**u** i Tobagu... Otkako je grupa kupila železaru u Trinidad i Tobagu...
d) Bahrein će igrati sa Trinidad**om** i Tobagom u plej-ofu... Odbrambeni fudbaler propustio je meč koji je Engleska igrala sa Trinidad i Tobagom

Figure 11. The examples form 'Trinidad and Tobago" corpus: a) number of the compound; b) compound in genitive case; c) locative case; d) instrumental case.

FST from Figure 12 shows that there are two outputs, one that establishes the compound *Trinidad i Tobago* as singular, and the other as plural. There are also two paths: the upper path generates the forms where both *Trinidad* and *Tobago* inflect, the lower part generates the form in which only *Tobago*. The lower path uses only one output, since in this case the compound can only be singular. As a

result, the FST form Figure 12 would generate three morphologically different forms for instrumental case:

Trinidad i Tobagom,Trinidad i Tobago.N+Top:ms6q
Trinidadom i Tobagom,Trinidad i Tobago.NTop:mp6q
Trinidadom i Tobagom,Trinidad i Tobago.NTop:ms6q

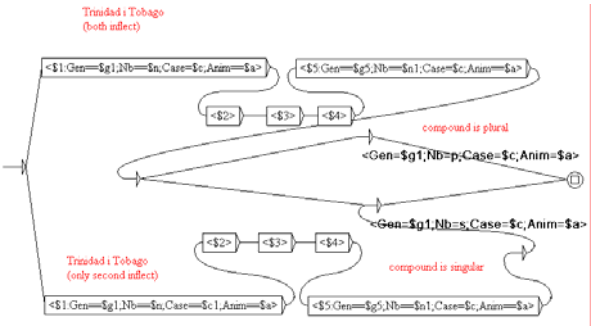


Figure 12. FST for *Trinidad i Tobago*

4. The Examples of Usage

The main contribution of the presented approach is that once the text has been tagged the simple words and compounds are treated equally in all subsequent text processing applications, such as formulation of queries or development of syntactic grammars.

Consider the example of money amounts that are in newspapers and agency news usually expressed by a numeral followed by the name of the currency. The numeral, however, can be simple or compound, expressed by digits, alphabetic characters, or combination of both. The simple query formulated by the graph represented at the top of the Figure 13 retrieves from the text all numerals – the syntactic category <NUM> - followed by some sequence recognized by the sub-graph valute 'currencies'. This sub-graph recognizes in Serbian text all the major world currencies. It takes into consideration that when preceded by numerals the currencies have to be in certain grammatical forms – either genitive plural or genitive paukal. Since the syntactic category <NUM> is attached both to the simple word numerals found in e-dictionaries and to compound numerals recognized by FSTs described in subsection 2.2 the money amounts can be correctly retrieved, as shown in concordance lines in Figure 14.

The FST in Figure 13 is oversimplified – it would be suitable for information retrieval cases, since it would retrieve even the grammatically incorrect usages (for instance, the incorrect usage of paukal). For correct syntactic modeling more complex FSTs are produced and used.

, a od te sume oko 100 milijardi jena (900 miliona dolara) predvi
u Sloveniji košta 159,3 tolara, ili 0,66 dolarski centi. Vlad
mesecu je iznosila 7.257 dinara, plata medicinske sestre u domu
partneru iznosi 250 miliona američkih dolara. Kosovo: 11
iznosi milijardu 96 miliona i 275 hiljada dinara podeljen
biće izdvojeno milion i 500 hilxada evra. Jugoslovensko
vom vecxom od pet milijardi kuna povecxale su svoj udeo

Figure 14. Concordance lines for money amounts

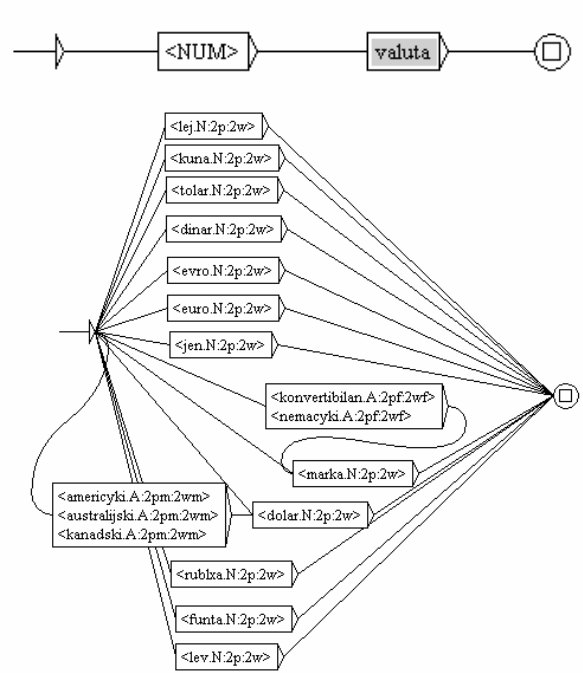


Figure 13. The simple graph for the recognition of money amounts

5. References

Courtois, B., Silberztein, M. (eds.), 1990. Dictionnaires électroniques du français, Langue française 87, Paris: Larousse

Krstev, C., Vitas, D., Savary, A. 2006. Prerequisites for a Ccomprehensive Dictionary of Serbian Compounds, in Proceedings of 5th International Conference FinTAL, August 23-25, 2006, Turku, Finland, pp. 552-563.

Laporte, E. (2003). *The RELEX Network* (http://infolingu.univ-mlv.fr/ - link 'Reseau International')

Maurel, D. and Guenthner, F., 2005. *Automata and Dictionaries*, Texts in Computing Seies, King's Colleague.

Paumier, S. (2002): *Manuel d'utilisation du logiciel Unitex.* IGM, Université de Marne-la-Vallée. http://www-igm.univ-mlv.fr/~unitex/manuelunitex.pdf

Savary, A., 2005. Towards a Formalism for the Computational Morphology of Multi-Word Units, in Proceedings of 2nd Language & Technology Conference, April 21-23, 2005, Poznan, Poland, ed. Zygmunt Vetulani, pp. 305-309

Silberztein, M. 2004. INTEX Manual, v. 4.33. (http://intex.univ-fcomte.fr/downloads/Manual.pdf)

Silberztein, M., 2005. NooJ's Dictionaries. In the Proceedings of LTC 2005, Poznan University

Vitas, D., Krstev, C., 2005, Regular derivation and synonymy in an e-dictionary of Serbian, in *Archives of Control Sciences*, Volume 15(LI), No. 3, pp. 469-480, Committee of Automation and Robotics, Polish Academy of Sciences.

Vitas, D., Krstev, C., Obradović, I., Popović, Lj., Pavlović-Lažetić, g., 2003. An Processing Serbian Written Texts: An Overview of Resources and Basic Tools ", in Workshop on Balkan Language Resources and Tools, 21 Novembar 2003, Thessaloniki, Greece, eds, S. Piperidis and V. Karkaletsis, pp. 97-104

The Role of the Lexicon in Lexical-Functional Grammar - Example on Croatian

Sanja Seljan

University of Zagreb – Faculty of Humanities and Social Sciences
Department of Information Sciences
Ivana Lučića, 10 000 Zagreb, Croatia
sseljan@ffzg.hr

Abstract

The LFG model is based on an enriched lexicon, which contains associations between grammatical functions and their arguments, enabling decomposition on characteristic features and is suitable for formal description of languages with a rich morphological structure and a relatively free word order. Rejecting syntactic movement of constituents as the mechanism for realization of the surface syntactic structure, it is based on the idea of grammatical functions presented in the lexicon. In the paper the role of the lexicon is presented, as well as its close interaction with constituent and functional structures. In a formalization of Croatian sentence structure based on LFG, a new type of lexicon organization has been proposed, containing grammatical functions relating to argument structure, new features and constraints. Theoretical models are verified through educational version of the LFG model.

Vloga leksikona pri formalnem modelu LFG – primer hrvaščine

Model LFG temelji na obogatenem leksikonu, ki vsebuje povezave med slovničnimi vlogami in argumenti, kar omogoča razčlembo na značilke, ki se kažejo kot ustrezne za formalni opis jezikov z bogatim oblikoslovnim ustrojem in relativno prostim besednim redom. Model zavrača skladenjske pretvorbe konstituentov kot načina za uresničitev površinske skladenjske zgradbe in za osnovo vzame idejo o slovničnih funkcijah, zajetih v leksikonu. V članku je predstavljena vloga leksikona kot tudi tesnih povezav med gradniki in funkcijsko zgradbo. Pri postopku formalizacije hrvaških stavkov je bil predlagan nov tip organizacije leksikona, ki vsebuje slovnične vloge povezane z zgradbo argumentov, novimi značilkami in omejitvami. Teoretični modeli so preverjeni z izobraževalno verzijo modela LFG.

1. Introduction

As the name says by itself, this formal model is based on the idea of representing grammatical functions in the lexicon. In Bresnan (1982) grammatical functions are defined as universal syntactic primitives of the grammar classified according to two main criteria: subcategorization principle and semantic restriction principle.

Starting form the idea that grammatical functions, such as subject and object, exist in all or most of natural languages, the LFG model rejects the syntactic movement of constituents and the surface syntactic realization, but accepts the idea of grammatical functions and enriched lexical component. Grammatical functions are situated as an interface between the lexicon and the syntax.

Since the LFG model integrates linguistic knowledge with computer application and aims to be suitable for description of highly structured languages, as well as for languages with free word order, this model has been applied for various linguistic phenomena in various languages (English, German, French, Italian, Dutch, Icelandic, Russian, Warlpiri, Bantu, Greek, Chinese, Icelandic, etc.), according (Abeillé, 1993).

2. Role of the lexicon

In the Lexical-Functional grammar (LFG), the lexical process determines multiple set of associations of arguments (Agent, Theme, etc.) with grammatical functions (Subject, Object, etc.), according Neidle (1994).

A surface structure is realized by constituent (c-) structure, enriched by the lexical component which exists simultaneously with functional (f-) structure that integrates information from the c-structure and from the lexicon.

As the c-structure reflects the surface syntactic structure, it varies across languages, while the f-structure tends to be universal when describing the same language phenomena through different languages.

J. Bresnan (1982) gives arguments in favour of the lexical account, supporting relations between grammatical functions and arguments, rather than syntactic movements.

In languages with relatively free word order, like Croatian, as one of the Slavic languages that has rich morphological system, the possibility to formalize language phenomena through grammatical functions in lexicon that can have various positions in the sentence obtains preference in favor of this formal model.

As the LFG model is based on enriched lexical component containing grammatical functions, enabling decomposition of categories on characteristic features, incorporation of contextual elements and adding various constraints, this model tends to be suitable for formal description of the Croatian sentences having rich morphological structure and relatively free word order.

3. Lexical entry

Containing grammatical relations between predicate-argument structure, grammatical functions and characteristic features, the lexicon plays in the LFG model an important role.

The lexicon contains following types of information:

- form of the item (*on, oni, djeca, čitaju, knjigu*, etc.)
- part of speech (N, V, Adj, etc.)
- functional schemata containing information about meaning inside of quotes ‘ ‘ and grammatical functions (Subj, Obj, etc.) interrelated with thematic roles (Agent, Theme, etc.)
- other characteristic features (attribute-value pairs)

čita V [eng. *is reading*]
(↑PRED)='čitati<(SUBJ) (SUBJ, OBJ)
(SUBJ, IOBJ, OBJ)>'
(↑NUM) = SG
(↑PRS) = 3
(↑TNS) = PRES
knjigu N [eng. *book*]
(↑PRED)= 'zadaća'
↑GND) = FEM
(↑NUM) = SG
(↑CASE)= ACC

These type of equations called constituent equations are incorporated into functional structure, contrary to the constraining equations that serve only to verify the truth (for e.g. to verify agreement between demonstratives or adjectives with noun inside of NP). While constituting equations are incorporated into f-structure, constraining equations in the following example (marked with =c) serve as the control mechanism and ensure the proper well-formeness of constructions respecting the constraint (e.g. agreement with subject in number and person).

čita V [eng. *is reading*]
(↑PRED)='čitati<(SUBJ, IOBJ, OBJ)>'
(↑SUB NUM) $=_C$ SG
(↑SUB PRS) $=_C$ 3
(↑TNS) = PRES

Constraining equations could be used in the process of agreement which is in Croatian of the considerable importance (e.g. agreement between subject and verb in person and number, in complex tenses, inside of NP in case, number, gender, then between subject and past participle, etc.).

3.1. Lexical entry related to c- and f-structures

In the LFG model the sentence is represented by three interrelated levels of representation: lexical structure, constituent (c-) structure and functional (f-) structure, which exist simultaneously (Kaplan & Bresnan, 1982), although other levels of representation have been afterwards added, such as argument (a-) structure (Bresnan & Kanerva, 1989) and morphological (m-) structure (Frank, 2000; Kaplan, 2000).

As grammatical functions are associated by the mechanism of annotation of phrase structure rules with lexical items and its syntactic positions, they mediate between the lexicon and the syntax, i.e. c-structure.

Constituent structure reflects the superficial syntactic structure and encodes linear order, hierarchy and syntactic categories of constituents (in the form of context-free rules enriched with functional annotations or in the form of the annotated tree). This structure varies from one language to another. C-structure corresponds to the superficial phrase structure and works closely with an enriched lexical component.

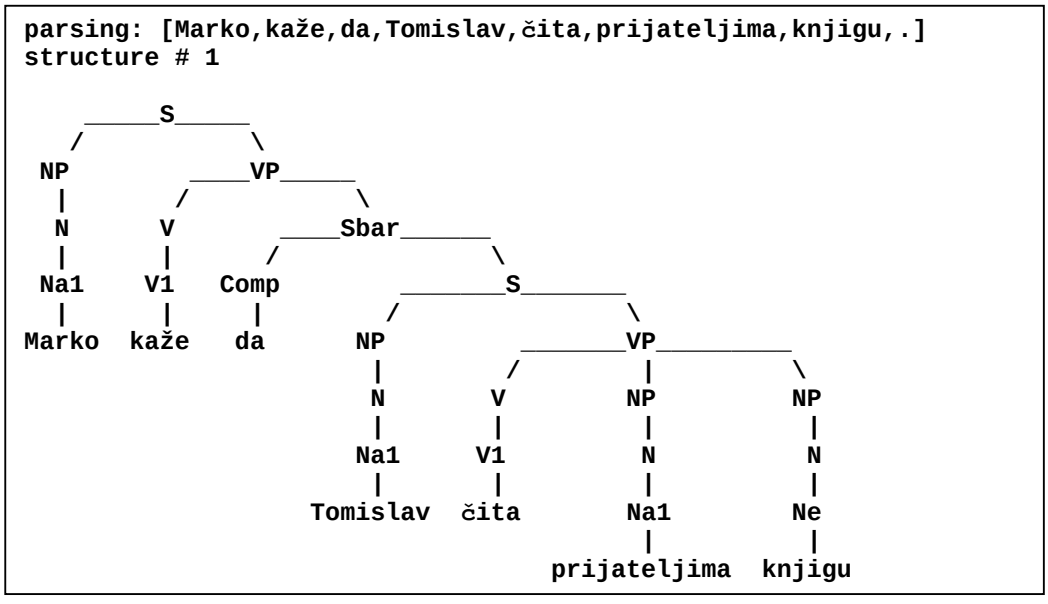


Figure 1. C-structure for the sentence
Marko – kaže - da – Tomislav – čita – prijateljima - knjigu
[eng. *Marc - says - that - Tom - is reading – to friends - a book.*]

C-structure exists simultaneously with functional f-structure. Information contained in the lexicon is integrated together with structural information into functional (f-) structure and presented in the form of matrix, as presented in the following example:

```
PRED  Kazati<SUBJ,COMP>
  TNS  PRES
  SUBJ  PRED  Marko
        NUM  SG
        PRS  3
        GND  MASC
        CASE  NOM
COMP PRED  Citati<SUBJ,IOBJ,OBJ>
        TNS  PRES
        SUBJ  PRED  Tomislav
              NUM  SG
              PRS  3
              GND  MASC
              CASE  NOM
        IOBJ  PRED  Prijatelj
              NUM  PL
              GND  MASC
              CASE  DAT
        OBJ   PRED  Knjiga
              NUM  SG
              GND  FEM
              CASE  ACC
```

Figure 2. F-structure for the sentence
Marko kaže da Tomislav čita prijateljima knjigu
[eng. *Marc - says - that - Tom - is reading – to friends - a book.*]

3.2. Lexical entry and argument structure

As each lexical entry contains grammatical relation between argument structure (thematic or non-thematic roles) and grammatical functions, the principle of Function-Argument Biuniqueness requires that each argument can be associated with the unique grammatical function, i.e. that no grammatical function can occur more than once.

'udariti (agent, theme)' ← argument structure
(SUBJ , OBJ) ← gram. functions

Lexical entry of the verb *udariti* [eng. *hit*] for the unit might look as follows:

(↑PRED)= ' udariti < SUBJ , OBJ > '

indicating that the PRED feature has as its value the meaning of the verb which is a two-place predicate.

The sucategorization frame of the verb *kuhati* [eng. *cook*] in active and *kuhan* [eng. *cooked*] in passive forms could be the following:

(↑PRED)= ' kuhati < SUBJ , OBJ > '
(↑PRED)= ' kuhan < OBJ, ∅ > '

If the grammatical function is not directly associated to the logical argument (e.g. when the agent is not alive or in open complements with functional control principle), the subject is called to be non-thematic. In the sentence *Treba pričekati* [eng. *It is necessary to wait*, fr. *Il faut attendre*] in Croatian the subject doesn't have even the form, as in French *Il* or in English *It*.

Treba pričekati.
(↑Pred)= 'trebati <(XComp) > (Subj)'
(↑Subj Form)=$_c$ ∅
fr. *Il faut attendre.*
(↑Pred)='falloir <(Xcomp)> Subj'
(↑Subj Form)= il
eng. *There are some books.*
(↑Pred)='be<Obj> Subj'
(↑Subj Form)=$_c$ there

In the following sentence *On ju smatra dobrom prijateljicom* [eng. *He considers her a good friend*] the object of the verb *smatrati* [eng. *consider*] is in the same time the logical subject of the open complement – *dobrom prijateljicom* [eng. *a good friend*], which is indicated by placing the function Obj outside the angled brackets.

On ju smatra dobrom prijateljicom
[eng. *He considers her a good friend*]
smatra, V [eng. *considers*]
(↑PRED)= 'smatrati <(SUBJ, XCOMP)> (Obj)'
(↑OBJ)=(↑XCOMP SUBJ)

In the Croatian it is also necessary to make an agreement between the subject *on* [eng. *he*] and the verb *smatra* [eng. *considers*] in person and number, between the object *ju* [eng. *her*] and *dobrom prijateljicom* [eng. *a good friend*] in gender, although they are in different cases (*ju* in accusative and *dobrom prijateljicom* in instrumental). Therefore, *dobrom prijateljicom* [eng. *a good friend*] is treated as an XComp - the open complement whose subject is controlled grammatically by the Object function.

4. Formalization process

Since this work represents attempt to describe formally subset of Croatian natural language sentences at different levels and to verify computationally the theoretical models, informatics and linguistic approaches are closely interrelated. The presented approach distinguishes from the classic Croatian grammars, which is conditioned by the LFG model itself, by the program which is educational by purpose, by the specific language demands, supporting relatively simple morphological component Although there are much better solutions for Croatian, especially at morphological level, the advantage of this model represents an attempt to describe the sentence at different levels.

Very rich inflectional and derivational morphological system enables relatively free word order. For the purpose of formal description, a recursive binary structure has been proposed. The concatenate morphology, different form traditional grammar, allows association of the first part of the word, marked as word category, with the last part, i.e. with endings and, eventually, with prefixes. Endings are marked with standard characteristic features, such as case, number and gender for NP, and with special features introduced because of formal distinction between cases having the same form, but different meanings.

Since the form of the word (morphological component) can not be considered separately from the structure and word relations (syntax), neither independently from the context (semantics),

morphological component is strongly related in the Croatian to the syntax and to the semantics.

In order to formalize certain language segments, the following steps were undertaken:

- Definition of syntactic groups (NP, VP, PP, AP, AdvP)
  - As formal constituents (noun, adjective, pronoun, etc.)
  - As functional constituents (determiners, premodifiers, head, postmodifiers)
- Definition of parts of the speech and subgroups
- Definition of attribute-value pairs
- Lexicon organization (part of speech, paradigms, constraints)
- Generative rules and constraints

4.1. Syntactic groups, subgroups, features

In the Croatian words are basically divided into changeable (that change the form through the paradigms) and unchangeable (that do not change the form through the paradigm). Changeable types of words are those that change through cases (nouns, adjectives, pronouns, noun numbers behaving like nouns, adjective numbers behaving like adjectives) and verbs that change through persons. All changeable types of words are described by characteristic features. Unchangeable types of words are adverbs, number adverbs, prepositions, exclamations and particles. Some Croatian authors consider adverbs unchangeable type of word (Raguž, 1997; Batnožić, Ranilović, Silić, 1996; Anić, 1994), some authors consider it as partly changeable (Barić et al., 1979:65), and some consider the adverb as changeable type of word (Tadić, 1994.) with regular changes in gradation.

Although the basic division into parts of speech has been retained, some formal distinctions, subgroups and additional features have been introduced, because of formal constraints.

1) Determiners are formally subdivided into indefinite PDet: *neki, svaki* [eng. *some, every,* etc.], referential Det: *ovaj, taj, onaj* [eng. *this, that*], relative DetRel: *tko, što, koji* [eng. *who, what, whose,* etc.], quantifiers DetQ - number adjectives: *prvi, prva, prvo* [eng. *first* in 3 genders] and possessives Dposs - possessive adjectives, possessive pronouns, possessive and reflexive pronoun. Distinction has been introduced: determiners are not recursive, opposed to ex. modifiers, but subgroups of determiners can be combined among themselves.
2) Premodifiers in NP are recursive (nouns, adjectives, past participles).
3) In the Croatian formal model, past participle has two forms:

a) past participle in active form formally marked as Adj_Part (Adjective Participle), having different suffixes denoting gender and number as *čitao-čitala-čitalo*, eg. *on je čitao, ona je čitala, ono je čitalo* [eng. *he/she/it was reading*]

b) past participle in the passive form formally described as Adj_Part, PASS +, ASP FIN/PROG having also different suffixes denoting 3 genders , eg. *čitan, čitana, čitano* [eng. *was read*], containing also features of progressive (*čitan*) or finite aspect (*pročitan*). The basic form of the word is related to the paradigm number, giving information on gender and number

4) Numbers are subdivided into following groups: noun numbers behaving like nouns, eg. *stotina, tisuća* [eng. *hundred, thousand*], adjective numbers that agree with noun in gender, number and case, eg. *prvi, prva, prvo* [eng. *first*], adverb nouns that do not change and behave like adverbs (cardinal numbers)
5) Possessive and reflexive pronouns - *se, sebe* [eng. *himself, herself, itself*] are defined as clitics Cl, placed inside of VP, but distinguishing strong and weak form Str= +/ -.
6) Demonstrative pronouns marked by proximity PROX= 1/ 2/ 3 denoting 3 distances
7) Interrogative pronouns are marked with QU=+
8) Relative pronouns are marked with REL=+ and ANI=+ to distinguish the same form *koji* [eng. *which*] between animate in nominative case and inanimate in accusative
9) Collective nouns have characteristic feature COLL=+
10) Instrumental case is marked with SOC=+ and THG=+ in order to distinguish between dative and instrumental when having the same form.
11) Past participles are formally divided into:

a) past participle in active form (Adj-Part) that can have prefix marking finite aspect (ASP FIN) or suffix marking gender and number, which must agree with subject

b) past participle in passive form, finite aspect (Adj-Part, PASS +, ASP FIN)

c) past participle in progressive aspect Adj-Part, PASS +, ASP PROG

12) In the formal description of preterit and future, auxiliaries subcategorize XComp function, and are marked by tense, number, person, optionally as negative or strong forms. Although the author adopts the m-structure, the older version of formal composite tenses was adopted because of formal reasons.
13) Adverbs in this work are treated as unchangeable type of words, marked with degree level (Deg = pos/ com/ sup).

4.2. Case-marking

One of the central questions for representing the Croatian language is case-marking and agreement. The term ‘case’ is used in LFG in a traditional sense, in order to describe use of inflections, which in the Croatian encode syntactic and semantic relations. In LFG syntactic case is associated with specific grammatical function and a morphological form that comes from the lexicon with the suitable case inflection. Case-marked forms are generated in the lexicon. The suitable case form is inserted into c-structure and then appropriate use verified in the f-structure.

In Croatian there are seven cases (nominative, genitive, dative, accusative, vocative, locative, instrumental), some of them having the same written form (genitive - accusative, dative - locative, dative - instrumental). In the formalization process some additional constraints have been introduced for the purpose of distinguishing homographic cases.

4.3. Lexicon organization

Word composition in the highly flective languages with rich morphological system represents one of the dominant questions in creation of the electronic lexicon. What are smaller parts, their meanings, how do they combine and differences regarding to traditional grammar in formal description. The presented method combines linguistic and informatic approaches.

In the classical grammar, next to the lexical morpheme, eg. *prijatelj* [eng. *friend*]several grammatical morphemes (*-ic, -a*) denoting gender feminine, number singular and case nominative can be added. The morpheme *-ic* is not always a morpheme, eg. m*ajic-a* [eng. *T-shirt*) but can be part of the stem. The same morpheme *-a* and the same lexical unit *majica* can also have also another meaning (gender feminine, number plural, case genitive). Differences in the meaning between nominative singular and genitive plural are reflected in the grammatical functions.

What is proposed is a delimitation inside of words, which are to be divided in two parts: the first and the last part, or beginning and end. The last part would be formalized in the sense of declination endings and the first part as part of speech. Therefore, morphemes are not used in the traditional sense in this formal analysis and lexical units are formally delimited in the following way:

- Flective unit – stem, marked as part of speech (which can be subdivided)
- Paradigms containing prefixes or suffixes with characteristic features
- Irregular forms that are marked separately, not related to the paradigm

In this formal model there are 135 lemmas related to 260 allomorph basic forms. Using the concatenation principle by adding suffixes of paradigms to stems, there are around 1.370 different generated forms (e.g. *knjig* and *knjiz* represent two allomorph basic forms – stems of one lemma *knjiga* [eng. *book*], generating 14 forms, some which are the same, differentiating in case and in number.

While some consider the paradigm of endings to be marginal, but diagnostically relevant, others consider it an important morphological phenomena. Vincent and Börjars (1996) indicate that if certain morphological system requires to be analyzed in terms of morphosyntactic representations which consist of feature bundles rather than X'-projections and if there is inevitable continuity between morphology and syntax, that in consequence the best model of the morphology-syntax interface would be featural rather than configurational.

5. Computational model

Theoretical models are verified through the computational model using LFGW code (Andrews, 1991.). LFGW is a basic LFG system adapted for doing small (homework-assignment sized) grammar fragments. It includes simple morphology and lexical inheritance, as well as an `error-tolerant parsing' facility (to help find mistakes in the grammar), but does not implement any of the advanced features of recent LFG theories (functional uncertainty, anaphoric binding), and also lacks a workable treatment of long-distance dependencies. LFGW is licensed for free use for any educational or noncommercial purpose. This educational version of the LFG model is used for description of certain linguistic phenomena of Croatian language (case-marking and some agreement phenomena).

The program gives for the sentence in the Croatian two structures:

a) Constituent (c-) structure using tree form and defining the surface level with parts of speech and cases

b) Functional (f-) structure in the matrix form unifying information form the constituent structure and from the lexicon, which has to satisfy principles of uniqueness, coherence and completeness.

Sentences are firstly generated by syntactic rules, passing then constraining tests introduced in the lexicon or added in generative rules. The program surely does not represent the best solution (especially on the morphological level) but one possible model of sentence analysis, where special attention is given to the lexicon.

6. Conclusion

Being context-sensitive, non-transformational grammar using constraints and unification principle, the LFG formal model aims to be suitable for description of various linguistic phenomena in various languages, and therefore in Croatian.

As the LFG model is based on grammatical functions and enriched lexical component, permitting also decomposition on characteristic features and introduction of new constraints, which are in meta-language reflected as attribute-value pairs, it has shown to be adequate for description of some language phenomena of the Croatian, like case-marking and agreement. Lexical component, containing information about meaning, characteristic features and subcategorization principles, and closely relating to constituent, argument and functional structures, enable description of the language through lexical and functional component. Having also possibility to add new features and constraints, characteristic for the specific language, this formal model becomes suitable for description of various languages, including Croatian with rich morphological system and relatively free word order. The LFG could be seen as the bridge between linguistics and informatics helping us to better understand our proper language in order to approach the theoretical linguistic models and practical application.

7. References

Abeillé, A., 1993. Les nouvelles syntaxes: Grammaires d'unification et analyse du Français. Paris: Armand Colin.

Andrews, A., 1991.LFGW System. University of Brisbane (http://www-csli.stanford.edu/~andrews/lfgw.html)

Anić, V., 1994. Rječnik hrvatskoga jezika, 2nd ed. Zagreb: Novi Liber.

Austin, P., 2001. Lexical-Functional Grammar. //International Encyclopedia of the Social and Behavioural Sciences. Smelser, N.J., Baltes, P. (eds). Elsevier: 8748 – 8754. (http://www.linguistics.unimelb.edu.au/contact/staff/peter/Elsevier.pdf)

Barić E. et al., 1979. Priručna gramatika hrvatskoga književnog jezika. Zagreb:Školska knjiga.

Batnožić, S., Ranilović, B., Silić, J., 1996. Hrvatski računalni pravopis: gramatičko-pravopisni računalni vodič: spelling-checker. Zagreb: SYS, Matica hrvatska.

Bresnan, J., 1982. The Mental Representation of Grammatical Relationsl. MIT Press, Cambridge, Massachusetts.

Bresnan, J, 2001. Lexical-Functional Syntax. Blackwell Publishers.

Briffault, X., Chibout, K., Sabah, G., Vapillon J., 1997. A Linguistic Engineering Environment using LFG (Lexical Functional Grammar) and CG (Conceptual Graphs). Proceedings of the LFG97 Conference. CSLI Online Publications. http://csli-publications.stanford. edu/LFG/2/briffault/briffault-lfg97.html

Butt, M. The Treatment of Tense. Proceedings of the LFG01 Conference. CSLI Online Publications. http://csli-publications.stanford.edu/LFG/6/ lfg01butt.pdf

Butt, M., Dipper, S., Frank, A., Holloway King, T., 1999. Writing Large-Scale Parallel Grammars for English, French and German. Proceedings of the LFG99 Conference. CSLI Online Pub. (ftp://ftp.ims.uni-stuttgart.de/pub/users/dipper/ papers/ lfg99.pdf)

Chomsky, N., 1965. Aspects of the Theory of Syntax. Cambridge: MIT Press.

Chomsky, N., 1972. Syntactic Structures. Paris: Mouton.

EAGLES, 1996. Synopsis and comparison of morphosyntactic phenomena encoded in lexicons and corpora: A common proposal and applications to european languages. Technical Report EAT-LWG-Morphsyn, ILC-CNR, Pisa. (http://www.ilc.pi.cnr.it/ EAGLES96/morphosyn/morphosyn.html)

Falk, Y., 2001. Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax. Lecture Notes No 126. Stanford: CSLI.

Frank, A., 2000. Syntax and Morphology of Tense in LFG. Proceedings of the LFG00 Conference. http://www.xrce.xerox.com/people/frank/papers.html

Kaplan, R. M., 1996. The Formal architecture of Lexical-Functional Grammar./ Dalrymple, Kaplan, Zaenen (eds). Formal Issues in Lexical-Functional Grammar, pp.7-27. Stanford: Center fo the Study of Language and Information – CSLI, 1996.

Kaplan, R.M., Bresnan, J., 1995. Lexical-Functional Grammar: A Formal System for Grammatical Representation. Bresnan, J. (ed.) The Mental Representation of Grammatical Relations. MIT Press. Reprint: Dalrymple, M., Kaplan, R.M., Maxwell III, J. T., Zaenen, A. (eds.) Formal Issues in Lexical-Functional Grammar. Stanford: CSLI: 29-130.

Kaplan, R.M., Bresnan, J., 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation: 173-281.

Kaplan, R.M., Newman, P.S., 1997. Lexical Resourde Reconciliation in the Xerox Linguistic Environment. (http://acl.ldc.upenn.edu/W/W97/W97-1508.pdf)

Kaplan, R.M, Butt, M., 2002. The Morphology-Syntax Interface in LFG. (Abstract) Proceedings of the LFG02 Conference (http://csli-publications. stanford.edu/ LFG/7/lfg02kaplanbutt-abs.html)

King, T. H., 1995. Configuring Topic and Focus in Russian. Stanford: Center for the Study of Language and Information CSLI.

Neidle, C., 1994. Lexical-Functional Grammar. The Encyclopedia of Language and Linguistics. New York, Pergamon Press: 2147-2153. Reprinted: K. Brown, J. M. (eds.)., 1996. Concise Encyclopedia of Syntactic Theories. Oxford: Elsevier.

Raguž, D., 1997. Praktična hrvatska gramatika. Zagreb: Medicinska naklada, 1997.

Rosen, V., Zaenen, A., 1999. Grammar Writing in LFG: Introduction. Proceedings of the LFG99 Conference. CSLI Online Pub. (http://www2.parc.com/istl/ groups /nltt/)

Seljan, S., 2003. Lexical-Functional Grammar of the Croatian Language: Theoretical and Practical Models. Doctoral thesis. University of Zagreb.

Tadić, M., 1994. Računalna obrada morfologije hrvatskoga književnog jezika. Disertacija. Sveučilište u Zagrebu. (http://www.hnk.ffzg.hr/txts/mt-dr-le.pdf)

Vincent, N., Börjars, K., 1996. Suppletion and syntactic theory. // Proceedings of the LFG96 Conference. CSLI Online Publications. (http://lings.ln.man.ac.uk/ Info/staff/ KEB/Papers/Grenoble/Grenoble.html)

Wescoat, M.T. Practical Instructions for Working with the Formalisms of Lexical Functional Grammar. Online University of Essex. (http://www.fb10.uni-bremen.de/ linguistik/khwagner/lfg/pdf/wescoat.pdf)

Mining actions from reports on flood

Luboš Popelínský, Jan Blaťák

Knowledge Discovery Lab
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
{popel,xblatak}@fi.muni.cz

Abstract

This paper focuses on mining in short reports that describe a situation in a given area and actions performed as reaction to that situation. Such texts are frequent in crisis management in situations like earthquake, fire or flood. For further analysis it is necessary to filter the relevant pieces of text. We found that common machine learning algorithms fail for filtering such sentences. We describe a novel method based on inductive logic programming which yields in high precision and recall. This method has been successfully used for analysis of reports on flood in Central Europe in 2002. We also discuss different domain knowledge and also various natural language processing tools that we used for preprocessing the documents.

Učinki rudarjenja po poročilih o poplavah

Članek se osredotoča na rudarjenje po dokumentih, ki opisujejo razmere v določenem območju in delovanje kot posledico tovrstnih razmer. Taka besedila so pogosta v kriznem menedžmentu, v razmerah, kot so potresi, požari ali poplave. Za nadaljno analizo je potrebno filtrirati določeno informacijo. Pri razvrščanju besedil se ponavadi dobro obnesejo algoritmi strojnega učenja, kot je naivni Bayesov klasifikator. Ugotovili smo, da pri filtriranju stavkov, ki opisujejo delovanje, ti algoritmi niso uspešni. Opišemo novo metodo, ki temelji na induktivnem logičnem programiranju in daje rezultate z visoko točnostjo in pokritjem. Metoda je bila uspešno uporabljena pri analizi poročil o poplavah v Srednji Evropi l. 2002. Prav tako razpravljamo o različnih specializiranih znanjih in orodjih za obdelavo naravnega jezika, ki smo jih uporabili pri procesiranju dokumentov.

Keywords text filtering, information extraction, term extraction

1. Text mining in crisis management

Exploratory data analysis in geographical domains should not be limited only to data with explicit spatial and temporal information. As bigger and bigger data sources contain data different from that in geographic information systems – e.g. text, hypertext, audio and video sequences, it is necessary to look for tools that have been developed for this kind of data and adapt them for specific purposes in geographical domains.

In crisis management, like flood, earthquake or fire management, a big amount of messages and reports is being exchanged between the parties that participate in the recovery process. Any tool that decreases this amount or even extract the relevant information can be helpful. An example is text filtering (Blaťák and Popelínský, 2004a; Sebastiani, 2002) where each document is classified into one of two classes, e.g. `INTERESTING` and `NON-INTERESTING`. When using such a tool, the recipient obtains only the relevant messages or messages relevant with a high confidence. In (Popelínský and Blaťák, 2006) we showed that methods based on the state-of-the-art propositional learning techniques can reach high accuracy when classifying whole document or a document paragraph.

However, in all these experiments whole document was supposed to belong to one class. Unfortunately it is not the case in reality because messages may contain short pieces of text with information on different topics. E.g. in the case of reports on flood a message consists of description of a current situation as well as description of actions performed. In (Popelínský and Blaťák, 2006) it was demonstrated that good performance can hardly be reached with propositional learning algorithms like Naive Bayes or Support Vector Machines without user intervention, namely without new features construction. One reason is the poor language for building the classifier which is actually built upon propositional logic only. Another reason is the small length of the information that are to be filtered – one sentence, one clause in a sentence or even a subpart of a clause.

In this paper we show that knowledge-intensive learning techniques, namely inductive logic programming (Cussens and Džeroski, 2000; Džeroski and Lavrač, 2001) that exploits predicate logic, can help to solve this problem. We aim at building a tool that gives a trustful answer to some of classification queries and maybe leaves some queries unanswered. The main goals of this work were

- to find an appropriate representation for this kind of tasks
- to find feasible natural language processing tools for pre-processing the text data and for enriching domain knowledge
- and eventually to find a method that reach high precision

Domain knowledge contain, for each word, information about its position in the sentence, a part-of-speech tag, a syntactic category and also hyperonymas in a domain-dependent ontology.

We demonstrate our approach on processing reports on flood in Central Europe in 2002. The problem is displayed in Section 2. The data used in experiments are introduced in Section 3. In Section 4. we introduce natural language processing (NLP) tools that we used for text pre-processing. Section 5. contains description of data transformations and several variants of domain knowledge. Description of the method can be found in Section 6. and results in Section 7. We conclude with discussion in Section 8. and with plans for future work in Section 9.

2. Reports on flood

News reports on flood, like the example below

> *In the Czech Republic the capital Prague is bracing for a major flood, just days after storms in the south of the country killed six people. "The forecast is bad," said Josef Novotny of the Prague crisis committee, warning that the Vltava river could burst its banks overnight. Floods affected some parts of Prague on Friday, but Mr Novotny said twice as much water was now bearing down on the city. Several southern towns are already cut off by water, and some have been evacuated. "Trains are not running, because bridges have fallen, and buses are not running, because roads are damaged," the mayor of the southern town of Prachatice, Jan Bauer, told Czech radio. Officials called on residents of the UNESCO-protected town of Cesky Krumlov – the second most popular tourist destination in the country – to leave.*
>
> *(Radio BBC Archive)*

usually contain two kinds of information. The first one concerns description of the current situation, the other describes an action performed, e.g. by an emergency unit. For instance the sentence

> *In the Czech Republic the capital Prague is bracing for a major flood, just days after storms in the south of the country killed six people.*

describes a situation whilst the sentence

> *Officials called on residents of the UNESCO-protected town of Cesky Krumlov – the second most popular tourist destination in the country – to leave.*

an action. It is evident that a sentence (or more generally, a part of the message) can concern both, or be irrelevant. Then the goal of a classification can be defined as an assigning a label from the set {`SITUATION, ACTION, BOTH, IRRELEVANT`} to each part of the given news report. The class `BOTH` contains sentences that concern both the current situation and the action performed. Then the label `IRRELEVANT` is assigned to all sentences that cannot be classified to none of these classes because the sentence brings no information relevant to a situation or to an action.

This work is the first step to fully understand such kind of reports. If we know that a sentence concerns, e.g., an action, a goal of the next step is understanding this action, e.g., learning the subject – agent(s) and target(s) or spatial and temporal relations. Such knowledge can be then used directly for decision support.

3. Data

In our experiments we used the summary report on flood in 2002 that has been manually collected (Andrienko, 2001). For each day there are two paragraphs, one describing the situation in the region affected with flood and the other referring about actions performed. The part of the description of the first day of the flood follows.

> *9 August 2002*
> *Situation*
> *Unusually heavy rains falling over a broad area of Central Europe have resulted in widespread flooding. In Austria, Bulgaria, the Czech Republic and Romania the floods have been particularly severe. The weather forecast for the next few days threatens even more rain. A rain dense and very slow moving front is lingering over the area, heading toward the Black Sea*
> *...*
> *Actions*
> *In Austria, the Red Cross has been working together with the fire brigade and the military to aid those affected by the floods. A 24 hour around the clock operation helped to ensure that those at risk were rescued. While efforts are continuing, it is believed that all of those who were in immediate danger have now been assisted. However, water levels remain dangerously high, with the risk of more rain at any moment. The Red Cross also organized mobile kitchens, providing hot food and drinks to those affected.*

This report was collected from texts on web – BBC, CNN, France Press, Reuters, Deutsche Welle, The Associated Pres Situation reports of OCHA (United Nations Office for the Coordination of Humanitarian Affairs), ReliefWeb, Emergency appeals and reports of humanitarian organizations: Salvation Army, Red Cross, a report of ENVIS – the Prague Information System on the Environment and an event report of RMS – Risk Management Solutions, Inc.

4. NLP tools

Memory-based shallow parser Memory-based shallow parser (MBSP) (Daelemans and van den Bosch, 2005) splits each sentence into chunks – name phrases, verb phrases or prepositional phrases. Moreover it can recognize borders of the subject and the object part in the sentence. Memory-based part-of-speech tagger that is a part of MBSP returns for each word its morphological category.

Topic maps We also used topic maps, namely Ontopoly from Ontopia[1] for building ontology for actions in flood management. We grouped all terms (mostly one- or n-words noun phrases) into classes of terms and also defined associations between these terms and verbs that appeared in the documents. In the work reported here we exploited only the hierarchy of terms. For each term, we add a pointer to its hyponym or to ANY. The list of classes that contain more than one term consists of *accessories, actions, area, authorities, chemical, doing, impulse, mobileEquipment, organization, state, valuables.*

WordNet Besides the hand-coded hierarchy mentioned above we also employed the WordNet semantic lexicon[2], namely synsets and collection of hyponyms. We generated for each word in documents (not only for the terms) its synset code(s) and its hyponyms.

5. Data representation and domain knowledge

5.1. Data representation

Each document has been morphologically and syntactically tagged with the memory-based shallow parser and then transformed into three relations

```
word(SiD, WordOrder, Word)
tag(SiD, WordOrder, PartOfSpeechTag)
chunk(SiD, WordOrder, Chunk)
```

where `SiD` is the unique sentence identifier and `WordOrder` identifies the position of a word in the sentence `SiD`. This *flat data representation* is then used in the domain knowledge predicates described below.

5.2. Domain knowledge

We use the term "domain knowledge" in the way commonly used in machine learning or inductive logic programming as knowledge that is not or cannot be expressed by learning examples themselves. This notion is more general than a feature description language which actually transforms data into propositional form. In domain knowledge predicates we are capable to describe any dependency between variables in those predicates without explicit building a feature for each dependency.

In (Blaťák, 2005) we described two different sets of background knowledge predicates for text documents, $\mathcal{B}^1$ and $\mathcal{B}^2$. They consist of predicates which specify general properties of a given focus word (*focusWord/2*), for example, that a given position in the sentence is a punctuation (*isPunct/2*), a quotation mark (*isQuot/2*) or that the first letter is capital (*begCap/2*).The difference between $\mathcal{B}^1$ and $\mathcal{B}^2$ lies in a manner of exploring the context of the analyzed word. $\mathcal{B}^1$ uses a literal *hasWord/3* whose first argument determines the relative position of a word with respect to the focus word (e.g. -3 means the third word to the left). The background knowledge $\mathcal{B}^2$ does not use information about a position of a word in the sentence and only introduces an arbitrary word from a context.

For a need in this work we extended $\mathcal{B}^2$ with temporal logic. Each sentence is seen as a sequence of events – words. $\mathcal{B}^3$ domain knowledge thus consists of all predicates in $\mathcal{B}^2$ and temporal predicates

```
follows(SiD,W1,W2)
after(SiD,W1,W2)
precedes(SiD,W1,W2)
before(SiD,W1,W2)
```

that have the meaning "in the sentence `SiD`, word `W2` immediately follows/is after/immediately precedes/is before the word `W1`". An example of a formula in $\mathcal{B}^3$ is below.

focusWord(S,B), after(S,B,C), begCap(S,C), hasTag(S,C,'NNP'), after(S,C,D), hasTag(S,D,'CC').

in the sentence A, there is a word B,
somewhere on the right there is the word C which starts with a capital letter
and has tag 'NNP'
and somewhere right from the word C there is the word D with tag 'CC'

Example:
"...[between/IN]^B the/DT United/NNP ^{C} States/NNP and/CC^D China/NNP ..."

6. Experiments

6.1. Aleph

The Aleph[3] is an ILP learner that can learn from noisy data. It chooses one or more positive examples from a training set and constructs their least general generalizations – so called a bottom clause – with respect domain knowledge. Then using literals in the bottom clause, Aleph builds new rules in general-to-specific manner and employs a covering paradigm: it learns one clause a time and after finding it, Aleph removes all positive examples covered by this clause. This repeats until all (but a small fraction of) positive examples are covered and none (but a small fraction of negative) examples are not covered. The degree of incorrectness and inconsistence is driven by user-defined threshold. parameters.

6.2. Description of the method

As positive examples we used sentences that describe an action, the rest has been used as negative examples. Each sentence was enriched with output from memory-based morphological tagger and shallow parser. Further we added the information form hand-coded ontology and information from WordNet - synsets and hyponyms for each word.

The goal was to find a definition of the predicate `s(SiD, Subj, Verb, Obj)`. Arguments of the predicate `s(SiD, Subj, Verb, Obj)` brings information about the sentence (SiD, sentence identifier), a noun that appears in the subject part (Subj), a non-auxiliary verb (Verb), and a noun that appears in the object part (Obj). In

[1] `http://www.ontopia.net/`
[2] http://wordnet.princeton.edu/
[3] http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/

general, there can be more then one learning example per sentence: it may happen e.g. when the subject part contains more than one noun.

The average number of literals was 127.29 (standard deviation 44.95, max 222, min 57).

We used Aleph for finding all rules that cover a minimal number (between 5 and 25) of positive examples in the learning set[4] and then used these rules for classifying unseen test data. The bottom limit was set to 5 because coverage smaller than 5 examples resulted in over-fitting. We used 200, 300, 400, 500 and 600 examples for learning, the rest for testing. The clause length (number of literals in the rule) varied from 3 to 6.

For description of results we use the usual characteristics, precision, recall and the F-1 measure.

All experiments were performed on AMD Athlon™ XP 2500+ computers with 756 MB of memory.

7. Results

Summary of results Precision and recall for different cardinality of learning set are displayed in Fig. 1 and Fig. 2. On X-axis, minpos stand for the minimal coverage. On Y-axis there is precision and recall, respectively. All other characteristics for the case of 500 learning examples are in Table 1.

The fact that precision is increasing with increasing number of learning examples (see Figures 1 and 2) is not surprising. More important is the fact that for 400, 500 and 600 examples differences in precision are very small.

The most important result is the fact that even more significant increase of precision has been observed for increasing minimal coverage. Minimal coverage is the minimal number of positive examples from the learning set that has to be covered by each rule.. When looking at Table 1 it is true that precision for more than 300 examples in the learning set, is always high. But there are also many situations that are incorrectly classified – recall for situations is high. From this respect, the best choice will be higher minimal coverage of rules. We can see that for minimal coverage=22 the recall for situations is half of that for lower values of mininimal coverage.

min_cov.		Prec. (%)	Rec. (%)	F-1 (%)	Acc. (%)
5	act.	87.69	49.31	63.12	56.12
	sit.	32.48	22.11	26.31	
10	act.	88.05	52.69	65.93	58.52
	sit.	33.80	22.85	27.27	
18	act.	89.28	55.08	68.13	60.75
	sit.	35.47	21.13	26.48	
22	act.	93.56	51.38	66.36	60.28
	sit.	36.35	11.30	17.24	

Table 1: Learning from 500 examples

[4]This is called a minimal support in learning association rules

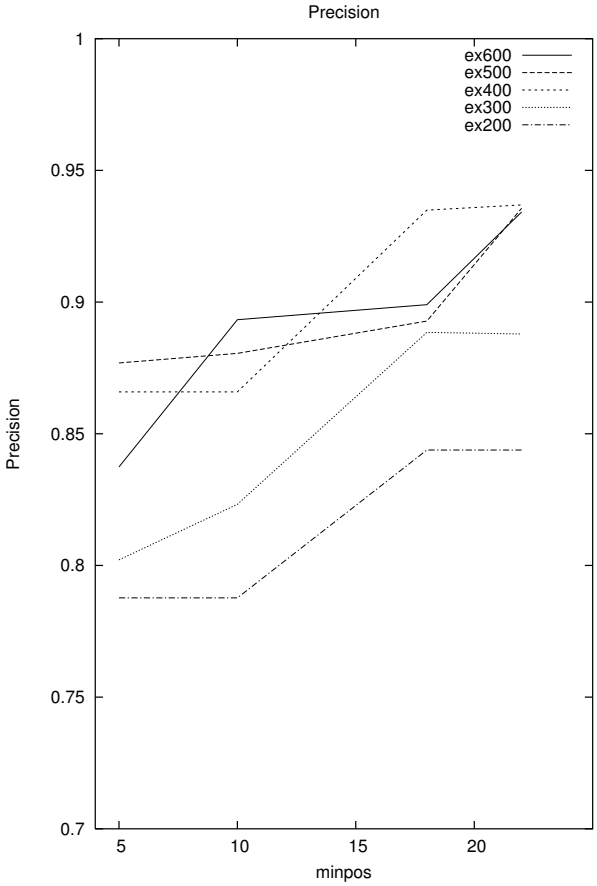


Figure 1: Precision for different learning sets

Rules Examples of the most interesting rules (600 examples in the learning set, clause length=5) are in Fig. 3.

8. Discussion

Best parameters settings We observed that the best clause length was 5 literals. Longer rules did not result in a significant increase of precision.

Dependency on the domain knowledge We also checked how the precision is influenced by the domain knowledge – $\mathcal{B}^1, \mathcal{B}^2$, and $\mathcal{B}^3$ – used. Not surprisingly, precision is increasing with the complexity of the domain knowledge. The same trend, but much more faster, has been observed for recall.

Use of WordNet The use of data from WordNet did not result in increase of accuracy. Information about a synset did not appear in the learned rule at all. Info on hyperonyma appears in less then 5% of rules, and always together with hyperonyma from the hand-coded ontology. It is obvious because the hand-coded ontology is domain-specific and contains more information specific to our task.

State-of-the-art Up to our knowledge, this is the first work on classification of short texts and action recognition. Technically, it is of course a part of the research stream on text filtering (Sebastiani, 2002). Similar goals are solved in the series of workshops on Event Extraction and Synthesis. See e.g. `http:\\www.ics.uci.edu/~ashish/ee.htm`.

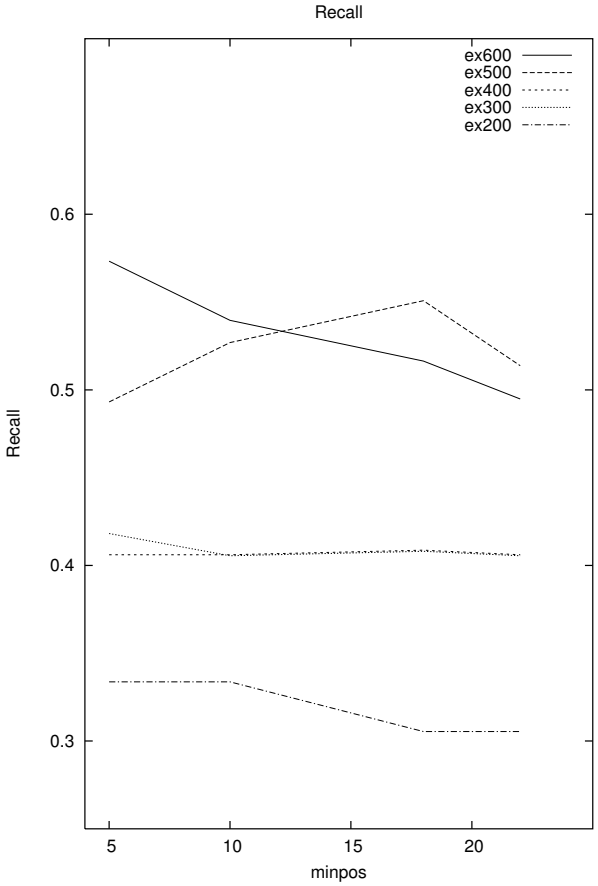


Figure 2: Recall for different learning sets

9. Conclusion and future work

We developed and experimentally confirmed a novel method for filtering small pieces of text that is based on inductive logic programming framework. In filtering sentences that brings information abour actions during floods, the precision overcome 90%.

In future, we want to use this method for term recognition. First results, with propositional learning algorithms has been introduced in (Popelínský and Blaťák, 2006). First-order logic rules learned with Aleph contains even more information. Another way is to use frequent patterns (Blaťák and Popelínský, 2004b) (also called large itemsets) for finding new features. We also plan to exploit other relations defined in the Topic Maps ontology.

We believe that this work can be helpful in automatic information extraction in the process of crisis management. As a small step to understanding the contents of a message, our approach can help to find an equilibrium between a need of understanding and necessary formalization of messages.

Acknowledgement

We thanks Natalia Andrienko for providing the report on flood and Petr Výmola for building the ontology. This work has been partially supported by the Faculty of Informatics, Masaryk University in Brno and by the Grant Agency of the Czech Republic under the Grant No. MSM0021622418 Dynamic Geo-visualization in Crisis Management.

10. References

N. Andrienko. 2001. A report on flood in central europe 2001. Manuscript.

J. Blaťák and L. Popelínský. 2004a. Fragments and text categorization. In Blanche P. and Rodrigues H., editors, *Proceedings of the ACL-2004 Interactive Posters/Demonstrations Session, Barcelona 2004.*

J. Blaťák and L. Popelínský. 2004b. Mining first-order maximal frequent patterns. *Neural Network World*, 5:381–390.

Jan Blaťák. 2005. First-order frequent patterns in text mining. In *Proceedings of the 12th Portuguese Conference on Artificial Intelligence, EPIA'05*, pages 344–350. Institute of Electrical and Electronics Engineers, Inc., December.

J. Cussens and S. Džeroski. 2000. *Learning Language in Logic*. Springer-Verlag.

W. Daelemans and A. van den Bosch. 2005. *Memory-based language processing*. Cambridge University Press.

S. Džeroski and N. Lavrač. 2001. *Relational Data Mining*. Springer-Verlag, September.

L. Popelínský and J. Blaťák. 2006. Mining situations and actions from news. In *Proceedings of Znalosti'06, Czech-Slovak Conference on Artificial Intelligence*, pages 1–3, Feb.

F. Sebastiani. 2002. Machine learning in automated text categorization. In *ACM Computing Surveys*, volume 34, pages 1–47, March.

Rule 1 Pos cover = 128 Neg cover = 0
s(A,B,C,D) :- hasWord1(also,A,E), hasWord1(to,A,F).

Rule 2 Pos cover = 83 Neg cover = 0
s(A,B,C,D) :- precedes(A,D,E), before(A,E,F), isPoS(A,F,'VB'), isVP(A,E).

Rule 3 Pos cover = 184 Neg cover = 0
s(A,B,C,D) :- hasWord1(actions,A,E), before(A,E,F), isPoS(A,F,'NNS'), isPoS(A,E,'NN').

Rule 4 Pos cover = 40 Neg cover = 0
s(A,B,C,D) :- before(A,D,E), isPoS(A,E,'VBZ'), before(A,C,F), isPoS(A,F,'RP').

Rule 5 Pos cover = 24 Neg cover = 0
s(A,B,C,D) :- precedes(A,B,E), isString(A,E,of), before(A,E,F), isPoS(A,F,'VBG').

Rule 6 Pos cover = 174 Neg cover = 0
s(A,B,C,D) :- hasWord1(leave,A,E).

Rule 7 Pos cover = 124 Neg cover = 0
s(A,B,C,D) :- hasWord1(have,A,E), hasWord1(city,A,F).

Rule 8 Pos cover = 49 Neg cover = 0
s(A,B,C,D) :- begCap(A,B), precedes(A,B,E), isString(A,E,were).

Rule 9 Pos cover = 152 Neg cover = 0
s(A,B,C,D) :- hasWord1(12,',',A,E), before(A,C,F), isPoS(A,F,'JJ'), isOBJ(A,F).

Rule 10 Pos cover = 128 Neg cover = 0
s(A,B,C,D) :- hasWord1(popular,A,E).

Rule 11 Pos cover = 59 Neg cover = 0
s(A,B,C,D) :- before(A,B,E), isSBJ(A,E), precedes(A,D,F), isPoS(A,F,'NNS').

Rule 12 Pos cover = 126 Neg cover = 0
s(A,B,C,D) :- hasWord1(12,',',A,E), hasWord1(city,A,F), isPoS(A,B,'NNS').

Rule 13 Pos cover = 83 Neg cover = 0
s(A,B,C,D) :- hasWord1('Prime',A,E).

Rule 14 Pos cover = 125 Neg cover = 0
s(A,B,C,D) :- hasWord1(medieval,A,E).

Figure 3: Rules

Towards Combining Finite State, Ontologies, and Data Driven Approaches to Dialogue Management for Multimodal Question Answering

Daniel Sonntag

DFKI GmbH – German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3
D-66123 Saarbrücken, Germany
sonntag@dfki.de

Abstract

Information-providing dialogue systems typically use one of the following dialogue strategies: finite-state based, frame-based, or agent-based. Recent extensions are concerned with hybrid models, where mixed automaton and information state approaches are combined. We report on our multimodal mobile Semantic Web access system SMARTWEB which uses a more rigid dialogue management strategy to ensure operability and robustness of the system while allowing for flexible dialogical interaction in a question answering scenario. In addition, a strategy to incorporate information state approaches into the running system to be extended towards a machine learning scenario for very particular, e.g. domain or language specific, dialogue management decisions is proposed.

Korak h kombiniranju končnih avtomatov, ontologij in pristopov na podlagi podatkov za vodenje dialoga pri multimodalnem odgovarjanju na vprašanja

Sistemi dialoga za dajanje informacij običajno uporabljajo eno od strategij, značilnih za dialog: takšno, ki temelji bodisi na končnih avtomatih, na okvirih ali agentih. Najnovejše širitve se ukvarjajo s hibridnimi modeli, pri katerih se kombinirata pristopa avtomatov in informacijskega stanja. V prispevku predstavljamo svoj multimodalni mobilni sistem SMARTWEB za dostopanja do mreže Semantic Web, ki uporablja bolj togo strategijo vodenja dialoga in s tem zagotavlja operabilnost in robustnost sistema, hkrati pa dovoljujemo prilagodljivo dialoško interakcijo v scenariju odgovarjanja na vprašanja. Poleg tega predlagamo strategijo za vključevanje pristopov informacijskega stanja v delujoč sistem, ki bi se razširil v scenarij strojnega učenja za zelo specifične odločitve vodenja dialoga, npr. za določeno področje ali jezik.

1. Introduction

Dialogue systems often use finite-state-automata (FSA) based dialogue management strategies where the dialogue flow is represented by a path through a finite-state machine. More flexible strategies are frame-based (frame slots are filled dynamically), or agent-based (the interaction is free as far as possible according to some dialogue objectives, e.g. user objectives) (Chu et al., 2005; McTear, 2002). Recent extensions are concerned with hybrid models, whereby automaton and information state (IS) approaches are combined (e.g. (Horacek and Wolska, 2005)). In context of the SMARTWEB system (Wahlster, 2004; Reithinger et al., 2005), we extend hybrid dialogue models for mobile multimodal interaction and explore, how information states can be extracted from dialogue processing data, in particular from ontology structures. The goal is to integrate data-driven approaches to dialogue management.

In our approach, Semantic Web (Fensel et al., 2003) structures form the representation basis of dialogue processing data which allows for extracting machine learning features for dialogue adaptations in a specific application scenario: SMARTWEB aims to develop a context-aware, mobile and multimodal interface to ontology servers, composed Web Services and open-domain question answering (QA) systems. In the main application scenario, the user carries a PDA and is able to pose multimodal questions about football games, teams, and players at a visit to the football World Cup in Germany — using speech, pen, and gesture as input modalities. The displays of these mobile devices are small (320*240 pixel for T-Mobile's MDA3 or 480*640 pixel for the MDA4), and the pocket computer has very limited computational power. Nonetheless, the user should be able to interact with the system in different modalities such as speech and gesture and refer to the displayed results for further inspection or posing a new query.

In SMARTWEB dialogue objectives and hence the dialogue reaction behaviour is governed by the general QA scenario, which means that almost all dialogue and system moves relate to questions, follow-up questions, clarifications, or answers. As these dialogue moves can be regarded as adjacency pairs, a standard dialogue behaves according to some finite-state grammar for QA, which makes a basic FSA appear reasonable for dialogue management. A finite state approach generally enhances robustness and portability and allows to demonstrate dialogue management capabilities even before more complex information states are available to be integrated into the reaction and presentation decision process. The paper is organised as follows: in section 2. the interaction requirements are discussed, followed by the general system architecture. In section 3. the reaction and presentation module design is introduced, how the FSA for QA looks like, what kind of ontology structures are used, and what kind of meta data can be made available for automatic adaptation. In section 4. we give concluding remarks.

2. Mobile Interaction Requirements

Interaction requirements are discussed in terms of reaction and presentation requirements to provide a basis for implementing a multimodal mobile human-computer-interface (HCI).

2.1. From Storyboard to HCI Implementation

Basically SMARTWEB allows the user to send multimodal requests to various services linked by a Semantic Web framework. The partners in the project share implementation experience from earlier multimodal intreaction projects like Verbmobil and SmartKom (Wahlster, 2000; Reithinger et al., 2003). Like others, we used some guidelines (Oviatt, 1999; Alexandersson et al., 2004) in the development of the storyboard and the specification of interaction possibilities.

The user should be able to

- ask simple factoid and enumeration questions, and inspection questions or commands (search, explore, inspect).
- control the system. She can ask for status information, or cancel a running query.

On the other hand, the system can take the initiative to

- clarify or cancel user requests.
- add and replace results.
- provide status information and hints.

Interesting decisions in dialogue management are concerned with these system initiatives.

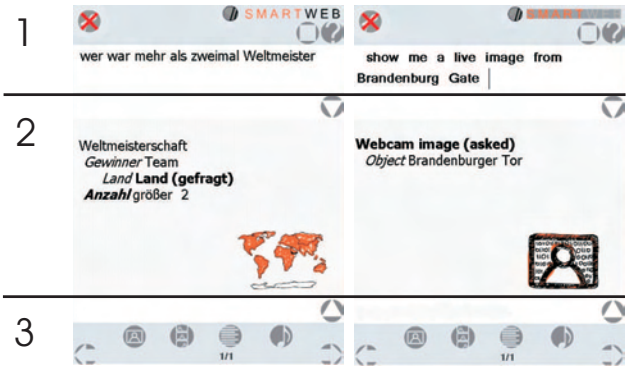


Figure 1: In (1) we display the output of the automatic speech recogniser, (2) shows the corresponding multimodal semantic paraphrase. The query paraphrase can also be listened to by *on ear* audio output. While audio repetition plays, barge-in is possible which leads to correction modes for single words or the complete query.

Paraphrasing a semantic query and displaying it to the user is one of the key elements for implicit user feedback (figure 1). The ontological structures resemble typed feature structures (TFS) (Carpenter, 1992) common in formal NLP. The paraphrase, which is presented to the user and sent to the Semantic Web in RDF representation, is constructed on the basis of the results of the question analysis. Ontology query instances are communicated between the dialogue server and the Semantic Web knowledge bases (figure 2). The nested predicate-argument structure shows the interpretation of the user utterance *Who won the football World Cup more than twice?* Note that the paraphrase is fully specified and contains the unfilled template slot *varName* for the winner name and the expected focus type (Team in text medium). Team itself is an underspecified concept which can be instantiated by a more specific instance according to the domain, e.g., a *FootballNationalTeam* instance. If the user completed his utterance by *Are pictures there?*, the *mediaTypes* slots would have changed to comprise text and image media. In this way we establish multimodal access to the Semantic Web.

```
[ discourse#Query
      text: "wer war mehr als zweimal Weltmeister"
      dialogueAct: [ InterrogQuestion ]
      focus: [ Focus
         focusMediumType:[ mpeg7#Text ]
         ...
         contextObject: [ FIFAWorldCup
                          winner:Team
                          origin: [ sumo#Country  ... ]
                        ]
         contextObject: [ GreaterThan
                          constraintRightArg: "2"
                        ]
         varName: ?X
         ]
]
```

Figure 2: Semantic queries serve as input to the Semantic Web knowledge bases.

2.2. Dialogue System Architecture

A flexible dialogue system platform is required to support audio transfer and other data connections between the mobile device and a remote dialogue server. We developed a new framework complementing other approaches (Cheyer and Martin, 2001; Herzog et al., 2004; Bontcheva et al., 2004) for Semantic Web based data structures for both dialogue system-internal and system-external communication. The dialogue system instantiates and sends requests to the so-called *Semantic Mediator*, which provides the umbrella for all different access methods to the Semantic Web we use: a knowledge server, a Web Service composer, semantically wrapped Web pages, and a QA system. To integrate the dialogue components we developed a Java-based hub-and-spoke architecture (Reithinger and Sonntag, 2005). The speech interpretation component (SPIN) (Engel, 2005), the modality fusion and discourse component (FADE) (Pfleger, 2005), the context-module SITCOM to resolve GPS coordinates, the natural language generation module (NIPSGEN), and the system reaction and presentation component (REAPR) are attached to it (figure 3). An exemplary data flow is $SPIN \rightarrow FADE \rightarrow REAPR \rightarrow SemanticMediator \rightarrow REAPR \rightarrow NIPSGEN$, which gets more complicated if, e.g., misinterpretations or clarifications are involved.

Having received a result list of multimodal items as answers to a question after query processing, we have to decide which responses are appropriate to be presented. The last point concerns content selection, medium selection, and selection of the visual presentation metaphors engaged. In this contribution we focus on the reaction behaviour, including the decisions of accepting a proposed semantic paraphrase we coded into a FSA structure. We put emphasis on the REAPR component in the remainder of the text.

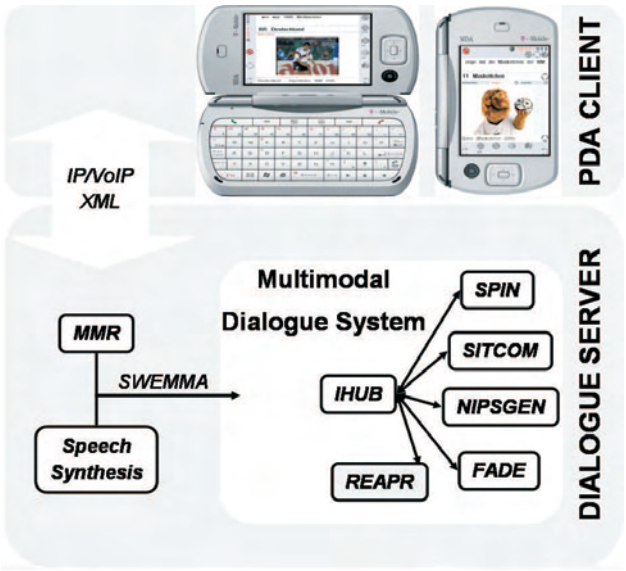


Figure 3: SMARTWEB's mobile dialogue system architecture: the PDA client and the dialogue server which comprises the dialogue manager. MMR stands for multimodal recogniser.

3. The Reaction and Presentation Module

REAPR manages dialogical interaction, i.e., the reaction and presentation behaviour, for the supported dialogue phenomena such as flexible turn-taking, incremental processing, and multimodal fission/fusion of system output. REAPR is based on the FSA shown in figure 4

3.1. General Discourse Obligations and Structures

1. The primary role to fulfill in information-providing dialogue systems is to elicit all relevant information from the user to pose a very specialised query for which getting the right answer is very probable. This role gains even more importance if the queries must be transformed into explicit semantic representations, i.e., ontological query instances.

2. Whenever users have the freedom to formulate statements, understanding may be difficult. In such cases the strategy is, for first, to produce useful reactions, and for second, to give hints or examples to the user on how to reformulate the question.

The general discourse obligations towards mixed and system initiative dialogue system behaviour are coded into the non-deterministic FSA structure, the multiple outgoing arcs at important input processing dialogue nodes, such as *query completion*.

3.2. User Correction Model

One important question in the user interaction model with respect to dialogue management decisions is how to correct invalid user input stemming from speech recognition errors or from errors that occur while interpreting user utterances. This becomes even more relevant in the context of composite multimodality, where the dialogue system must understand and represent the multimodal input.

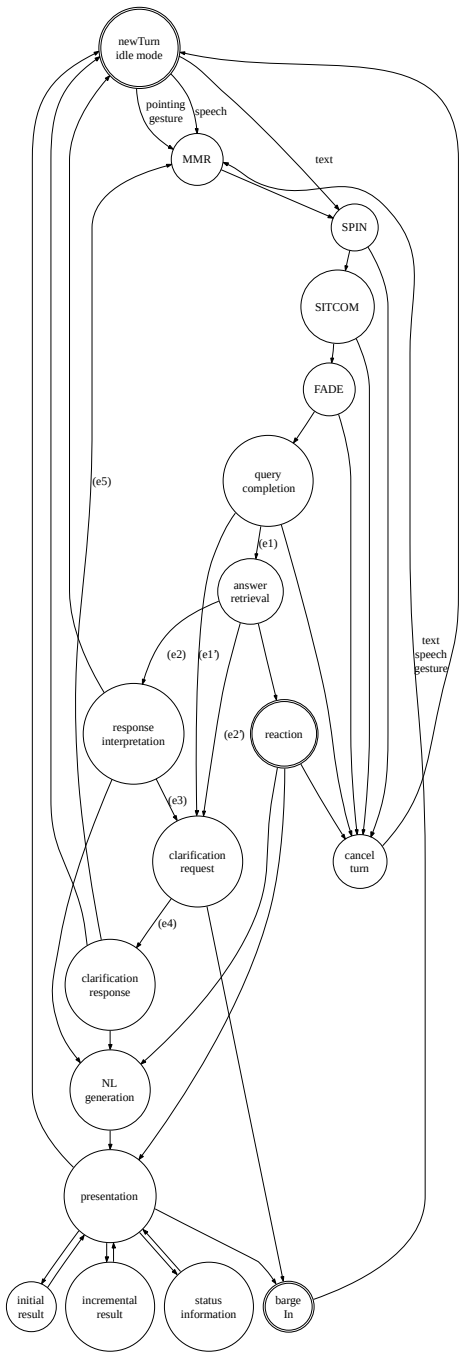


Figure 4: FSA structure as REAPR's ground control

When the system displays the query paraphrase, the user should have the possibility to interrupt the audio output and edit the query: A click on a word or word group directly enables the correction mode of the word(s), whereby the navigation through the displayed query is provided via keyboard or pen. Pen is preferred because it allows intuitive word selection on screen. For example, the user could simply click on or underline an incorrect word. The queries for a Semantic Web search have to be as accurate as possible, and correcting flawed speech recognition output is of paramount importance in the Semantic Web context. Practically this means a lot of manual corrections to be done by the user. In order to minimise the number of correc-

tions to be done by the user, two system initiative strategies can be explored: To induce missing parts at the NLU stage and/or to apply an independent (binary) classificator for dialogue management to find out the underspecified queries with high or low success chances. Inducing missing parts at NLU is a language and domain-dependent task for the language and domain experts developing the NLU module. Applying a classificator depends on the suitability of meta data that can be extracted during dialogue processing. The design of REAPR is tailored toward an decision making process using automatic classifiers (section 3.2.).

FSA Structure as REAPR's Ground Control The FSA makes up the integral part of the dialogue managment decisions in the specific QA domain we model. The dialogue structure that is embedded and committed by the transitions of the FSA allows for a declarative control mechanism for reaction and presentation behaviour.

The initial node of the FSA is $N_{newTurn}$ which represents the system's idle mode while awaiting user input. The second starting node is $N_{reaction}$, the system takes initiative and informs the user or cancels the current turn. The third starting node is $N_{bargeIn}$ which is the user-initiative action as counterpart to system-initiative action. Every user action while processing a query can be seen as barge-in and is interpreted by the MMR component. This concerns the speech input, the selection of result words and sentences, and other gesture input such as poiting gestures on images, whereas new textual queries directly go to SPIN. The reason for that is simple, we do not fuse textual and image pointing gestures, since cross-modal fusion (speech and gesture) is much more convenient.

Starting with a new speech input, the example flow: $SPIN \rightarrow FADE \rightarrow REAPR \rightarrow SemanticMediator \rightarrow REAPR \rightarrow NIPSGEN$ can be mapped onto the actual FSA states: After $N_{queryCompletion}$ the answer is processed and retrieved $N_{answerRetrieval}$, generated $N_{NLgeneration}$, and presented $N_{presentation}$. However, we focus on the interesting part around the possibly cyclic paths containing $N_{clarificationRequest}$ (table 1). All cyclic paths containing $N_{clarificationRequest}$ are designed to elicit additional query information by feedback from the user to recover from query problems.

Path Name	Path Nodes
$P_{interpretation}$	$(e1) \cdot (e2) \cdot (e3) \cdot (e4) \cdot (e5)$
$P_{annotation}$	$(e1) \cdot (e2') \cdot (e4) \cdot (e5)$
$P_{induction}$	$(e1') \cdot (e4) \cdot (e5)$

Table 1: Cyclic paths containing $N_{clarificationRequest}$

$P_{interpretation}$ is the simplest QA dialogue processing path in which each retrieved answer is tested for semantic correctness (e.g. correct answer type). According to the meta data obtained from the answer status (figure 5), REAPR decides whether the answer is appropriate for presentation or not. An empty answer or false answer type initiates a system response, to either ask a different question or reformulate the question.

$P_{annotation}$ relies on the semantic annotation of the Semantic Web access. For example, the composed Web Service module can question missing parts matched to input descriptions of individual web services. In this case, the decision to pose a clarification question is dynamically deflected to the answering services, in this case the Web Services. The entity to be asked for is explicitly marked-up in the result obtained from the Web Services. The composed service module automatically sets a *ClarificationResponse* dialogue act instead of an *Answer* into the result structure.

$P_{induction}$ corresponds to the attempt to adapt dialogue processing towards recoverability decisions at a very early stage of processing. REAPR itself is responsible to decide if a query is valid and to be transferred to the Semantic Mediator. Although the right choice among the different strategies can be predicted by adaptable systems in many different ways, the $P_{induction}$ is a very interesting one: It is possible to infer patterns from the available meta data material, or abstractions from domain-ontological question-answer instances to judge a query as yet unsuitable, i.e., too underspecified, too specific, unsupported, untrusted, hence with little change of success. Effective selection and mining of ontological process data is a precondition and includes the question what kind of ontological meta data is available and suitable for feature spaces in machine learning environments (cf. section 3.3.).

The simple but effective FSA ground structure allows for adding new knowledge-driven functionality if necessary without the need for expensive training data following a fully empirical approach. On the other hand, tuning and adaptation can be easily integrated by casting the decision, which path in the undeterministic FSA to follow, into a classification problem to be solved by any suitable supervised machine learning classifier. Since the ontology structures are tailored toward semantically-rich information items, a unsupervised classification experiment can be complemented by association rule mining.

Ontological result structures can be seen in figure 5. The features obtained from the *AnswerStatus* resemble the features used in previous experiments to classify user models (Komatani et al., 2003) for dialogue system adaptivity. The feature *finishedSearchComponent* reveals the source of the obtained results. Additional scores for single utterances can be obtained from the recogniser (recogniser score), SPIN (speech interpretation score) and ratios of correct answering processes. The number of filled slots in the query and their names can also be added to the IS to extract patterns from. A semantic ontological result delivers meta data about the current dialogue state to be incorporated into the IS.

According to the undeterministic FSA, the dialogue management component has to decide on-the-fly whether a clarification dialogue is to be initiated, or a confirmation is needed, or the query is being sent without any confirmation. In this way we try to obtain optimised dialogue prompts in specific data situations. This optimisation toward more natural interaction should be obtained by mining the IS.

3.3. Information States for QA

Information state theory of dialogue modelling consists basically of a description of informal components (e.g.,

```
[ Result
    status: [ AnsweringStatus
           derivedFromQuery: discourse#Query
           elapsedTime:  "7"
           resAmount: "1"
           resForm: "nonincremental"
           finishedSearchComponent: "knowledgebase"
            ]
    ...
    content: [ FootballNationalTeam ... ]
    answerType: "FootballNationalTeam"
]
```

Figure 5: Semantic result: answer status, content, and answer type information

obligations, beliefs, desires, intentions) and their formal representation (Larsson and Traum, 2000). IS states as envisioned here do not declare update rules and an update strategy (for e.g. discourse obligations (Matheson et al., 2000)) because the data-driven approach is pattern-based, using directly observable processing features, which complements an explicit manual formulation of update rules. Since the dialogue ontology is a formal representation model for multimodal interaction, multimodal MPEG7 result representations (Sonntag and Romanelli, 2006), result presentations (Sonntag, 2005), dialogue state, and (agent) communication with the backend knowledge servers, large information spaces can be extracted from the ontological instances describing the system and user turns in terms of realised dialogue acts.

The turn number represents our first FSA extension to IS with the result of more flexibility to user replies. Replies which are not specified in a pathway, are not considered erroneous by default, since the IS now contains a new turn value. Ontologicial features for IS extraction under investigation are summarised in table 2.

Feature Class	IS State Features
MMR	*Listening, Recording, Barge-in, Last-ok, Input dominance (text or voice)*
NLU	*Confidence, Domain relevance*
Query	*Dialogue act, Focus medium, Complexity, Context object, Query text*
Fusion	*Fusion act, Co-reference resolution*
Answer	*Success, Speed, Answer streams, Status, Anser type, Content, Answer text*
Manager	*Turn/Task numbers, Idle states, Waiting for Results, User/system turn, Elapsed times: input/output, Dialogue act history (system and user) e.g. reject, accept, clarify*

Table 2: IS Feature Classes and Features

In previous existing work on dialogue management adaptations (Walker et al., 1998; Singh et al., 2002; Rieser et al., 2005), reinforcement learning was used for which large state spaces with more than about five non-binary features a hard to deal with. As seen in table 2, more than five relevant features can easily be declared. Since our optimisation problem can be formulated at very specific decisions in dialogue management due to the FSA ground control, less training material for larger feature extractions is to be expected. Relevance selection of ontology-based features is the next step for ontology-based dialogue managment adaptations.

Ontological Infrastructure SMARTWEB's ontological infrastructure is realised by merging concepts from two established foundational ontologies, DOLCE (Gangemi et al., 2002) and SUMO (Niles and Pease, 2001) into a new one (SWIntO) (Cimiano et al., 2004). Domain specific knowledge is modelled in sub-ontologies. SWIntO integrates question answering specific knowledge, interpretations of user utterances (modelled by the EMMA[1] extension SWEMMA), dialogue acts, and HCI concepts in a discourse ontology (DISCONTO). The DISCONTO also contains concepts for the communication between the *Dialogue Server* and the *Semantic Mediator*. The SWIntO and DISCONTO provide semantic representation structures for natural language understanding, generation, and dialogue management.

4. Concluding Remarks

We presented the interaction requirements and intermediate development steps of the reaction and presentation module REAPR for the second demonstrator of the SMARTWEB system[2]. The current dialogue model is FSA-based with user barge-in capabilities. SMARTWEB as multilingual (German and English), multimodal QA system was successfully demonstrated in the context of the football World Cup 2006 in Germany. The knowledge base (Swinto ontology) comprises 2308 concept classes, 1036 slots, and 90522 instances.

Semantic Web-based dialogue and data models are convenient models towards language-independence and multilinguality of HCI technologies. Operating on semantic ontological instances, knowledge-intensive processing modules within the dialogue system, such as REAPR, can be language-independent. Language-dependent modules (SPIN, FADE) operate on the same ontological instances, but exploit further language-dependent information provided by a multilingual lexicon model LingInfo (Buitelaar et al., 2005; Buitelaar et al., 2006). REAPR's FSA model works completely language-independent, for $P_{induction}$ our next step is to include linguistic features into the IS model. Currently, 19992 instances of the 90522 instances such as games, players, goals support linguistic information, and 6002 LingInfo instances have been created for German and English, which motivates machine learning experiments to incorporate linguistic, probably language-dependent, features into REAPR's IS dialogue model for adaptable dialogue management on top of a more structured but robust language-independent non-deterministic FSA-based dialogue management model.

5. Acknowledgments

The research presented here is sponsored by the German Ministry of Research and Technology (BMBF) under grant

[1] http://www.w3.org/TR/EMMAreqs

[2] http://www.smartweb-project.org

01IMD01A (SMARTWEB). We thank our student assistants and the project partners. Norbert Reithinger provided valuable comments on earlier versions. The responsibility for this papers lies with the author.

6. References

Jan Alexandersson, Tilman Becker, Ralf Engel, Markus Löckelt, Elsa Pecourt, Peter Poller, Norbert Pfleger, and Norbert Reithinger. 2004. Ends-based dialogue processing. In *Proceedings, Second International Workshop on Scalable Natural Language Understanding*, Boston.

Kalina Bontcheva, Valentin Tablan, Diana Maynard, and Hamish Cunningham. 2004. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10. Special issue on Software Architecture for Language Engineering.

Paul Buitelaar, Michael Sintek, and Malte Kiesel. 2005. Feature representation for cross-lingual, cross-media semantic web applications. In Thierry Declerck and Siegfried Handschuh, editors, *Workshop on Knowledge Markup and Semantic Annotation (SemAnnot2005)*, 11.

Paul Buitelaar, Thierry Declerck, Anette Frank, Stefania Racioppa, Malte Kiesel, Michael Sintek, Ralf Engel, Massimo Romanelli, Daniel Sonntag, Berenike Loos, Vanessa Micelli, Robert Porzel, and Philipp Cimiano. 2006. Linginfo: Design and applications of a model for the integration of linguistic information in ontologies. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, May 24–26.

B. Carpenter. 1992. The logic of typed feature structures.

Adam J. Cheyer and David L. Martin. 2001. The Open Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 4(1–2):143–148.

Shiu-Wah Chu, Ian ONeill, Philip Hanna, and Michael McTear. 2005. An approach to multi-strategy dialogue management. In *Proceedings of INTERSPEECH*, pages 865–868.

Philipp Cimiano, Andreas Eberhart, Pascal Hitzler, Daniel Oberle, Steffen Staab, and Rudi Studer. 2004. The smartweb foundational ontology. Technical report, Institute for Applied Informatics and Formal Description Methods (AIFB) University of Karlsruhe, Karlsruhe, Germany. SmartWeb Project.

Ralf Engel. 2005. Robust and efficient semantic parsing of free word order languages in spoken dialogue systems. In *Proceedings of 9th Conference on Speech Communication and technology*, Lisboa.

Dieter Fensel, James A. Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. 2003. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press.

Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schcneider. 2002. Sweetening Ontologies with DOLCE. In *In 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, volume 2473 of Lecture Notes in Computer Science, page 166 ff, Sigünza, Spain, Oct. 1–4.

Gerd Herzog, Alassane Ndiaye, Stefan Merten, Heinz Kirchmann, Tilman Becker, and Peter Poller. 2004. Large-scale Software Integration for Spoken Language and Multimodal Dialog Systems. *Natural Language Engineering*, 10. Special issue on Software Architecture for Language Engineering.

Helmut Horacek and Magdalena Wolska. 2005. A hybrid model for tutorial dialogs. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2003. Flexible guidance generation using user model in spoken dialogue systems. In Erhard Hinrichs and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 256–263.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering, Cambridge University Press*.

C. Matheson, M. Poesio, and D. Traum. 2000. Modelling grounding and discourse obligations using update rules. In *Proceedings of NAACL 2000*, May.

Michael F. McTear. 2002. Spoken dialogue technology: Enabling the conversational user interface. *ACM Computing Survey*, 34(1):90–169, March.

Ian Niles and Adam Pease. 2001. Towards a Standard Upper Ontology. In Chris Welty and Barry Smith, editors, *Proc. of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine, October 17–19.

Sharon Oviatt. 1999. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81.

Norbert Pfleger. 2005. Fade - an integrated approach to multimodal fusion and discourse processing. In *Proceedings of the Dotoral Spotlight at ICMI 2005*, Trento, Italy.

Norbert Reithinger and Daniel Sonntag. 2005. An integration framework for a mobile multimodal dialogue system accessing the semantic web. In *Proc. of Interspeech'05*, Lisbon, Portugal.

Norbert Reithinger, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löckelt, Jochen Müller, Norbert Pfleger, Peter Poller, Michael Streit, and Valentin Tschernomas. 2003. SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications. In *Proc. of the 5th Int. Conf. on Multimodal Interfaces*, pages 101–108, Vancouver, Canada. ACM Press.

Norbert Reithinger, Simon Bergweiler, Ralf Engel, Gerd Herzog, Norbert Pfeger, Massimo Romanelli, and Daniel Sonntag. 2005. A Look Under the Hood Design and Development of the First SmartWeb System Demonstrator. In *Proceedings of 7th International Conference on Multimodal Interfaces (ICMI 2005)*, Trento, Italy, October 04-06.

Verena Rieser, Kruijff Kruijff-Korbayova, and Oliver Lemon. 2005. A framework for learning multimodal clarification strategies. In *Proceedings of the International Conference on Multimodal Interfaces (ICMI)*.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research (JAIR), Volume 16, pages 105-133*.

Daniel Sonntag and Massimo Romanelli. 2006. A multimodal result ontology for integrated semantic web dialogue applications. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, May 24–26.

Daniel Sonntag. 2005. Towards interaction ontologies for mobile devices accessing the semantic web - pattern languages for open domain information providing multimodal dialogue systems. In *Proceedings of the workshop on Artificial Intelligence in Mobile Systems (AIMS). 2005 at MobileHCI*, Salzburg.

Wolfgang Wahlster, editor. 2000. *VERBMOBIL: Foundations of Speech-to-Speech Translation*. Springer.

Wolfgang Wahlster. 2004. Smartweb: Mobile applications of the semantic web. In Peter Dadam and Manfred Reichert, editors, *GI Jahrestagung 2004*, pages 26–27. Springer.

M. Walker, J. Fromer, and S. Narayanan. 1998. Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email.

Towards clustering-based word sense discrimination

Darja Fišer*, Špela Vintar*, Ljupčo Todorovski†

* Oddelek za prevajalstvo, Filozofska fakulteta, Univerza v Ljubljani
Aškerčeva 2, SI – 1000 Ljubljana
{spela.vintar, darja.fiser1}@guest.arnes.si
† Fakulteta za upravo
Gosarjeva 5, SI – 1000 Ljubljana
ljupco.todorovski@fu.uni-lj.si

Abstract

This paper describes a series of experiments conducted to group similar words using context features derived from a corpus. The goal is to find an approach that would be suitable for cleaning the fuzzy WordNet synsets obtained by automatic translation of Serbian synsets into Slovene. Similar techniques have been used successfully by a number of researches already and they are attractive particularly because they are knowledge-lean and based on evidence found in simple raw text. A selection of features and settings are tested on sample test sets with an unsupervised machine learning method called hierarchical clustering. In the final part of the paper, the obtained results are analyzed and the optimal set of features is selected, followed by a discussion of the results and some further research plans.

Poskus uporabe hierarhičnega razvrščanja v skupine za določanje pomena besed

Prispevek opisuje niz eksperimentov, s katerimi smo na podlagi okolice besed, ki smo jo izluščili iz korpusa, skušali besede združiti v skupine glede na njihov pomen. Cilj naloge je bil najti pristop, ki bi bil primeren za čiščenje avtomatsko prevedenih sinsetov v slovenskem semantičnem leksikonu. Uporabljene tehnike so pred nami uspešno uporabili že številni avtorji in so priljubljene predvsem zato, ker zanje razen besedilnih zbirk posebni jezikovni viri niso potrebni. V eksperimentih smo na vzorčnih primerih sinsetov preverili različne nize atributov z metodo nenadzorovanega strojnega učenja, imenovanega hierarhično razvrščanje v skupine. Prispevek analizira optimalen niz atributov, predstavlja in vrednoti rezultate razvrščanja in podaja načrte za prihodnost.

1. Introduction

Words in natural language often have multiple distinct meanings which can only be determined by considering the context in which they occur. Given a target word used in a number of different contexts, its senses can be grouped together by determining which contexts are the most similar to each other.

The approach, commonly referred to as word sense discrimination (e.g. Agirre and Edmonds 2006), does not categorize words on a pre-existing sense inventory but clusters words based on their contexts observed from corpora. It is attractive primarily because it is knowledge-lean and thus does not rely on sense-tagged corpora or other manually crafted knowledge resources that are difficult and expensive to obtain. Furthermore, because it is data-driven, it does not fall victim to an absolute view of word meanings encoded in sense inventories (see Kilggarrif 1997, Hanks 2000) and is adaptable and portable across languages.

Word sense discrimination can be carried out either in a mono- or multilingual setting. The distributional approaches make distinctions between word meanings based on the assumption that words which appear in similar contexts have similar meanings (Harris 1968, Miller & Charles 1991). They do not assign but discriminate word meanings based on their distributional similarity found in monolingual corpora. On the other hand, approaches taking advantage of translational equivalence found in word-aligned parallel corpora use the sense-dependent translations of a word as a kind of sense inventory for that word in the source language (Brown et al. 1991, Gale et al. 1992, Ide et al. 2002).

A further distinction between the approaches in the word-sense discrimination domain is whether we are interested in identifying sets of related words by measuring similarity between word co-occurrence vectors (type-based), such as Latent Semantic Analysis (Deerwester et al. 1991), Hyperspace Analogue to Language (Burgess and Lund 2000) and Clustering by Committee (Lin and Pantel 2002). If, however, we aim to distinguish among the senses of a word in multiple contexts by clustering all the contexts of a word, we need to look into token-based approaches (Schütze 1998). In our case, the envisaged application was the construction of the Slovene Wordnet, where one of the tasks includes the validation of automatically translated synsets. It was hoped that the text mining algorithms based on first-order features, word co-occurences and POS would cluster similar words together and help us find the odd ones out. This is why we adopted an approach similar to McQuitty's Similarity Analysis (Pedersen & Bruce 1998).

The paper is organized as follows. In section 2 we briefly describe the process of building the Slovenian Wordnet and give an example of the 'fuzzy' synsets we attempt to clean. Section 3 describes the corpus and the methods used to construct the datasets. Section 4 presents the text mining methods and the distance measures selected for our experiments. Finally, sections 5 and 6 present the results obtained with different settings and discusses them.

2. Building Slovene WordNet

WordNet (Fellbaum 1998) is an extensive lexical database in which words are divided by part of speech and organized into a hierarchy of nodes, where each node represents a concept. Words denoting the same concept

are grouped into a synset, together with links to other relevant synsets (e.g. antonyms).

In recent years, WordNet has become one of the most valuable resources for a wide range of NLP research and applications which initiated the development of WordNets for many other languages (e.g. EuroWordNet[1], BalkaNet[2]). One of such enterprises is the building of Slovene WordNet (see Erjavec & Fišer 2006).

Being limited in the resources and manpower at our disposal, the expand model (Vossen 1998) seemed like the most suitable approach. Synsets were taken from the existing WordNet and were translated into Slovene. We used the Serbian WordNet (SWN) as the closest relative of Slovene in the WordNet family because we believe that concepts and relations among them overlap across languages better if the languages are closely related.

The Jurančič Slovene / Serbo-Croatian bilingual dictionary was inverted to give pairs of Serbo-Croatian / Slovene lemmas. This lexicon was then used to automatically translate Serbian synset literals from Base Concept Sets 1 and 2.

A typical error occurred in translations of polysemous literals where they were translated with equivalents that would be acceptable for some senses but not for this particular one (see Figure 1). Attempts are being made to detect and correct such errors with a clustering technique presented in this paper.

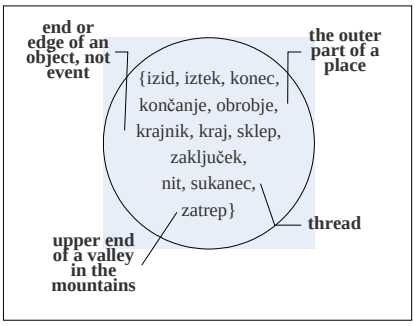


Figure 1. Example of a fuzzy synset: Eng. ending, conclusion, finish (event whose occurrence ends something)

3. Feature and settings selection

3.1. Test sets

As the main aim of our experiments was to test whether our text mining algorithms would prove reliable in grouping together similar words represented by their contexts, we first constructed two controlled Test Sets where each consisted of two distinct groups of synonyms.

- **Test Set 1**: *profesorica*[+], *učiteljica*[+], *tovariš*[+], *tovarišica*[+], *mentor*[+], *učitelj*[+], *profesor*[+]; *veselje*[*], *radost*[*], *sreča*[*], *zadovoljstvo*[*] (Eng. [+]: teacher, [*]: happiness)
- **Test Set 2**: *mož*[+], *fant*[+], *moški*[+], *možak*[+], *deček*[+]; *gora*[*], *hrib*[*], *vzpetina*[*], *grič*[*] (Eng. [+]: man, [*]: mountain)

These two Test Sets were considered "easy" because each consisted of only 2 target clusters, with clearly distinguishable meanings. They were used primarily to define the optimal context features and tune the clustering algorithm to the task at hand.

For a more realistic set of experiments, we adapted some unedited synsets from the Slovenian Wordnet in which the typical polysemy error explained above was observed in order to see whether the clusters proposed by the algorithm would detect the different word senses and whether it could also be used to validate the automatically translated synsets:

- **Test Set 3**: *panoga*[+], *stroka*[+], *disciplina*[+], *veja*[*], *odrastek*[*] (Eng. [+]:branch, division, [*]: tree branch)
- **Test Set 4**: *konec*[+], *kraj*[+], *krajnik*[*], *obrobje*[*], *nit*[*], *sukanec*[*], *zaključek*[+], *sklep*[+], *zatrep*[*] (Eng. [+]:end, conclusion, [*]: other)

3.2. Context features

Each word in a Test Set is described by a number of parameters, where a parameter is defined as a word appearing within the same sentence as the test word, i.e. its context. The parameters and their values were collected from the FidaPlus[3] corpus, a 100-million reference corpus of Slovene (Gorjanc 1999).

A subcorpus was extracted for each dataset in order to speed up the clustering but also because some sort of normalization of the corpus was required. We observed that without a normalization the occurrence of frequent words is disproportional compared to that of infrequent words to the extent that it completely overrides both the selection and distribution of parameters. This is why the same number of sentences for all instances from the datasets were included. If an instance was more frequent than the instance with minimum occurrence, only the number of sentences corresponding to the instance with minimum occurrence were randomly selected and included in the subcorpus.

We were also interested in finding the optimal number of parameters used for clustering. This is why we ran the tests in two different settings. In one we included all the parameters found in the corpus, and in the other we sorted the parameters in the descending order and only included 500 most frequent ones.

Our assumption was that some context features have greater importance for sense discrimination than others. We therefore tested and evaluated several variants of context selection:

- all (lemmatized) tokens within the same sentence (ALL)
- verbs and nouns (VN)
- only verbs (V)
- only nouns (N)
- only adjectives directly preceding the noun (A1)
- only verbs following the noun in question (V1)

The computed parameters were represented either by:

- the Binary Frequency, returning the values 0 or 1 according to the non-occurrence or occurrence of the context word in the corpus (BIN)
- the TFIDF measure, returning a value between 0 and 1 that is computed by multiplying Term Frequency by Inverse Document Frequency

The Inverse Document Frequency (IDF) weighing method is one of the most popular ones in text retrieval

[1] http://www.illc.uva.nl/EuroWordNet/
[2] http://www.ceid.upatras.gr/Balkanet/
[3] http://www.fidaplus.net

methods and language processing techniques (Robertson 2004). It was first proposed by Karen Spärk Jones (1972) and is based on counting the number of documents in the collection which contain the term in question. A query term which occurs in many documents is not a good discriminator and should therefore be given less weight than the one which occurs in few documents. Term Frequency (TF) which is the frequency of a given term in the document itself. In this case, the higher the frequency, the higher the importance of the term in this document.

$$\mathrm{tf} = \frac{n_i}{\sum_k n_k}$$

n_i: no. of occurrences of the considered term
$\sum_k n_k$: no. of occurrences of all terms

$$\mathrm{idf} = \log \frac{|D|}{|(d_i \supset t_i)|}$$

$|D|$: total no. of documents in the corpus
$|(d_i \supset t_i)|$: no. of documents where the term t_i appears (that is $n_i \neq 0$)

$$\mathrm{tfidf} = \mathrm{tf} \cdot \mathrm{idf}$$

Figure 2. The TFIDF function (adapted from Salton & Buckley 1988)

4. Text mining methods

Word sense discrimination has been a popular topic of interest in the past decade (see Schütze 1998; Pedersen & Bruce 1998). Its core problem is finding classes of similar contexts such that each class represents a single word sense. Contexts that are grouped in the same class represent a particular word sense.

As opposed to related methods, the McQuitty's Similarity Analysis produces a relatively small feature vector of a target word's morphological features, POS of the surrounding words and co-occurence features. A first-order vector is created for each context. These are then compared according to how many features they have in common.

4.1. Hierarchical clustering

Clustering is an unsupervised learning method. Given data about a set of instances, a clustering algorithm creates groups of objects following two criteria. Firstly, instances are close (or similar) to the other instances from the same group (internal cohesion) and secondly, they are distant (or dissimilar) from instances in the other groups (external isolation) (Vintar et al. 2003).

A particular class of clustering methods studied and widely used in statistical data analysis are hierarchical clustering methods. Their main advantage is that the number of clusters does not need to be specified in advance. The agglomerative hierarchical clustering is a bottom-up algorithm that merges clusters into larger and larger units. It starts with assigning each instance to its own cluster, and iteratively joins together the two closest (most similar) clusters. The distances between instances are provided as input to the clustering algorithm. The iteration continues until all instances are clustered into a single cluster (Manning et al. 2006).

The output of the hierarchical clustering algorithm is a hierarchical tree of clusters or dendrogram (see Figure 4) that illustrates the order in which instances are joined together in clusters. Initial clusters, consisting of a single element, form the leaves of the tree and each internal node represents a cluster that is formed by joining its children nodes. The height of the node is proportional to the distance between the joined clusters.

In the final step of the hierarchical clustering algorithm, the dendrogram is cut into sub-trees, producing separate clusters from elements in each sub-tree. Cutting the same dendrogram at different heights produces different number of clusters. The optimal "cut point" that produces clusters with maximal internal cohesiveness and minimal external isolation from a given dendrogram is where the difference between heights of two successive nodes in the tree is maximal (Todorovski et al. 2002).

4.2. Distance measures

4.2.1. Distance measures between data points

For any clustering the choice of measuring the distance between objects and clusters of objects is very important. The most commonly used distance measures are those which define distance between two n-dimensional vectors of real numbers. In the presented experiment, the Manhattan distance measure that captures the difference in the scale and baseline between objects (the sum of distances) was used (see Table 5).

4.2.2. Distance measures between clusters

Variants of the HAC algorithm differ in how similarity is defined. The most widely known distance measures used with hierarchical agglomerative algorithms are single (minimum), average and complete (maximum) links, also known as UPGMA (Purandare & Pedersen 2004). In single-link clustering, distance between two clusters is the distance between the nearest neighbors in those clusters. This single-link merge criterion is local. Attention is paid solely to the area where the two clusters come closest to each other. Other, more distant parts of the cluster and the clusters' overall structure are not taken into account. In complete-link clustering, the distance between two clusters is the distance between the furthest points in those clusters. This complete-link merge criterion is non-local; the entire structure of the clusters is taken into account. In average-link clustering the distance between two clusters is the average of the distances between all the points in those clusters. It is a compromise between the sensitivity of complete-link clustering to outliers and the tendency of single-link clustering to form long chains (Manning et al. 2006).

Shütze (1998) points out that single link clustering tends to place all instances into a single elongated cluster, whereas Purandare (2003) shows that average links produce satisfactory results. In order to determine which function would produce the best result in our setting, we decided to test all three.

5. Results and discussion

This section presents the results for a number of feature configurations, context representations and clustering algorithms. All the Test Settings that were run on each of the four Test Sets from Section 3.1 are listed in Table 1.

Test Settings

normaliz. of subcorpus	yes	no				
no. of parameters	500	unlimited				

context selection	ALL	NV	N	V	V1	A1
weighing method	BIN	TFIDF				
UPGMA link	single	average	complete			

Table 1. Test settings used for clustering

At an early stage of experiments it turned out that words in the Test Sets had highly uneven frequency distributions in the FidaPlus corpus. For example, the lemma *učitelj* occurred 13.241 times and the lemma *tovarišica* 253 times. This meant that overall, the parameters of the frequent lemma had much higher scores than the parameters of the rare one, and even though they shared many of the parameters, they were not treated as similar by the clustering algorithm.

This is why we normalized the subcorpus by including the same number of sentences for each word in a given Test Set, regardless of their number of occurrence in the corpus. First, the word with the least occurrences was identified and all the sentences in which it occurred were included in the subcorpus. As far as the rest of the words from the Test Set are concerned, only the same number (i.e. the number of the most infrequent word in the test set) of randomly selected sentences were included in the subcorpus. This approach yielded much better results, which is why the rest of the Test Settings were compared only on the normalized subcorpus.

The second feature we were interested in was to determine the extent of the context that is the most useful for successful clustering. The first motivation was a practical one; to speed up processing time. The second reason was of a more serious nature; because data sparseness is a well-known problem in clustering, the number of parameters should not be too low. On the other hand, too much noise can have a negative effect on the results as well. Bearing this in mind, we tested the feature with two different settings; once the parameters were sorted in the descending order according to the frequency of their co-occurrence with the words from the Test Sets and only used the 500 most frequent ones, and then all the parameters were used to compute distance measures.

Limiting the number of frequencies turned out to be useless when context selection was very restrictive (i.e. only verbs that appear directly after a given noun from the Test Set) because in most cases, there were less than 500 parameters in the first place. We also found that after having normalized the subcorpus, the datasets became much smaller so that processing time played no role. For example, the total number of parameters in Test Set 2 when all words from all parts of speech were used was 18.309. We believe that any noise brought into the data in this way could be reduced with the appropriate weighing method. We therefore decided not to limit the number parameters in further tests.

Next, we examined what context selection performs best. Our intuition was that different kinds of words co-occurring with the words from the test sets or patterns have varying impact on displaying their semantic (dis)similarity. Since our test sets contain nouns only, we tried to find the best carriers of semantic distance between them.

We repeated tests a number of times, allowing different context features each time: all the words that appear in the same sentence as a given word from the data set (ALL), only nouns and verbs from the same sentence (NV), nouns only (N), verbs only (V), adjectives that directly precede a test word (A1) only and verbs that directly follow a test word only (V1).

Before the corpus was normalized, none of the results were satisfactory but the A1 setting was by far the best. In the normalized corpus the ALL NV and N settings performed much better. The results in the A1 setting did not change while the V and V1 settings turned out to be the worst of all. When examining the distribution of parameters across different POS it becomes clear that good performance of the three settings can be explained by a rich representation of nouns in ALL and NV. In both Test Sets, nouns represent more than half of all the parameters in ALL and almost 80 per cent of all the parameters in NV (see Table 2). This means that the parameters in these two settings do not differ much from the N setting, explaining the similarity of the results obtained by clustering.

Test Set 1 (ALL)

N	8213	53,81%
A	3200	20,96%
V	2153	14,11%
other	1698	11,12%
total	15264	100,00%

Test Set 2 (ALL)

N	9293	50,76%
A	4098	22,38%
V	2540	13,87%
other	2378	12,99%
total	18309	100,00%

Test Set 2 (NV)

N	8240	79,28%
V	2153	20,72%
total	10393	100,00%

Test Set 2 (NV)

N	9327	78,60%
V	2540	21,40%
total	11867	100,00%

Table 2. Parameters across POS for Test Sets 1 and 2

Due to disproportionate frequencies of the words in our Test Sets a weighed representation of parameters found in the corpus was necessary. Two popular weighing methods were used; BIN and TFIDF. The latter performed significantly better in all the tests we ran. It is very interesting that both BIN FREQ. and TFIDF measure resulted in consistent clusters of male/female pairs for *teacher: učitelj/učiteljica, profesor/profesorica, tovariš/tovarišica*. BIN separates the frequent expressions (e.g. *učitelj/učiteljica, profesor/profesorica*) from their less frequent synonyms (e.g. *tovariš/tovarišica*). TFIDF does not repeat the same mistake and treats all the three pairs equally.

A comparison of graphs created based on shared features and Test Settings, with the three UPGMA options being the only distinction, reveals that single link produces the least satisfactory results. As already reported by Schütze (1998), all the single link graphs have a distinct cascade-like structure and are as such useless for our word task. Much better results were obtained from the average and complete links, with the complete performing slightly better in all the cases.

Table 3 shows clustering reports for the two Test Sets, the parameters for which were obtained from the normalized subcorpus and represented with the TFIDF measure. The HAC algorithm used the Manhattan distance measure and complete link. In both cases, 8 clusters were obtained. But this should not be considered as a bad result since the default cut point can be raised to the level which would leave us with two clusters only.

Finally, because we were interested in the capacity to deal with naturally fuzzy synsets, we selected the ones which contain mistakes resulting from polysemous literals in the source language. The two synsets were pre-edited in order to omit literals which do not occur in the FidaPlus corpus or which are very rare (less than 10 occurrences).

Test Set 1 (ALL, NV)

Manhattan DM (complete) - 8 cl.

cl. 1 (2 el.):	učitelj, učiteljica
cl. 2 (2 el.):	professor, profesorica
cl. 3 (2 el.):	tovariš, tovaršica
cl. 4 (1 el.):	mentor
cl. 5 (1 el.):	sreča
cl. 6 (1 el.):	veselje
cl. 7 (1 el.):	radost
cl. 8 (1 el.):	zadovoljstvo

Test Set 2 (ALL, NV)

Manhattan DM (complete) - 8 cl.

cl. 1 (2 el.):	mož, možak
cl. 2 (1 el.):	fant
cl. 3 (1 el.):	moški
cl. 4 (1 el.):	deček
cl. 5 (1 el.):	hrib
cl. 6 (1 el.):	grič
cl. 7 (1 el.):	vzpetina
cl. 8 (1 el.):	gora

Table 3. Clustering reports for Test Sets 1 and 2

The best results for Test Sets 3 and 4 were obtained by using the normalized subcorpus and an unlimited number of parameters. Context features were NA or N, weighed with the TFIDF measure. The Manhattan DM was computed and a complete link used in the algorithm. It is interesting that BIN performed well in these cases too. This combined with the fact that the ALL setting was less accurate with Test Sets 3 and 4, we took a closer look at their parameters (see Table 4). It turns out that (due to low occurrence of the lemma *odrastek* and consequently a small normalized corpus) there are very few parameters (1705). The nouns contributed less to the overall POS distribution in the ALL setting which influenced the final result. The output of the clustering algorithm is a graph of three clusters which could be cut at a slightly higher point, thus creating two correct clusters (see Figure 4).

Test Set 3

POS of parameters

N	727	42,64%
A	327	19,18%
V	213	12,49%
other	287	18,47%
total	1705	100,00%

Manhattan DM (complete) - 3 clusters

cl. 1 (2 el.):	odrastek, veja
cl. 2 (2 el.):	disciplina, stroka
cl. 3 (1 el.):	panoga

Table 4. Results for Test Set 3

6. Conclusions and Future Work

This paper presents a series of experiments aimed at grouping similar words using context features derived from the reference corpus of Slovene language. The motivation was to try to get around the knowledge-acquisition bottleneck by finding a suitable knowledge-lean approach that would be suitable for cleaning the fuzzy WordNet synsets obtained by automatic translation of Serbian synsets into Slovene. A number of features and settings were tested, after which the data was fed into the agglomerative hierarchical clustering algorithm. The analysis of the results has shown that the approach is promising and should be carried out on a larger scale.

All the experiments conducted performed better with normalized subcorpora and the TFIDF measure. Nearly all of them give better results with an unlimited number of parameters and with nominally strong contexts (ALL, NV, N). Although both average and complete links gave similar results, the height difference between sub-clusters was greater in complete links, making it easier to determine the appropriate cut-off point for clusters.

It must be noted here that with smaller quantities of data, there is less possibility of finding words which share the same contexts. They are more often conceptually related than lexically the same. This is why the approach is not as successful in such cases.

The presented approach is only a preliminary feasibility study and therefore only took into consideration nominal synsets. This is why the optimal feature and settings selection might not hold for verbs, adjectives and adverbs. We plan to carry out a comprehensive survey of optimal feature and settings selection for other parts of speech in the future.

Last but not least, an evaluation method of clustering results needs to be employed in order to enable a comprehensive comparison and evaluation of the results. This is not a trivial task as it is known to be challenging to evaluate the results without manually inspecting them or comparing them to a gold standard (e.g. Schütze 1998, Pedersen & Bruce 1998).

7. References

Agirre, E.; Edmonds, P. (2006): *Word Sense Disambiguation*. Algorithms and Applications. Dordrecht: Springer.

Brown, P.; Della Pietra, S. A.; Della Pietra, V. J.; Mercer, R. L. (1991): *Word-sense disambiguation using statistical methods*. In: Proceedings of the 29th Meeting of the Association for Computational Linguistics (ACL), Berkeley, U.S.A.

Burgess, C.; Lund, K. (2000): *The dynamics of meaning in memory*. Cognitive Dynamics: Conceptual Representational Change in Humans and Machines, ed. by Dietrich E.; Markman, A. 117-156. Mahmah, U.S.A.: Lawrence Erlbaum Associates.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; Harshman, R. (1990): *Indexing By Latent Semantic Analysis*. Journal of the American Society For Information Science, 41:391-407.

Erjavec, T.; Fišer, D. (2006): *Building Slovene WordNet*. In Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC '06, Genoa, Italy.

Fellbaum, C. (ed.) (1998): *WordNet: An Electronic Lexical Database*. MIT Press.

Gale, W.; Church, K. W.; Yarowsky, D. (1992): *Using bilingual materials to develop word sense disambiguation methods*. In: Proceedings of the 41th International Conference on Theoretical and Methodological Issues on Machine Translation, Montreal, Canada.

Gorjanc, V. (1999): *Korpusi v jezikoslovju in korpus slovenskega jezika FIDA.* In Proceedings of the 35st Seminar of The Slovene Language and Literature. Ljubljana, Slovenia. 47-59.

Gorjanc, V., Logar, N. (2005): *Od splošnih do specializiranih korpusov - načela gradnje glede na njihov namen*. In the Proceedings Razvoj slovenskega strokovnega jezika, Ljubljana, Slovenia, p. 16.

Hanks, P. (2000): *Do word meanings exist?* In Computers and the Humanities, 34(1-2): 205-215.

Harris, Z. (1968): *Mathematical Structures of Language*. New York: Interscience Publishers.

Ide, N.; Erjavec, T.; Tufis, D. (2002): *Sense discrimination with parallel corpora*. Proceedings of the ACL SIGLEX Workshop on Word Sense Disambiguation: Recent Successes and Future Directions, Philadelphia, U.S.A.

Kilgarrif, A. (1997): *I don't believe in word senses*. In Computers and the Humanities, 31(2): 91-113.

Lin, D. (1998). *Automatic retrieval and clustering of similar words*. In the Proceedings of COLING-ACL98, Montreal, Canada.

Lin, D.; Pantel, P.(2002): *Concept discovery from text*. Proceedings of the 19th International Conference on Computational Linguistics (COLING), Taipei, Taiwan 577-583.

Manning, C.; Prabhakar, R.; Schütze, H. (2006): *An introduction to information retrieval*. Draft. Cambridge University Press.

Miller, G. A.; Charles, W. G. (1991): *Contextual correlates of semantic similarity*. Language and Cognitive Processes. 6(1):1-28.

Pedersen T.; Bruce, R. (1998): *Knowledge lean word sense disambiguation*. In Proceedings of the Fifteenth National Conference on Empirical Methods in Natural Language Processing. Providence, RI, pp. 197-207.

Purandare, A (2003): *Discriminating among word senses using McQuitty's similarity analysis*. In Proceedings of the HLT-NAACL 2003 Student Research Workshop, Edmonton, Alberta, Canada. pp. 19-24.

Purandare, A.; Pedersen, T. (2004): *Word Sense Discrimination by Clustering Contexts in Vector and Similarity Spaces*. In the Proceedings of the Conference on Computational Natural Language Learning (CoNLL), Boston, USA

Robertson, S. (2004): *Understanding Inverse Document Frequency: On theoretical arguments for IDF*. Journal of Documentation, 60:503-520.

Salton, G.; Buckley, C. (1988): *Term-weighting approaches in automatic text retrieval*. Information Processing & Management 24(5):513-523.

Schütze, H. (1998): *Automatic word sense discrimination*. Computational Linguistics. 24(1):97-123.

Spark Jones, K. (1972): *A statistical interpretation of term specificity and its application in retrieval*. Journal of Documentation, 28:11-22.

Todorovski, L.; Cestnik, B.; Kline, M.; Lavrač, N.; Džeroski, S. (2002): *Qualitative Clustering of Short Time-Series: A Case Study of Firms Reputation Data*. In the Proceedings of the ECML/PKDD'02 workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning. Helsinki, Finland. pp 141-149.

Vintar, Š.; Todorovski, L.; Sonntag, D.; Buitelaar, P. (2003) *Evaluating Context Features for Medical Relation Mining*. In Proceedings of the Workshop on Text Mining and Data Mining for Bioinformatics, ECML/PKDD 2003.

Vossen, P. (ed.) (1998): *EuroWordNet: A multilingual database with lexical semantic networks*. Kluwer Academic Press.

	profesorica	učiteljica	tovariš	tovarišica	mentor	učitelj	profesor	veselje	radost	sreča	zadovoljstvo
profesorica	0	1.1834	1.4745	1.3767	1.0859	1.1252	0.5471	1.4238	1.5401	1.3118	1.3651
učiteljica	1.1834	0	1.4110	1.3103	1.1375	0.4871	1.1099	1.3551	1.4727	1.2742	1.3035
tovariš	1.4745	1.4110	0	0.8018	1.4432	1.3723	1.4585	1.4774	1.6137	1.3838	1.4460
tovarišica	1.3767	1.3103	0.8018	0	1.3625	1.3069	1.4134	1.4149	1.5217	1.3542	1.3832
mentor	1.0858	1.1375	1.4432	1.3625	0	1.0425	1.1046	1.3880	1.5193	1.2347	1.3286
učitelj	1.1252	0.4871	1.3723	1.3069	1.0425	0	0.9754	1.3120	1.4476	1.1995	1.2503
profesor	0.5471	1.1099	1.4585	1.4134	1.1046	0.9754	0	1.4286	1.5554	1.2965	1.3743
veselje	1.4238	1.3551	1.4774	1.4149	1.3880	1.3120	1.429	0	1.2023	1.1484	1.1696
radost	1.5401	1.4727	1.6137	1.5217	1.5193	1.4476	1.5554	1.2023	0	1.2850	1.3671
sreča	1.3118	1.2742	1.3838	1.3542	1.2347	1.1995	1.2965	1.1484	1.2850	0	1.1181
zadovoljstvo	1.3651	1.3035	1.4460	1.3832	1.3286	1.2503	1.3743	1.1696	1.3671	1.1181	0

Table 5. Manhattan distance measures for Test Set 1 (times 10^{-4})

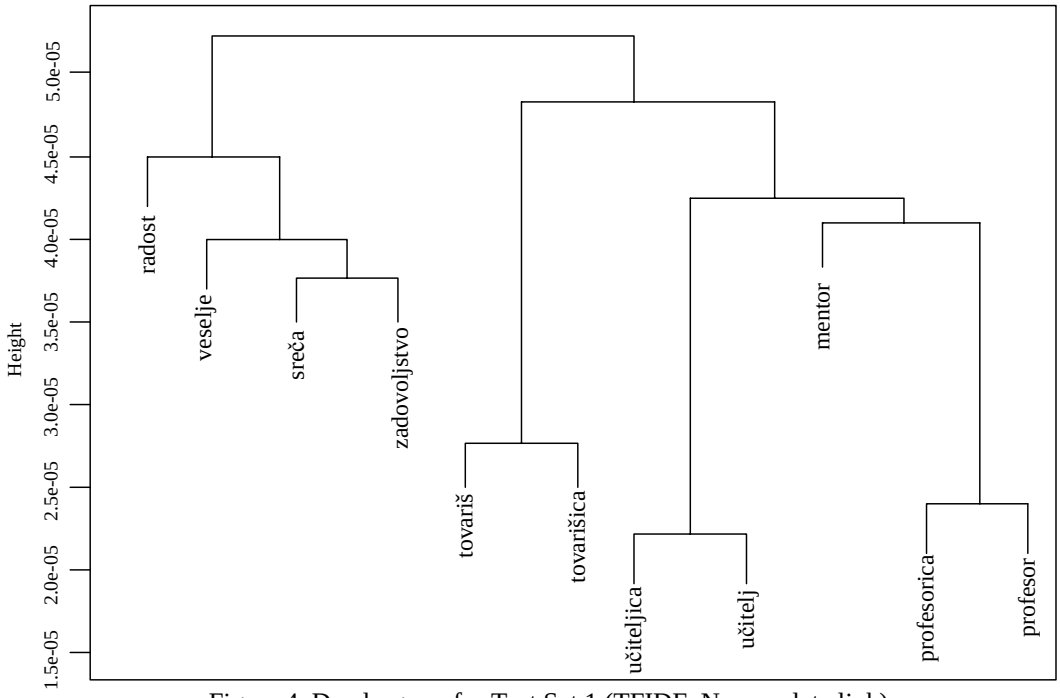


Figure 4. Dendrogram for Test Set 1 (TFIDF, N, complete link)

Slovenian to English Machine Translation using Corpora of Different Sizes and Morpho-syntactic Information

Mirjam Sepesy Maučec,* **Janez Brest,†** **Zdravko Kačič***

*Institute of Electronic and Telecommunication
Faculty of Electrical Engineering and Computer Science
University of Maribor
Smetanova 17, SI-2000 Maribor
mirjam.sepesy@uni-mb.si, kacic@uni-mb.si

†Institute of Computer Science
Faculty of Electrical Engineering and Computer Science
Smetanova 17, SI-2000 Maribor
janez.brest@uni-mb.si

Abstract

Word based statistical machine translation has emerged as a robust method for building machine translation systems. Inflective languages point out some problems with the approach. Data sparsity is one of them. It can be partly solved by enlarging the training corpus and/or including richer linguistic information: lemmas and morpho-syntactic features. Acquisition of a large bilingual parallel corpus for the desired domain and language pair requires a lot of time and effort. In this paper we report the performance comparison on training corpora of different sizes: 1k, 10k and 100k. Experiments were performed on small to middle-sized sentences of IJS-SVEZ corpus.

Strojno prevajanje iz slovenščine v angleščino s korpusi različnih velikosti in morfo-sintaktičnimi oznakami

Statistično strojno prevajanje na osnovi besed se kaže kot zelo obetavni pristop na področju strojnega prevajanja. Težavnost pregibnih jezikov je razpršenost podatkov. Delno jo rešujemo z večanjem korpusov za učenje in z uporabo dodatnih jezikovnih informacij: leme, in morfosintaktične oznake. V pričujočem članku analiziramo vplive različnih tipov jezikovnih informacij in različnih velikosti učnih korpusov. Pri eksperimentih smo uporabili IJS-SVEZ korpus.

1. Introduction

Research in statistical machine translation was pioneered at IBM (P. F. Brown and Mercer, 1993). They developed a language-independent framework, which was later re-implemented, improved, and the software has become freely available. Given these tools and a parallel corpus, a statistical machine translation system can be built in a relatively short time. The quality of the system closely depends on the features of the training corpus.

The historical enlargement of the EU has brought many new challenging language pairs for machine translation. A lot of work has been done on Czech (Čerjek et al., 2003), Polish (Jassem, 2004), Croatian (Brown, 1996), Serbian (Popović et al., 2004) and not at last Slovenian (Vičič and Erjavec, 2002; Romih and Holozan, 2002). This paper studies the translation direction Slovenian to English.

Acquisition of a large bilingual parallel corpus for the desired domain requires a lot of time and effort. Therefore, investigation of statistical machine translation with a small amount of training data is receiving more and more attention (Popović et al., 2004). In this paper we analyse statistical translation systems built on the largest Slovenian-English parallel corpus IJS-SVEZ (Erjavec, 2006). We analyse the results obtained with different amounts of training data, extracted from the same corpus.

2. Statistical Machine Translation

Statistical machine translation uses a notation of a source string $f_1^J = f_1...f_j...f_J$, which is translated into a target string $e_1^I = e_1...e_i...e_I$. In our experiments a source string is a Slovenian sentence and a target string is an English sentence. I is the length of the target string and J is the length of the source string. Among all possible target strings, the string with the highest probability as given by the Bayes' decision rule is chosen:

$$\hat{e}_1^I = \arg\max_{e_1^I} P(e_1^I|f_1^J) = \arg\max_{e_1^I} P(e_1^I) \cdot P(f_1^J|e_1^I) \quad (1)$$

$P(e_1^I)$ is the language model (of the target language) and $P(f_1^J|e_1^I)$ is the translation model. The $\arg\max$ operation denotes the search problem. In this paper we will focus on a translation model, which is based on an alignment model.

3. Translation Model

In the translation model the terms 'target language' and 'source language' are reversed. In the translation model the term 'target language' refers to the Slovenian language and the 'source language' refers to the English language. The translation model is based on word alignment. Given an English string e and a Slovenian string f, a word alignment is a many-to-one function that maps each word in f onto exactly one word in e, or onto the NULL word. The

NULL word is an invisible word in the initial position of an English sentence e_0. It accounts for Slovenian words that have no counterpart in the English sentence. More than one Slovenian word can be mapped onto the same English word. In the Slovenian string of words, we distinguish the heads from the non-heads. The head is the leftmost word of the group mapped to the same English word. All subsequent words in the same group are non-heads. A group of Slovenian words does not always contain neighbouring words. A sample of word alignment is shown in Figure 1. Each Slovenian word has its counterpart in an English sentence. Two Slovenian words ('Bil' and 'je') are mapped to the same English word ('was'). The word 'Bil' is a head word and 'je' is a non-head word. In this example, these two words are neighbouring words, but it is not always the case.

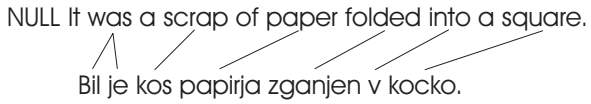


Figure 1: A sample alignment of sentence-pair.

An additional sample of word alignment is shown in Figure 2. The NULL word is an artificial construct in the initial position of an English sentence.

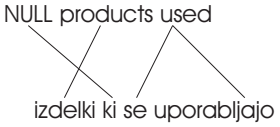


Figure 2: A sample alignment using a NULL word as a counterpart of Slovenian words that have no translation in English.

Word-for-word alignments of the translated sentences are not known. All possible alignments for a given sentence pair (e, f) are taken into account. An alignment for a sentence pair is denoted by a.

A series of five translation models (Model 1 to Model 5) were proposed by IBM (P. F. Brown and Mercer, 1993). Models 4 and 5 are the most sophisticated. We will focus on Model 4. Model 4 computes the probability $P(a, f|e)$ of a particular alignment and a particular sentence f given a sentence e. This probability is a product of five individual decisions:

$t(f_j|e_i)$ - translation probability. It is the probability of Slovenian word f_j being a translation of English word e_i.

$n(\phi_k|e_i)$ - fertility probability. An English word can be translated into zero, one or more than one Slovenian word. This phenomenon is modelled by fertility. The fertility $\phi(e_i)$ of an English word e_i is the number of Slovenian words mapped to it. The probabilities of different fertility values ϕ_k for a given English word are estimated.

p_0, p_1 - fertility probability for e_0. Instead of fertilities $\phi(e_0)$ of a NULL word, one single parameter $p_1 = 1 - p_0$ is used. It is the probability of putting a translation of a NULL word onto some position in a Slovenian sentence.

$d_1(\Delta j|\mathcal{A}(e_i), \mathcal{B}(f_j))$ - distortion probabilities for the head word. Δj is the distance between the head of current translation, and the previous translation. It may be either positive or negative. Distortion probabilities model different word order in the target language in comparison to the word order in the source language. Classes of words are used instead of words.

$d_{>1}(\Delta j|\mathcal{B}(f_j))$ - distortion probabilities for the non-head words. In this case Δj denotes the distance between the head and non-head word.

Model 4 has some deficiencies. Several words can lie on top of one another and words can be placed before the first position or beyond the last position in the Slovenian string. An empty word also causes problems. Training results in many words being aligned to the empty word. Model 5 is a reformulation of Model 4, in order to overcome some problems. An additional parameter is trained which denotes the number of vacant positions in the Slovenian string. It is added to the parameters of the distortion probabilities. In our experiments Models 4 and 5 will be trained, but only Model 4 will be used when decoding. Model 5 is not yet supported by the decoding program.

This was a short overview of the translation model. Readers interested in a more detailed description are referred to the paper (P. F. Brown and Mercer, 1993).

4. Adding Morphological Information

Previous work has shown that, for highly inflective languages, morphological information may be quite useful (Popović et al., 2004). The question arises, how much can be gained by adding morphological information.

A very basic way to modify input data using morphological information is by replacing each word-form with associated lemma. We expect this transformation would lead to an improvement in translation quality due to the restriction of data sparsity.

Since lemmatisation removes some useful information, we proceed by adding information from morpho-syntactic tags. These tags provide values along several morphological dimensions, such as part of speech, gender, number, etc. First only POS (Part Of Speech) tag is used, afterwards the complete MSD (Morpho-Syntactic Description) code is attached (Erjavec, 2004). In the latter case, data sparsity is increased because of homographs.

The translation model uses words grouped into classes. We analyse the influence of morpho-syntactic information on word grouping. The comparison was carried out between monolingual automatic clustering based on mutual information and clustering based on MSD codes.

In the following section four different sets of experiments are described, which differ in the ways the Slovenian lemma and morpho-syntactic tags are used.

The contribution of morphological information is closely related to the amount of training data and to its domain adequacy. We compare translation models, trained on different amounts of training data.

5. Experiments

5.1. SVEZ-IJS corpus

All experiments were performed on SVEZ-IJS corpus, a large parallel annotated English-Slovenian corpus. It contains approx. 10 million words of legal texts of the European Union, the ACQUIS Communautaire. The corpus is encoded in XML (according to TEI P4) and linguistically annotated at word-level. Tagging was performed by using TnT trigram tagger. Tagging accuracy for Slovenian was approx. 90%. CLOG (which is based on machine learning) was used for automatic lemmatisation. The estimated accuracy was approx. 95%. All corpus processing steps were performed by authors of the corpus and are described in some details in (Erjavec, 2006).

We discarded sentences longer than 15 words from the corpus, because of the computational complexity. The test set contained 25,000 sentences, taken at regular intervals from the corpus (homogeneous partition). The experiments were performed using three train sets, which differed in size (measured in sentences): 1k, 10k and 100k. There was no overlapping between the train and test sets. The vocabulary contained all units with occurance frequency (in the train set) greater than 2. All singletons (in training set) are mapped to the unique symbol UNK.

5.2. Tools

The experiments were performed using only publicly available third-party tools. The language model was trained by using the CMU-SLM toolkit (Rosenfeld, 1995). Classes of words were automatically created by means of the tool presented in (Maučec, 1997) and developed for language modelling. Translation model was trained using GIZA++ (Och and Ney, 2003). The decoding of test sentences was performed by the ISI ReWrite Decoder (Germann, 2003). Translations were evaluated using Word Error Rate (WER) and Bleu score (Papineni et al., 2001).

5.3. Translation model based on words

In our first set of experiments all word forms appeared as unique tokens and were exposed as candidates for word-to-word alignments. The Slovenian vocabulary (determined by the largest train set) contained 46,475 units (words). This vocabulary resulted in 5.0% OOV rate.

Before training, Slovenian words were mapped into 1000 classes and English words into 100 classes. A conventional trigram language model was built for the English language. The language model remained the same in all experiments. 10 iterations of training were performed for each translation model (1-5). The numbers of iterations were fixed for all experiments. Translation results are in Table 1. As expected, the error rate of the system trained on extremely small amounts of corpus is high. Using the 10-times larger train set the Bleu score improved by 89% relatively. When we used a train set of 100k sentences we obtained additional improvement of the Bleu score by 45%.

Train Set Size	WER [%]	Bleu [%]
1k	78.2	15.31
10k	61.0	28.92
100k	46.6	41.97

Table 1: Translation results. Translation model is based on word-forms.

5.4. Translation model based on lemmas

The purpose of the second set of experiments was the reduction of data sparsity. Here we used the lemmatised Slovenian part of the corpus. The English part remained unchanged. The Slovenian vocabulary (determined by the largest train set) contained 29,384 units (lemmas). The Slovenian vocabulary was reduced by 36% relatively (in comparison to the word-based translation model). This vocabulary resulted in a 2.7% OOV rate, which is 2.3% (absolute) lower than in the case of the word-based translation model. The translation results are in Table 2. A relative improvement is calculated to each value of evaluation metric (comparing the results with word-based baseline system). We achieved some improvements in the first two experiments, where data sparsity problem is more evident. In the last experiment we had worse results, because some information is lost by lemmatisation.

Train Set Size	WER [%]	Imp. [%]	Bleu [%]	Imp. [%]
1k	76.4	2.3	15.40	0.6
10k	59.3	2.8	30.41	5.2
100k	47.5	-1.9	41.36	-1.5

Table 2: Translation results. Translation model is based on lemmas.

5.5. Translation model based on lemmas and POS tags

We wanted to further examine the influence of morphosyntactic information is the translation process. Each Slovenian word was replaced by its lemma and the POS tag attached to it. The Slovenian vocabulary (determined by the largest train set) contained 30,450 units (lemmas with POS tag). This vocabulary resulted in a 2.9% OOV rate. Translation results are in Table 3. A relative improvement is calculated to each value of evaluation metric (comparing the results with word-based baseline system). In the first two experiments the improvement was not as evident as in the previous set of experiments with lemmas. In the last case (using 100k sentences in training) worsening of the Bleu score is smaller, because less information went astray.

Train Set Size	WER [%]	Imp. [%]	Bleu [%]	Imp. [%]
1k	76.3	2.4	15.38	0.5
10k	59.8	2.0	29.52	2.1
100k	47.7	-2.3	41.82	-0.4

Table 3: Translation results. Translation model is based on lemmas and POS tags.

5.6. Translation model based on lemmas and MSD codes

In this set of experiments we wanted to observe the influence of complete morpho-syntactic information. Slovenian words were replaced by lemmas and MSD codes were attached to them. The Slovenian vocabulary (determined by the largest train set) contained 59,339 units (lemmas, with MSD code). This vocabulary resulted in a 6% OOV rate.

In these experiments we expose the problem of homographs. For example the word *gori* can be replaced either by *goreti_*[VMIP3S–N] or by *gori_*[RGP], depending on the context. In addition, the problem of data sparseness increases. The translation results are in Table 4. The results

Train Set Size	WER [%]	Imp. [%]	Bleu [%]	Imp. [%]
1k	83.3	-6.5	10.35	-32.4
10k	66.5	-9.0	24.51	-15.2
100k	49.0	-5.1	40.60	-3.3

Table 4: Translation results. Translation model is based on lemmas and MSD codes.

were again compared against the word-based baseline system. We can see that using complete MSD code "adds a lot of noise" to the translation process. It should be noted that this observation depends tightly on the language pair under consideration and the direction of the translation. For example most MSD codes add usefull information to lemmas if we translate from one highly inflectional language to the other. The same is true, if we change the translation direction in our experiments.

5.7. Translation model based on word-forms and MSD classes

In the last set of experiments we used word forms as modelling units once again, but replaced automatic classes with classes based on MSD codes. Each distinct MSD code defines one class. All words having the same MSD code were mapped to the same class. The vocabulary size and OOV rate are the same as in first set of experiments (see Section 5.3.).

Train Set Size	WER [%]	Imp. [%]	Bleu [%]	Imp. [%]
1k	78.8	-0.8	15.42	0.6
10k	60.9	0.2	30.56	5.7
100k	47.1	-1.2	42.55	1.3

Table 5: Translation results. Translation model is based on word-forms and MSD classes.

The translation results are in Table 5. Comparing Bleu scores against the word-based baseline system shows, that MSD codes contain some information about word reordering between the source and target languages.

6. Conclusion

This paper reports our first experiments using SVEZ-IJS corpus. We were interested in the influence of morpho-syntactic information on statistical machine translation using different amounts of training data. Lemmatisation reduces data sparsity significantly and improves the results when using small training corpus. In the case of a large training corpus the performance deteriorated, because some useful information was lost. Using complete morpho-syntactic information is unwise choice due to the increase in data sparsity. It seems that only a subset of morpho-syntactic features is important, which depends on the language pair under consideration. Our future work will proceed in the direction of extracting useful morpho-syntactic features by a data driven approach.

7. References

R. Brown. 1996. Example-based machine translation in the Pangloss system. *In Proceedings of COLING-96*.

M. Čerjek, J. Cuřin, and J. Havelka. 2003. Czech-English dependency-based machine translation. *In Proceedings of the European Chapter of the ACL*, Vol. 1.

T. Erjavec. 2006. The English-Slovene ACQUIS corpus. *In Proceedings of the conference LREC*, pp.: 2138–2141.

T. Erjavec. 2004. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. *In Proceedings of the conference LREC*, pp.: 1535–1538.

U. Germann. 2003. Greedy Decoding for Statistical Machine Translation in Almost Linear Time. *In Proceedings of the HLT-NAACL-2003*. URL: http://www.isi.edu/licensed-sw/rewrite-decoder/.

K. Jassem. 2004. Applying Oxford-PWN English-Polish dictionary to machine translation. *In Proceedings of the 9th EAMT Workshop*.

M. S. Maučec. 1997. Statistical language modeling based on automatic classification of words. *In Proceedings of the workshop: Advances in speech technology*.

F. J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*. URL: http://www.fjoch.com/GIZA++.html.

V. J. D. Pietra P. F. Brown, S. A. D. Pietra and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2001. Bleu: a Method for Automatic Evaluation of Machine Translation. RC 22176(W0109-022), IBM Research.

M. Popović, S. Jovičić, and Z. Šarić. 2004. Statistical Machine Translation of Serbian-English. *In Proceedings of the SPECOM-2004*.

M. Romih and P. Holozan. 2002. Slovensko-angleški prevajalni sistem. *In Proceedings of the conference Jezikovne tehnologije*.

R. Rosenfeld. 1995. The CMU Statistical Language Modeling Toolkit, and its use in the 1994 ARPA CSR Evaluation. *In Proceedings of the ARPA SLT Workshop*. URL: http://www.speech.cs.cmu.edu/SLM/toolkit.html.

J. Vičič and T. Erjavec. 2002. Vsak začetek je težak : avtomatsko učenje prevajanja slovenščine v angleščino. *In Proceedings of the conference Jezikovne tehnologije*.

A Software Tool for Semi-Automatic Part-of-Speech Tagging and Sentence Accentuation in Serbian Language

Milan Sečujski*, Vlado Delić*

* Faculty of Engineering, University of Novi Sad
Trg Dositeja Obradovića 6, Novi Sad, Serbia
{secujski, vdelic}@uns.ns.ac.yu

Abstract

This paper presents a software tool for semi-automatic part-of-speech tagging, annotation of morphological categories and accentuation of texts in Serbian language. The software tool described in this paper is used for very efficient development of tagged text corpora in Serbian language since the accuracy of automatic POS tag and morphological category assignment is 87,2%. This result was obtained by testing the algorithm on a text containing 36692 words, and has turned out to be highly dependent on the type of text. The same algorithm for automatic POS tag and morphological category assignment can be included in text-to-speech systems, enabling correct accentuation of sentences, which, in turn, leads to fairly natural prosody. Within the test mentioned above, accent type and position were determined for each word based on automatically assigned POS tag, morphology-related information, as well as certain syntax cues, and correct accentuation assignment rate of 97,2% was achieved.

Programsko orodje za polavtomatsko oblikoskladenjsko označevanje in pripisovanje stavčnega poudarka v srbskem jeziku

V članku je prestavljeno programje za polavtomatsko oblikoskladenjsko označevanje, pripisovanje oblikoslovnih kategorij in mesta naglasa/poudarka besedilom v srbskem jeziku. V članku predstavljeno programsko orodje je uporabljeno za zelo učinkovit razvoj označenih besedilnih korpusov srbskega jezika; natančnost pripisovanja oblikoskladenjskih oznak je namreč okrog 88 %. Rezultat je bil dosežen s preizkušanjem algoritma na besedilnem korpusu velikosti 36.692 besed, izkazal pa se je za v veliki meri odvisnega od tipa besedil. Isti algoritem za avtomatsko oblikoskladenjsko označevanje je lahko vključen tudi v sisteme pretvorbe zapisanega v govorjeno besedilo, saj omogoča pravilen pripis stavčnega poudarka, ki vodi k precej naravni prozodiji. V zgoraj omenjenem preizkusu je bila na osnovi avtomatično pripisanih oblikoskladenjskih oznak pri ugotavljanju naglasnega tipa in mesta naglasa pri posamezni besedi dosežena natančnost 97,2 %.

1. Introduction

Current methods in language technology rely heavily on the use of large speech and text corpora. Text corpus collection and annotation are costly and time-consuming processes, and their perpetual necessity is the problem which every language community, especially the smaller ones, is facing. Most of the existing text corpora in Serbian language are not annotated. Exceptions include the Corpus of Serbian Language developed at the Institute for Experimental Phonetics and Speech Pathology in Belgrade, containing 11 million manually annotated words (Kostić, 2001), as well as the Serbian translation of George Orwell's "1984", the centrepiece of the MULTEXT-East resources for Serbian, containing about 90000 words (Krstev, Vitas, Erjavec, 2004). However, information related to accent type and position, essential for use of these corpora in high-quality text-to-speech synthesis and automatic speech recognition, is missing. Most of it could be recovered using an adequate dictionary, containing both morphology and accentuation information, however, some of the phenomena related to accentuation cannot be captured in this way (e.g. stressed vs. unstressed personal pronoun forms, falling accent shifting onto the preceding clitic etc.). This is why the AlfaNum team for development of speech technologies at the Faculty of Engineering in Novi Sad, Serbia, opted for developing a software tool for semi-automatic POS tagging and accentuation, with intention to use it for development of a large annotated text corpus.

2. Goal of the paper

This paper presents a possible solution for automatic assignment of POS tags as well as tags related to values of morphological categories and accent type and position to words in Serbian language. The algorithm explained in this paper is used for efficient development of annotated text corpus in Serbian, within a software tool developed for that purpose. Input text is tokenized and annotated automatically, and the visually intuitive software enables very efficient manual correction of errors. In this way a correctly annotated text corpus is developed semi-automatically. At the same time the comparison between initial and manually modified tags gives an estimate of the accuracy of the algorithm and points out the most frequent error types, enabling further improvement of the algorithm.

3. On Serbian language

Owing to significant dependence of natural language processing techniques on target language, some attention should be given to general features of Serbian language before proceeding to specific details of the algorithm.

Serbian language is an Indo-European, South-Slavic language, with 10 million speakers in Serbia (11 million world-wide) (Grimes, 1996). Like other Slavic languages, it exhibits some interesting features that prove challenging to natural language processing technologies.

It exhibits a high degree of inflection – a complete overview of grammatical categories is too complex to be presented here, but the fact that a complete declension of adjectives consists of seven grammatical cases, three

genders and two numbers, including suppletive forms for neutral plural as well as separate forms for dual/paucal, can serve as an illustrative example. Derivation with the use of prefixes and suffixes is also quite common, and the word order has significant freedom.

Complexity of morphology in Serbian language makes the dictionary size extremely large, causing well-known sparse data problems to statistically oriented language models based on *N*-grams. This fact together with the significant freedom of word order makes basic *N*-gram based models an unlikely choice for practical use in POS tagging. Since in languages with relatively free word order the information needed for accurate POS tagging lies in morphological categories of the words rather than in word order, a strategy aiming at accurate POS tagging should include a grammatically controlled search in sentence parsing.

Regarding the use of this algorithm within text-to-speech systems, some other features of Serbian language should be taken into account. The accentuation system of Serbian language is rather complex. Serbian is a tonal language, meaning that words possess inherent pitch patterns. Accented syllables are termed either rising or falling, and contain a long or a short vowel. Traditional notation in grammars and dictionaries combines these two features using four accent marks. Complexity of the accentuation system in Serbian aggravates the task of automatic accent assignment since the number of options is much greater than the number of syllables. It is, however, very important that it be as accurate as possible, since in a tonal language such as Serbian many minimal word pairs that differ only in accent can be found. Such minimal pairs can exist both within the inflection of a single lemma, such as in *žèna* (*woman*, n. nom. sg.) vs. *žéna* (*woman*, n. gen. pl.) as well as across different lemmas, such as in *céne* (*price*, n. nom. pl.) vs. *cêne* (*appreciate*, v. pres. 3. pers. pl.)). The problem is further complicated by the fact that vowel timbre can also vary with accent type, and that errors in accent assignment can introduce vowel timbre errors in synthesized speech.

Although lexical accent type does not depend on syntax in general, in some situations grammatical status may be insufficient for correct accent assignment. The most notable cases are when a clitic has a nonclitic homographic alternant, as is the case with some personal pronouns (*nas*, *vas*), when a falling accent shifts onto the preceding clitic (in modern language this can happen when negative particle *ne* is added to a verb form), as well as within certaint frequent collocations.

4. Tagging algorithm

POS tagging process relies on a dictionary containing more than 80,000 lemmas. Since there are many irregularities in inflection of Serbian words (especially nouns and verbs), all the inflected forms were included in the dictionary as separate entries, each of them containing lemmatization and morphology information, as well as information regarding accent type and position. Another reason for including all inflected forms as separate entries was the fact that accent can vary along with inflections of the same word, and that those variations are predictable only to a certain extent. However, the number of entries in such a dictionary exceeds 3 millions. The number of possible combinations of different parts-of-speech and values of morphological categories used in the dictionary exceeds 1200. Whereas tagset size affects the accuracy of automatic tag assignment (Manning, Schütze, 1999), it is still important to keep the tagset complex enough so as to avoid losing important information that may be needed for a particular application such as accentuation of unknown texts (Hladká, 2000).

4.1. Tagging procedure step by step

POS tagging procedure can be divided into several steps. After the initial tokenization of the input text, the words are looked up in the dictionary and a list of all possible POS and morphologic category values that correspond to given inflectional forms is created. In languages with poor inflection, tags usually contain only POS information, whereas in highly inflective languages tags usually contain much more information. The next step consists of context analysis, which considers a word in its context and seeks to determine its tag given the possible tags of neighbouring words. The result of context analysis is a list of words with their corresponding tags, as well as accentuation pattern, which is even more important from the point of view of speech synthesis.

Each of the steps listed above is wrought with difficulties. To begin with, some of the words may not be found in the dictionary, as is the case with many proper names and words including nonstandard affixes. Therefore strategies for assigning correct lemmatization and morphology information must be defined. Some of the strategies for overcoming that problem include making analogies based on standard prefixes and suffixes and rhyming. For example, having failed to find the infrequent word *podleteti* (*v*. to fly under sth.), the system searches the dictionary sorted in "rhyming order" and very soon comes upon the infinitive verb *uleteti* (*v*. to fly into sth.). Knowing that both *pod-* and *u-* are standard verb prefixes, the system will conclude (correctly) that *podleteti* is also an infinitive verb. Such a procedure is not entirely error-free, but it performs well in practice, and provides us not only with morphologic information but with accent type and location as well, since words derived in the same way are likely to possess the same accentuation pattern.

The input data for context analysis consist of a list of possible tags of all words in the sentence. As it would be impossible to consider all tag combinations separately, an algorithm similar to dynamic programming is used, keeping the number of partial hypotheses under control.

Let us consider a sentence $W = w_1 w_2 ... w_N$. Each of the words w_i has a corresponding tag list:

$$T_i = \{t_{i1}, t_{i2}, ... t_{iN_i}\}, \quad (1)$$

and its actual tag t_i is one of the t_{ij}, $j = 1, 2, ... N_i$. Initially only the hypotheses of length one are considered, containing only the first word of the sentence:

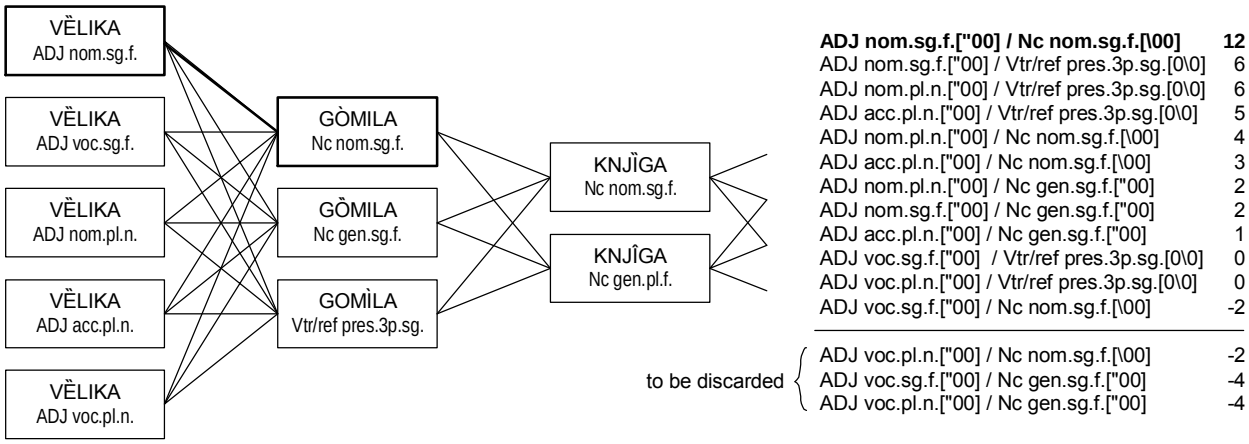


Figure 1. An example of a step in the disambiguation algorithm for the sentence "*Velika gomila knjiga stoji na stolu*" ("*A large | heap | of books | stands | on | the table*"). The diagram shows the situation after all the hypotheses of length two are considered, and three of them with lowest scores are to be discarded (in this example stack size limit is $L = 12$).

$$H_1 = \{(t_{11}), (t_{12}), \ldots (t_{1N_1})\}. \quad (2)$$

In every following step of the algorithm, each variant of the next word is combined with each of the existing partial hypotheses. A set of all hypotheses of length two is thus:

$$H_2 = \{(t_{1m}, t_{2n}) \mid m = 1, 2 \ldots N_1, n = 1, 2 \ldots N_2\}. \quad (3)$$

Each time a new word is appended in such a way, the score of each partial hypothesis is recalculated, based on the likelihood that a word with such a tag can follow. If the number of all hypotheses exceeds a previously set limit L, only L hypotheses with highest scores are retained, and all the others are discarded. The procedure continues until all words are included and the hypothesis with the highest score is selected as the estimate of actual tag sequence $T = t_1 t_2 \ldots t_N$. Fig. 1 shows an example of such analysis. The algorithm described here performs in time proportional to the length of the sentence, and one of its interesting features is that it produces partial results very quickly. The first word in the sentence is assigned its tag long before the analysis is over, which is consistent with the notion that, when reading a sentence, humans are usually able to start pronouncing it far before they reach its end, and that they organize the sentence into simple prosodic units which can be obtained from local analysis (Dutoit, 1997). Furthermore, this feature of the algorithm is especially useful from the point of view of speech synthesis, because synthesis of the speech signal can start as soon as the first partial results are obtained, which minimizes the delay introduced by context analysis.

The criteria for the actual scoring of the hypotheses are based on rules defined according to the statistics of different parts-of-speech in Serbian language, as found in (Jovičić, 1999), as well as most regular short-range dependencies among them, as found in (Stanojčić, Popović, Micić, 1989). For instance, since it is known that adjectives modifying a noun have to agree in gender, number and case with the noun in question, hypotheses where such pairs adjective-noun occur are considered more likely than hypotheses containing mismatching pairs. Further rules based on dependencies among specific words have also been defined in case it has been proven that such rules could eliminate a significant number of errors still present after the application of an algorithm based on rules of general type only. Some of the templates for rules of general type are as follows:

Award n points to a partial hypothesis $h = (w_1, w_2, \ldots w_l)$:

- If w_l is tagged t_i
- If w_l is tagged t_i and w_{l-1} is tagged t_j
- If w_k is tagged t_i, w_{l-1} is tagged t_j and w_{l-2} is tagged t_k
- If w_l is tagged t_i and w_{l-1} is tagged t_j and the value of a morphologic category c contained in the tag t_i is the same (is not the same) as the value of the corresponding morphologic category contained in the tag t_j
- If w_l is tagged t_i and w_{l-1} is tagged t_j and all of the values of morphologic categories $c_1, c_2, \ldots c_k$ contained in the tag t_i are the same (are not the same) as the values of corresponding morphologic categories contained in the tag t_j

After the (presumably) correct tag sequence has been discovered, the next step consists of modifying accent patterns to account for some words changing their accent type and/or location in a specific context, as described in previous section. If the algorithm is used within a text-to-speech synthesis system, this accent pattern will be used for obtaining a rich prosody structure, defining phoneme durations and variations of fundamental frequency and energy in time.

4.2. Testing the algorithm

Accuracy of POS tagging and assignment of morphological categories and accentuation patterns are of great importance for efficient development of language resources as well as for high quality and naturalness of synthesized speech. If a wrong accentuation pattern were assigned to the sentence, or if there were errors in identification of syntactic units, the resulting f_0 curve would

carry misleading prosodic information and a human listener would have trouble recognizing what had been said. The remarkable importance of lexical accent was shown in (Sečujski et al., 2002), where an experiment is described in detail, showing the improvement in human speech recognition caused by introducing accentuation-based prosody into synthesized speech in Serbian language. Twelve listeners were given synthesized sentences corrupted with noise and were asked to recognize what had been said. As many as 83% of the sentences with prosody based on accentuation only were correctly identified at once, compared to 52% of the sentences without any f_0 variations and 31% of the sentences with misleading accentuation. F_0 contours of the sentences were constructed by concatenating and postprocessing initial word f_0 contours based on accent type, location and position relative to a punctuation mark. Speech signal was synthesized by concatenation of prerecorded speech segments selected from a large speech database at runtime, according to (Beutnagel, Mohri, Riley, 1999) using TD-PSOLA model described in (Dutoit, 1997).

It is clearly of interest to establish the accuracy of such an algorithm on a large text corpus. The algorithm described in this paper was tested on a small text corpus containing 3064 sentences (36692 words). The corpus consists of three parts. Part 1 contains 1144 relatively short sentences of general type content (7054 words). Part 2 contains 915 medium length sentences from children's stories (11239 words). Part 3 contains 1008 relatively long sentences from encyclopedic articles (18399 words). Results presented in Table 1 show that 4.03% words in the entire corpus were assigned incorrect POS, 12.77% were assigned incorrect POS or values of one or more morphological categories, and 2.78% were assigned incorrect accent type and/or location. A certain dependence of accuracy on text type was also observed. Results of tests on Part 2 of the corpus were significantly inferior to the others. The subsequent analysis showed that main reasons for this included relatively free word order and somewhat archaic language with frequent use of aorist tense. Aorist forms of a number of verbs are heterophonous homographs to corresponding forms of present tense, and thus errors in morphologic category annotation lead to errors in accentuation. There is a similar relationship between present tense and imperative verb forms, and the number of errors that occur as a consequence depends on the type of text, since imperative forms are rare in encyclopedic articles, but quite often in children's stories.

	Part 1	Part2	Part3	Total
Words	7054	11239	18399	36692
POS	4.58%	4.09%	3.78%	4.03%
POS, morph.	12.94%	13.58%	12.21%	12.77%
Accentuation	2.55%	3.29%	2.55%	2.78%

Table 1: Percentage of error in POS, morphology and accentuation assignment.

5. Conclusion

In this paper a software tool for semi-automatic part-of-speech tagging and sentence accentuation in Serbian language was presented. The rule-based algorithm for initial automatic POS tagging is based on statistics of different parts-of-speech as well as regular short-range dependencies between them as found in available literature. Results of testing this algorithm on a text corpus containing 36692 words show that there is still some room for improvement as regards annotation of part-of-speech tags and morphology-related information. Reducing this error should lead to further reduction of the number of errors related to accentuation, which is of special importance for application of this algorithm within text-to-speech systems. One of the main sources of errors is the lack of more sophisticated knowledge of lexical short-range dependencies. The second one is the inherent inability of such an algorithm to capture long-range dependencies between words, which is one of the main shortcomings of *N*-gram based algorithms in general. The third and the most difficult one is lack of the "knowledge of the world", which still remains largely unsolved.

6. Acknowledgment

This work was supported in part by the Ministry of Science and Environment Protection of the Republic of Serbia within the Project “Interdisciplinary Research into Speech and Language Resources of Serbian Language”.

7. References

Beutnagel, M., Mohri, M., Riley, M., 1999. Rapid Unit Selection from a Large Speech Corpus for Concatenative Speech Synthesis. In Proc. of EUROSPEECH’99, Budapest, 607-610.

Dutoit, T., 1997. *An Introduction to Text-to-Speech Synthesis*. Dordrecht: Kluwer academic publishers, 149-152.

Grimes, B. F., 1996. *Ethnologue – languages of the world*. SIL International, SIL International.

Hladká, B., 2000. *Czech language tagging*. Ph.D. thesis, Institute of Formal and Applied Linguistics, Charles Univ., Prague.

Jovičić, S., 1999. *Speech communication: physiology, psychoacoustics and perception*. Belgrade: Nauka, 98-103.

Kostić, Đ., 2001. *Quantitative description of Serbian language structure: Corpus of Serbian language*, Institute for Experimental Phonetics and Speech Pathology, Faculty of Philosophy, Belgrade.

Krstev, C., Vitas, D., Erjavec, T., 2004. MULTEXT-East resources for Serbian. In Proc. of IS-LTC, Ljubljana, 108-114.

Manning, C., and Schütze, H., 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA, ch. 3.

Sečujski, M., Obradović, R., Pekar, D., Jovanov, Lj., and Delić, V., 2002. AlfaNum System for Speech Synthesis in Serbian Language. In Proc. of 5th Conf. Text, Speech and Dialogue, Brno, 8-16.

Stanojčić, Z., Popović, Lj., and Micić, S., 1989. *Contemporary Serbo-Croatian language and the culture of expression*. Belgrade: Zavod za udžbenike i nastavna sredstva.

The iTEMA E-mail Reader

Jerneja Žganec Gros[1], Vlado Delić[2], Darko Pekar[3], Milan Sečujski[2], Aleš Mihelič[1]

[1]Alpineon R&D, Iga Grudna 15, Ljubljana, Slovenia
[2] Faculty of Engineering, Trg D. Obradovića 6, 21000 Novi Sad, Srbija
[2] AlfaNum, Trg D. Obradovića 6, 21000 Novi Sad, Srbija

Abstract

The iTEMA project is outlined within the paper. The project aims to develop the iTEMA e-mail reader: a user-friendly solution to e-mail access over the telephone. By using iTEMA, users will be able to listen to the received e-mails in a variety of European languages, with an emphasis on ex-Yugoslav languages: Slovenian, Serbian, Croatian, Bosnian and Macedonian. They will also be able to choose between basic responses to the heard email and to save or delete individual messages. By the end of the project, a toll-free number will be provided in all the participating countries where free voice-enabled email access over telephone lines will be enabled for the chosen end-user target groups with disabilities, in particular for blind and visually-impaired persons.

Samodejni bralnik elektronske pošte iTEMA

V okviru projekta iTEMA nameravamo razviti aplikacijo računalniško podprte telefonije za telefonski dostop do elektronske pošte. Namen projekta je omogočiti skupini slepih in slabovidnih oseb ter ostarelim, da preko govornega vmesnika dostopajo do elektronske pošte. Telefonski bralnik elektronske pošte je zanimiv tudi za individualno in poslovno uporabo, saj omogoča oddaljen dostop do elektronske pošte preko telefona, denimo na poti v službo, na služenem potovanju, ipd. Govorna aplikacija bo dostopna slepim in slabovidnim uporabnikom na brezplačni številki in bo podpirala jezike, ki se uporabljajo na področju nekdanje Jugoslavije.

1. Introduction

Recent initiatives involving e-inclusion aim to prevent risks of 'digital exclusion', which is to ensure that disadvantaged people are not left behind and to avoid new forms of exclusion due to a handicap, lack of digital literacy or of Internet access. At the same time e-inclusion means also tapping new 'digital opportunities' for the inclusion of socially disadvantaged people and less-favoured areas.

The Information Society has the potential to distribute knowledge resources more equally and to offer new job opportunities, also by overcoming traditional barriers to mobility and geographic distance.

The strategic challenge for e-inclusion applications and services is thus twofold:

- to fully exploit the ICT potential to overcome traditional forms of social exclusion,
- to ensure that all citizens benefit from the Information Society.

The third objective of the recent initiative i2010 Communication: "an Information Society that is inclusive, provides high quality public services and promotes quality of life" [i2010], brings new challenges for e-inclusion. Furthermore, the year 2007 has been proclaimed by the EC as the “European year of equal opportunities for all”, as a joint effort to promote equal opportunities and to prevent discrimination [Delić05].

The iTEMA project directly tackles a large target group of end-users within the e-inclusion iniciative, namely the blind and visually impaired community, along with the elderly – the ageing population, which is often much more expirienced in using telephones than computers. Both target groups are not able to use one of the most widely spread basic communication means of our time – the electronic mail or e-mail.

First an overview of the iTEMA project goals is provided. We continue by describing the tenative system architecture and system modules. An implementation plan and evaluation plans are provided by the end of the paper.

2. iTEMA E-mail Reader

The iTEMA project will research and develop a multilingual CTI application for voice-enabled telephone access to user e-mails. Free access will be provided for user groups with disabilities, esp. to blind and visually impaired persons.

The project will focus on the following main technological issues:

· analyse the state-of-the-art of how people with visual disabilities and the elderly access voice-enabled services;

· analyse end-user requirements during the project and tune the resulting application according to the resulting findings of the research;

· specify standards for technologies and application programming interfaces that are being addressed within the project; with a view to enable easy access by disadvantaged groups of citizens;

· research, develop and utilize speech technologies, voice-enabled browser applications and e-mail access and processing techniques, which provide the building blocks of the proposed iTEMA system;

· implement an intelligent multilingual telephone e-mail reader using sophisticated text-to-speech engines to provide voice output of the user e-mail messages.

The main project goals and the planned implementation of the system are further outlined in the subsequent chapters.

3. User Requirements

State-of-the-art of the services and applications for users with disabilities and the elderly will be analysed from the point of view of usability in real practice in the countries of participants. Special attention will be paid to voice-enabled services.

User requirements for a voice-enabled e-mail reader application will be collected and digested. Based on their analysis a final set of recommendations for the development of an e-mail reader application will be compiled.

4. System Architecture

The system architecture of the e-mail reader application of the iTEMA system will be defined with respect to the results of the recommendations based on the user requirements. The system architecture will be upgraded from existing speech application architectures from the project partners.

Common standards for technologies and application programming interfaces with a view to enable easy access of disadvantaged groups of citizens will be identified and applied.

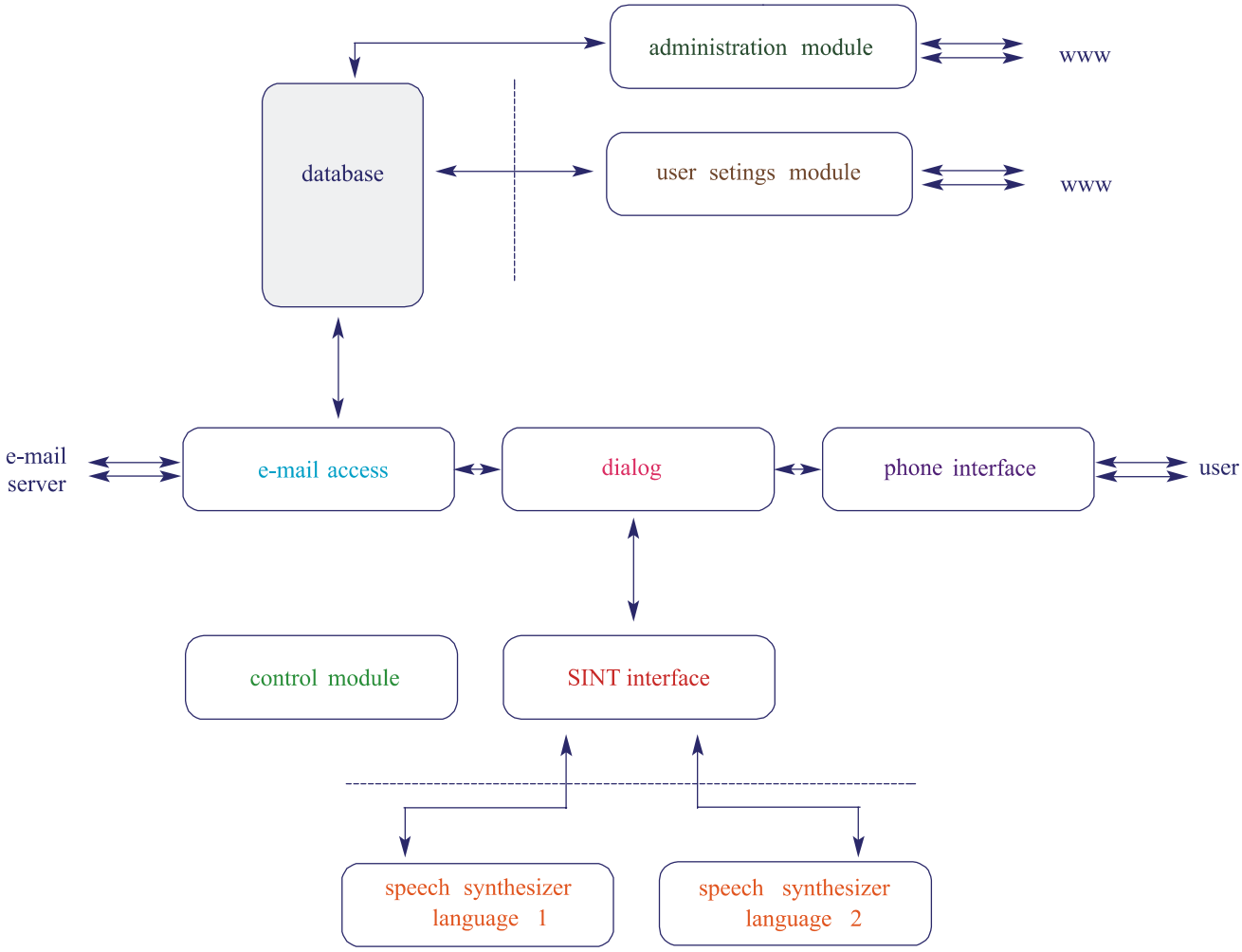


Figure 1. System architecture of the iTEMA e-mail reader.

A communication workflow and a failure behaviour protocol between the system modules will be defined.

A preliminary version of the iTEMA system architecture is depicted in Figure 1.

5. System Modules

According to Figure 1, the iTEMA system consists of the following modules:

a) speech synthesiser:
- converts input text to speech signal,
- saves speech signal into speech WAV files,
- for each language supported a TTS engine is needed.

b) SINT interface:
- coordinates requests for speech synthesis,
- performs language identification,
- performs time optimization for speech synthesis requests.

c) phone interface:
- accepts user calls and oversees the phone connection with the user,

- detects user commands,
- transmits messages to the user.

d) mailbox access:
- initializes a connection with the e-mail server (POP3 protocol),
- delivers e-mail data from the e-mail server,
- analyzes the content and format of the e-mail messages,
- accesses user settings from the database.

e) dialog module:
- controls the dialog between the application and the user,
- coordinates the actions of other parts of the system.

f) control module:
- oversees the correct start-up of the system,
- oversees the other system components and provides system self-repair.

g) administration module:
- enables viewing of user settings,
- enables the administrator to change the parameters within the system,
- enables inspection of system load and history of access to the system.

h) user settings web interface:
- enables setup of user settings.

In order to enable maximum quality speech output, two speech synthesis methods will be used in parallel. Application-specific voice message chunks will be prerecorded and concatenated in run-time to provide user menu guidance, give information on the number of new e-mails, the number of attachments, receipt dates, etc. All other input text will be rendered using language-specific text-to-speech synthesis methods.

Special attention will be devoted to the development of high-quality text-to-speech systems for all target languages (Slovenian, Serbian, Croatian, Bosnian, Macedonian). State-of-the art methods for text preprocessing, phoneme-to-allophone conversion, prosody modelling and speech production will be applied [Mihelič06], [Žganec Gros06], [Sečujski05].

Intelligent e-mail processing will be developed. E-mail messages will be parsed into various fields: sender, subject, receipt date, number of attachments, and most importantly, the message body. When processing the message body, it will be taken into account that some characters in the target languages (mainly those with diacritics, like č, ć, ž, š, đ), are not included in a standard ASCII codepage, and therefore many users tend to replace them with their closest ASCII relatives. This problem needs to be properly addressed in the speech synthesis module.

A n-gram language identification module will be developed which will classify and assign the input text to one of the target languages; it will support various code pages for different alphabets, e.g. Serbian Cyrillic.

The dialog module needs to be developed in a user friendly way so that the user can comfortably navigate through the application. It has to enable user log-in and application menu navigation.

The user should be able to choose to listen to her/his old/new e-mail messages, delete them or respond to the sender of the message using one of your pre-set e-mail replies. While listening to the content of an e-mail, the user can choose to: move to the start/end of the message, move to the next/previous sentence, listen to the next/previous message, delete/reply to a message, return to the menu. The dialog flow will be specified in XML.

6. User Settings

The multilingual User Settings web interface will enable remote setup of user profiles through HTTP clients. Before using the iTEMA system users will have to update their user profile through a web interface.

The following parameters will be mandatory – they will have to be set for a normal usage of the iTEMA system:

- User telephone number: telephone number that the user will most frequently use to access the iTEMA system, 10 digits;
- New password: PIN code;
- E-mail server address: name or IP address of the e-mail server storing user e-mails;
- E-mail server access protocol: e.g. POP3 or Secure POP3 (protocol for contacting the e-mail server);
- E-mail server access port: usually 110 (port for e-mail access on the e-mail server using a chosen protocol);
- User name on the e-mail server: e.g. test (user identification name on the e-mail server);
- User password on the e-mail server: ******* (user identification password on the e-mail server);
- SMTP server address for sending e-mail replies: server name or IP address of the server for sending e-mails;
- SMTP server access port: usually 25 (port for accessing the SMTP server);
- User reply ID: e.g. Test (reply-to user name – friendly name used for replies to e-mail messages).

On the iTEMA User Settings Pages users will be able to set additional parameters that will be optional: three preset answers for replies to the received e-mail messages and two filter lists for filtering the e-mail messages that the user wants to access via the iTEMA system:

- List of desired senders: this list contains e-mail addresses or a domain name of those senders whose e-mail messages the user explicitly wishes to access using the iTEMA system.

Examples:

*: all senders are permitted;

info@alpineon.com: only this sender is permitted.

- List of undesired senders: this list contains e-mail addresses or domain names of those senders whose e-mail messages the user does not wish to access using the iTEMA system - blocking sender list.

7. Implementation and Testing

Towards the end of the project, system integration of all modules will be performed, resulting in a 1st prototype of the iTEMA system.

A toll-free number will be procured for a test implementation. Testing will be performed by target user groups and their user-feedback will be collected. In parallel, system performance and robustness testing will be carried out.

A typical scenario of using the iTEMA system would be the following. The system log-in will include the following steps:

Step 1 The user dials a telephone number that will connect her/him to the iTEMA e-mail reader application. The system responds with a greeting and gives the user further instructions.

Step 2 In case the user makes the call not from his own telephone - that is used as his user ID in the system - but from another telephone, he first needs to enter his telephone number and press the pound # key. If the call is originated from the user's own telephone Step 2 can be skipped.

Step 3 After identifying the user the system invites the user to enter his password (PIN code). The user enters a 4-digit PIN code and presses the pound # key.

Step 4 After verification of the username and password the Main Menu will be prompted.

After initial verification of the username and password the Main Menu will be prompted. The Main Menu consists of the following submenus: New messages, Old messages, Settings, Help. For navigating the system menu structure the user has to use the Function Keys.

If the user chooses to listen to the messages, the system will advise the user on the number of new messages in the mailbox. The new messages are those that are new from the last usage of the system.

The system will start by reading the first message in the detected language (usually two language options are available) and will include the following information:

- Name of the sender;
- Timestamp;
- Subject;
- Content of the message;
- Information on attached files.

After completion of - and already while - listening to an e-mail, the user has the option to respond to the heard e-mail. The user has to pre-set default replies via the User Settings web interface. Upon selection of a pre-set reply the system accesses the original sender's e-mail address and creates a new e-mail reply. Each e-mail can also be deleted from the user's mailbox. The user can interrupt an e-mail at any given time and move back or ahead to another e-mail or another sentence in the current e-mail.

When the system is left waiting too long for the user's selection, it will automatically explain the various viable menu options. After two consecutive occurrences of non-response from a misunderstood command, the iTEMA system will automatically say 'goodbye' and disconnect.

Upon completion of a step within the system, the user will be given various options to choose from. The user will have the option of pressing the Help key, should she/he need any additional assistance. Basic instructions indicating how to use the system will be available to the user at all times.

Final system revisions will be performed before the iTEMA system service launch in all participating countries.

8. Conclusion

The iTEMA project outline has been provided within the paper. The project aims to develop the iTEMA e-mail reader: a user-friendly solution to e-mail access over the telephone. By using iTEMA, users will not only be able to listen to the received e-mails in a variety of European languages, with an emphasis on ex-Yugoslav languages, i.e. Slovenian, Serbian, Croatian, Bosnian and Macedonian, but will also be able to choose between basic responses to the heard email. In addition, they will be able to manage their email database by saving or deleting individual messages.

By the end of the project, a toll-free number will be provided in all the participating countries where free voice-enabled email access over telephone lines will be enabled for the chosen end-user target groups with disabilities, in particular for blind and visually-impaired persons.

9. Acknowledgements

The research and development work on the iTEMA system will be co-funded in scope of the Eureka E!3864 project iTEMA: Intelligent Telephone E-Mail Access [iTEMA06].

10. References

i2010, (2006). i2010 – A European Information Society for growth and employment. http://ec.europa.eu/information_society/eeurope/i2010/i2010/index_en.htm.

iTEMA06, (2006). iTEMA Eureka project information page, http://www.eureka.be/inaction/AcShowProject.do?id= 3582.

Mihelič, A., Žganec, M., Pavešić, N., Žganec Gros, J., (2006). Efficient subset selection from phonetically transcribed text corpora for concatenation-based embedded text-to-speech synthesis, Informacije MIDEM, Vol. 36, No. 1, pp. 19-24.

Žganec Gros, J., (2006). Text-to-speech synthesis for embedded speech user interfaces, In WSEAS Transactions on Communications, No. 4, Vol. 5, pp. 543-548.

Sečujski, M., (2005). Obtaining prosodic information from text in Serbian Language, In Proc. EUROCON – The Int. Conf. on "Computer as a Tool", pp. 1654-1658.

Delić, V., Vujnović, N., Sečujski, M., (2005). Speech-enabled computers as a tool for Serbian-speaking blind persons, In Proc. of EUROCON – The Int. Conf. on "Computer as a Tool", pp. 1662-1665.

The VoiceTRAN Speech Translation Demonstrator

Jerneja Žganec Gros[1], Stanislav Gruden[1], France Mihelič[2], Tomaž Erjavec[3], Špela Vintar[4], Peter Holozan[6], Aleš Mihelič[1], Simon Dobrišek[2], Janez Žibert[2], Tomo Korošec[5], Nataša Logar[5]

[1]Alpineon d.o.o., Ljubljana, Slovenia
[2]University of Ljubljana, Faculty of Electrical Engineering, Ljubljana, Slovenia
[3]Jožef Stefan Institute, Ljubljana, Slovenia
[4]University of Ljubljana, Faculty of Arts, Ljubljana, Slovenia
[5]University of Ljubljana, Faculty of Social Sciences, Ljubljana, Slovenia
[6]Amebis d.o.o., Kamnik, Slovenia

Abstract

This paper describes the design phases of the VoiceTRAN Communicator, which integrates speech recognition, machine translation, and text-to-speech synthesis using the Galaxy architecture. The aim of the work was to build a robust multimodal speech-to-speech translation system able to translate simple domain-specific sentences in the language pair Slovenian-English. The work represents a joint collaboration between several Slovenian research organizations that are active in human language technologies.

Govorni komunikator VoiceTRAN

Prispevek opisuje delo na razvoju govornega komunikatorja VoiceTRAN, ki združuje tehnologije prepoznavanja govora, strojnega prevajanja in sinteze govora. Podajamo opis arhitekture sistema ter posameznih sistemskih modulov. Nadalje opisujemo jezikovne vire, ki smo jih uporabili pri izgradnji sistema, ter preskus sistema. Sistem VoiceTRAN omogoča govorno prevajanje za jezikovni par slovenščina-angleščina na omejenem področju uporabe.

1. Introduction

Automatic speech-to-speech (STS) translation systems aim to facilitate communication among people that speak different languages [1, 2, 3]. Their goal is to generate a speech signal in the target language that conveys the linguistic information contained in the speech signal from the source language.

There are, however, major open research issues that challenge the deployment of natural and unconstrained speech-to-speech translation systems, even for very restricted application domains, due to the fact that state-of-the-art automatic speech recognition and machine translation systems are far from perfect.

In addition, in comparison to translating written text, conversational spoken messages are often conveyed with imperfect syntax and casual spontaneous speech.

In practice, when building demonstration systems, STS systems are typically implemented by imposing strong constraints on the application domain and the type and structure of possible utterances; that is, both in the range and in the scope of the user input allowed at any point of the interaction. Consequently, this compromises the flexibility and naturalness of using the system.

The VoiceTRAN Communicator was developed in a Slovenian research project involving 6 partners: Alpineon, the University of Ljubljana (Faculty of Electrical Engineering, Faculty of Arts, and Faculty of Social Studies), the Jožef Stefan Institute, and Amebis as a subcontractor.

The work has been co-funded by the Slovenian Ministry of Defense and the Slovenian Research Agency. The aim is to build a robust multimodal speech-to-speech translation communicator, similar to Phraselator [4] or Speechalator [5], able to translate simple sentences in the language pair Slovenian-English. It goes beyond the Phraselator device because it is not limited to predefined input sentences.

In the initial phase of the project a system demonstrator was developed. In further phases it will be wrapped into a stand-alone communicator and upgraded to new language pairs. The application domain is limited to common application scenarios that occur in peace-keeping operations on foreign missions when the users of the system have to communicate with the local population. More complex phrases can be entered via keyboard using a graphical user interface.

First an overview of the VoiceTRAN system architecture is given. We continue to describe the individual server modules. We conclude the paper by discussing the speech-to-speech translation evaluation methods and outlining plans for future work.

2. System Architecture

The VoiceTRAN Communicator uses the DARPA Galaxy Communicator architecture [6]. The Galaxy Communicator open source architecture was chosen to provide inter-module communication support because its plug-and-play approach allows interoperability of commercial software and research software components. It was specially designed for development of voice-driven user interfaces in a multimodal platform.

The VoiceTRAN Communicator consists of a Hub and five servers that interact with each other through the Hub as shown in Figure 1.

The Hub is used as a centralized message router through which servers can communicate with one another. Frames containing keys and values are emitted by each

server. They are routed by the hub and received by a secondary server based on rules defined in the Hub script.

Audio Server	Receives speech signals from the microphone and sends them to the recognizer. Sends synthesized speech to the speakers.
Graphic User Interface	Receives input text from the keyboard. Displays recognized source language sentences and translated target language sentences. Provides user controls for handling the application.
Speech Recognizer	Takes the signals from audio server and maps audio samples into text strings. Produces an N-best sentence hypothesis list.
Machine Translator	Receives N-best postprocessed sentence hypotheses from the speech recognition server and translates them from a source language into a target language. Produces a scored disambiguated sentence hypothesis list.
Speech Synthesizer	Receives rich and disambiguated word strings from the machine translation server. Converts the input word strings into speech and prepares them for the audio server.

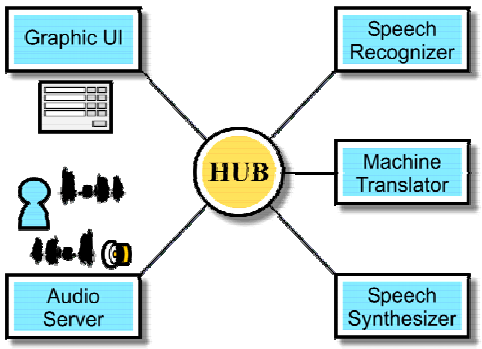


Figure 1. The Galaxy system architecture used in the VoiceTRAN communicator.

2.1. Audio Server

The audio server connects to the microphone input and speaker output terminals on the host computer and performs recoding of user input and playing prompts or synthesized speech.

Input speech captured by the audio server has been automatically recorded to files for posterior system training.

2.2. Speech Recognizer

The speech recognition server receives the input audio stream from the audio server and provides a word graph at its output and a ranked list of candidate sentences; the N-best hypotheses list, which can include part-of-speech information generated by the language model.

The speech recognition server used in VoiceTRAN is based on the Hidden Markov Model Recognizer developed at the Faculty of Electrical Engineering, University of Ljubljana [7]. It has been upgraded to perform medium-size vocabulary (10K words) speaker (in)dependent speech recognition on a wider application domain. A back-off class-based trigram language model is used. Given a limited amount of training data the parameters in the models have been carefully chosen in order to achieve maximum performance.

Because the final goal was a stand-alone speech communicator used by a specific user, the speech recognizer has been additionally trained and adapted to the individual user in order to achieve higher recognition accuracy in at least one language pair direction.

A common speech recognizer output typically has no information on sentence boundaries, punctuation, and capitalization. Therefore, additional postprocessing in terms of punctuation and capitalization has been performed on the N-best hypotheses list before it is passed to the machine translator.

The inclusion of a prosodic module was necessary in order to link the source language to the target language, but also to enhance speech recognition proper. Besides syntactic and semantic information, properties such as dialect, sociolect, sex and attitude etc are signaled by prosody. The degree of linguistic information conveyed by prosody varies between languages, from languages such as English, with a relatively low degree of prosodic disambiguation, via tone-accent languages such as Swedish, to pure tone languages. Prosody information helps to determine proper punctuation and sentence accent information.

2.3. Machine Translator

The machine translator (MT) converts text strings from a source language into text strings in the target language. Its task is difficult since the results of the speech recognizer convey spontaneous speech patterns and are often erroneous or ill-formed.

A postprocessing algorithm inserts basic punctuation and capitalization information before passing the target sentence to the speech synthesizer. The output string can also convey lexical stress information in order reduce disambiguation efforts during text-to-speech synthesis.

A multi-engine based approach was used in the early phase of the project that makes it possible to exploit strengths and weaknesses of different MT technologies and to choose the most appropriate engine or combination of engines for the given task. Four different translation engines have been applied in the system. We combined TM (translation memories), SMT (statistical machine translation), EBMT (example-based machine translation) and RBMT (rule-based machine translation) methods. A simple approach to select the best translation from all the outputs was applied. A bilingual aligned domain-specific corpus was used to build the TM and train the EBMT and the SMT phrase translation models.

The Presis translation system was used as our baseline system [8]. It is a commercial conventional rule-based translation system that is constantly being optimized and upgraded. It was adapted to the application domain by

upgrading the lexicon. Based on stored rules, Presis parses each sentence in the source language into grammatical components, such as subject, verb, object and predicate and attributes the relevant semantic categories. Then it uses built-in rules for converting these basic components into the target language, performs regrouping and generates the output sentence in the target language.

We continue the paper by describing the VoiceTRAN SMT experiment.

2.3.1. Statistical Machine Translation Experiment

Some initial machine translation attempts have been reported for the translation from Slovenian into English [8], [9], however, very little has been done for the opposite direction, from English into Slovenian. We have performed experiments in both translation directions, where the latter proved to be an especially complex and demanding task due to the highly inflectional nature of the Slovenian language.

The SMT experiments were performed on a joint corpus, consisting of 3 parallel corpora: the VoiceTRAN application-specific corpus, the SVEZ-IJS [10] and the IJS-ELAN corpus [11], where the words in all three corpora contain automatically assigned context-disambiguated lemmas and morphosyntactic descriptions (MSDs). Sentences longer than 25 words were discarded from the joint corpus.

The freely available GIZA++ tool [12] was used for training the SMT model. The CMU-SLM toolkit [13] was used for generating the language model. The ISI ReWrite Decoder [14] has been applied for the translation of test sentences.

Two different types of test sets were used. The first test set was extracted from the joint corpus. The test sentences were chosen at regular intervals, one out of every 1000 sentences. For the second test we used the sentences from one of the components of the IJS-ELAN corpus, the ORWL file (Orwell's "1984"), which is of a significantly different text type from the rest of the joint corpus.

This set-up enabled us to test the system with sentences, which were highly correlated to the training data, as well as on those that had low correlation to the training set. The test sentences were excluded from the training material for the SMT and language models.

The SMT experiments were performed in two ways. First, we implemented the 'simple' procedure, where the sentences used for training the SMT system were taken directly from the joint corpus, without any prior modifications.

The second or 'combined' procedure was more complex. From the joint corpus we have derived two corpora. In the first corpus, the sentences in both languages have been modified so that the words were replaced by their lemmas, using the lemmatization information provided in the source corpora. In order to derive the second corpus, all original word forms have been replaced by their corresponding morphosyntactic descriptors.

These two corpora were then separately fed to the training system.

The decoding was performed as follows: every test sentence was preprocessed into two sentences, where words had been replaced by lemmas in the first sentence, and by MSDs in the second sentence.

Then we traced how each pair lemma+MSD in the source language changed to the corresponding pair lemma+MSD in the target language. The resulting pair lemma+MSD was ultimately combined to construct the final word in the target language. Our goal was to decrease the data sparsness of the training corpus. That was achieved by translating lemmas instead of original words. By translating MDSs separately, we wanted to preserve the MSD information without affecting the translation of the lemmatized text.

Further improvement was expected by adding a dictionary corpus to the joint corpus in the training phase. We have used only pairs of single words (no multi-word expressions were included at this stage). In cases when one English word had many Slovenian translation equivalents, one entry was added for each of these translations to the dictionary corpus, which ended up by containing approximately 140.000 entries.

We introduced many additional corrections, for which we expected they might improve the translation performance. For example, less important features of the MSDs were replaced by a wild-card character, etc. Further, if the decoding algorithm decided that the MSD gender value of the target word was dual and the dual word form was not found in our word base, plural was used instead, similarly as in [9]. All tokens in the corpus containing numerals were in the initial phase replaced by a unique token, which further reduced the data sparseness. Words marked in English as proper nouns, were handled in a similar way. By tracing word translation, we were able to replace these unique tokens, which appeared in the Slovenian text with the corresponding original token from the English language test sentence. Finally, one additional monolingual annotated text corpus was added to train the Slovenian language modeling tool, the FDV-IJS corpus.

Since the sources of the training data had been automatically tagged with lemmas and MSDs, the resulting imperfections in the training material had negative effects, especially on the combined translation method results. Therefore, we intend to re-tag the source corpora in the continuation of the project.

2.4. Speech Synthesizer

The last part in a speech-to-speech translation task is the conversion of the translated utterance into its spoken equivalent. The input target text sentence is equipped with lexical stress information at possible ambiguous words.

The Proteus unit-selection text-to-speech system is used for this purpose [15]. It performs grapheme-to-phoneme conversion based on rules and a look-up dictionary and rule-based prosody modeling. Domain-specific adaptations include new pronunciation lexica and the construction of a speech corpus of frequently used in-domain phrases.

Special attention was paid to collocations as defined in the bilingual dictionary. They were treated as preferred units in the unit selection algorithm.

We are also exploring how to pass a richer structure from the machine translator to the speech synthesizer. An input structure containing information on POS and lexical stress information resolves many ambiguities and can result in more accurate prosody prediction.

The speech synthesizer produces an audio stream for the utterance. The audio is stream is finally sent to the speakers by the audio server. After the synthesized speech has been transmitted to the user, the audio server is freed up in order to continue listening for the next user utterance.

2.5. Graphical User Interface

In addition to the speech user interface, the VoiceTRAN Communicator provides a simple interactive user-friendly graphical user interface, as shown in Fig. 2. Input text in the source language can also be entered via keyboard or selected by pen input.

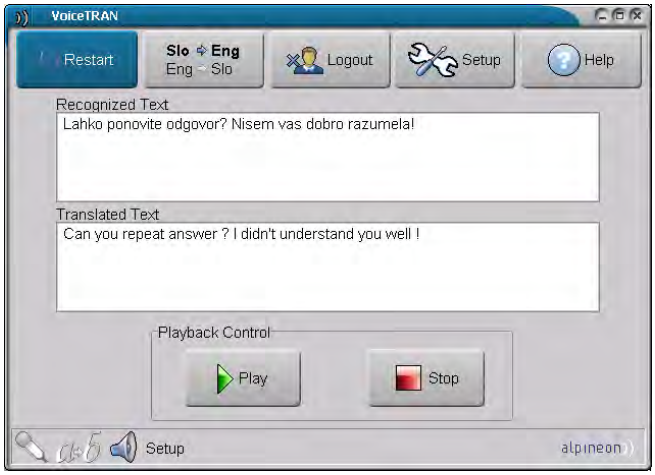


Figure 2. Screenshot of the graphical user interface in the VoiceTRAN communicator application. The source language text provided by the speech recognition module and the translated text in the target language are displayed.

Recognized sentences in the source language along with their translated counterparts in the target language are displayed.

A push-to-talk button is provided to signal an input voice activity, and a replay button serves to start a replay of the synthesized translated utterance. The translation direction can be changed by pressing the translation direction button.

The setup menu enables the user to customize the application according to his needs. It also provides the possibility to choose between different text-to-speech engines.

3. Language Resources

Some of the multilingual language resources needed to set up STTS systems and include Slovenian are presented in [16]. For building the speech components of the VoiceTRAN system, existing speech corpora have been used [17]. Since the speech corpora have been collected from different sources, adaptations have been carried out. The language model has been trained on a domain-specific text corpus that was collected and annotated within the project.

The Proteus pronunciation lexicon [15] has been used for both speech recognition and text-to-speech synthesis. Speech synthesis is based on the Proteus speech corpus. It has been expanded by the most frequent in-domain utterances.

As mentioned in section 2.3., for developing the machine translation component, a dictionary of military terminology [18], and various existing aligned parallel corpora were used [10], [11]. We have syntactically annotated an in-domain large Slovenian monolingual text corpus, the FDV-IJS that was collected at the Faculty of Social Studies, University of Ljubljana. This corpus contains over 5.5 million words and has been used for training the language model in the speech recognizer, as well as for inducing relevant multiword units (collocations, phrases, and terms) for the domain.

An aligned bi-lingual in-domain corpus with 300,000 words – the VoiceTRAN corpus – has been collected within the project. The compilation of the corpus involved selecting the digital original of the bi-texts, re-coding to XML TEI P4, sentence alignment, word-level syntactic tagging, and lemmatization [19]. The corpus has been used to induce bi-lingual single word and phrase lexica for the MT component, and as direct input for SMT and EBMT systems. It was also used for training of the speech recognizer language model.

4. Evaluation

The evaluation tests of a speech-to-speech translation system serve two purposes:

1. to evaluate whether we have improved the system by introducing improvement of individual components of the system;
2. to test the system acceptance by the end users in field tests.

We have performed individual component tests in order to select the most appropriate methods for each application server. Speech recognition was evaluated by computing standard word error rates, which were below 10%. The TTS system was evaluated using ITU-T recommendations for subjective performance tests. The results are reported in [15].

		Relative changes in the average values of the MT metrics with a tested system configuration in comparison to the baseline system configuration			
		Δ WER [%]	Δ GTM [%]	Δ NIST [%]	Δ BLEU [%]
1)	Tested configuration: combined MT method Baseline system: simple MT method Test sentences: from the joint corpus	+3 to +5	-20 to -8	-10 to -5	-25 to -10
2)	Tested configuration: combined MT method Baseline system: simple MT method Test sentences: ORWL corpus	-3 to -2	-5 to +6 [1] -8 to -1 [2]	-2 to 0	+20 to +80
3)	Tested configurations: various additional corrections from 2.3.1 Baseline system: simple/combined MT method without additional corrections Test sentences: joint corpus, ORWL	-2 to 0	+5 to +10	+5 to +10	+25 to 200

Table 1: Evaluation results for the SMT system. Relative changes in the average values of the MT metrics with a tested system configuration in comparison to the baseline system configuration are given. In experiments 1) and 2), the range of percentages reflects the following variations in system configuration: with or without the dictionary, with or without the additional corrections from 2.3.1. In experiment 3), the range of percentages reflects the following variations in system configuration: with or without dictionary, combined method/simple translation method, different test sets.

For the machine translation component, initial objective evaluation tests were performed, which we describe in the following subsection.

4.1. SMT Evaluation Results

To measure the 'closeness' between the SMT-generated hypothesis and human reference translations, standard objective MT metrics were used: Word Error Rate (WER), General Text Matcher (GTM) [20], NIST and BLEU [21].

The SMT evaluation efforts were centered on three system variation impacts:

1) the impact of the choice of the translation method, i.e. simple or combined,
2) the impact of the addition of a dictionary, and
3) the impact of the combination of the additional corrections, described by the end of chapter 2.3.1.

In Table 1, relative changes in evaluation scores (WER, GTM, NIST and BLEU) of the tested MT system and training set configuration versus the baseline system are given. The obtained values for the BLEU score were so small that the obtained results have not been considered as reliable.

In comparison to the simple translation method, the combined translation method did not perform well for test sentences extracted from the unprocessed joint corpus.

The combined method performed better when ORWL test sentences were used, proving its potential for translation of out-of-domain sentences.

Surprisingly, in all cases, the NIST score was slightly better for the simple translation method.

The simple translation method apparently adapted well to inflected Slovenian words, some of which were frequent enough in the training material to allow for sufficient training of the statistical model. As a consequence, when testing on test sentences from the joint corpus, which were well correlated to the training corpus, the test set translations were translated rather well. As expected, the combined translation method performed better when translating texts, which were very different from the training sentence set, as was the case with the ORWL test corpus.

For every test configuration we found that the addition of a dictionary had a minor and more or less random influence on the translation quality. The dictionary contained many entries, which translated one English word to more than one Slovenian word candidates, which proved to be an obstacle in the training process. One of these words usually dominated and the system too often picked it as a result.

From the other corrections, introduced in 2.3.1, only the special treatment of numeric tokens and proper nouns has yielded a better performance, whereas the addition of the IJS-FDV corpus to the language model has not.

The scores for translation quality using the standard metrics were generally low. We would like to stress that we found that these evaluation methods are not suitable for evaluating translations into Slovenian. These tools are all based on an exact comparison of entire words, which works well for English. Due to the rich inflectional paradigms in Slovenian, words, which are semantically correctly translated, but their ending is wrong, have the calculated score of zero. A method, which attributes score points for finding a correct word stem would provide a much better translation quality estimation. Nevertheless, the used evaluation methods were suitable for the purposes of our research since we were only interested in an indicator for improvement or deterioration when using various MT system and training set configurations.

[1] without dictionary
[2] with dictionary

5. Conclusion

The implementation concept of the VoiceTRAN communicator demonstrator has been discussed in the paper. It is able to translate simple domain-specific sentences in the language pair Slovenian-English.

The chosen system architecture makes it possible to test a variety of server modules. The end-to-end prototype was evaluated in and is ready for end-user field trials.

6. Acknowledgements

The work presented in this paper was supported by the Slovenian Ministry of Defense and the Slovenian Research Agency under contract no. M2-0019.

7. References

[1] A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavaldà, T. Zeppenfeld, and P. Zhan, "Janus-III: Speech-to-Speech Translation in Multiple Languages," Proceedings of the ICASSP, Munich, Germany, 1997, pp. 99–102.

[2] W. Wahlster, Verbmobil: Foundation of Speech-to-Speech translation, Springer Verlag, 2000.

[3] A. Lavie, F. Metze, R. Cattoni, E. Costantin, S. Burger, D. Gates, C. Langley, K. Laskowski, L. Levin, K. Peterson, T. Schultz, A. Waibel, D. Wallace, J. McDonough, H. Soltau, G. Lazzari, N. Mana, F. Pianesi, E. Pianta, L. Besacier, H. Blanchon, D. Vaufreydaz, and L. Taddei, "A Multi-Perspective Evaluation of the NESPOLE! Speech-to-Speech Translation System," Proceedings of the ACL 2002 Workshop on Speech-to-Speech Translation: Algorithms and Systems, Philadelphia, PA, 2002.

[4] A. Sarich, "Phraselator, one-way speech translation system," available at http://www.sarich.com/translator/.

[5] A. Waibel, A. Badran, A.W. Black, R. Frederking, D. Gates, A. Lavie, L. Levin, K. Lenzo, L. Mayfield, L. Tomokyo, J. Reichert, T. Schultz, D. Wallace, M. Woscsyna, and J. Zhang, "Speechalator: Two-Way Speech-to-Speech Translation on a Consumer PDA," Proceedings of the Eurospeech'03. Geneva, Switzerland, 2003, pp. 369–372.

[6] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, P. and V. Zue, "Galaxy-II: A Reference Architecture for Conversational System Development," Proceedings of the ICSLP'98, Sydney, Australia, pp. 931–934, available at http://communicator.sourceforge.net/, 1998.

[7] S. Dobrišek, "Analysis and Recognition of Phrases in Speech Signals," PhD Dissertation, University of Ljubljana, Slovenia, 2001.

[8] M. Romih and P. Holozan, "A Slovenian-English Translation System," Proceedings of the 3rd Language Technologies Conference, Ljubljana, Slovenia, 2002, p. 167.

[9] J. Vičič and T. Erjavec. "Vsak začetek je težak : avtomatsko učenje prevajanja slovenščine v angleščino," Proceedings of the conference Jezikovne tehnologije, Ljubljana, Slovenia, 2002, pp. 20-27.

[10] T. Erjavec, C. Ignat, P. Pouliquen, and R. Steinberger, "Massive Multi-lingual Corpus Compilation: Acquis Communautaire and Totale," Proceedings of the 2nd Language and Technology Conference, Poznań, Poland, 2005.

[11] T. Erjavec, "The IJS-ELAN Slovene-English Parallel Corpus," International Journal on Corpus Linguistics, Vol. 7, 2002, pp. 1–20.

[12] F. J. Och and H. Ney. "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003, available at http://www.fjoch.com/GIZA++.html.

[13] R. Rosenfeld. "The CMU Statistical Language Modeling Toolkit, and its use in the 1994 ARPA CSR Evaluation," Proceedings of the ARPA SLT Workshop, available at http://www.speech.cs.cmu.edu/SLM/toolkit.html.

[14] U. Germann. "Greedy Decoding for Statistical Machine Translation in Almost Linear Time," Proceedings of the HLT-NAACL-2003, available at http://www.isi.edu/licensed-sw/rewrite-decoder/.

[15] J. Žganec Gros, "Text-to-speech synthesis for embedded speech user interfaces," WSEAS trans. commun., Mar. 2006, vol. 5, iss. 3, pp. 543-548.

[16] D. Verdonik, M. Rojc. Jezikovni viri projekta LC-STAR. Proceedings B of the 7th International Multi-Conference Information Society IS 2004: Jezikovne tehnologije, October 2004, Ljubljana, Slovenia, pp. 24-47.

[17] F. Mihelič, J. Žganec Gros, S. Dobrišek, J. Žibert, and N. Pavešić, "Spoken Language Resources at LUKS of the University of Ljubljana," Int. Journal on Speech Technologies, Vol. 6., No. 3, 2003, pp. 221–232.

[18] T. Korošec, "Opravljeno je bilo pomembno slovarsko delo o vojaškem jeziku," Slovenska vojska, Vol. 10, No. 10, 2002, pp. 12–13.

[19] T. Erjavec and S. Džeroski, "Machine Learning of Language Structure: Lemmatising Unknown Slovene Words," Applied Artificial Intelligence, Vol. 18, No. 1, 2004, pp. 17–41.

[20] Joseph P. Turian, Luke Shen, and I. Dan Melamed, "Proteus technical report #03-005: Evaluation of Machine Translation and its Evaluation," available at http://nlp.cs.nyu.edu/eval/.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation," 2001. RC 22176(W0109-022), IBM Research.

Combining Efforts for Improving Automatic Classification of Emotional User States

Anton Batliner[1], Stefan Steidl[1], Björn Schuller[2], Dino Seppi[3], Kornel Laskowski[4], Thurid Vogt[5], Laurence Devillers[6], Laurence Vidrascu[6], Noam Amir[7], Loic Kessous[7], Vered Aharonson[8]

[1]**FAU:** Lehrstuhl für Mustererkennung, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
batliner@informatik.uni-erlangen.de, steidl@informatik.uni-erlangen.de

[2]**TUM:** Institute for Human-Machine Communication, Technische Universität München, Germany, *schuller@tum.de*

[3]**ITC:** ITC-irst, Trento, Italy, *seppi@itc.it*

[4]**UKA:** interACT, University of Karlsruhe, Germany, *kornel@cs.cmu.edu*

[5]**UA:** Multimedia Concepts and their Applications, University of Augsburg, Germany, *vogt@informatik.uni-augsburg.de*

[6]**LIMSI:** Spoken Language Processing Group, LIMSI-CNRS, Orsay Cedex, France, *devil@limsi.fr, vidrascu@limsi.fr*

[7]**TAU:** Dep. of Communication Disorders, Sackler Faculty of Medicine, Tel Aviv University, Israel,
noama@post.tau.ac.il, kessous@post.tau.ac.il
[8]**AFEKA:** Tel Aviv academic college of engineering,Tel Aviv, Israel, *vered@nexsig.com*

Abstract

Classification performance of emotional user states found in realistic, spontaneous speech is not very high, compared to the performance reported for acted speech in the literature. This might be partly due to the difficulty of providing reliable annotations, partly due to suboptimal feature vectors used for classification, and partly due to the difficulty of the task. In this paper, we present a co-operation between several sites, using a thoroughly processed emotional database. For the four-class problem *motherese*/*neutral*/*emphatic*/*angry*, we first report classification performance computed independently at each site. Then we show that by using all the best features from each site in a combined classification, and by combining classifier outputs within the ROVER framework, classification results can be improved; all feature types and features from all sites contributed.

Združevanje sil za boljše samodejno razvrščanje čustvenih stanj uporabnika:

Uspešnost samodejnega razvrščanja čustvenih stanj uporabnika, ki jih najdemo v realističnem, spontanem govoru, je v primerjavi s kakovostjo, ki jo v literaturi navajajo za igrani govor, precej nižja. To je lahko delno posledica težav pri zagotavljanju zanesljive anotacije, delno posledica uporabe podoptimalnih vektorjev značilk pri razvrščanju, delno pa posledica težavnosti te naloge. V prispevku predstavljamo sodelovanje med različnimi ustanovami na temeljito obdelani bazi podatkov. Za štiristopenjski problem *govor otroku*/*nevtralno*/*poudarjeno*/*jezno* najprej navedemo kakovost razvrščanja, kot so jo izračunali neodvisno na vsaki od sodelujočih ustanov. Nato poka-žemo, da lahko izboljšamo rezultate razvrščanja z uporabo najboljših značilk vsake izmed ustanov in z združevanjem rezultatov razvrščevalnikov znotraj ogrodja ROVER.

1. Introduction

In this paper, we present a co-operation between several sites dealing with classification of emotional user states conveyed via speech; this initiative was taken within the European Network of Excellence HUMAINE under the name CEICES (**C**ombining **E**fforts for **I**mproving automatic **C**lassification of **E**motional user **S**tates); as for an overview of emotion recognition in human-computer-interaction, cf. (Cowie et al., 2001). The database used is a German corpus with recordings of 51 ten- to thirteen-year old children communicating with Sony's AIBO pet robot. Conceptualization, design and recordings were done at the 'originator' site FAU[1]; results have been reported for example in (Steidl et al., 2005; Batliner et al., 2005b; Batliner et al., 2005a). The approach to be followed within CEICES looked like this: the originator site provided speech files, phonetic lexicon, manually corrected word segmentation (and, in the future, manually corrected F0 values), emotional labels, definition of train and test samples, etc. The data was annotated at the word level. We aimed at two different classification tasks: word-based and turn-based classification; for the latter, we mapped the word-based labels onto turn-based ones. All partners committed themselves to share with all the other partners their extracted feature values together with the necessary information (which feature models which acoustic or linguistic phenomenon, format of feature values, classifier used, etc.). Thus each site could assess the features provided by all other sites, together with their own features, aiming at a repertoire of optimal fea-

[1]The abbreviations for all sites can be found in the affiliations given in the title of this paper; AFEKA is subsumed under TAU.

tures. In this work, we look not only at acoustic but also at linguistic features.[2]

2. Material and annotation

The general framework for the database reported on in this paper is child-robot communication, and the elicitation and subsequent recognition of emotion-related user states. The robot is Sony's (dog-like) AIBO robot. The basic idea has been to combine a new type of corpus (children's speech) with 'natural' emotional speech within a Wizard-of-Oz task. The speech is intended to be 'natural' since children do not disguise their emotions to the same extent as adults. However, it is of course not fully 'natural' as it might be in an unsupervised setting. Furthermore the speech is spontaneous; the children were not told to use specific instructions but to talk to the AIBO as they would to a friend. In this experimental design, the child is led to believe that the AIBO is responding to his or her commands, but the robot is actually being controlled by a human operator, using the 'AIBO Navigator' software over a wireless LAN (the existing AIBO speech recognition module is not used). The wizard causes the AIBO to perform a fixed, pre-determined sequence of actions, which takes no account of what the child says. For the sequence of AIBO's actions, we tried to find a good compromise between obedient and disobedient behaviour: we wanted to provoke the children in order to elicit emotional behaviour but of course we did not want to run the risk that they break off the experiment. The children believed that the AIBO was reacting to their orders — albeit often not immediately. In fact, it was the other way round: the AIBO always strictly followed the same plot, and the children had to align their orders to its actions.

The data was collected from 51 children (age 10 - 13, 21 male, 30 female). The children are from two different schools (25 children from 'MONT' and 26 from 'OHM'); the recordings took place in the respective classrooms. The only persons in the room were the child, a supervisor who initially instructed the children, the wizard (behind the children, pretending to be doing the recordings) and a third assistant.[3] Each recording session took some 30 minutes. Because of the experimental setup, these recordings contain a huge amount of silence (the reaction time of the AIBO), which caused a noticeable reduction of recorded speech after raw segmentation; ultimately we obtained about 9.2 hours of speech. More details are given in (Steidl et al., 2005; Batliner et al., 2005b; Batliner et al., 2005a).

Five labellers (advanced students of linguistics) listened to the recordings and annotated independently of each other each word as *neutral* (default) or as belonging to one of ten other classes which were designed during earlier inspection of the data; we do not claim that these classes represent children's emotions in general, only that they are adequate for the modelling of these children's behaviour in this specific scenario. We resorted to majority voting (henceforth MV): if three or more labellers agreed, the label was attributed to the word; if four or five labellers agreed, we assumed a sort of prototype. The following raw labels were used — in parentheses, the number of cases with MV is given: *joyful* (101), *surprised* (0), *emphatic* (2528), *helpless* (3), *touchy*, i.e., irritated (225), *angry* (84), *motherese* (1260), *bored* (11), *reprimanding* (310), *other*, i.e. non-neutral, but not belonging to the other categories (3), and *neutral* (39169). 4707 words had no MV; all in all, there were 48401 words. *joyful* and *angry* belong to the 'big' emotions, the other ones rather to 'emotion-related/emotion-prone' user states and by that, to 'emotion' in its broader meaning.

The state *emphatic* has to be commented on especially: based on our experience with other emotion databases (Batliner et al., 2003), any marked deviation from a neutral speaking style can (but need not) be taken as a possible indication of some (starting) trouble in communication. If a user gets the impression that the machine does not understand her, she tries different strategies – repetitions, reformulations, other wordings, or simply the use of a pronounced, marked speaking style. Such a style does not necessarily indicate any deviation from a neutral user state, but it suggests a higher probability that the (neutral) user state will be changing soon. Of course, it can be something else as well: a user idiosyncrasy, or a special style — 'computer talk' — that some people use while speaking to a computer, like speaking to a non-native listener, to a child, or to an elderly person who is hard of hearing. Thus the fact that *emphatic* is observed can only be interpreted meaningfully if other factors are considered. There are three further — practical — arguments for the annotation of *emphatic*: firstly, it is to a large extent a prosodic phenomenon, and can thus be modelled and classified with prosodic features. Secondly, if the labellers are allowed to label *emphatic*, it may be less likely that they confuse it with other user states. Thirdly, we can try and model emphasis as an indication of (arising) problems in communication (Batliner et al., 2003).

Some of the labels are very sparse; if we only take labels with more than 50 MVs, the resulting 7-class problem is most interesting from a methodological point of view, cf. the new dimensional representation of these seven categorical labels in (Batliner et al., 2005a). However, the distribution of classes is very unequal. Therefore, we downsampled *neutral* and *emphatic* to **N***eutral* and **E***mphatic*, respectively, and mapped *touchy*, *reprimanding*, and *angry* onto **A***ngry*[4], as representing different but closely related kinds of negative attitude. This more balanced 4-class problem, which we refer to as AMEN, consists of 1557 words

[2] We expect improved recognition rates from this co-operation. However, it is an educated guess that, for instance, manual segmentation yields more reliable results for emotion recognition than automatic segmentation — we simply do not know yet whether and to what extent this will turn out to be a fact.

[3] Speech was transmitted with a wireless headset (UT 14/20 TP SHURE UHF-series with microphone WH20TQG) and recorded with a DAT-recorder. The sampling rate of the signals was 48 kHz, quantized at 16 bits. The data was downsampled to 16 kHz prior to processing.

[4] The initial letter is given boldfaced and recte; this letter will be used in the following for referring to these four cover classes. Note that now, **A***ngry* can consist, for instance, of two *touchy* and one *reprimanding* label; thus the number of **A***ngry* cases is far higher than the sum of *touchy*, *reprimanding*, and *angry* MV cases.

for **A***ngry* (**A**), 1224 words for **M***otherese* (**M**), 1645 words for **E***mphatic* (**E**), and 1645 for **N***eutral* (**N**) (Steidl et al., 2005). Cases where less than three labellers agreed were omitted as well as those cases where other than these four main classes were labelled. Interlabeller agreement is dealt with in (Steidl et al., 2005).

A last note on label names and terminology in general: names for the non-cognitive phenomena that we are dealing with are known not to be unequivocal or agreed upon. There is wide disagreement as to whether affect encompasses emotion or the other way around. In this paper, we follow a terminology widely adopted by HUMAINE. Some of our label names were chosen for purely practical reasons: we needed unique characters for processing. We chose *touchy* and not *irritated* because the letter 'I' has been reserved in our labelling system for *ironic*, cf. (Batliner et al., 2005a). Instead of *motherese*, some people use 'child-directed speech'; this is, however, only feasible if there is in the respective database no negative counterpart such as *reprimanding* which is 'child-directed' as well. **A***ngry* was not named **N***egative* because we reserved **N** for **N***eutral*.

3. Pre-processing of the data

The word is a simple and rather unequivocal concept in speech processing; the basic unit of emotional speech might not be the word — nor the sentence — but something in between (clauses, noun phrases, etc.). By annotating words, we are able to map sequences of words onto larger emotion units later on. An automatic reverse top-down splitting — from turns to clauses — would not be possible. The processing of emotional speech, however, at almost any other site, resorts to turns as units which have been labelled as such. We therefore decided to start with turns as units of investigation; these 'turns' are physically stored and distributed as speech files which were extracted out of the recordings of the sessions using longer pauses as the automatic segmentation criterion.[5] This leaves us with the task of mapping word-based labels onto turn-based labels: a simple 50% threshold — for instance, if an **A** turn has 10 words, then 5 or more words have to be labelled as **A** — would be suboptimal because some words, especially function words, are likely not to be produced in an emotional manner; moreover, a longer turn can consist of one neutral clause, and one emotional clause — then chances are that the whole turn will be wrongly mapped onto neutral.

For the mapping onto turn-based labels, we employed the following strategy: fragments and auxiliaries were used as stop words.[6] For each turn, we pool together the labels given by our 5 labellers (for a turn of n words, we obtain 5 x n labels). For the turn to be mapped onto neutral, 70% of the labels have to be neutral; *joyful* and the other spurious labels are not taken into account for this computation. If 30% or more are non-neutral, then the turn is **A**, **M**, or **E**. If at least 50% of the non-neutral labels are **M**, the turn is mapped onto **M**. If **A** and **E** are equally distributed, the turn is mapped onto **A**. The remaining turns, which are neither **A** or **M**, are declared to be **E**. This means that we employ a sort of 'markedness' condition: **M** is more marked than **A**, and **A** is more marked than **E**, and all are more marked than **N**. This strategy yields the following turn-based label counts: 868 **A** (21.7 %), 1347 **E** (33.7 %), 495 **M** (12.4 %), and 1280 **N** (32.0 %), summing up to 3990 (100 %) turn labels.

Especially for the word-based classification to be reported on in a future work, to avoid automatic segmentation errors which certainly will be different at different sites, the automatic segmentation of all words belonging to these 3990 AMEN turns conducted at FAU was manually corrected by the first author. We hope that this will eliminate performance differences that might be traced back to different automatic segmentations.

4. Classification

For classification, we used 2-fold cross-validation: MONT vs. OHM and vice versa, and then average the two results. This way, we can guarantee strict speaker independence and, at the same time, easily compare results across sites by visual inspection — which would not be possible if we resorted to leave-one-speaker-out (i.e. 51-fold cross-validation). This 2-fold cross-validation is a more conservative strategy yielding lower recognition performance than leave-one-speaker-out. It might be argued that, in addition, we should define a validation sample, and that we should deal with the multiplicity effect, i.e. the repeated use of the same data, through significance testing using the Bonferroni adjustment. A practical argument against a validation sample is that it would reduce the number of cases — which is already low. There are some theoretical/methodological arguments against the Bonferroni adjustment (Pernegger, 1998); however, in our situation, when we are pursuing rather *I-wonder-what-will-happen* instead of *I-bet-this-will-happen* hypotheses, the Bonferroni adjustment might be appropriate — but only if we were to **claim** significance for our results. We prefer to conceive our experiments as what they indeed are: collecting cumulative evidence for trends that have to be corroborated anyway with other (types of) data. In Tables 1 to 3, we report the overall recognition rate RR (number of correctly classified cases divided by total number of cases or weighted average) and CL (a 'class-wise' computed recognition rate, i.e. the mean along the diagonal of the confusion matrix in percent, or unweighted average).

4.1. Separate Classification

In this section, we report on those initial experiments that were conducted at each site with different features and different classifiers, thereby providing a baseline for different automatic classification strategies. Essentially, one and

[5] Such a criterion is of course not based on syntactic considerations. Full-fledged sentences are, however, rather sparse in the register *'giving commands to a robot'*.

[6] For the turns containing our 6070 AMEN words, this means 17618 words in 3996 turns; stop words consisted of 596 fragments and 196 auxiliaries (some words both); this results in 16856 remaining words. Note that we could have identified more stop words, but this would be rather data-dependent and we chose to avoid that. For six turns containing only stop words, no turn-based labels were generated.

Site	# of features		# per type of features						domain		classification				
	original (4024)	selected (381)	prosodic	spectral	MFCC	POS	lexical	genetic search	turn	word	classifier	RR	CL	ROVER	features combined
FAU	303	87	19	-	-	6	62	-	√	√	Neural Networks	55.8	55.3	√	√
TUM	980	103	9	17	22	2	50	3	√	-	SVM	59.3	56.4	√	√
ITC	32	32	26	-	-	6	-	-	√	√	Random Forest (RF)	57.6	55.8	√	√
UKA	1320	25	6	-	5	-	14	-	√	-	Linear Regressor	59.1	54.8	-	√
UA	1289	84	10	1	73	-	-	-	√	-	Naive Bayes	50.9	52.3	√	√
LIMSI	76	26	9	9	-	5	3	-	√	-	SVM	54.9	56.6	√	√
TAU	24	24	24	-	-	-	-	-	√	-	Rule-based	48.9	46.6	-	√

Table 1: *Features and classifiers: per site, # of features before/after feature selection; # per type of features, and their domain; classifier used, weighted average recognition rate RR and non-weighted class-wise averaged recognition rate CL; used or not used (-) in ROVER and in classifiaction with all features; SVM = Support Vector Machines, POS = part-of-speech.*

the same database is independently used by each authoring site reporting different results. This effectively defines a range of performance for this task.

For the results given in Table 1, the 3990 cases, the labels, and training and test sets were identical across all sites; only the features and classifiers differed. The types of features included[7]:

- **prosodic**: F0, energy, duration, and other types of supra-segmental information such as jitter and shimmer;
- **spectral**: modelling Harmonics-to-Noise ratio, formants with band-width etc.;
- **MFCC**: the usual MFCC features plus derivatives;
- **part-of-speech (POS)**: based on coarse word classes such as nouns, particles, etc. provided by FAU;
- **lexical**: single words, or bag-of-word classes (Joachims, 1997);
- **genetic search**: features generated automatically, based on evolutionary alteration and combination.

Irrespective of the types of features and classifiers used, the results are roughly of the same order of magnitude; these figures are, for a 4-class problem and for realistic, spontaneous speech which does not only contain prototypical, very clear cases, in the expected range.[8] Our heuristic threshold of 70% for the definition of MV cases, cf. above, may have resulted in lower classification performance than a threshold of 50%. However, we were not interested in manipulating the data to obtain the highest possible recognition rates, but rather in a realistic setting which takes into account possible applications. For the same reasons, we avoided focusing on only those turns in which the labellers fully agreed, which could have led to a classification performance of up to 80% for our 4-class problem.

The results in Table 1 illustrate an initial range of performance for this task; they should not be conceived of as competing with each other. We found it hard to control all aspects of processing at the different sites which used, e.g. different feature normalization and selection procedures.[9] '√' in the last two columns means that these classifier outputs (cf. columns 13–14) and the features from columns 4–9 were put into ROVER and into a classification which combines features from all sites respectively. Our intention was that with this step, each site can reduce its own large feature set (sometimes $>$ 1000 features) to a smaller set with most of the relevant features.

4.2. Combining Classifiers

When multiple classifiers are available, it is possible to combine their independent results to obtain a composite output whose classification performance is higher than that of the individual systems. In automatic speech recognition (ASR), this is normally achieved using the ROVER framework described in (Fiscus, 1997). Basically, ROVER per-

[7]Note that at times, assignments of a feature to one of these feature cover classes is not unequivocal.

[8]There are some studies available describing realistic speech with two or three classes. As for the very few with four classes and classification performance (CL) well above 60%, it can be shown that the results were 'fine-tuned' somehow; such strategies are dealt with in (Batliner et al., 2005b).

[9]The results reported by TAU are obtained with one specific type of prosodic feature (intonation model pitch features) whereas the other sites used multiple prosodic feature types. FAU and ITC followed a two-stage strategy: they first computed word-based features using the manually corrected segmentation; in a second step, turn-based features were computed based on these word-based features, cf. column 11 in Table 1. Some of the LIMSI features were speaker-normalized. FAU independently selected acoustic/part-of-speech and lexical features each with sequential feature selection (SFS), LIMSI used several different methods, TAU none, all others used SFS based on all feature types. Feature selection has been done independently for the two computations in the 2-fold cross-validation, then the set union of the features was used again; these results are reported in Table 1. This procedure yields sub-optimal performance but guarantees that all possibly relevant features will be kept for the combined classification. As for features modelled and/or classifiers used, cf., in addition to the other references, (Schuller et al., 2005; Vogt and André, 2005; Devillers et al., 2005; Kießling, 1997).

forms a word alignment among independent ASR outputs, and later combines the best hypotheses and their confidence measures to find the most probable word. For our purposes, the alignment step can be skipped, while the scoring step is almost identical to that described in (Fiscus, 1997): the final label e^* is chosen using the following:

$$e^* = \arg\max_e \left[\alpha \cdot \left(\frac{N(e,i)}{\sum_i N(e,i)} \right) + (1-\alpha) \cdot C_k(e,i) \right]$$

where $N(e,i)$ is the frequency of label e in the i outputs, $C_k(e,i)$ is their combined confidence measure, and α is a weighting factor. $C_k(e,i)$ has been evaluated in $k = \{1,2,3\}$ different ways: the straightforward method assumes no weighting ($\alpha = 1$); C_2 is the mean of the confidence scores while C_3 is their maximum. For both these last two systems, α is usually chosen using a cross-validation data set. Due to data scarceness, we chose values of α that maximize RR on the training set, obtaining values for α between 0.7 and 0.9. In other words, when testing on the OHM subset of the data, we selected α by maximizing RR on the MONT subset, and vice-versa. As the original confidences for UKA and TAU were not available, we used altogether the output of five classifiers, cf. Table 1, column 15; results are given in Table 2.

confidence	k	α	RR	CL
C_1, none	1	1.0	62.8	61.9
C_2, mean	2	0.7 - 0.8	63.1	62.2
C_3, max	3	0.8 - 0.9	63.5	62.4

Table 2: *ROVER results obtained by combining the outputs and the confidences of 5 classifiers, cf. Table 1.*

4.3. Combining Features

We now report on classifications with all 381 'most relevant' features from all sites, cf. Table 1, columns 3–9. In Table 3, RR and CL are given for three different classifiers. Feature selection was performed independently for the two training sets MONT and OHM in the 2-fold cross-validation; the number (#) of 'surviving' features is given in columns 2–3. SVM and RF classifiers, using the surviving features, outperform all results obtained independently at each site. These two more sophisticated classifiers perform some percent points — but not considerably — better than the out-of-the-box LDA classifier which used considerably fewer features. The difference in performance may become more pronounced if we were to use a leave-one-speaker-out strategy.

A lack of space makes it prohibitive to fully explore the possible gain in knowledge from combining features and classifier outputs, but we attempt a cursory analysis in Table 4. We first give the number of features per type used by the three classifiers in the two 2-fold cross-validations MONT and OHM; the last line shows the number of features per type summing up to 381. Each feature type has been used throughout, and for each run, features from all sites were used. Note that the 'original' 4024 features were obtained with quite different methods – some by 'brute force' and automatic selection, some using prior knowledge. The SVM and LDA classifiers appear to use more lexical features in relation to RF which uses more acoustic features. Even if each additional feature contributes only negligibly in terms of performance — the size of the feature vectors in Table 3 grows much faster than classification accuracy — they may be valuable for subsequent interpretation.

classifier	# selected features		RR	CL
	MONT	OHM		
LDA	53	67	58.8	56.3
SVM	159	150	61.8	57.9
RF	299	284	60.8	58.7

Table 3: *Classification performance, combining 381 features from all sites, feature selection for 2-fold cross-validation on the training set, with 3 different classifiers; LDA = Linear Discriminant Analysis.*

classifier	training set	prosodic	spectral	MFCC	POS	lexical	gen. search
LDA	MONT	16	5	6	4	21	1
	OHM	19	2	11	3	31	1
SVM	MONT	47	14	37	7	53	1
	OHM	34	14	33	8	59	2
RF	MONT	102	27	100	18	49	3
	OHM	101	27	100	15	38	3
# original features		103	27	100	19	129	3

Table 4: *# of features used per type/per classifier/per training set.*

5. Discussion and Future Work

It is not very difficult to fine-tune classifier performance and obtain considerably higher recognition rates than those reported in this paper, by concentrating on prototypical cases for example — in (Batliner et al., 2005b), up to 75.5 % CL for the same 4-class problem with an LDA classifier — and/or by using leave-one-speaker-out. For prototypical exemplars, we could focus on only those cases where a majority of 4 or 5 out of 5 labellers agreed. In our opinion, to start with, it is more important to establish solid baselines such as those shown in Tables 2 and 3. With ROVER, we have shown an absolute improvement of up to 5.8 % with respect to the best independent site result for CL, cf. Table 2 versus Table 1. By combining features from all sites, we achieved up to 2.1 % absolute improvement for CL, cf. Table 3 versus Table 1. It appears that the combination of different classifiers with different (types of) features which is used by ROVER can model the distribution better than just the use of all 'surviving' features in one and the same classifier.[10]

In future work, we hope to address the following topics:

[10]Note that RFs are an exception, as they are actually a multi-classifier system (Breiman, 2001) composed of a large set of classification trees, each one working on a randomly sampled sub-

- pre-processing: various strategies such as automatic versus manual segmentation and F0 extraction, and forced alignment versus processing based on word-hypothesis graphs[11];

- units and context: turn- versus word-based processing; mapping chunks of words onto 'emotionally significant' units; taking into account of session context in turn-based classification;

- phonetic and linguistic 'substance': which features and types of features are most relevant, and which are not, and why is this the case?

- pattern classification: optimization of classifiers, comparison of performance with and without a loss matrix; possibly automatic feature generation, genetic programming and boosting, and decorrelation of features with PCA; using other knowledge sources such as language models.

6. Concluding Remarks

The idea behind this CEICES endeavour has been to co-operate closely by assembling and evaluating together all kinds of features, both acoustic and linguistic, rather than to compete between sites as in the more common assessment and evaluation procedures (Gibbon et al., 1997). The small performance differences between the authoring sites (Table 1) have to be traced back to differences in either features, classifiers, or feature space optimization. We have shown that co-operation leads to improvements if we simply accumulated and evaluated all features from all sites together at the input level of classification, cf. Table 3. However, results were 'only' up to some two percent points better than the best results obtained at any single site. Further impovements are possible by combining different sets of features with different types of classifiers, cf. the results obtained with ROVER in Table 2. Markedly better classification peformance might not be possible with a further fine-tuning of features and classifiers; in addition we should take into account some of the aspects mentioned in section 5.

7. Acknowledgements

This work was partly funded by the EU in the projects PF-STAR under grant IST-2001-37599 and HUMAINE under grant IST-2002-50742. The responsibilty lies with the authors.[12]

set of features. It is surprising that they are outperformed by the ROVER framework. A possible interpretation is that the diversity of classifier types, as used by ROVER, is crucial. Furthermore, we must stress that the two approaches — in general all results reported in Tables 2 and 3 — cannot be directly compared as they do not rely on the same features, cf. columns 15–16 in Table 1.

[11]For fully automatic processing, some features such as bag-of-words or part-of-speech have to be extracted from a word-hypothesis graph and will not always be correct.

[12]The ROVER computation of section 4.2. was done at ITC, the combined classification of section 4.3. at FAU, TUM, and ITC.

8. References

A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Nöth. 2003. How to Find Trouble in Communication. *Speech Communication*, 40:117–143.

A. Batliner, S. Steidl, C. Hacker, E. Nöth, and H. Niemann. 2005a. Private Emotions vs. Social Interaction - towards New Dimensions in Research on Emotion. In *Proceedings of a Workshop on Adapting the Interaction Style to Affective Factors, 10th International Conference on user Modelling*, pages 8 pages, no pagination, Edinburgh.

A. Batliner, S. Steidl, C. Hacker, E. Nöth, and H. Niemann. 2005b. Tales of Tuning – Prototyping for Automatic Classification of Emotional User States. In *Proceedings of Interspeech 2005*, pages 489–492, Lisbon.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18(1):32–80.

L. Devillers, L. Vidrascu, and L. Lamel. 2005. Challenges in real-life emotion annotation and machine learning based detection. *Neural Networks*, 18:407–422.

J. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *ASRU*, Santa Barbara, USA.

Dafydd Gibbon, Roger Moore, and Richard Winski, editors. 1997. *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, Berlin.

T. Joachims. 1997. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Technical report, LS-8 Report 23, Dortmund, Germany.

A. Kießling. 1997. *Extraktion und Klassifikation prosodischer Merkmale in der automatischen Sprachverarbeitung*. Shaker, Aachen.

Thomas V. Pernegger. 1998. What's wrong with Bonferroni adjustment. *British Medical Journal*, 316:1236–1238.

B. Schuller, R. Müller, M. Lang, and G. Rigoll. 2005. Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensembles. In *Proceedings of Interspeech 2005*, pages 805–808, Lisbon, Portugal.

S. Steidl, M. Levit, A. Batliner, E. Nöth, and H. Niemann. 2005. "Of All Things the Measure is Man" - Classification of Emotions and Inter-Labeler Consistency. In *Proceedings of ICASSP 2005*, pages 317–320, Philadelphia.

Thurid Vogt and Elisabeth André. 2005. Comparing Feature Sets for Acted and Spontaneous Speech in View of Automatic Emotion Recognition. In *Proceedings of IEEE International Conference on Multimedia & Expo (ICME 2005)*, Amsterdam, The Netherlands.

A Taxonomy of Applications that Utilize Emotional Awareness

Anton Batliner[†], Felix Burkhardt[*], Markus van Ballegooy[*], Elmar Nöth[†]

[†]Lehrstuhl für Mustererkennung, Universität Erlangen–Nürnberg, Martensstr. 3,
91058 Erlangen, Germany
{batliner,noeth}@informatik.uni-erlangen.de
∗T-Systems Enterprise Services GmbH, Goslarer Ufer 35, 10589 Berlin, Germany
{Felix.Burkhardt,Markus.van-Ballegooy}@t-systems.com

Abstract

This paper deals with human-computer interaction applications that utilize emotional awareness. We will confine our discussion on speech-based applications. Prerequisites — training data, annotations — as well as state of the art in recognition and synthesis are addressed focusing on usability in possible applications and keeping restrictions in industrial environments in mind. We will present a taxonomy of applications using criteria such as online/offline, mirroring/non-mirroring, emotional/non-emotional and critical/non-critical system reactions. Based on a list of prototypical applications we check the consistency and usefulness of this taxonomy.

Taksonomija aplikacij, ki uporabljajo čustveno zavedanje

Prispevek se ukvarja z aplikacijami za interakcijo med človekom in računalnikom, ki uporabljajo čustveno zavedanje. Diskusijo omejujemo na govorne aplikacije. Obravnavamo predpogoje — učni podatki, označevanje — in najnovejša dognanja v razpoznavanju in sintezi govora, osredotočamo se na uporabnost možnih aplikacij ob upoštevanju omejitev v industrijskih okoljih. Predstavljamo taksonomijo aplikacij glede na to, ali omogočajo sprotni odziv ali ne, ali so zrcaljene ali ne, ali so čustvene ali ne ter ali so reakcije sistema kritične ali ne. Na podlagi seznama prototipskih aplikacij preverjamo doslednost in uporabnost te taksonomije.

1. Introduction

Taking into account emotional and emotion-related (affective) states of users interacting with automatic systems – be this automatic dialog systems or automatic systems in general – has been an emerging topic during the last years. The idea behind is simple: even if automatic systems were much better than state-of-the-art systems are today at interacting with human users, there is still a certain quality missing, namely recognizing and dealing with not only the semantics of the user's message but his/her emotional state as well. Often it is supposed that only then, the machine is really on an equal footing with the human communication partner.

The channels monitored and used by the system have probably been the most obvious way of telling apart different types of applications – alone because traditionally, they belong to different disciplines: speech (and by that, acoustic and/or linguistic information), facial expressions, gestures, body postures, and background knowledge (i.e. context in its broader sense). In this paper, we want to concentrate on systems that deal with speech only (recognition and generation/synthesis), and we want to introduce some further characteristics which can be used for a tentative taxonomy. By that, we concentrate mostly on choices made by the system designers and much less on the choices which are made – more or less consciously – by users of such systems.

In the history of emotion research, some prototypical application examples emerged, cf. e.g. (Picard, 1997). We introduce such a list of different application-sketches here and will use them as examples in the coming chapters of this article, representing substitutes or prototypes for similar applications that may be used in different fields.

Emotional Monitoring: E.g. anger detection can be used to soothe disgruntled users or for automatic quality monitoring.

Emotional Mirror: Speech analysis can be used for self-training (e.g. '*Do I sound boring?*')

Understanding Tutor: Teacher-student communication can be enhanced enormously by emotional channels in order to monitor and augment motivation.

Emotion-aware Surrounding: Quite an old idea is a computer controlled environment that adapts automatically on the user's mood by e.g. playing '*just the right music*' or adjusting automotive system reaction.

Believable Agent: The naturalness of an artificial '*being*' and the appearance of intelligence (we will not go into philosophical questions here) is highly altered by emotional expressions; especially gaming applications can benefit.

Emotional Chat: The high success of the so-called '*Emoticons*' shows how strong the human desire is to express emotion in mode-restricted computer mediated communication (CMC). Special channels can be provided to facilitate this and analysis can be used to automate emotional labeling.

This article is structured as follows: section 2. will discuss important aspects for data collection from the application point of view. The following Section 3. deals with technical potentialities of recognition and synthesis. Because applications will not be greenfield developments, we include a subchapter on industrial requirements. In Section 4. we propose our taxonomy and classify our example applications accordingly. We conclude with final remarks in Section 5.

2. Models need Data

Emotion-aware applications are based on models and models rely on data. Thus we will sketch some central issues with respect to data-collection from the application's perspective in this section. Although it is trivial to remark that the data should be as close to the intended application as possible (a self-training application would be perfect), it is of course not economic to collect data for each application. In order to reuse data-collections, standardized ways to annotate data are required and will emerge.

In contrast to other speaker characterizations like age or gender, emotion is a fuzzy topic. The performance of a human labeler can be measured by comparison with other labelers. Standardized ways to control performance of labelers, measurements of inter-labeler agreement (Steidl et al., 2005) and finding unified labels will be essential for application deployment. In many applications, e.g. the anger-detecting voice portal, the recognition of emotion and the system's reaction must play hand-in-hand, and dialog designer and labeler must rely on a common emotion-coding language.

Because emotional expression depends highly on speaker idiosyncrasies, speaker-dependent modeling should be preferred, if possible. Such personalized applications could require an emotion-recognition training process just like dictation systems do nowadays.

One of the problems that arises in data-collection and makes Heisenberg's uncertainty principle come into mind, comes from the fact that people, if they know they will be monitored, react differently then if they had not be aware of the monitoring; this phenomenon has been called 'observer's paradox' (Labov, 1970). On the other hand hidden monitoring is often difficult, probably unethical and generally prohibited by law.

3. State of the Art in Speech Technology

3.1. Emotion Recognition

The state of emotion recognition in general still suffers from the prevalence of acted laboratory speech as object of investigation. The high recognition rates of up to 100% reported for such corpora cannot be transferred onto realistic, spontaneous data. For realistic databases, performance for a two-class problem is typically $< 80\%$, for a four-class problem, $< 60\%$, cf. (Batliner et al., 2005); normally, acoustic (mostly prosodic and/or MFCC based features) and some plain linguistic features such as bag-of-words are employed. Performance can be improved

- by employing highly sophisticated classifiers,
- by concentrating on prototypical, clear cases ((Batliner et al., 2005) report up to 77.5% for a four-class problem),
- by mapping onto cover classes (for instance, only taking into account positive vs. negative valence),
- by taking into account cost functions (for instance, penalizing only 'severe' confusions),
- by resorting to speaker-dependent[1] modeling, as indicated above, if this is suitable for the resp. application,
- and by using other, additional knowledge bases (for instance, dialog and/or interaction history).

Larger databases, i.e. more training data, seem to be a must but are difficult to obtain because the reference (ground truth, i.e. the phenomena that have to be recognized) cannot be obtained easily: for word recognition, a simple transliteration will suffice; for emotion recognition, manual annotation is normally necessary, time-consuming and costly. In some few scenarios, it might be possible to resort to external evidence, allowing a sort of automatic annotation; for instance, in a car driving scenario, actual speed and movements of the steering wheel could be monitored automatically and used for labeling training data.

3.2. Approaches in Speech Synthesis

First attempts to simulate emotional speech by means of speech synthesis started soon after the first mature speech synthesizers were developed. For an overview on the history of emotional speech synthesis, the reader may be referred to (Schröder, 2001). Most of today's research concerning emotional speech synthesis is still dealing with a small set of basic emotions (e.g. the so-called 'big *n*', *n* being a small number like 4,5,6) such as anger, sadness, fear, or joy.

Because speech synthesis until now still has to solve more pressing problems than the simulation of emotional speech, and applications for emotional synthesis are yet more in the future than for emotion-recognition[2], the research is less advanced. The simulation of realistic and natural emotional speech expression is vital for most application scenarios, e.g. for applications to enhance the believability of talking heads. Some ideas of yet mainly unsolved problems that might be required by prospective applications are summarized in the following items:

- Simulation of a larger set of discrete emotions that are displayed more subtle than 'the big *n*' performed in a cartoon-like style.
- Blending between two or more emotions, like e.g. anger and sadness, and finding models for the transition from one emotion to a different one during one utterance.
- Finding acoustic correlates for other models than discrete emotions like emotional dimensions or stimulus evaluation checks in order to support different emotion models directly.
- There is still a big gap between system-modeling formant-synthesis (flexible but unnatural) and

[1] For speaker-dependent modeling, spectral features might come into play as well which might not be suitable for speaker-independent modeling.

[2] Consider that applications that utilize emotional synthesis often depend on artificial intelligence in order to know when to speak with which emotion.

manipulation-avoiding speech-concatenating non-uniform unit-selection (high quality but inflexible). Solving the problem of the discrepancy between naturalness and inflexibility of non-uniform unit selection approach is vital for emotion-simulation, a fact that becomes manifest e.g. in voice-quality modeling.

For the time being emotional high quality synthesis constrains in either adding some emotional interjections to the data, cf. (Eide et al., 2003), or reducing the set of emotions to a binary choice, e.g. agitated speaking style vs. normal. Note that this is easier in low-quality / small footprint approaches that include signal manipulation like formant- or diphone-synthesis.

3.3. Industrial Requirements

Emotion recognition in an industrial environment has to fulfill a set of requirements that should be considered while planning an emotion-aware application:

- The emotion aware modules must integrate into the existing architecture and should use standardized interfaces as much as possible.
- In an HCI system the delay caused by the processing must not obstruct the dialog flow.
- For most applications a classification task must be based solely on automatically gained features.
- The recognition algorithms have to work often on highly noisy data and performance results from laboratory studies are not directly applicable.
- The procedures must attend to economic issues, e.g. algorithms that are IPR (intellectual property rights) protected must be avoided and manual labor should be restricted.

With a growing dispersion of emotional applications a market of emotion-aware components operating on different processing steps will emerge; data vendors will collect emotion-annotated data to train the models, speech core-technology vendors will offer emotion recognition and simulation components, integrators will offer 'emotion-modules' for dialog platforms and gaming engines. This market will rely on a set of standards yet to be developed.

4. Applications

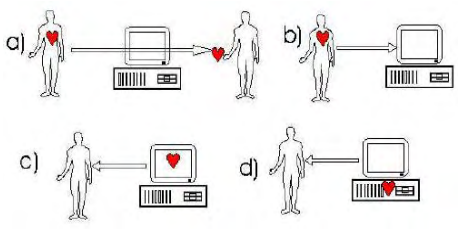


Figure 1: Different uses of emotional processing in computer systems.

Emotional applications can be thought up in an arbitrary number. This Section will give a frame to classify applications based on common features in order to facilitate the process of thinking up new useful application scenarios and identify reuse of common modules for existing ones.

Emotions can be processed in several places of an information-processing system (Picard, 1997). Figure 1 displays several possibilities:

a) Broadcast: Emotional expression is an important channel of information in human communication. In telecommunication it might be desirable to provide for a special channel for emotional communication. A popular example are the so-called '*emoticons*' used in e-mail communication.

b) Recognition: The human emotional expression can be analyzed in different modalities, and this knowledge is used to alter the system reaction.

c) Simulation: Emotional expression can be mimicked by the system in order to enhance a natural interface or to access further channels of communication, like e.g. uttering urgent messages in an agitated speech style.

d) Modeling: Internal models of emotional representations can be used to represent user- or system states or as models for artificial intelligence, e.g. influence decision making.

Of course systems may combine several of these features for an integrated application like the above mentioned emotional tutoring system. As we are focused on speech processing in this article, we will not follow up on the AI-possibilities but confine on the recognition, transport and simulation for communicative reasons.

Instead of identifying system components, we can use different features to tell apart emotional systems: In Table 1, we present four binary features telling apart different types of applications: +/- {*online, critical, mirroring, emotional*}. In principle, this would result in 16 possible combinations, but some combinations are less likely than others, e.g. an *offline* application cannot show *mirroring* as defined in the table and need not be *critical* as there can be a human check on the performance. In order to check the validity of our taxonomy we will classify our example applications accordingly. Most of these applications are already on the market or in a prototype stadium.

Emotional Monitoring: One of the most often expressed ideas for emotional speech monitoring are voice portals that use detection of negative feelings such as anger to appease by *mirroring* their expressions (Burkhardt et al., 2005). This application is *critical* because users could really get angry if they were – wrongly – 'accused' being angry but are not. It is an *online* application because the system reacts directly but could be imagined in an *offline* scenario, where the customer satisfaction is measured later by classifying the call logs. Depending on the system reaction it might be *emotional*, i.e. try to soothe the user

features	description
	system design
online	system reacts (immediately/delayed) while interacting with user
offline	no system reaction, or delayed reaction after actual interaction
mirroring	user gets feedback as for his/her emotional expression
non-mirroring	system does not give any explicit feedback
emotional	system reacts itself in an emotional way
non-emotional	system does not behave emotionally but 'neutral'
	meta-assessment
critical	application's aims are impaired if emotion is processed erroneously
non-critical	erroneous emotion processing does not impair application's aims

Table 1: Criteria for Taxonomy

by adequate dialog strategies; in a *non-emotional* variant, it can simply transfer to a human agent. A *non-mirroring* variety might result in ethical concerns: it could be used to monitor call center agents or psychotic patients and enable supervisors intervene if necessary. Another *non-mirroring* variety consists of automatically identifying untrustworthy customers, e.g. in an insurance portal.

Emotional Mirror: The emotional mirror was often suggested by R. Picard's team and was developed in the Jerk-o-meter application (Madan et al., 2005). A person's speech is monitored for emotional expression which can be used for training reasons. This application is definitely *mirroring* and could be used *online* as well as *offline*. It is *critical* because emotion recognition is the central aspect of the application which will, however, not react itself *emotional*. Note that for this application, training data could perhaps be acted because the user's intention might be to sort of act him/herself.

Understanding Tutor: Automatic tutoring is an interesting topic in a growing information society and emotional strategies are important to enhance motivation. Several systems have been suggested already, e.g. (Poel et al., 2004). This application must be *online* and *mirroring* to react directly, e.g. on the pupils boredom, and will be *emotional* to motivate the pupil but is not necessarily *critical*, i.e. reactions could be quite subtle.

Emotion-aware Surrounding: Quite an old idea is a computer controlled environment that adapts automatically on the user's mood by e.g. playing '*just the right music*' or adjusting automotive system reaction. This set of applications are of course *online* and *mirroring* but are not *emotional* themselves. Whether they are *critical* would be a distinction between an emotional CD-player and a car reacting on a stressed driver.

Believable Agent: The naturalness of artificial '*beings*' and the appearance of intelligence is highly altered by emotional expression; especially gaming applications can benefit. This application could be regarded as generic term for the understanding, emotional tutor and can consistently be classified just as this.

Emotional Chat: In an avatar-based chat system the avatar's emotional expression could be controlled by the user deliberately. An example where the emotional expression gets measured automatically would be Picard's classroom barometer application (Picard, 1997) where the tele-conference teacher is informed about the pupils attention via affective jewelry. It is an *online* (if automatically gained), *mirroring*, *non-emotional* application likely to be *non-critical* (if self-reported).

5. Concluding Remarks

In this article, we sketched necessary prerequisites for emotional systems such as data, recognition, and synthesis; in the resp. sections, we could not go into detail and only addressed some pivotal topics. Our main contribution is a tentative taxonomy of emotional applications. Taxonomy as such makes life easier: we know what we are looking for, and what we should decide between. It can be a sort of roadmap what features such as given in Table 1 to incorporate or to disregard for a new application, and to help thinking of new ways of incorporating emotional awareness. Such features or feature combination can characterize modules that can be turned on or off, depending on user characteristics, dialog step, and confidence measures.

So far, we have mostly talked about technology.[3] As long as emotional applications are in their infancy, this might seem OK. What has not been addressed is ethical issues – a topic which will be more pressing the better such systems perform. Emotion aware and expressive applications might seem more intelligent and capable than they really are. Thus it might not always be the best idea to make a system react emotionally, and it might not always be desirable because of ethical reason to make the user conceive of the system as human-like – not to speak of other ethical questions that will arise. Last but not least: just like convictions should not be based on lie-detectors, decisions that impact peoples' welfare severely should not be based on emotion recognition.

[3] Some more basic questions are dealt with in (Picard, 1997; Picard, 2003).

Acknowledgments: This work was partly funded by the EU in the framework HUMAINE (*http://emotion-research.net/*) under grant IST-2002-507422, and by the German Federal Ministry of Education and Research (*BMBF*) in the framework of SmartWeb (Grant 01IMD01F). The responsibility for the contents of this study lies with the authors.

6. References

A. Batliner, S. Steidl, C. Hacker, E. Nöth, and H. Niemann. 2005. Tales of Tuning – Prototyping for Automatic Classification of Emotional User States. In *Proc. 9th Eurospeech - Interspeech 2005*, pages 489–492, Lisbon.

F. Burkhardt, M. van Ballegooy, R. Englert, and R. Huber. 2005. An emotion-aware voice portal. In *Proc. Electronic Speech Signal Processing ESSP*, pages 123–131.

E. Eide, A. Aaron, R. Bakis, W. Hamza, M. Picheny, and J. Pitrelli. 2003. A corpus-based approach to expressive speech synthesis. In *Proc. ISCA ITRW on Speech Synthesis, Pittsburgh*, pages 79–84.

W. Labov. 1970. The Study of Language in its Social Context. *Studium Generale*, 3:30–87.

A. Madan, R. Caneel, and A. Pentland. 2005. Voices of attraction. In *Proc. Augmented Cognition, HCI 2005, Las Vegas*.

R. Picard. 1997. *Affective computing*. MIT Press.

R. Picard. 2003. Affective Computing: Challenges. *Journal of Human-Computer Studies*, 59:55–64.

M. Poel, R. op den Akker, D. Heylen, and A. Nijholt. 2004. Emotion based agent architectures for tutoring systems: The ines architecture. In *Cybernetics and Systems 2004. Workshop on Affective Computational Entities (ACE 2004)*.

M. Schröder. 2001. Emotional speech synthesis - a review. In *Proc. Eurospeech 2001, Aalborg*, pages 561–564.

Stefan Steidl, Michael Levit, Anton Batliner, Elmar Nöth, and Heinrich Niemann. 2005. "Of All Things the Measure is Man" - Classification of Emotions and Inter-Labeler Consistency . In *Proc. of ICASSP 2005*, pages 317–320.

Context-Dependent Acoustic Modelling of Croatian Speech

Sanda Martinčić – Ipšić[*)] and Ivo Ipšić[+)]

[*)] Department of Informatics,
Faculty of Philosophy
University of Rijeka,
Omladinska 14, 51000 Rijeka, Croatia
Phone: (385) 51-345 046 Fax: (385) 51-345 207
E-mail: smarti@ffri.hr

[+)] Technical Faculty,
University of Rijeka,
Vukovarska 58, 51000 Rijeka, Croatia,
Phone: (385) 51-651 421
E-mail: ipsic@riteh.hr

Abstract

This paper presents experiments of Croatian speech modelling used in speech recognition as well as in speech synthesis. The proposed acoustic model is based on context-dependent triphone hidden Markov models and Croatian phonetic rules. For speech recognition and speech synthesis experiments a common Croatian speech corpora is used. The experiments have shown that Croatian speech corpora, Croatian phonetic rules and hidden Markov models as the modelling formalism can be used to develop speech recognition and speech synthesis systems in parallel. The proposed procedures for Croatian acoustic modelling were developed as speech interfaces in a spoken dialog system.

Kontekstno odvisno akustično modeliranje hrvaškega govora

V članku so opisani postopki akustičnega modeliranja, ki so bili uporabljeni pri razpoznavanju in sintezi hrvaškega govora. Predlagani akustični model temelji na kontekstno odvisnih trifonskih modelih in na fonetičnih pravilih hrvaškega govora. Za učenje akustičnih modelov je izbran formalizem prikritih Markovovih modelov in korpus hrvaškega govora. Razviti postopki za razpoznavanje in sinetezo hrvaškega govora so del enotnega sistema za govorni dialog.

1. Introduction

The paper describes procedures for acoustic modelling of Croatian speech. The proposed context-dependent acoustic model is used in the speech recognition as well as in the speech synthesis module of a spoken dialog system for Croatian speech in the domain of weather forecasts. Using such a system a user can ask questions about weather conditions and forecasts. The dialog system would provide information about weather in different regions of Croatia and for different time periods, collecting the information from the available web sites over the Internet (Žibert et al., 2003). The spoken dialog system includes modules for speech recognition, spoken language understanding and speech synthesis. The speech recognition and synthesis module relay on data-driven statistical and rule-based knowledge approach. Data driven statistical approach is based on large quantities of spoken data collected in speech corpora. Both approaches must be combined in a spoken dialog system because there is not enough speech data to statistically model the human speech and there is not enough knowledge about processes in human mind during speaking and understanding.

Since the main resource in a spoken dialog system design is the collection of speech material the Croatian domain related speech corpora is presented. Further the acoustic modelling procedures of the speech recognition system including phonetically driven state tying procedures are given. Conducted speech recognition experiments and speech recognition results are presented in the third section. The fourth part explains the Croatian trainable speech synthesis, which is based on the same context-dependent acoustic model as the one used in the speech recognition experiments. Some advantages of the same acoustical modelling approach for Croatian speech recognition and speech synthesis are discussed. We conclude with the description of current activities and future plans in Croatian speech technologies

2. The Croatian speech database

The Croatian speech corpora VEPRAD includes weather forecasts and reports spoken within broadcast news of national radio (Martinčić-Ipšić et al., 2004). The collected speech material is divided in several groups: weather forecasts read by professional speakers within national radio news, weather reports spontaneously spoken by professional meteorologists, other meteorological information spoken by different reporters and radio News.

The VEPRAD corpus is a multi-speaker speech database and contains 13 hours of the transcribed speech spoken in the studio acoustical environment and the telephone speech. The spoken utterance has its word level transcription. The corpora statistics is shown in Table 1

The first part, VEPRAD radio database, of the collected speech material consists of transcribed weather forecast. This is a multi-speaker database, which contains speech utterances of 11 male and 14 female professional

speakers. VEPRAD radio part consists of 3566 utterances and lasts 6 hours and 17 minutes. The transcribed sentences contain 57896 words, where 1354 are different. Relatively small number of different words shows that the VEPRAD speech database is strictly domain oriented.

From the VEPRAD radio database one male speaker was selected for speech synthesis voice. For the selected speaker additional 85 minutes of radio news speech was recorded and transcribed. The synthesis part database includes 1111 utterances with 3840 different words.

The third part, VEPRAD telephone database, contains weather reports given by 7 female and 5 male professional meteorologists daily over the telephone. The 158 transcribed weather reports are lasting 5 hours and 39 minutes and contain 1803 different words in 3223 utterances. Most of the speech captured in the VEPRAD telephone database can be categorized as semi-spontaneous. This data is very rich in background noises such as door slamming, car noise, telephone ringing and background speaking and contains noise produced by channel distortions and reverberations. All this special events and speech disfluencies and hesitations are annotated in transcriptions by < >.

The transcribing process involved listening to speech parts until a natural break is found. The utterances or parts of speech signals were cut out and a word level transcription file was generated. The speech file and the transcription file have the same name with different extensions. In the process of generating speech files and their transcription we used the Speech Viewer from the CSLU Speech Toolkit (Sutton et. al. 1998) and Transcriber (Barras et al. 2000). Manual correction of automatically segmented phones was performed using the Wavesurfer tool (KTH, 2004). An utterance example z07060702102 is shown in Table 2.

```
z07060702102
postupna naoblaka <uzdah> mjestimice s
pljuskovima i grmljavinom vjetar slab <sil>
a na jadranu povremeno umjeren
jugozapadnjak i jugo <uzdah>
```

Table 2. Example of one transcribed utterance.

VEPRAD	Dur. [min]	No.			Speakers	
		Sentences	Words	Diff. words	Male	Female
RADIO						
Radio weather forecasts	377	3566	57896	1354	11	14
Radio news-(synthesis)	85	1111	12265	3840	1	
Overall RADIO	**462**	**4677**	**70161**	**4504**	**11**	**14**
TELEPHONE						
Telephone weather reports	339	3223	51187	1803	5	7
Overall VEPRAD	**801**	**7900**	**121348**	**5344**	**16**	**21**

Table 1. Croatian speech database statistics.

3. Context-dependent modeling

The Croatian speech recognition and speech synthesis system is based on continuous hidden Markov models of monophones and triphones. The training of speech recognition system was performed using the HTK toolkit (Young et al., 2002), while for speech synthesis training the HTS tool (HTS, 2004), which is as an extension of the HTK, was used. Croatian weather forecasts speech database VEPRAD was used for training of all acoustic models.

3.1. Monophones

The training of speech recognition and synthesis acoustic models started with defining the Croatian phoneme set according to SAMPA (Bakran and Horga, 1996). For each Croatian phoneme a context-independent monophone hidden Markov model was defined.

Initially the monophone models with continuous Gaussian output probability functions described with diagonal covariance matrices were trained. Each monophone models consists of 5 states, where the first and last states have no output functions. The initial training of the Baum-Welch algorithm on HMM monophone models resulted in a monophone recognizer, which was used for the automatic segmentation of the speech signals.

The automatic segmentation of the speech signal to the phone level is performed using the forced alignment of the spoken utterance and the corresponding word level transcriptions. The number of mixtures of output Gaussian probability density functions per state was increased to 20 in the used monophone recognizer.

Further, the monophone models were trained by 10 passes of the Baum-Welch algorithm and the resulted monophone models were used for the initialization of context-dependent triphone hidden Markov models.

Additional models for silence, breath, restarts, hesitations, cough, telephone ringing, modem noise, car beeping and driving, paper turning, door slamming, background speaking and mispronounced words where made. The resulting monophone set has 5 additional models for handling unexpected events and noise in radio speech and 7 additional models for the telephone speech. Since telephone speech is rich in channel distortions and additive environmental noise for telephone data 2 additional models were trained. Additive noise can be

stationary (like computer ventilators, cars or air conditioners) and has a power spectral density that does not change over period of time. Nonstationary noise, in contrary, changes over time and is produced by door slamming, background speaking, telephone ringing, coughing, breathing etc. The channel distortions can be caused by reverberation in telephone network or distortions in broadcasted radio signal. This unexpected events handling approach is known as the explicit noise modelling approach and enables explicit handling of noise errors and acoustic events (Ward, 1989).

3.2. Triphones

In the next step we trained context-dependent cross-words triphone models with continuous density output functions (one to six mixture Gaussian density functions), described with diagonal covariance matrices. The triphone models also consist of 5 states, where the first and last states have no output functions.

The number of cross-word seen triphones in the training data used for radio speech recognition training is 6054 (about 16% of all possible triphones), for telephone speech recognition training is 6054 (about 12% of all possible) and the number in speech synthesis is 8290 (about 13% of the number of all possible triphones) (Martinčić-Ipšić and Ipšić, 2006a).

Therefore there is evidently not enough acoustical material for modelling all possible triphone models. The severe undertraining of the model can be a real problem in the recognizer performance. The lack of speech data is overcome by a phonetically driven state tying procedure.

3.2.1. Croatian phonetic rules and decision trees

The state tying procedure proposed in (Young et al., 1994) allows classification of unseen triphones in the test data into phonetic classes and tying of the parameters for each phonetic class. In our system 216 Croatian phonetic rules are used to build phonetic decision trees for HMM state clustering of acoustic models. The phonetic rules are describing the class of the phonemes according to their articulatory and acoustic characteristics. Some defined Croatian phonetic rules used for the training of phonetic classes are shown in Table 3.

Vowel	a, e, i, o, u, a:, e:, i:, o:, u:, r:
Accented Vowel	a:, e:, i:, o:, u:, r:
High Vowel	i, u, i:, u:
Medium Vowel	o, e, o:, e:
Back	k, g, h, o, u
Affricate	c, C, cc, dz, DZ
Velar	k, g, h
Glide	j, v
Apical	t, d, z, s, n, r, c, l
Strident	v, f, s, S, z, Z, c, C, DZ
Constant Consonant	v, l, L, j, s, S, z, Z, f, h
Unvoiced Fricative	f, s, S, h
Compact Consonant	N, L, j, S, Z, C, cc, dz, DZ, k, g, h

Table 3. Examples of Croatian phonetic rules.

En example of a phonetic decision tree for the Croatian phoneme /h/ is presented in Figure 1. It classifies triphones with the phoneme /h/ in the middle in eight possible classes. At each node the binary question about left and right context is asked and YES/NO answers are possible. The triphones in the same class are sharing the same parameters of state output probability density functions of HMMs.

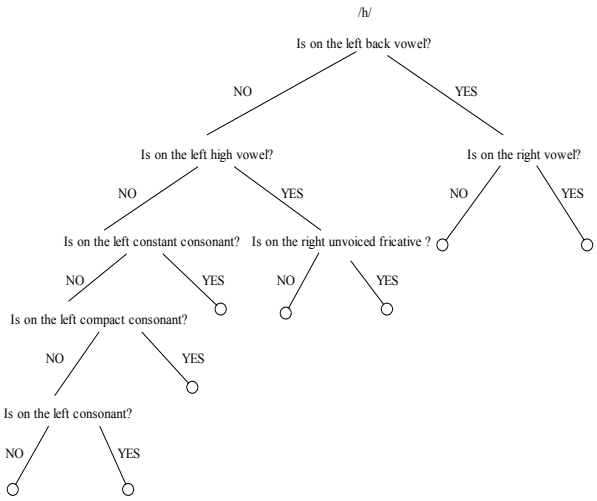


Figure 1: The decision tree of phonetic questions for the left and right context for phoneme /h/.

3.2.2. State tying

State tying enables clustering of the states that are acoustically similar, which allows all the data associated with one state to be used for more robust estimation of the model parameters (Gaussian distribution mean and variance). This enables more accurate estimating mixtures of Gaussian output probabilities and consequently better handling of the unseen triphones.

In the speech recognition state clustering procedure a separate decision tree for initial, middle and final states of each triphone HMM is built using a top-down sequential optimization procedure (Odell, 1995). Initially all relevant states are placed in the root node. So, all states are initially tied together and log likelihood is calculated for this node. The tying procedure iteratively applies phonetic rules to the states of the triphone models and partitions the states into subsets according to the maximum increase in log likelihood. When the threshold is exceeded the tied states are no further partitioned.

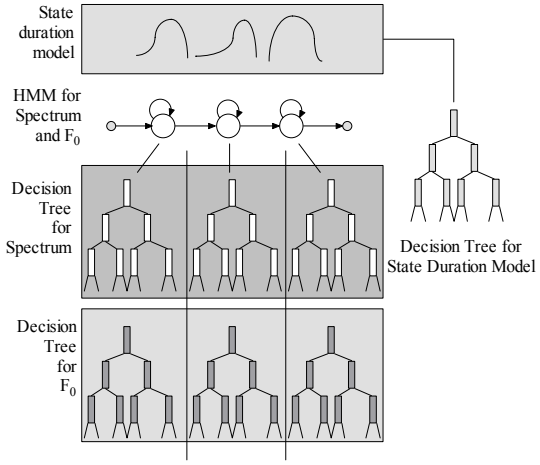


Figure 2. Decision trees for spectrum, pitch and duration in the Croatian speech synthesis system.

In the speech synthesis system the same 216 Croatian phonetic rules used in the speech recognition system were used. The clustering trees for spectral parameters, fundamental frequency F0 and duration were built separately as shown at Figure 2 (Yoshimura, 2000). The clustering trees were built separately because different context clustering factors are relevant for spectral part clustering, pitch clustering and duration clustering but the same Croatian phonetic rules were used.

4. Speech recognition experiments

So far we have performed speech recognition experiments using the VEPRAD radio (Martinčić-Ipšić and Ipšić, 2004) and VEPRAD telephone speech database (Martinčić-Ipšić and Ipšić, 2006a).

In VEPRAD radio speech recognition system 4135 (71%) utterances form 8 male and 8 female speakers were used for training and 1712 (29%) utterances from 3 male and 6 female speakers were used for testing.

In VEPRAD telephone speech recognition system 1982 utterances (61%) where used for acoustic modelling and parameter estimation of context dependent phone models and 1241 utterances (39%) were used for recognition. Speech from 3 female and 3 male meteorologists was used for training and speech from 2 male and 4 female meteorologists was used for testing.

In all experiments bigram language model was used. Estimated perplexity of the VEPRAD radio bigram language model is 13.23 and perplexity of the VEPRAD telephone is 18.09.

Table 4 compares data used for separate training and testing of radio and telephone speech recognition system and for speech synthesis experiments. Bottom part of the table compares the number of monophone models used in each subsystem as well as the number of seen triphones compared to the number of all possible triphones. Number of monophone models trained for speech synthesis is expanded for accented vowels including the occurrence of r as a vowel in Croatian language, and additional models for silence, breathing noises, mispronounced words and noise.

	VEPRAD		
	RADIO	TELEPHONE	SYNTHESIS
# diff. words	4504	1803	3840
perplexity	13.23	18.09	23.6
training			
# utterances	4135	1982	1111
# speakers	8m+8f	3m+3f	1m
testing			
# utterances	1712	1241	41
# speakers	3m+6f	2m+4f	1m
#monophones	30+5	29+7	36+5
triphones			
# all	36756	37597	60521
# seen	6054	4610	8290
% seen	16.47%	12.26%	13.70%

Table 4. The comparison of the data used for training and testing of radio and telephone speech recognition and for speech synthesis.

4.1. Speech feature vector

For speech recognition the speech signal feature vectors consist of log energy, 12 mel-cepstrum features and their derivatives and acceleration coefficients. The feature coefficients were computed every 10 ms for a speech signal frame length of 20 ms.

4.2. Speech recognition results

Speech recognition results for context-dependent and speaker independent recognition of the "clean" radio and noisy telephone speech are presented respectively in the Table 5. The number of different words is in the first raw. Results are given in terms of correctness and accuracy per different number of tied states and different number of Gaussian mixtures, always using the same proposed 216 Croatian phonetic rules set.

	VEPRAD			
	RADIO		TELEPHONE	
#words	**4504**		**1803**	
	% Corr.	**% Acc.**	**% Corr.**	**%Acc.**
# state	**1011**		**646**	
mix 1	78.58	75.74	86.18	72.14
mix 3	79.82	77.52	88.47	76.13
mix 6	**80.60**	**78.86**	**89.30**	**77.54**
# state	**1235**		**980**	
mix 1	78.39	75.41	86.62	72.31
mix 3	79.75	77.51	88.86	74.94
mix 6	80.34	78.51	89.16	76.92
# state	**1977**		**1872**	
mix 1	78.23	74.70	86.73	71.39
mix 3	79.50	76.75	88.15	74.02
mix 6	80.09	77.68	87.99	73.64
# state	**3099**		**2273**	
mix 1	78.03	74.34	86.49	70.45
mix 3	79.34	76.37	87.99	73.25
mix 6	79.61	76.93	87.08	72.02

Table 5. Speech recognition results for radio and telephone data in terms of correctness and accuracy.

The results for radio and telephone speech recognition are in the same error range. At the firs glance this is surprising, but this was actually expected since the number of different word is more then double (4504) as the number of different word in the radio data (1803). This indicates that use of trigram language models should be considered for the radio speech recognizer. Further, since the access to the weather information spoken dialog system is planned by telephone, the speech recognition accuracy for the telephone data is quite promising. The word error rate for telephone data, for the same reason, must be bellow 20% which will be achieved by incorporating more telephone speech in the acoustical model training procedure. And finally both recognition systems performed better when the number of tied states was reduced (using the same phonetic rules) and the number of Gaussian mixtures increased which indicates that more speech should be incorporated in the training of both recognizers for the use in the spoken dialog system.

5. Speech synthesis experiments

The hidden Markov model based trainable speech synthesis use speech corpora for the training of context-

dependent acoustic model, and uses HMM as a generative model for speech production. Similar speech synthesis systems were already developed for Japanese and English (Tokuda, 2002), Slovene (Vesnicer, 2004) and Portuguese (Baross, 2005).

The Croatian speech synthesis system (Martinčić-Ipšić and Ipšić, 2006b) was trained on selected male speaker speech, as presented in the third column of the Table 4. In the synthesis part 1111 utterances of selected male speaker from the VEPRAD radio were used for speech synthesis training and 41 for testing. For 3840 words phonetic dictionary contains accented words and phonetic transcriptions. The speech synthesis system, in contrary to the speech recognition system, differentiates between accented and non accented vowels. Accented vowels are marked by a : including the occurrence of r as a vowel.

5.1. Speech feature vector

The speech signals where windowed using a 25 ms Blackman window and 5 ms frame shift. The feature vector consists of spectral and excitation (pitch) parameters. The spectral feature vector consists of 25 mel-cepstral coefficients including the zeroth coefficient and its delta and acceleration coefficients. The pitch feature vector consists of logF0 and its dynamic parameters (delta and acceleration).

The HMMs were embedded-trained on the features vectors consisting of spectrum, pitch and their dynamic features simultaneously in a unified framework of multi-space probability distribution HMMs and multi-dimensional Gaussian distributions. Since the observation sequence of fundamental frequency is composed of one dimensional continuous function for voiced voices and a constant for the unvoiced speech segments, multispace probability distribution is used (Tokuda et al., 2000). The HMM state output feature vector consists of spectrum and excitation in a multispace probability distribution part as shown in Figure 3. State duration densities for the speech feature vectors generation are estimated by probabilities obtained in the last iteration of embedded reestimation.

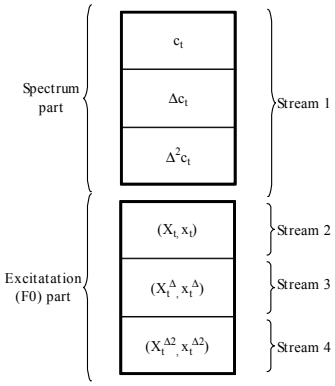


Figure 3. The HMM state output feature vector.

5.2. Speech signal generation

The speech synthesis part used prepared context-dependent HMMs, and state duration and pitch trees for generating the sequence of feature vectors for the test text. Since the last step in the training procedure was HMM parameters generation for unseen triphones, according to their classification in the phonetic decision trees, the unseen triphones can be synthesized as well.

According to the phoneme sequence in text labels the context-dependent HMMs were concatenated. State durations of the sentence are determined by maximizing the likelihood of state duration densities. According to the obtained state the sequence of mel-cepstral coefficients and F0 values including voiced/unvoiced decisions are determined by maximizing the output probability of HMM. State duration densities were modeled by multivariate Gaussian distribution. The dimensionality of state duration density is equal to the number of states of corresponding HMM. Finally the speech is synthesized from generated mel-cepstral feature vectors and pitch values using the MLSA filter (Tokuda et al. 1995).

5.3. Speech synthesis results

The text-to-speech test included 41 Croatian sentences. The text labels were transformed into triphone format. For each sentence the speech in raw format, pitch and duration were generated. Figure 4 presents the result of generated speech for the sentences:

"*Vjetar u unutrašnjosti većinom slab, na Jadranu umjerena i jaka bura. <uzdah> Najviša dnevna temperatura od minus jedan do plus tri stupnja na Jadranu od deset do petnaest.*"

From the top the pitch, spectrogram and raw signal are shown.

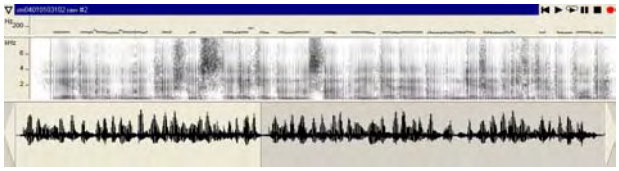


Figure 4. Pitch and spectrogram of generated speech signal for utterance sm04010103102.

Figure 5 shows the pitch and spectrogram of the corresponding part of original signal.

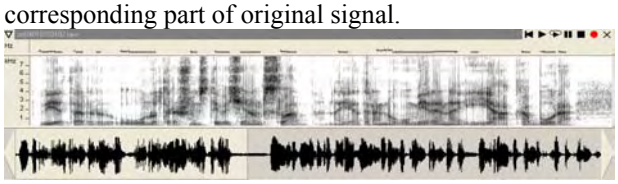


Figure 5. Pitch and spectrogram of original signal for utterance sm04010103102.

This approach where HMMs are used as generative model for speech production is very effective in rapid development of the TTS system for new language and for a new domain of interest within the same language. Although the quality of generated speech is "vocoded" buzzy speech it can be understood.

Since this speech synthesis will be incorporated in Croatian weather information spoken dialog system some improvement in the TTS quality should be considered.

In order to improve the context-dependent phone models used for synthesis more Croatian speech material

for selected speaker will be recorded and annotated. Further quality improvements will be done by manually inspecting the time boundaries of automatically segmented phones, since the overall automatically segmented phones correctness is 78.62%. And finally intelligibility and naturalness of synthetic speech can be improved also by using different Croatian speaker's speech in the speech synthesis system acoustic model training (Latore et al., 2006).

6. Conclusion

In the paper we described the context-dependent acoustic modelling of Croatian speech in the speech recognition and speech synthesis systems. The same Croatian speech corpora and Croatian phonetic rule were used for context-dependent hidden Markov models based speech recognition and speech synthesis. Presented speech recognition system for radio and telephone data and HMM based speech synthesis are planed for use in the Croatian weather information spoken dialog system.

Speech recognition experiments using context-independent and context-dependent acoustic models were prepared for "clean" radio and for noisy telephone speech. The fact that recognition accuracy for telephone speech is in expected range are very promising for further actions in development of the dialog system.

Since the telephone access to the spoken dialog system is planed, further improvements in speech synthesis quality must be considered. When the quality of the speech synthesis is satisfactory further work on evaluation of intelligibility, naturalness and functionality of synthetic speech will be done. The human experts and users will evaluate the system. The rate for intelligibility, overall quality, naturalness and functionality will be collected.

In spoken dialog system development the actions toward linguistic and semantic analysis are in progress. The Wizard-of–OZZ experiments for collecting the possible dialog scenarios are planned as the first stage of the dialog manager development.

This work showed a common approach for speech recognition and speech synthesis context-dependent acoustical modeling. Main advantage of the used approach is in fact that can be efficiently and rapidly ported to the other domains of interest under condition that adequate Croatian speech and language corpora is included.

7. References

Bakran, J., Horga, D.(1996). SAMPA for Croatian. *Govor*. XIII, Vol.1-2, p. 99-104.

Barras C, Geoffrois E, Wu Z and Liberman M. (2000). Transcriber: use of a tool for assisting speech corpora production. Speech Communication special issue on Speech Annotation and Corpus Tools. Vol. 33, No. 1-2,

Barros, M. J., et al. (2005). HMM-Based European Portuguese TTS System*, INTERSPEECH '05*, Lisbon, Portugal, p.p. 2581-2584.

Department of Computer Science, Nagoya Institute of Technology, HTS HMM Based Speech Synthesis System 1.0. http://hts.ics.nitech.ac.jp/, Japan, 2004. [09.2005.]

Fukada, T., Tokuda, K., Kobayashi, T., and Imai, S. (1992). An adaptive algorithm for melcepstral analysis of speech*, Proc. of ICASSP,* vol.1, pp.137–140, 1992.

Latorre, J., Iwano, K., Furui, S. New approach to pologlot synthesis: how to speak any language with anyone's voice. XXXX.2006.

Martinčić-Ipšić, S.; Matešić, M.; Ipšić, I. (2004). Croatian Speech Corpora. Govor: časopis za fonetiku. XXI (2); 135-150. (in Croatian).

Martinčić-Ipšić, S., Ipšić, I. (2004). Recognition of Croatian Broadcast Speech. *XXVII. MIPRO 2004*, Opatija, Vol. CTS + CIS , p. 111-114.

Martinčić-Ipšić, S., Ipšić, I. (2006a). Croatian Telephone Speech Recognition. *XXIX. MIPRO 2006,* Opatija, Vol. CTS + CIS, p. 182-186.

Martinčić-Ipšić, S., Ipšić, I. (2006b). Croatian HMM Based Speech Synthesis. 28th International Conference on Information Technology Interfaces, ITI 2006, Cavtat, Croatia. 2006.

Odell, J. The Use of Context in Large Vocabulary Speech Recognition, PhD Thesis, Queen's College, University of Cambridge, Cambridge, 1995.

Sutton S, et. al. (1998). Universal Speech Tools: The CSLU Toolkit. Proc. of the International Conference on Spoken Language Processing 1998 (ICSLP98), vol. 7, p. 3221-3224.

Tokuda, K. et al. (1995).An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. *EUROSPEECH 95*, 1:p. 757-760.,

Tokuda, K., Zen, H., Black, (2002) A. An HMM-Based Speech Synthesis System Applied to English. IEEE *TTS Workshop 2002*. Santa Monica. California, USA.

Tokuda, K. et al. (2000). Speech Parameter Generation Algorithm for HMM-Based Speech Synthesis. HMM, *Proc. ICASSP*., Vol. 3. p. 1314-1318.

Vesnicer, B., Mihelič, F. (2004). Sinteza slovenskega govora z uporabo prikritih Markovovih modelov. *Elektrotehniški vestnik*., vol. 71, no. 4, str. 223-228.

Ward, W., (1989). Modelling Non-Verbal Sounds for Speech Recognition, Proc Speech and Natural Language Workshop, Cape Cod, Morgan Kauffman, pp.311-318.

WaveSurfer, ver. 1.7.5., Centre for Speech Technology (CTT),KTH, Stocholm, Sweden, 2004. http://www.speech.kth.se/wavesurfer/

Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T. (1999) Simoultaneous Modelling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis. Eurospeech 99, Budapest, pp. 2347-2350.

Young S, et. al. (2002). The HTK Book (for HTK Version 3.2). Cambridge University Engineering Department, Cambridge, Great Britain.

Young, S., Odell, J., Woodland, P.C. (1994). Tree-Based State Tying for High Accuracy Acoustic Modelling, ARPA *HLT Workshop*, Plainsboro, NJ, Morgan Kaufaman Publishers, p307-312.

Žibert, J., Martinčić-Ipšić, S., Melita, H., Ipšić, I., Mihelič, F.. (2003). Development of a Bilingual Spoken Dialog System for Weather Information Retrieval. *EUROSPEECH '03*. Geneva, Switzerland. Vol. 1, p. 1917-1920.

A Review of AlfaNum Speech Technologies for Serbian, Croatian and Macedonian

Vlado Delić*, Milan Sečujski*, Darko Pekar[&], Nikša Jakovljević*, Dragiša Mišković[&]

* Faculty of Engineering, University of Novi Sad
Trg Dositeja Obradovića 6, Novi Sad, Serbia
{vdelic, secujski, jakovnik}@uns.ns.ac.yu

[&] AlfaNum – Speech Technologies
Trg Dositeja Obradovića 6, Novi Sad, Serbia
{darko.pekar, dragisa.miskovic}@alfanum.co.yu

Abstract

This paper gives a brief review of the development of systems for automatic speech recognition and text-to-speech synthesis in Serbian, Croatian and Macedonian language, at the Faculty of Engineering, University of Novi Sad, Serbia. The systems developed within this project enable two-way communication between humans and machines. These systems are predecessors to many commercial services such as voice portals and interactive voice response systems. Some original features of these systems, related to certain particularities of south-Slavic languages, will be illustrated as well. Available APIs and interfaces designed for using these software components in custom applications will also be described.

Pregled govornih tehnologij za srbščino, hrvaščino in makedonščino skupine AlfaNum

Članek poda kratek pregled razvoja sistemov samodejnega razpoznavanja govora in samodejnega tvorjenja govora iz besedil za srbski, hrvaški in makedonski jezik na Fakulteti za inženiring Univerze v Novem Sadu. Sistemi, ki so bili razvit pri tem projektu, omogočajo dvosmerno komunikacijo med človekom in računalnikom. Tovrstni sistemi so predhodniki mnogih komercialnih storitev, kot so glasovni portali in interaktivni govorni odzivniki. Ponazorjene bodo tudi nekatere izvirne značilnosti sistemov, ki so povezane s posebnostmi južnoslovanskih jezikov. Ob tem bodo opisani tudi razpoložljivi programski vmesniki (API) in vmesniki, ki smo jih razvili za uporabo programskih komponent pri uporabniških aplikacijah.

1. Introduction

Since speech is the most natural means of communication between humans, people have been trying to develop system that would enable them to extend this interface to communication with machines as well. Using automatic speech recognition (ASR) and text-to-speech synthesis (TTS), humans can talk to devices in their midst, such as household appliances, industry machines, cars, toys or remote computers, even via telephone. One of the important applications of these technologies is also providing independence in computer access to the physically impaired (particularly the visually impaired).

However, speech technologies are language dependent and in some regions they can hardly ever be imported from abroad as most other technologies. They have to be developed for each language separately, as is the case with languages such as Serbian, Croatian and Macedonian, having in mind all their peculiarities. It cannot be expected that some of the world's biggest companies dealing with speech technologies would decide to develop all resources needed for high-quality ASR and TTS for such a small and closed market in the near future.

A group at the Faculty of Engineering, University of Novi Sad, Serbia, called *AlfaNum* has been dedicated to the development of these technologies for ten years, and their results include phoneme-based automatic speech recognition over the telephone line with accuracy varying between 95% and 99% depending on vocabulary size and sound quality, as well as the first text-to-speech system in Serbian, Croatian and Macedonian taking into account linguistic information and thus greatly enhancing intelligibility and naturalness of synthesized speech. These systems are the basis of a number of applications, such as audio libraries and a speech-enabled web site for the visually impaired, as well as many commercial applications including interactive voice response systems and call centres.

2. AlfaNum Speech Synthesizer

During the development of the first high-quality TTS for Serbian language, this team has encountered many problems linked to bridging the gap between plain text and synthesized speech with all its typical features such as intelligibility and naturalness. There is no explicit information in a plain text concerning phone durations, pitch contours nor energy variations. These factors also depend on the meaning of the sentence, emotions and speaker characteristics, which further aggravates the task of attaining high naturalness of synthesized speech (Dutoit, 1997). While Serbian and Croatian are tonal languages, having high-low pitch patterns permanently associated with words, Macedonian is a pitch-accented language with antepenultimate stress on most words, excluding clitics, words of foreign origin as well as some other word groups. Nevertheless, a uniform dictionary-based strategy for lexical stress assignment has been successfully employed in all three cases.

The TTS engine has two main functions: text analysis and synthesis of the speech signal. Text analysis includes text processing such as expanding abbreviations, as well as resolution of morphological and syntactic ambiguities based on a comprehensive accentuation dictionary as well as rule-based syntax analysis. A separate dictionary and syntax analysis techniques were required for each language. Lexical stress assignment, as one of the most important factors influencing the shape of the pitch contour, is performed using a rule-based algorithm

described in (Sečujski, Delić, 2006). This approach has proved to produce reasonably correct stress pattern, with word error rate 2,8% for Serbian language. No equivalent tests have been carried out for either Croatian or Macedonian so far. The synthesized speech in all three languages is highly intelligible and reasonably natural-sounding, much more than any other attempts at speech synthesis in Serbian, Croatian and Macedonian so far. An objective test showing the improvement in TTS quality introduced by accent-based prosody is described in (Sečujski et al., 2002).

For the development of a multilingual TTS, separate speech databases are generally required for each language, although the Serbian database is at the moment used for Macedonian. This leads to a quite insignificant decrease in speech quality, due to the fundamental similarity between phonetic inventories of these two languages. The Macedonian speech database is expected to be recorded soon. If phonetic inventories are similar enough, as is the case for Serbian, Croatian and Macedonian language, it is appropriate accentuation and appropriate pitch contours that will make synthesized speech sound naturally, almost regardless of the original language of the database.

As to speech signal synthesis, the concatenative approach has been selected as the most promising. The AlfaNum R&D team has recorded a large speech database and labeled it using visual software tools specially designed for that purpose. By keeping score of every phone in the database and its relevant characteristics, use of phones in less than appropriate contexts was avoided, which further contributed to overall synthesized speech quality. This synthesizer is not diphone-based as almost all other speech synthesizers developed for related languages are. The TTS engine can use larger speech segments from the database, according to both phonetic and prosodic requirements, and select them at runtime in order to produce the most intelligible and natural-sounding utterance for a given plain text (Beutnagel et al., 1999).

The most significant application of this system so far is *anReader*, a speech synthesizer for the visually impaired that, combined with software known as *screen-readers*, offers them complete independence in computer access. The number of the visually impaired computer users in Serbia has increased significantly since *anReader* was presented for the first time, and its popularity in Croatia and FYR Macedonia is also growing.

3. AlfaNum System for Automatic Speech Recognition

The goal of ASR is to recognize spoken words in a speech signal, independently of the speaker, the input device, or the environment. A recognized sequence of words W_{ASR} for a given acoustic observation sequence X and all expected word sequences W is usually estimated using Bayes rule:

$$W_{\mathrm{ASR}} = \arg_W \max P(W|X) = \arg_W \max P(W) \cdot P(X|W)$$

where $P(W)$ is the *language model* estimated using *n*-gram statistics and $P(X|W)$ is the *acoustic model* represented by a Hidden Markov Model (HMM), trained using maximum likelihood estimation. HMM encodes the acoustic realisation of speech and its temporal behaviour, while prior probabilities for word sequences $P(W)$ lead to a choice of the word sequence hypothesis with the maximum posterior probability given the models and observed acoustic data. The best word sequence W_{ASR} is computed using a pattern recogniser based on a standard Viterbi decoder. A conventional approach to front-end signal processing of 30 *ms* frames, every 10 *ms*, results in a feature vector X that captures primarily spectral features of the speech signal estimated as cepstrum and energy, along with their first- and second-order time derivatives. A finite vocabulary defines the set of words (sequences of phone units) and phrases that can be recognised by the speech recogniser. The size of the recognition vocabulary plays a key role in determining the accuracy of a system, typically measured in Word Error Rate (WER), including insertion, deletion, and substitution errors.

R&D for Serbian, Croatian and Macedonian ASR has been concentrated on four aspects that define the quality of a speech recognition technique (Gilbert et al., 2005):

- *Accuracy* – WER is less than 5% for small and medium-sized vocabulary continuous ASR; it is achieved by developing acoustic models trained with 40 hours of speech database; good results for large vocabulary continuous ASR in these languages are expected when a more complex language model and more comprehensive post-processing are implemented.
- *Robustness* – channel distortions are compensated by CMS (Cepstral Mean Subtraction), background noise spectrum is subtracted and speaker variations are treated by gender separation and speaker adaptation based on VTN (Vocal Tract Normalization).
- *Efficiency* – long work on software code optimization has resulted in fast decoder and small memory footprint. The ASR engine consumes 2% or more of CPU time on a Pentium IV PC, depending on vocabulary size.
- *Operational performance* – The ASR engine gives a useful confidence scoring and implements barge-in capability, improving operational performance. On the other hand, features such as rejection of out-of-vocabulary speech have not yet been enabled.

Due to the complexity of the problem, a system for isolated word recognition in Serbian language was developed initially. It was later upgraded into a system for connected word recognition. Eventually a system for continuous speech recognition (CASR) was developed, based on recognition of phonemes in particular contexts. An elementary HMM model is a triphone model, representing a phoneme in a particular left and right context. In case there are too few instances of a triphone in the database, model-tying procedures are performed (Pekar, 2002). The advantage of phoneme-based approach is that users can define an arbitrary set of words (vocabulary) for each recognition at initialization time. The system takes into account lexical stress (particularly vowel length), assigning greater significance to stressed vowels at recognition time.

This system can be used for speech recognition in all three aforementioned languages because it is phoneme-based and because of the similarity of phonetic inventories of these three languages. No significant drop in performance for languages other than Serbian has been observed, but actual experiments will be carried out as soon as adequate ASR speech databases in Croatian and Macedonian are available.

Even state-of-the-art ASR systems cannot be successful enough if they are based on acoustic features only. In

order to achieve natural dialogs in speech applications, AlfaNum ASR has to apply some post-processing such as Spoken Language Understanding (SLU), as well as a lot of experience in both machine learning and design of front-end technology. The goal of SLU is to extract the meaning of recognized speech in order to identify a user's request. Dialog Manager (DM) evaluates the SLU output in context of the call flow specifications, which results in dynamic generation of the next dialog turn. The DM may apply a range of strategies to control dialog flow according to different application tasks. To provide a successful dialog progress, intelligent speech applications have to handle problematic situations caused by system failures or absence of concise or accurate information in a speech utterance. Post-processing makes it viable to adopt natural language dialog applications without having to achieve perfect recognition accuracy and without dictating what a user should say.

4. API for AlfaNum ASR and TTS

So far we have elaborated some general features of recognizer and synthesizer (ASR and TTS engine) and their capabilities. However, in order to be able to integrate ASR and TTS engine into a specific application, it is necessary to implement appropriate interfaces. Depending on the application and programming language in which it was designed, one of the forms of the application programming interface (API) is chosen.

For that reason, several versions of interfaces to the software component have been implemented, and a potential application designer can decide which one to use. Among other interfaces, standard MS SAPI4 and MS SAPI5 interface (speech APIs proposed by Microsoft) have been implemented. The full compatibility of ASR and TTS engines with MS SAPI4 and MS SAPI5 means that any application can access ASR and TTS engines via SAPI functions. Other interfaces implemented include a custom C++ library, socket communications and COM interface.

4.1. API for AlfaNum ASR

As mentioned above, AlfaNum ASR works with small and medium vocabularies. In order to make the system recognize specific words, a grammar must be defined. This is accomplished using regular expressions, which will be explained later. It is clearly possible to define several grammars and to decide which one to use for recognition at each moment. A specific way how to do this depends on the interface used. The input of the recognizer always consists of the speech signal and the name of the grammar. The output consists of two arrays. The first one is the array of recognized words (strings), such as ['DAJTE', 'MI', 'LOKAL', 'TRI', 'PET', 'DVA']. The second one is the array of numeric values, each of them defining the reliability of recognition of the corresponding word, i.e.: [93.1, 73.2, 90.0, 86.7, 91.2, 93.5]. Reliability values lie in the range 0-100. The exact format of these arrays depends on the interface applied.

Grammars are defined using Backus-Naur form which will be explained through an example.

Let us consider a grammar designed for recognition of a telephone extension number. In this grammar a user can, but does not have to, say the word "lokal" (extension), and after that a sequence of digits is expected. Before and after any speech activity some noise can occur, and the entire spoken sequence is optional (i.e. the user can stay silent).

```
digit = NULA | JEDAN | DVA | TRI | CHETIRI
| PET | SHEST | SEDAM | OSAM | DEVET;
lokal = LOKAL;
gr = <gar>;
main = [$gr] [[$lokal] <$digit>] [$gr];
```

Several elements can be observed:

variables – digit, lokal, gr, main. Variable "main" is the only reserved word and denotes the main sequence, i.e. what is to be recognized. For that reason it is defined at the end. Other variables can be referenced in any of the following definitions, which can be accomplished by using the prefix "$".

- **mark "|" –** denotes a choice. The recognizer will choose one of all given words.
- **angle brackets "<>" –** surrounded sequence can occur once or several times.
- **square brackets "[]" –** denotes an optional sequence. The recognizer can pass through this word (or the whole rule), or skip it.
- **reserved word "gar"** – denotes noise model used for noise itself as well as some sounds that could be produced by the speaker but are not qualified as orthographic words.

4.1.1. AlfaNum ASR Server

All interfaces which will be mentioned here rely on ASR server, which contains the recognition engine. All client applications communicate with the server over the IP protocol. This enables remote access to the server and distribution of multiple ASR servers (implemented on several computers) in case there is a need for large number of simultaneous recognitions. All interfaces first connect to the server, then send a command and wait for a response. Commands are usually recognition requests, although the protocol supports many other commands as well.

An application designer can always use low level IP communication with server. However, we have developed a higher level library in C++, enabling fully functional communication with the server using a very small number of functions, with no need for low level protocol details. The library contains the following functions and fields:

- **void AddHost (const string &host_name, float host_load) –** adds computer **host_name** to the list of computers having ASR server capabilities. **host_load** represents the load coefficient of a particular server. Servers will be used according to these coefficients.
- **void Connect () –** connects to ASR server.
- **void Disconnect () –** disconnects from ASR server.
- **void RecognizeFromFile (const string &grammar, const string &file_name, float timeout_s)** – Recognizes the utterance recorded in the file **file_name** in accordance with **grammar**. The result is stored in fields **results** and **reliabilities**.
- **void AddGrammarAsync (const string &name, const string &grammar_file_name, const string &transcriptor, const string &pronunciation, const string &postprocessor, int timeout_s = -1)**

– Starts a process of adding a new grammar named **name**, defined in the file **file_name**. Since this operation can be time-consuming, the process is asynchronous in order to avoid the client being blocked during initialization.

- **vector <string> results –** string vector with results of the last recognition.
- **vector <float> reliabilities –** float vector with values denoting reliability of the latest recognition.

The COM interface is similar to the C++ interface, but it contains a set of methods and properties which can be used in virtually any programming language. Most descriptions of functions and properties are the same as the ones given for the C++ interface:

- **AddHost(host_name As String, host_load As Single)**
- **Connect ()**
- **Disconnect ()**
- **RecognizeFromFile(grammar As String, file_name As String , timeout_s As Double)**
- **GetRecoResults As RecoResults –** returns an object of **RecoResults** type, with a string array containing spoken words.
- **GetResultsReliabilities As ResultsReliabilities** – returns an object of type **ResultsReliabilities** which contains reliabilities of recognized words.

Microsoft SAPI5 interface also relies on the ASR server, and on the client side contains methods and properties defined in MS SAPI5 standard. All details on this standard can be found at www.microsoft.com/speech.

4.2. API for AlfaNum TTS

The basic functionality of the AlfaNum TTS system is to transform an input text into speech. Textual input is usually an unicode string, while the output is usually a wav file. Beside this basic functionality, it is possible to set other input parameters such as pitch, speed, speaker, accentuation manner, etc.

It is possible to obtain an audio stream from the synthesizer, which means that the client side can get parts of audio signal even before the whole text has been processed. In this way, delays are much shorter than in case the signal is available only after the whole synthesis has been completed.

All interfaces which will be mentioned rely on TTS server. It contains text-to-speech engine, and client applications communicate with the server over the IP protocol, in a way rather similar to the ASR server.

An efficient C++ library for communication with the server was developed, similar to the one developed for communication with ASR server. Examples of TTS-specific functions and fields are given below:

- **void Synth (const string &file_name, const wstring &text, int timeout_s)** – synthesizes sentence given in unicode string text and puts it in the file file_name.
- **float speed** – defines the speed of the synthesized speech.
- **float pitch** – defines the pitch of the synthesized speech.
- **string speaker** – defines the speaker to be used for synthesis.
- **bool read_punctuation** – defines if punctuation marks should be read out.
- **bool read_abbreviations** – defines if abbreviations should be spelled.
- **bool letter_by_letter** – defines if the entire text should be spelled.
- **bool prosody_override –** defines if the user is allowed to specify his/her own preferred accentuation of a specific word.
- **bool character_substitution –** defines if, in the absence of letters with diacritics (č, ć, š, ž...), there should be an attempt to replace them with appropriate letters with diacritics.

Examples of methods and properties of the COM TTS interface as well as the Microsoft SAPI4 and SAPI5 interface are the same as for the ASR server, with addition of TTS-specific methods and properties related to the functions and fields given above.

5. Conclusion

Two speech technologies developed for the Serbian, Croatian and Macedonian language have enabled two-way communication between humans and machines in these languages. This communication can be direct or remote (e.g. via telephone), which introduces the possibility of building speech-enabled intelligent systems. This is a step of a human-to-machine interface in the regions where these languages are spoken from touch-tone prompts toward multimedia and multimodal interface. While both these technologies are still under development, they are already implemented in many commercial applications, and are also in wide use as aid for people with visual disabilities in Serbia, Croatia, Bosnia-Herzegovina and FYR Macedonia. Owing to a number of interfaces developed as well as appropriate reference manuals, these software components can be used by third party programmers who want to develop their own speech-enabled applications.

6. References

Beutnagel, M., Mohri, M., Riley, M., 1999. Rapid Unit Selection from a Large Speech Corpus for Concatenative Speech Synthesis. In *Proc. of EUROSPEECH'99*, Budapest, 607-610.

Dutoit, T., 1997. *An Introduction to Text-to-Speech Synthesis.* Dordrecht: Kluwer academic publishers, 149-152.

Gilbert, M.; Wilpon, J. G.; Stern, B.; Di Fabbrizio, G., 2005. Intelligent Virtual Agents for Contact Center Automation, *IEEE Signal Processing Magazine*, Vol. 22, 5:32-41.

Pekar, D.; Obradović, R.; Delić, V., 2002. AlfaNumCASR – a system for continuous speech recognition. In *Proc. of 3rd Conference DOGS*, Bečej, Serbia, 49-56.

Sečujski, M., Obradović, R., Pekar, D., Jovanov, Lj., and Delić, V., 2002. AlfaNum System for Speech Synthesis in Serbian Language. In *Proc. of 5th Conf. Text, Speech and Dialogue*, Brno, 8-16.

Sečujski, M., Delić V., 2006. A software tool for semi-automatic part-of-speech tagging and sentence accentuation in Serbian language. In *Proc. of IS-LTC*, Ljubljana.

www.microsoft.com/speech

www.microsoft.com/downloads, Speech SDK 5.1

Slovak TTS - From Rule Based To Unit Selection

Rusko Milan, Trnka Marian and Darjaa Sakhia

Institute of Informatics of the Slovak Academy of Sciences, Bratislava, Slovakia
{milan.rusko, trnka, utrrsach}@savba.sk

Abstract

Different types of synthesizers that have been developed at the Department of Speech Synthesis and Analysis of the Institute of Informatics of the Slovak Academy of Sciences from 1990 up till now are described.

The rule based synthesizer - Kempelen O.1 developed in 1990 was a memory-footprint optimized system with PC-speaker and parallel port outputs using a unique method of signal compression preserving transients and synthesizing stable parts of phonemes by repetition of the same microsegment (pitch period).

The Kempelen 1.x engine was based on concatenation of pre-recorded diphones with signal post-processing for intonation and rhythmical contours implementation. Some interesting features were added for commercial applications, such as multilinguality, singing voice synthesis and illustrative sounds (acousticons). These synthesizers have been used in several professional applications, such as voice operated information systems, interactive voice response systems and teleservices of the Slovak telephone operators as well as in special tools for visually handicapped people.

The Kempelen 2.x synthesizer is based on unit-selection. The speech synthesis database design is described in the paper as well as the experience resulting from the design and testing of Kempelen 2.0. A new approach in Kempelen 2.1 uses pre-selection of element-candidates based on a phonological analysis of the orthoepic transcription of the text. It is aimed at elimination of eventual concatenation in problematic areas of speech signal and on selection of candidate elements according to the phonetical context. Acoustical aspects are taken into account in the second run of the selection process.

Pretvarjanje besedila v govor za slovaščino – od sintetizatorjev na osnovi pravil do sintetizatorjev, ki temeljijo na izbiri govornih enot

Opisane so različne vrste sintetizatorjev, ki so jih od leta 1990 do danes razvili na Oddelku za sintezo in analizo govora Inštituta za informatiko Slovaške akademije znanosti. Sintetizator na osnovi pravil, Kempelen 0.1, ki so ga razvili 1990, je bil sistem za osebni računalnik, optimiziran na čim manjšo pomnilniško zasedbo, uporabljal je zvočnik osebnega računalnika in izhode na paralelnih izhodnih vratih, pri tem je uporabljal lastno metodo stiskanja signala, tako da je ohranjal prehodne in sintetiziral stabilne dele fonemov s ponavljanjem istega mikrosegmenta (osnovne periode). Kempelen 1.x je temeljil na združevanju vnaprej posnetih difonov z naknadno obdelavo signala za oblikovanje intonacije in izvedbo ritmičnih vzorcev. Za komercialne aplikacije so dodali nekatere zanimive lastnosti, kot npr. večjezičnost, sintezo pojočega glasu in ilustrativnih zvokov (akustičnih ikon). Te sintetizatorje so uporabili v več različnih (opomba: raje črtamo ali pa pustimo izraz profesionalnih) aplikacijah, kot so npr. govorno voden informacijski sistem, interaktivni govorni odzivniki in telekomunikacijske storitve za slovaške telefonske operaterje, kot tudi v posebnih orodjih za slepe in slabovidne. Sintetizator Kempelen 2.x temelji na izbiri osnovnih govornih enot. V prispevku sta predstavljeni zasnova podatkovne baze za sintezo govora in izkušnja načrtovanja in testiranja Kempelena 2.0. Nov pristop Kempelena 2.1 uporablja vnaprejšnje izbiranje kandidatov za osnovne govorne enote na podlagi fonološke analize pravorečne transkripcije besedila. Cilj tega je preprečevanje združevanja osnovnih govornih enot na problematičnih delih govornega signala in izbor kandidatov za osnovne govorne enote na podlagi fonetičnega konteksta. V drugem delu procesa izbire se upoštevajo še akustični vidiki.

1. Introduction

Early experiments with speech synthesis in Slovakia were made on RPP 16 mainframe computer developed in eighties at the Institute of Technical Cybernetics (which was later renamed to Institute of Informatics). The fist hardware formant synthesizer was built at the same institute in 1987. It was developed using a PC (IBM compatible PC PRAVEC, made in Bulgaria). The quality of the synthesized speech was not bad, but the hardware synthesizer board was expensive and the operation was not user-friendly. In that time a Department of speech analysis and synthesis was founded and led by a distinguished Slovak phonetician, Prof. Ábel Kráľ. His phonetic knowledge in combination with programming capabilities and signal processing skills of engineers from this department gave a birth to the first generation of software synthesizers in Slovakia.

2. Rule based synthesizer – intelligible, but robotic

The development of the first generation TTS - Kempelen 0.1 speech synthesizer – started in 1989. The early PCs, equipped with two floppy disks and no hard disk, had 512 kB of operational memory, so the engine of our phoneme-based concatenative synthesizer was designed to require only 80 kB of operational memory for code and additional 80 kB was needed for the data. To keep the memory footprint as small as possible a unique method of signal compression was used. The stable parts of voiced phonemes were synthesized by repetition of the same microsegment (pitch period). Some unvoiced consonants and transients were kept uncompressed.

The synthesis process of the voiced phonemes merely consisted of concatenating the phoneme transients (the beginning and the ending segment) and the looped central “steady” part of the phoneme.

The full set of the Slovak transients was categorized into several classes and only one transient was chosen to represent the entire class in the database of elements. The transient also served as a joint with the neighboring phoneme. For better naturalness some of the problematic phonemes were stored as a whole.

With no soundcard available the PC-speaker and the parallel port equipped with simple resistor D/A converter were used as outputs.

In spite of the fact, that the repetition of central microsegment made the sound of the synthesizer considerably robotic, it was well understandable. According to the opinion of the users from the Slovak Union of Blind, who tested it, the generated speech quality was much better than that of the Czech speech synthesizer of the EUREKA computer that they had in use.

Kempelen O.1 was monotonous in its basic configuration. However the fact, that it had its samples stored in a form of pitch-periods (microsegments), made it relatively easy to manipulate the melodic and rhythmical contours. Simple deletion of the last samples of the period was used to shorten the period and zero padding was used to lengthen it. The first experiments with singing voice synthesis were accomplished. As the pitch shifts were realized mainly on vowels and voiced consonants with high degree of periodicity, the voice sounded a bit like sung by two people – one singing vowels and second one singing consonants.

3. Diphone synthesizer – versatile, but still a bit unnatural

The research on diphone synthesis and development of a concatenative speech synthesizer started in Slovakia approximately in the year 1994. It brought a synthetic speech of better comprehensibility and higher naturalness, together with elaborated interface that made this generation of synthesizers suitable even for professional telecommunication applications.

3.1. The diphone concatenative synthesizer

The second generation of Slovak TTS - Kempelen 1.x - was based on concatenation of small elements of a pre-recorded speech signal, mainly diphones. An algorithm similar to Time Domain Pitch Synchronous Overlap and Add (TD-PSOLA) (Hamon, Moulines, Charpentier, 1989) was used for concatenation. We have also developed a Linear prediction (LP) and Residual Excited Linear prediction (RELP) (Macchi et al., 1993) versions of the synthesizer.

Listening tests were carried out in order to evaluate three versions of our diphone text-to-speech system. The three synthesizers were based on linear predictive (LP) synthesis, residuum excited LP synthesis (RELP) and time-domain pitch synchronous overlap-and-add synthesis (PSOLA), respectively. All of them were in two versions – female and male voice. We tested the overall quality of voices and our aim was to reach MOS values for these synthetic speech signals. (Cernak 2005)

All the ten decades of Test words for Slovak audiometry (Bargár et. al. 1986) were synthesized by all the synthesizers and played from the PC to the test participant via Sennheiser HMD 25 closed-system headphones in laboratory conditions.

The subjects taking part in listening tests belonged to the normal PC using population, with the provisos that:

a) they have not been directly involved in the work connected with assessment of the performance of speech synthesizers, or in related work;

b) they have not participated in any subjective test whatever for at least the previous six months and not in any listening-opinion test for at least one year;

c) they have never heard the same word lists before.

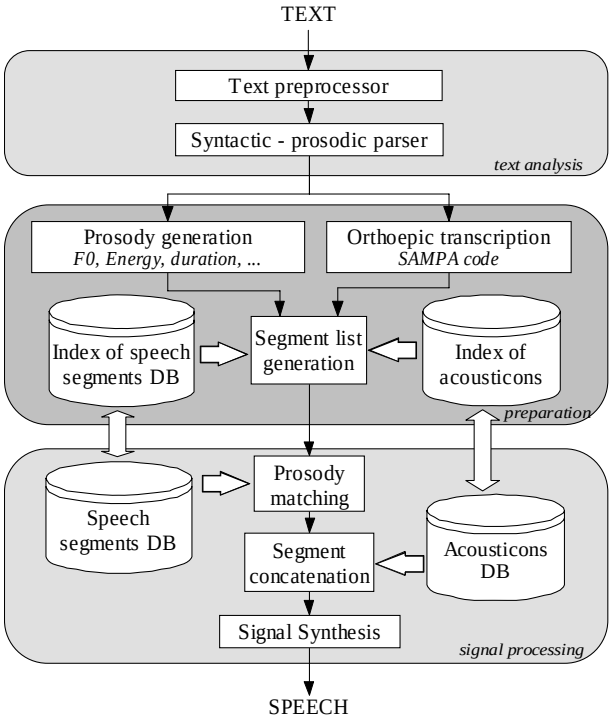


Figure 1. Schematic diagram of the Kempelen 1.x diphone concatenative synthesizer

8 subjects (2 males and 6 females) aged from 16 to 73 took part in the experiment.

Synthesis method	Mean opinion score (MOS)
LP-female	1.60
RELP-female	3.05
PSOLA-female	3.23
LP-male	1.53
RELP-male	2.84
PSOLA-male	3.34

Table 1: Subjective evaluation of the Kempelen 1.x synthesizers

The availability of synthesizers with more voices and different quality of speech made it possible to carry out experiments on voice quality measurement and to develop a method for objective synthetic speech measurement using PESQ measure (Cernak, Rusko, 2005).

The acceptable quality of the synthesized speech made it possible to use the synthesizer generated words as first draft templates for DTW word recognizer. These experiments were promising and the recognizer with male voice templates was able to recognize a majority of the words of its 1000 words vocabulary even when it was tested by female speaker. Anyway it of course could not compete with new technology - recognizers based on statistical models.

3.1.1. Rule based pronunciation

The pronunciation was controlled by the block of orthographical-to-orthoepical conversion (grapheme to phoneme) based on a sophisticated set of rules supplemented by a pronunciation vocabulary and a list of exceptions (Darjaa, Franěková, Rusko, 1994). This elaborated unit has proven to be more reliable than our similar data driven system based on CART trees (Cernak et al., 2003).

3.1.2. New voices

It generally takes several weeks to build a new professional quality diphone voice. To get an idea how the new voice will sound, we have designed a program that interactively records a set of nonsense words uttered by the tested speaker and immediately after a 10 minutes long recording session it automatically finds the needed diphones in the signal and creates a database for a draft new voice. The timbre of the new draft voice is the same as it will be in the definitive version of the new voice, only the appearance of concatenation discontinuities and rhythmical mistakes is much higher. So one can decide if the speaker is suitable for building a new voice.

Final recordings of the new voice were then realized in a studio under a permanent supervision of linguistic expert.

The diphone database building proceeded in two steps. A draft automatic phonetic alignment using a combined DTW/Rule-based recognizer. This had later to be checked and refined manually by a human expert to achieve a fluent and relatively natural voice.

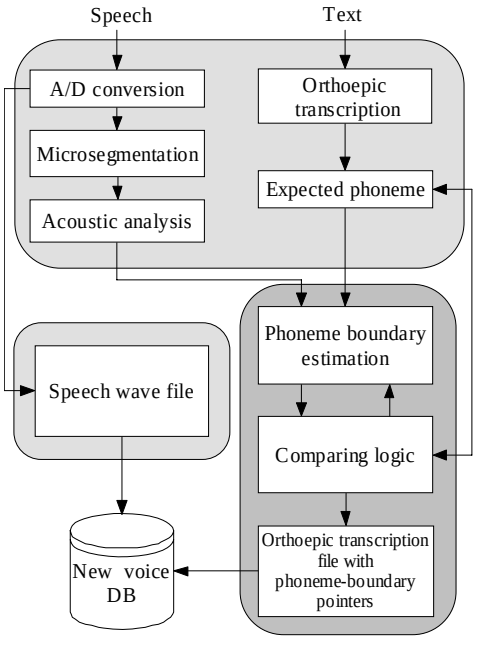


Figure 2. Diagram of automatic phonetic alignment used for generating a new voice

3.1.3. Multilinguality

One of the main trends in telecommunication services, information systems and computer speech interfaces is multilinguality. Developing English, German or French version of our synthesizer would probably not give any sense, as there are lots of high-quality synthesizers available for these languages, developed by reputable companies which have incomparably better financial and personal capacities for there development. We have however decided to make a Hungarian version of our synthesizer to broaden the rank of possible users by the Hungarian speaking fellow-citizens. We have used our synthesis engine and with a help of the students of Hungarian nationality and the employees of the Department of Hungarian language of the Comenius University in Bratislava we have defined rules and designed a database of synthesis elements as well as a block of pronunciation for Hungarian. As a result we have a synthesizer in two languages.

We think it would be interesting to have the source speech for synthesis recorded by one bilingual speaker in both languages, which would help to avoid timbre differences in the two languages.

3.1.4. Singing voice synthesis

Singing voice synthesizers have in general different purpose than speech synthesizers and they work on different principals. They are designed to provide enjoyable singing voice where intelligibility is not of highest importance. They may employ principals of music samplers, advanced methods of pitch processing and time stretching algorithms etc.

We decided to use the simplest and cheapest way – that is „to force the speech synthesizer to sing“. The basic formula for tempered tuning is:

$$f_{n+k} = kqf_n \qquad (1)$$

where q = 1,05946309, which is the twelfth root of two and k is the number of half-tones between f_n and f_{n+k}

It is obvious, that a direct mathematical representation of a note code does not give an acceptable pitch contour for the singing voice synthesis. Our analyses of the pitch contours of recorded songs had shown that at least several phenomena should be taken into account, such as rise and fall times of the tones, and vibrato, its depth, envelope and frequency. The introduction of these changes improved the synthesized singing significantly (Darjaa, Trnka, Rusko, 1999).

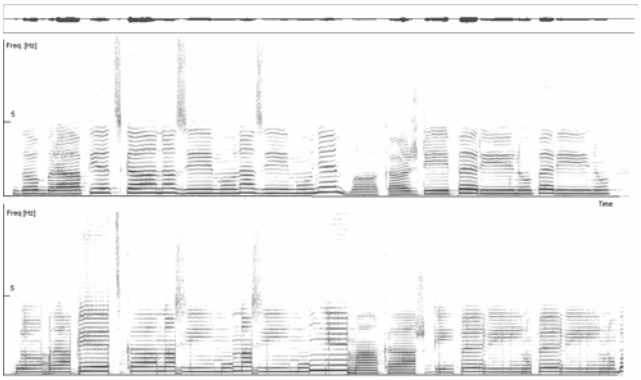


Figure 3. The spectrograms of a part of a song sung by a female singer (upper) and of a signal synthesized from the elements of speech of the same woman (lower). Only a simple rule for tone onset pitch changes and no vibrato is applied in this version.

In spite of imperfection of the solution we consider our singing voice feature to be fully functional and suitable to enrich the SMS to Voice service as a new entertaining feature.

4. The unit selection synthesizer – problems in unlimited domain

Advances in speech synthesis in the world led us to the decision that the third generation of the Kempelen synthesizers should be based on unit-selection from a speech database.

4.1. Speech database for synthesis

At the beginning of this project there was no annotated speech database available for unit-selection speech synthesis in Slovak. So it was an inevitable condition to design a professional-quality, one speaker, general purpose database for research, experiments and application building in unit-selection speech synthesis which would be extendible, but also down-scalable (e.g. for limited domain experiments).

4.1.1. Recording the database

The database consists of recordings of one male, non-professional speaker, experienced in speech processing. The recording took place in an unechoic room of a professional studio specialized to speech recording (radio commercials, dubbing etc.). The sessions lasted typically about two hours and were realized in irregular intervals from one week to one month. A Neumann U 87 cardioid condenser microphone with Focusrite Trackmaster pre-amplifier and a hard disk recording system equipped with AARK 20/20+ sound board was used in the sessions. The sampling frequency was 44.1 kHz and resolution was 16 bit.

4.1.2. Choice of the source text material, database content

In spite of the fact, that we plan to extend the speech database in future, the initial elementary structure of the database had to be clearly defined first. Our ambition was to design a general-purpose database being at the same time suitable for experiments in limited domain synthesis. The other contradictive requirement for the database was not to be too big, but to be representative enough from the phonetical, phonological, and other points of view. Therefore we decided to design the database as a combination of several more or less independent parts:

4.1.3. Phonetically rich sentences

- Set of words covering all Slovak diphones
- Sentences covering intonation phenomena
- Spontaneous speech record (General topic story, Application oriented story)
- Set of prompted application-oriented phrases and embedded application commands
- Numerals

4.1.4. Database annotation

The annotation consists of several levels of information. In the case of need new levels of annotation can be added. Annotation techniques and choice of annotation levels belong to the subjects of research to be accomplished on this database, therefore the mentioned annotation levels serve only as a reference, as an initial annotation to start with.

4.1.5. Annotation levels

There are two text annotation levels:

- orthographic text
- orthoepic text (in SAMPA)
- Signal annotation levels are the following:
- microsegmental information – pointers to single pitch periods
- phoneme boundaries information
- diphone boundaries information
- syllable boundaries information
- whole words and phrases information

Suprasegmental annotation level consists of:

- melody contour information - smoothed f0 value, intonation phrase boundaries
- accent information

4.1.6. Automatic annotation

Automatic annotation consists of orthographical to orthoepical conversion, microsegmentation – pitch marking and segmentation to diphones

4.1.7. Orthographical to orthoepical conversion

The text in the orthographic form was transcribed to the orthoepic form by the block of pronunciation developed for earlier versions of our synthesizers [4]. The orthoepic text generated automatically was then manually checked and corrected by an expert with a degree in linguistics.

4.1.8. Microsegmentation – pitch marking

Microsegmentation – pitch period boundaries detection was accomplished by a rule based routine, which works well on a clean studio-quality full range speech signal (Darjaa, Rusko, 1997).

With a help of an orthoepically transcribed text and a rule-based phoneme recognizer based on pitch synchronous analysis (Darjaa, Kráľ, Rusko, 1993) correspondence of every microsegment to a particular phoneme was recognized and its boundaries were estimated.

4.1.9. Segmentation to diphones

One of the levels of annotation splits the speech signal into parts (elements - mainly diphones) which inventory matches to the set of the elements used in our diphone synthesizer Kempelen 1.4. The boundaries of the elements which the signal was generated from are known for the synthesized signal. Making use of the fact that we have a synthesizer with the voice of the same speaker, we applied a DTW algorithm in one of our phonetic alignment algorithms to automatically label element (diphone) boundaries in the recorded signal by mapping the labels from the synthetic speech to the recorded one.

5. Experimental synthesizer

We used Baum-Welch training to build complete ASR acoustic models from a part of the database. The HMM recognizer with these models was then used to label data. The whole labeling was realized in FestVox framework,

where Carnegie Mellon University's SphinxTrain and Sphinx speech recognition system are used (Huang et. al., 1993).

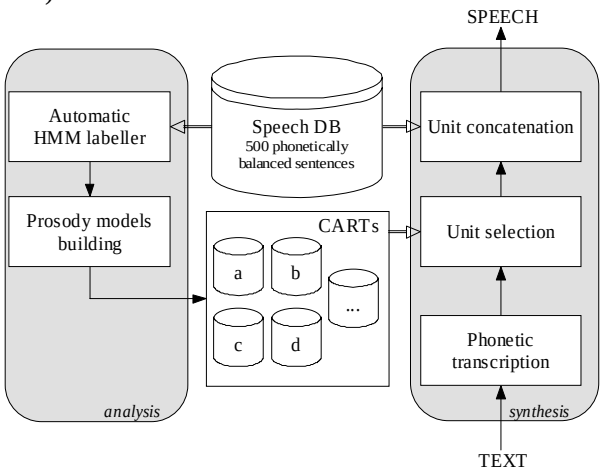


Figure 4. Schematic diagram of the experimental synthesizer

We used 500 phonetically balanced utterances for training and labeling. An experimental Slovak corpus-based speech synthesizer was built using the labeled data. The approach uses a technique of automatic clustering of similar units to build a CART for each phoneme with questions from NLP block in its nodes. We used duration model with average durations of phones. Then we applied simple multiplicative factors for the phones in phrase final and phrase initial positions. (Fig. 4)

5.1. Recent version - Kempelen 2.1 synthesizer

Recent version of the synthesizer, Kempelen 2.1. does not use any third-party components. It fully relies on our own annotation method, pre-selection of elements and unit selection algorithm. (Fig. 5)

5.1.1. Unit preselection

Generally speaking a syllable was taken for a basic element in our synthesizer. However the phoneme boundaries are annotated in the database and in the case of need smaller units than syllables are chosen for synthesis as well.

The aim of our preselection is to avoid using improper joint points just by employing phonological knowledge. The phonetical context is checked carefully. If an element of the required context is not available, the database is searched for an element with a context belonging to the same phonetic category as the desired one. Different phonetic contexts are allowable only in the worst case, as they usually cause audible disfluences in timbre at the concatenation point. This approach is similar to that of Taylor and Black (1999).

In some of the triphones an extremely strong coarticulation at the central phoneme can be expected and it is very unlikely for the automatic annotation program to find the boundaries of such a phoneme correctly. Therefore we have defined a list of "forbidden joint point triphones" which can be split only if no other solution is possible. Typical representatives are VCV combinations with sonorants l, L, r, j, or fricative h (in SAMPA) as their central phoneme. The preselection takes into account also a syllable position – word initial, word center, word final and sentence final.

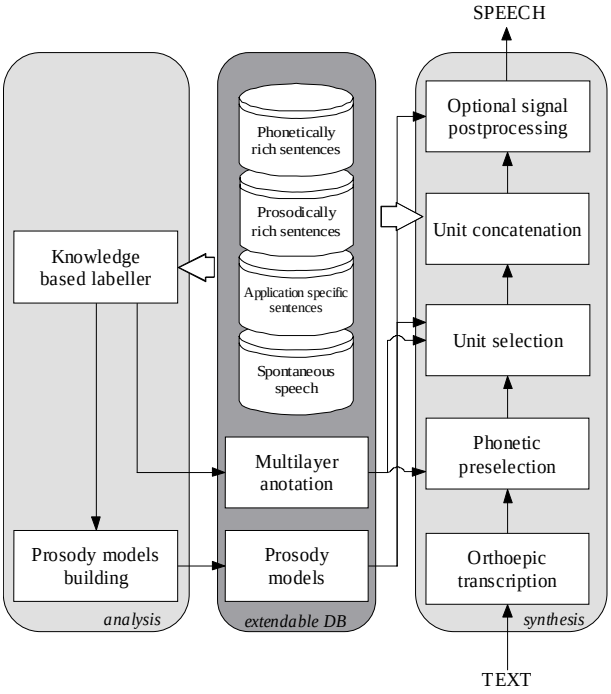


Figure 5. Schematic diagram of the Kempelen 2.1 synthesizer

6. Discussion

The rule "the more data the better" does not seem to hold in unit selection speech synthesis in all cases. Especially in the initial phases of the research it is necessary to have a smaller speech database recorded with relatively stable vocal effort and flat prosody. Only after the unit selection algorithm is well tuned, it is advisable to enrich the database with speech data covering intonation phenomena of expressive speech and rhythmically and phonetically problematic spontaneous speech. In this point it becomes even more important to have a reliable annotation method. We think that automatic HMM phoneme labelers should always be checked for typical errors and supplemented by knowledge based corrective algorithms. In our approach to phonetic alignment we strongly rely on secure identification of anchor points in the speech signal which are of three main categories:

- Vowels (high energy, periodicity, sharp formant structure)
- Fricatives (noisy spectrum with high frequency components)
- Plosives (pause plus burst structure)
- Phoneme boundary finding is always based on iteration.

In our recent approach to synthesis we apply a phonological unit preselection which reduces the universality and openness of the classical unit selection approach, but it excludes the most significant concatenative problems in advance, before the calculation of concatenative and unit costs has even started.

7. Conclusion

The paper presents a brief survey of research and development in speech synthesis in Slovakia.

The first generation of Kempelen speech synthesizers has proven a capability of software synthesizers to produce intelligible speech under very low computational expense. The know-how, from the first generation represented appropriate initial conditions for building a second generation with better performance, intelligibility, stability and versatility.

These reliable synthesizers have been integrated into voice services of all the three Slovak telephone operators. They are also in use by some members of the Slovak Union of Blind and Visually Impaired for screen reading and some of special tools for visually impaired are delivered with Kempelen 1.6. synthesizer too.

The Kempelen 2.1 synthesizer is the most recent of our products at the moment, which is still under development. We find it to be a promising successor of the popular Kempelen 1.x synthesizers and we hope, that the companies in Slovakia will discover the advantages of the unit-selection synthesis approach soon.

8. Acknowledgements

This work has been funded by the Ministry of Education of the Slovak Republic, task number 2003 SP 20 028 01 03 and Slovak Agency for Science, VEGA, grant No. 2/2087/22.

9. References

Hamon, C., Moulines, E., Charpentier, F., 1989. A diphone synthesis system based on time-domain prosodic manipulations of speech. *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, 238.

Macchi, M., Altom, M.J., Kahn, D., Singhal, S., Spiegel, M.F. 1993. Intelligibility as a Function of Speech Coding Method for Template-Based Speech Synthesis. *Proceedings of Eurospeech 93*, Berlin, 893-896

Cernak, M., Rusko, M. 2005. An evaluation of a synthetic speech using the PESQ measure. *Proceedings of Forum Acusticum 2005*, Budapest

Darjaa, S., Franěková, L., Rusko, M. 1994. Conversion and Synthesis of the Slovak Speech. (in Slovak), *Jazykovedný časopis, 45,(Bratislava) 1994, No. 1*, 31-34.

Cernak, M., Rusko M., Trnka, M., Darjaa, S., 2003. Data-Driven Versus Knowledge-Based Approaches to Orthoepic Transcription in Slovak. *Proceedings of ICETA 2003, Kosice (Slovak Republic)*, 95-97.

Darjaa, S., Trnka, M., Rusko, M., 1999. The Application of Text-to-Speech System in Slovak to Singing Voice Synthesis. *Proceedings of SPECOM'99, Moscow*, 162-165.

Dutoit, T., 1997. An Introduction to Text-To-Speech Synthesis, *volume 3 of Text, Speech and Language Technology*. Kluwer Academic Publishers, The Netherlands.

Huang, X.D., et. al., 1993. The SPHINX-II Speech Recognition System: An Overview, *Computer Speech and Language (1993)*, 137-148.

Lee, K.F., 1989. *Automatic Speech Recognition: The Development of the SPHINX SYSTEM*, Kluwer Academic Publishers, Boston, 1989.

Rusko, M., 2000. Definition of corpus, scripts and standards for Fixed Networks – Slovak. *SpeechDat-E deliverable-ED1.2.3, http://www.fee.vutbr.cz/SPEECHDAT-E*

Kráľ, A. 1996. *Pravidlá slovenskej výslovnosti*, Slovenské pedagogické nakladateľstvo Bratislava, 163 – 200

Rusko, M., Darjaa S., Trnka M., Petriska M., 2000. SpeechDat-E, the First Slovak professional-quality telephone speech database, *In: Research Advances in Cybernetics.*, ELFA Publishing House, Košice, 187-211

Moulines, E., Charpentier, F., 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones, *Speech Communication 9*, 453-467.

Darjaa, S., Kráľ, Á., Rusko, M., 1993. Phoneme-oriented Approach to Speech Recognition in Slovak. *in D. Mehnert (Hrsg.): Elektronische Sprachsignalverarbeitung in der Rehabilitationstechnik, Berlin*, 83-89,

Darjaa, S., Rusko, M. 1997. Automatic Labelling of Speech Signal for Slovak Speech Database Building, *Proceedings of the 31st Int. Conf. ACOUSTICS-High Tatras 97,* 124-125.

Rusko, M., Darjaa, S., Trnka, M., 2001. Databases for speech recognition and synthesis in Slovak. *In: Proceedings of the conference SLOVKO - Slovenčina a čeština v počítačovom spracovaní*, Bratislava , VEDA, 88-97

Rusko, M., Darjaa, S., Trnka,M., 2002. Automatic design of the elements database for speech synthesizer in Slovak, (in Slovak), *in Proceedings of the conference Noise and vibrations in practice - Kočovce 2002*, Slovak Technical University Bratislava, 75-78

Black, A.W. and P. Taylor 1997. Automatically clustering similar units for unit selection in speech synthesis. *in Proc. of the European Conference on Speech Communication and Technology*. Rhodos, Greece.

Bargár Z., Kollár A., 1986. *Praktická audiometria*. Osveta, 159-160 [In Slovak].

Taylor P. and Black A. W., 1999. Speech Synthesis by Phonological Structure Matching, *in Proceedings of Eurospeech'99*, 623-626.

A Flemish Voice for the Nextens Text-To-Speech System

Wesley Mattheyses, Lukas Latacz, Yuk On Kong and Werner Verhelst

Vrije Universiteit Brussel
Dept. ETRO-DSSP
Pleinlaan 2, B-1050 Brussels, Belgium

{wmatthey, llatacz, ykong, wverhels}@etro.vub.ac.be

Abstract

Nextens is an open source text-to-speech system that can be used to convert Dutch text into speech as spoken in The Netherlands. Flemish is the variant of Dutch as spoken in Flanders. These two languages have the same written form, but they sound clearly different. This paper describes how we transformed the Nextens system into a Flemish speaking application. In order to achieve this goal, a high-quality acoustic diphone synthesizer has been developed as the new back-end. This synthesizer is based on a very simple and effective overlap-add technique that can be used to simultaneously solve the problem of waveform concatenation and to perform the necessary prosodic modifications. In addition, some post-lex rules have been adapted to the Flemish speaking style. The resulting Flemish diphone synthesis system has a quality that is comparable to that of a commercial diphone synthesis system.

Flamski govor za sistem Nextens za pretvarjanje besedila v govor

Nextens je odprtokodni sistem za pretvarjanje besedila v govor. Uporabljamo ga lahko za pretvarjanje besedila v nizozemšcini v govor, kakršnega govorijo na Nizozemskem. Flamšcina je razlicica nizozemšcine, ki jo govorijo na Flamskem. Podobno kot britanska in ameriška angleščina imata ta dva jezika isto pisno obliko, zvenita pa razlicno. V prispevku je opisano, kako smo spremenili sistem Nextens v flamsko govoreco aplikacijo. Za dosego tega cilja je bil razvit zelo kakovosten akusticen difonski sintetizator kot novi zaledni del. Sintetizator temelji na zelo preprosti in ucinkoviti tehniki prekrivanja in dodajanja, ki jo lahko uporabljamo za socasno reševanje problema združevanja valovnih oblik in za izvajanje zahtevanih prozodicnih prilagoditev. Poleg tega so bila nekatera postleksikalna pravila prilagojena flamskemu nacinu govora. Kakovost dobljenega flamskega difonskega sistema za sintezo je primerljiva s kakovostjo komercialnih difonskih sistemov za sintezo.

1. Introduction

Dutch is the common name for the main language in both The Netherlands and in Flanders, the northern part of Belgium. The grammatical rules and spelling are the same for both regions, but the pronunciation of the Dutch language differs clearly between them (comparable to the difference between British and American English)[1]. We will refer to the language as spoken in The Netherlands as 'Northern Dutch', and to the language as spoken in Flanders as the Flemish language.

A text-to-speech system (TTS system) is an application that converts a written text into a speech signal. The development of such systems has been a topic of research for many years but unfortunately only few open-source TTS projects, usable for research, are available. Regrettably, no open-source TTS system has yet been developed for Flemish. Nextens (Nextens, 2006) is an open-source TTS system for Northern Dutch, which we used as a starting point for developing our own TTS system for the Flemish language.

This paper starts with a short introduction to TTS systems and Nextens in section 2. There, we also introduce our strategy for changing the Nextens voice to a Flemish sounding voice. Since the difference in pronunciation is caused by discrepancies in phonetics, we decided to record a new diphone database. To assure compatibility with the database and to permit future enhancements we also designed a new back-end. This state of the art acoustic diphone synthesizer will be described in section 3., which will be the main part of this paper. The quality of the result is evaluated in section 4. and finally the conclusions are drawn in section 5.

[1]Note that in contrast to the variants of the English language, there is only one correct spelling for both variants of Dutch

2. Strategy for adapting Nextens to Flemish

In this section we give a short summary of the different modules and functionalities found in a TTS system like Nextens, after which we explain which modules need to be replaced in order to obtain a Flemish version of the Nextens system.

2.1. A text-to-speech system

Figure 1 illustrates the different modules found in most common TTS systems. Such a system can be split-up in two main parts. The text input is first handled by a linguistic front-end, which starts by *normalizing* the input text and converting it into a set of known *tokens* (e.g., abbreviations and numbers written down with numerals are converted to plain words). Then, a *part-of-speech tagging* will take place, which delivers information about the position of the nouns, the verbs, etc. in the sentence. Hereafter, a *syntactic parsing* provides data about the inter-word relationships. All this information is used to create an accurate *prosody model* for the speech. In this part of the TTS synthesis, this model will mostly be expressed by means of 'tone-and-break indices' (ToBi), which indicate the variations in speech rate and pitch going from word to word or from syllable to syllable.

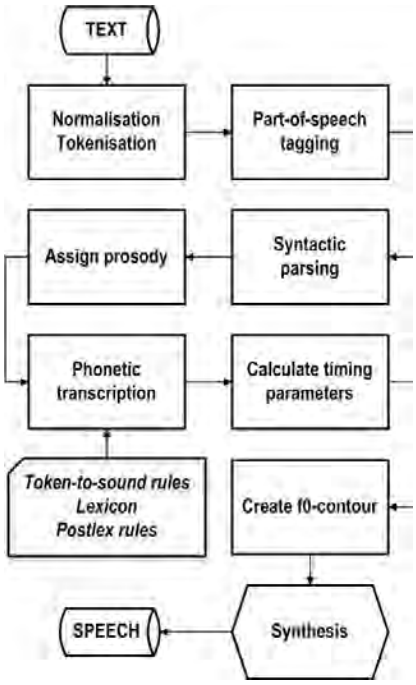


Figure 1: Overview of the different steps involved in the conversion of plain text into speech.

Obviously, the plain sentences have to be transcribed into their *phonetic transcriptions* before the corresponding speech can be generated. The system accomplishes this by using a phonetic *lexicon*, in combination with *token-to-sound* rules which are applied when the target word is missing in the lexicon (e.g., for names and foreign expressions). In the last stage, the *phonetic transcriptions* are modified by the *postlex-rules*, which are based on phone co-pronunciation properties, and the prosodic information is converted in more useful data that can be applied directly to the *synthesis* module of the TTS system. The *timing parameters* describe the optimal durations of the phonemes in the speech and the *f0-contour* indicates the most favorable fundamental frequency (or pitch) for the output signal at all time instances.

In the acoustic part of the TTS process, the prosodic and phonetic information is used as input for the synthesis module, which constructs the physical speech signal. In a concatenative system, the target speech is assembled by joining multiple sound records from a database in an appropriate way. Typical examples are the diphone systems which employ a database consisting of diphones, speech signals containing two consecutive phonemes (i.e., a diphone actually starts in a characteristic point of the first phoneme (e.g., in its most stable part) and ends in a characteristic point of the second phoneme). Nowadays, higher quality TTS synthesis can be accomplished with so-called 'non-uniform unit selection systems' which use much larger speech databases and can better account for contextual variations, for example. Nevertheless, diphone systems are still important for their possible application in small mobile devices.

2.2. Modification of the original system

The Nextens project provides the Dutch equivalent for all the front-end modules shown in Figure 1 and is equipped with an MBROLA synthesizer (MBROLA, 2006). In order to achieve a Flemish sounding output, a modification of the synthesizer will be necessary. In any case, a new database with speech recordings needs to be created and provided to the synthesis module. It is obvious that by registering a new diphone set by means of Flemish carrier words, a big step toward a Flemish TTS output is realized. Furthermore, one can opt to implement a new synthesizer in order to assure a maximum compatibility between the dataset and the used synthesis algorithms, which will undoubtedly have a positive influence on the output quality.

Since the Dutch grammatical rules and spelling apply for Northern Dutch as well as for Flemish, only minimal revision of the front-end will be necessary. The phonetic transcription of the input text is accomplished by using a language-dependent lexicon. After the lexicon-lookup, the phonetic transcriptions are handled by the 'postlex-rules'. These rules are also used to modify the transcriptions in order to attain the intended regional accent, hence a modification of some of these postlex-rules will be obligatory to facilitate the adjustment of the Nextens voice from Northern Dutch to Flemish. Note that by 'accent', we understand in this context the differences in pronunciation of the official Dutch language, which are comparable to the differences between spoken British English and American English. For the conversion of a TTS system to a real dialectic voice (where non-standard sounds, words and expressions are used), much more effort would be required (for example, changing the grapheme to phoneme conversion module would be needed, the lexicon would need major revisions, etc.).

A few examples of Northern Dutch postlex-rules that needed to be discarded for the Flemish language are shown in table 1.

Postlex-rule	Example
G-r $\rightarrow x - r$	begrip
G-l $\rightarrow x - l$	begluren
N-G $\rightarrow N - x$	mongool
l-G $\rightarrow l - x$	algebra
b-d $\rightarrow p - d$	abdij
b-n $\rightarrow p - n$	abnormaal
Z-w $\rightarrow S - w$	bourgeois

Table 1: Northern Dutch postlex-rules that were discarded to adapt the system to the Flemish speaking style.

3. A high-quality acoustic diphone synthesizer

A new diphone synthesizer has been implemented in order to achieve maximum compatibility with the new Flemish diphone database that we recorded (around 1800 recordings were included in the dataset). This section will explain how this synthesizer constructs an output speech signal by the concatenation of elements from the diphone database, followed by the assignment of the prosody defined by the

parameters delivered by the linguistic front-end. We believe that the strength of our synthesizer mainly resides in its high quality and low complexity that was achieved by using an overlap-add technique for both the segment concatenation and the prosodic modification, in accordance with the source filter interpretation of pitch synchronized overlap-add (PSOLA) (Moulines and Charpentier, 1990), as introduced in (Verhelst, 1991). As will be explained further in this section, according to this interpretation, the synthesizer can make use of the series of pitch markers that is defined for each diphone signal to fulfill the concatenation.

3.1. Pitchmarking

The pitch markers are a set of sample indices which indicate the local pitch periods in a speech signal. This implies that the distance between two consecutive pitch markers is in fact a local pitch measure for the signal. The prosody in our synthesizer will be assigned by using the pitch-synchronous overlap-add technique (PSOLA), which needs a series of good pitch markers to attain quality output. The quality of the synthesizer will thus greatly depend on the correctness of these markers. Therefore we designed an efficient and robust algorithm to accomplish this pitch epoch detection, as described in (Mattheyses et al., 2006) and summarized below.

Our algorithm is an extension of a previous technique (Lin and Jang, 2004) that is based on a dynamic programming approach applied to voiced segments. In our approach, we start by performing a frame-based voiced/unvoiced decision on the speech signal. This is necessary because unvoiced frames, due to their noise-like behavior, have to be treated differently than the voiced speech segments, which contain a clear periodicity.

In the voiced regions of the signal, the markers are systematically placed at signal peaks or at signal troughs. Note that the choice for peaks or troughs has to be the same for all the diphone signals in the database. This peak/trough decision is made corresponding to whichever minimizes an error measure between the local pitch values (obtained as the difference between consecutive pitch markers) and the global pitch contour obtained from an AMDF pitch detection algorithm.

If we assume that the markers are to be positioned at signal peaks, the algorithm continues by searching for the maximum sample present in the frame. By using the AMDF pitch measure in combination with this highest sample index, several search-regions can be defined. These correspond to those parts of the signal in which the other pitch markers in this frame can be assumed to be located. Next, a set of candidate markers is selected for each search region, based on two properties. The candidates have to represent a sample value which is as high as possible, while we also require that successive candidates are separated by a given minimum distance. This results in a set of possible markers per search region, each representing a different signal peak. In a final stage, each candidate is given a score based on its height and another score is associated with the transition between two candidates. The algorithm selects one candidate in each region as final pitch marker by maximizing the total score, summed over all selected candidates and transitions. For more details, the reader is referred to (Mattheyses et al., 2006). Figure 2 shows the first steps of the voiced pitch marking process. As illustrated in the last panel of the figure, selecting the highest candidate as the final marker would not always result in a consistent set of pitch markers, which explains the introduction of the transition scores.

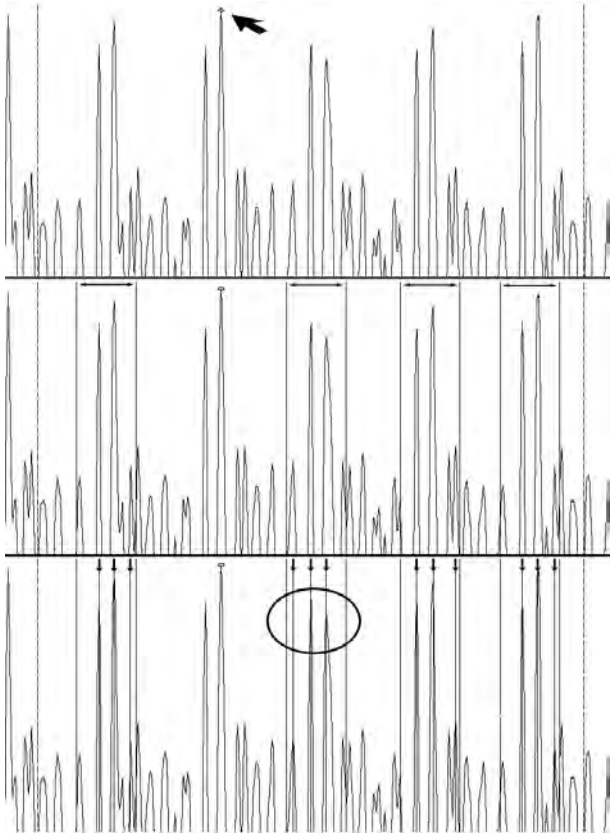


Figure 2: Finding pitch mark candidates in voiced speech. The largest peak of the signal is detected (upper panel) and search regions are defined lying at multiples of a global pitch measure from the largest peak (middle panel). The three largest peaks in each search region are the pitch mark candidates.

In contrast to many other pitch marking algorithms, which simply place the pitch marks in unvoiced signal regions at regular time intervals, we opted to position the unvoiced markers in a well-thought manner. We found this to be necessary as a frame could be classified as unvoiced, but still contain part of a voiced signal, include voiced/unvoiced transitions, etc. Such signals contain some residual periodicity, which should be indicated by the final set of pitch markers. Therefore, in the unvoiced regions of the speech signal, we determine the pitch markers by positioning them according to the neighboring voiced pitch markers. Figure 3 shows a detail of a speech signal and its trough-based pitch markers. It illustrates the correctness of the pitch marks for voiced parts of the signal as well as for unvoiced parts and for voiced/unvoiced transitions. As reported in (Mattheyses et al., 2006), the pitch marking

algorithm has been tested and evaluated and it provides a series of consistent markers, which are suitable for application in a TTS system. Note that, although not really necessary, one could also choose to hand-correct the pitch marks since pitch marking of a TTS database is done offline.

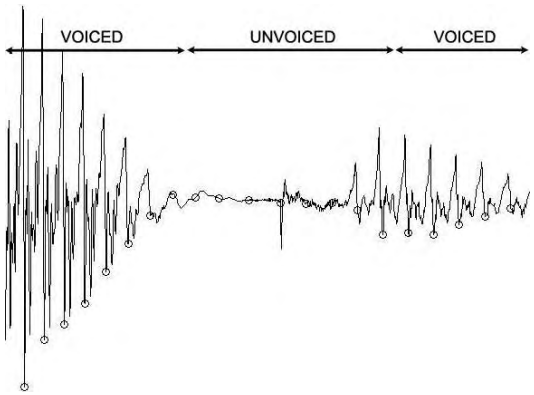


Figure 3: Open circles illustrate the result of automatic trough-based pitch marking in a transitory speech segment.

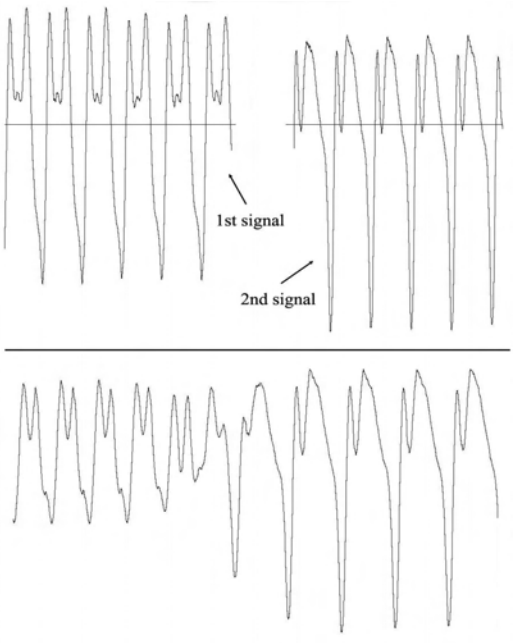


Figure 4: Illustration of typical problems that occur with straightforward non optimized diphone concatenation.

3.2. Segment concatenation

The acoustic synthesizer has to concatenate the diphone recordings in order to construct the desired speech signal. To achieve a fluent and intelligible speech, the diphones have to be concatenated in an appropriate way. Figure 4 illustrates the concatenation of the Dutch diphones 'b-o' and 'o-m'. It shows that there is a quite large dissimilarity between the two signals, although both represent the same phoneme 'o'. In an ideal diphone database, every phoneme would have been recorded at a same speech rate and having a same pitch value. It is obvious that in reality only an approximation of this ideal can be achieved. Therefore, the concatenation technique has to smooth the transition between the two signals over a certain time, otherwise these transitions will appear to be too abrupt and the concatenated speech would not sound very fluent, but chopped.

While joining two voiced speech signals, we have to make sure that the resultant signal shows a continuous periodicity. A shortcoming of many concatenation techniques is that they introduce anomalous pitch periods at the diphone transitions, which has a harmful influence on the output quality. In the second panel of figure 4, such a bad concatenation result of the 'o' phoneme is shown. As one can see, the transition between the two consecutive 'o' signals is not smooth and at the transition point abnormal pitch periods appeared.

Since we have a series of pitch markers for each diphone signal, we can exploit the benefits of the use of this pitch-information in joining the diphone segments. A diphone database contains information about the most optimal cut-points in the diphone recordings (this is referred to as the 'segmentation' of the database). This information is derived offline and obviously can not take into account exactly which two segments will be concatenated. By choosing a pitch marker as the diphone cut-point, we can assure that the periodicity of the speech signal will not be disrupted by the concatenation procedure. The most straightforward technique would be to select that pitch marker that is closest to the segmentation-cut-point as the 'cut-marker'. In order to further enhance the concatenation quality, we designed an optimization method which selects the best cut-marker according to the minimization of a MEL-scale spectral distance, as suggested in (Conkie and Isard, 1994). This technique selects for each join a pitch marker from the first and from the second diphone in such a way that the transition will occur where there is as much similarity between the two speech signals as possible.

Once the cut marks are determined, the actual concatenation problem is tackled by a pitch-synchronous window/overlap technique. First, a number of pitch periods (typically 5) is selected from the end cut-marker and from the beginning cut-marker of the first and second diphone, respectively. Then, the pitch of these two short segments is altered using the well known PSOLA technique, which will result in two signals having exactly the same pitch. The initial pitch value of these resulting signals is chosen equal to the pitch present in the original signal extracted from the first diphone. This pitch then varies smoothly along the length of the signals such that the final pitch value becomes equal to the pitch of the signal extracted from the second diphone. Finally, these two completely pitch synchronized signals are cross-faded using a hanning-function to complete the concatenation of both diphone recordings. By first assuring the pitch-synchronicity of both signals before applying the cross-fade, the introduction of irregular pitch periods is minimized and the periodicity is preserved as much as possible.

Figure 5 illustrates our concatenation method using the same diphones as in figure 4. To illustrate its robustness,

we used a first diphone recording that has a pitch value which is much higher than that of the second diphone, as one can see in the upper panel of the figure. The middle panel shows the pitch-alignment of the extracted pitch periods and the bottom panel shows the final concatenated 'o' phoneme. This last plot illustrates that in the concatenated speech signal the diphone transition is smoothed among a few pitch periods, which is necessary if a fluent output is to be obtained. In addition, the output does not suffer from irregular pitch periods.

The proposed concatenation technique delivers results of the same quality as some more complex concatenation methods found in the literature. The technique has been systematically judged against a spectral interpolation approach and it was concluded that the computationally more complex interpolation could not outperform the proposed overlap-add method. This can be explained by noting that the transition was actually realized as the result of three processes: the use of the pitch markers assures a maximum preservation of the periodicity, the pitch-synchronous overlap-add accomplishes the transition in pitch value from the first diphone to the second one, and finally the window/overlap operation creates the transition in waveform shapes between both diphones.

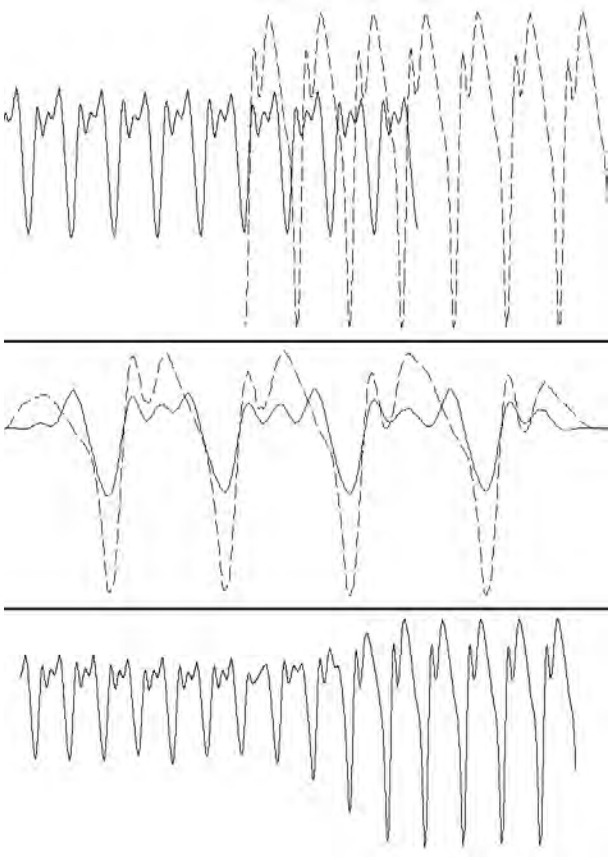


Figure 5: Pitch-synchronous concatenation. The upper panel illustrates the diphones to be concatenated, the middle panel illustrates the pitch-synchronized waveshapes, and the lower panel illustrates the result after cross-fading.

3.3. Adding prosody

At this point we need to apply the correct prosody to the concatenated signal. We opted to use the PSOLA technique to alter the timing and the pitch of the speech. During the concatenation process, the pitch markers of the synthesized speech signal can be computed from the diphone pitch markers. These will then be used as analysis-pitch markers for the PSOLA technique.

At the same time, each sample point that indicates a phoneme transition is memorized. By using these transition points the synthesizer calculates the length of each phoneme present in the concatenated signal. The Nextens front-end provides a set of timing parameters, indicating the optimal length of each phoneme in the final TTS output. Using these two sets of values, the amount of time-stretching that is necessary to provide the output speech with the correct timing properties is computed. Subsequently, the PSOLA algorithm will synthesize the output signal by using a time varying time-stretch value going from phoneme to phoneme. The synthesis-pitch markers used by the PSOLA operation determine the pitch of the final TTS output (Verhelst, 1991). Obviously, it suffices to calculate these pitch markers based on the pitch-parameters coming from the front-end (the 'f0-contour') to ensure that the imposed intonation curve is correctly assigned to the final speech signal. Note that only voiced parts of the speech require a pitch-shift. Since the PSOLA algorithm is constructing the output of a TTS system, we know at each point in time which phoneme corresponds to the current signal segment. This information can be used to decide whether a pitch-shift is desired or not.

4. Evaluation

In this section the performance of the TTS system will be discussed. First our system will be compared with the Nextens application and afterwards the overall TTS quality will be judged while possible explanations and solutions to improve the output quality will be stated. The evaluation is based on informal listening tests, conducted by people with experience in the field and by people without experience.

To achieve a Flemish TTS synthesis, our diphone synthesizer is provided with the prosodic parameters of the Nextens system. When the output of our Flemish application is judged against the original Nextens speech, we actually also compare our overlap-add synthesizer with the MBROLA synthesizer, which is resident in the Nextens system. It appears from our experiments that our synthetic voice definitely sounds as fluent as the MBROLA voice[2]. Both signals display very similar timing and pitch variations, which indicates that our acoustic synthesizer does apply the desired prosodic modifications in an accurate way. Due to the cut-point optimization, discussed in subsection 3.2., our voice is robust against small segmentation-errors of the diphone database and the pitch-synchronous concatenation technique makes it feasible for use with databases that contain inconsistent pitch levels. Further, the output of

[2]Note, however, that we could not compare our PSOLA synthesis method against the MBROLA synthesis technique using in both cases a same diphone database

our system sounds undoubtedly Flemish in contrast with to original Nextens voice which means that the main goal, the conversion of the language of the system, is achieved.

In general, the output of our Flemish TTS system is very intelligible. However, in most cases the speech possesses a sub-optimal prosody (coming from the Nextens system). The pitch variations are often to abrupt and sometimes syllable durations are to short. Especially this last imperfection can have a dreadful influence on the clarity of the output speech. We compared our TTS system with two commercial systems for Flemish, (Realspeak, 2006) and (Fluency, 2006). The first one is a system that uses a very large segment database instead of a small diphone database. As one would expect, the naturalness of its speech is much higher than with our system at the expense of a much larger footprint and computational load. These systems also achieve a higher output quality due to the presence of multiple instances of the same segment in the database. More appropriate is the comparison with the second commercial application, which is also a diphone system. The smoothness of this commercial system and the fluency of our TTS application are about the same. However, the output of the commercial system sounds more natural and is overall more intelligible than the output of our system. As mentioned before, a correct timing model is necessary to attain a highly intelligible output. The accuracy of the f0-contour has less influence on the clarity of the speech, although a precise intonation curve is needed to reach a natural sounding TTS output. The influence of a non-optimal prosodic model can be counteracted by lowering the speaking rate. However, this classic technique obviously has its limitations. To ensure enough naturalness, we suggest that the variations in the f0-contour are kept limited, since a more flat intonation will sound less disturbing than an incorrect one. It is also important to create an f0-contour with mean value around the original pitch present in the diphone recordings. This ensures that only minor pitch modifications are required from the PSOLA algorithm, which enhances the quality of the output speech. Another point of attention is the introduction of 'phrase breaks' in the speech signal. These are short pauses between two words, some of which, but not all, are determined by the punctuation. In contrast to the commercial systems, the Nextens front-end fails to predict these pauses accurately, as they are only placed according to punctuation (e.g., after a comma).

We performed some experiments in which we provided our synthesizer with a better set of prosodic parameters by manually measuring these values in the commercial TTS outputs. This resulted in speech signals of approximately the same quality as the commercial diphone system, which demonstrates the importance of the prosodic information in order to attain high-class output speech. Furthermore, we manually inserted the correct phrase breaks into the signal, which lead to an important enhancement of the clarity of the TTS output. This can be explained by noting that the extra pauses will slow down some parts of the speech and they will make it easier to distinguish between the different words in a sentence. These tests illustrated that a very good diphone TTS output is achievable by using our diphone synthesizer, provided that more optimal prosodic parameters are used in comparison to the prosody that the Nextens front-end can provide.

5. Conclusions

In this paper we discussed the conversion of a TTS system between two regional accents: Northern Dutch and Flemish. A new diphone synthesizer has been designed, which uses the PSOLA technique to impose the desired prosody on the output speech. The synthesizer also uses the PSOLA pitch markers to successfully maintain a maximum periodicity while concatenating the diphones. A cut-point optimization method proved useful to cope with small segmentation errors in the database. By combining the pitch-synchronous overlap-add technique with a simple cross-fade method, robust high quality concatenation was achieved.

The switch from Northern Dutch to Flemish was accomplished by providing a new set of diphones and a modification of some postlex-rules. Once the synthesizer produces fluent and intelligible speech, a revision of some of the linguistic modules of Nextens will be necessary in order to enhance the clarity and the naturalness of the output. The introduction of phrase breaks and the adjustment of the f0-contour can definitely contribute to achieve this goal. Our experiments have shown that high-class diphone synthesis is attainable by using our diphone synthesizer and a set of optimal prosodic parameters.

6. Acknowledgments

Parts of the research reported on in this paper where supported by the IWOIB project Link II - Voice Modification of the Brussels region, by the IWT projects SPACE (sbo/040102) and SMS4PA-II (O&O/040803), and by the research fund of the Vrije Universiteit Brussel.

7. References

A. Conkie in I. Isard. 1994. Optimal coupling of diphones. V: *Proc. SSW2 – 2nd ESCA/IEEE Workshop on Speech Synthesis.*

Fluency. 2006. http://www.fluency.nl/.

C. Y. Lin in J.S. Jang. 2004. A two phase pitch marking method for td-psola synthesis. *GETS Interbational transaction on Speech Science and Engineering*, 1(2):211–221.

Wesley Mattheyses, Werner Verhelst, in Piet Verhoeve. 2006. Robust pitch marking for prosodic modification of speech using td-psola. V: *Proc. IEEE Benelux Signal Processing Symposium, SPS-DARTS.*

MBROLA. 2006. http://tcts.fpms.ac.be/synthesis/mbrola.html.

E. Moulines in F. Charpentier. 1990. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9:453–467.

Nextens. 2006. http://nextens.uvt.nl/.

Realspeak. 2006. http://www.nuance.com/realspeak/.

W. Verhelst. 1991. On the quality of speech produced by impulse driven linear systems. V: *Proc. International Conference on Acoustics, Speech and Signal Processing*, str. 501–504.

Articulatory Manner Features Recognition with Linear and Polynomial Kernels

Jan Macek, Julie Carson-Berndsen

School of Computer Science and Informatics
University College Dublin
Belfield, Dublin, Ireland
{jan.macek, julie.berndsen}@ucd.ie

Abstract

A typical speech recognition system uses acoustic features to represent speech for its processing. Recently, articulatory features were introduced to serve the same purpose. They are motivated by linguistic knowledge and may therefore provide better or complementary representation of speech signal. We present research on recognition of such articulatory features by Support Vector Machines with three types of kernels—a linear kernel and two polynomial kernels. As input for recognizers we use standard set of Mel-frequency cepstral coefficients extended with values of formants and pitch of the speech signal. Performance is compared to recent results for the task based on other methods of machine learning. We conclude that for most of the articulatory features SVMs with a polynomial kernel give superior performance.

Razpoznavanje značilk artikulatornega načina z linearnimi in polinomskimi jedri

Tipičen sistem razpoznavanja govora uporablja pri procesiranju za predstavitev govora akustične značilke. V zadnjem času so se z istim namenom začele uporabljati tudi artikulatorne značilke. Uporabo leteh je motiviralo jezikoslovno znanje, zato lahko morda omogočajo boljšo ali komplementarno predstavitev govornega signala. V prispevku predstavljamo raziskavo o tem, kako z metodo podpornih vektorjev (MPV) razpoznavamo artikulatorne značilke s tremi vrstami jeder z linearnim jedrom in z dvema polinomskima jedroma. Kot vhodne podatke za razpoznavalnike uporabljamo standardno množico melodičnih frekvennih kepstralnih koeficientov, razširjenih z vrednostmi formantov in osnovnih period govornega signala. Kakovost izvedbe primerjamo z nedavnimi rezultati za isto nalogo na podlagi drugih metod strojnega učenja. Sklenemo z ugotovitvijo, da dajo za večino artikulatornih značilk polinomske MPV najboljše rezultate.

1. Introduction

Speech representations today are usually based on the acoustic information of the signal (Heřmanský, 1999). However, by relying only on this acoustic information, these speech representations seem to achieve only moderate success, especially, in adverse environments (noisy, out-of-task, out-of-vocabulary, etc). One of the ways to improve performance in such environments is to integrate linguistic knowledge as suggested in (Launay et al., 2002; Carson-Berndsen, 1998).

Articulatory features (AF) have been shown to improve word recognition accuracy under variable conditions of speech signal production. For example, in a multilingual environment, feature recognizers trained on data from different languages were shown to have the capability of improving the overall performance by ensemble recognizer or by crosslingual recognizer (Stüker et al., 2003). The AF representations have also been shown to perform well in noisy environment (Kirchhoff, 1999).

AF is thought to be a good compromise, offering better descriptions of the acoustic signal than phonemes yet still providing a linguistically interpretable symbolic annotation. Acoustic correlates of features have been described in the literature (Stevens, 2000; Stevens, 1980). The first detailed description of distinctive features (Jakobson et al., 1952) assumed that they had identifiable counterparts.

In this paper, Support Vector Machines (SVMs) with three types of kernels are presented for extraction articulatory features from the speech signal. The performance of the SVMs is compared among them and against referenced results of bagging that are reported as giving best results for this task among machine learning methods. We only refer to reported performance of Hidden Markov Models on this task (Kanokphara et al., 2006) where they do not provide good performance, apparently for the reasons of weaker probabilistic dependence between adjacent articulatory features in the speech signal. Our article extends the research reported in (Kanokphara et al., 2006) and (Macek et al., 2005). The SVM classifiers with variable kernels were run with the SVMLight implementation (Joachims, 1999).

Systematically, this paper is organized as follows. Section 2. explains the details of the experimental paradigm used in this paper, i.e. the corpus, the evaluation method and the feature table. Section 3. describes the general framework of support vector machines and Section 4. presents the results of experiments with SVM-based AF extraction. Finally, discussion and conclusions are presented in Section 5.

2. Experimental Setup

2.1. The Corpus

In the experiments we used the standard TIMIT corpus (Garofolo et al., 1993) consisting of 6300 sentences, 10 sentences spoken by each of 630 speakers, of which 462 are in the training set and 168 are in the testing set. There is no overlap between the training and testing sentences, except 2 SA sentences that were read by all speakers. The training set contains 4620 utterances and the testing set contains 1680 utterances. The core test set, which is the abridged version of the complete test set, consists of 192 utterances, 8 from each of 24 speakers. In this paper, the full training set with SA sentences is used as the training set while only the core test set without SA sentences is used as the test set.

Articulatory manner feature	Frequency in corpus	Phone (TIMIT transcription used)
approximant	8.12%	axr, r, w, y
closure	9.68%	bcl, dcl, gcl, kcl, pcl, tcl
flap	0.78%	dx, nx
fricative	16.47%	ch, dh, f, hh, hv, jh, s, sh, th, v, z, zh
lateral approx.	3.37%	el, l
nasal	5.72%	em, en, eng, m, n, ng, nx
stop	16.22%	b, bcl, d, dcl, g, gcl, k, kcl, p, pcl, q, t, tcl
vocalic	37.99%	aa, ae, ah, ao, aw, ax, ax-h, ay, eh, er, ey, ih, ix, iy, ow, oy, uh, uw, ux

Table 1: Assignment of articulatory manner feature classes to phonemes and their frequency in the TIMIT corpus

2.2. The Evaluation

The evaluation method used in this paper is a comparison of overall accuracy in terms of frame error rate (FER) together with recall, precision and F1-measure. FER is widely used for articulatory feature extraction evaluation (Chang et al., 2005). In our method the speech signal is represented as a sequence of numeric vectors where each vector represents speech in each time frame. Therefore, the AF extraction systems are evaluated on the frame level. Due to variable distribution of classes for each articulatory feature it is necessary to extend the performance measure of accuracy with the values of *precision*, *recall*, and *F1-measure* that are used in Tables 3, 4 and 5. The *precision* is defined as the ratio

$$\frac{\text{number of correctly classified instances of class } c}{\text{number of instances classified as class } c}$$

and the *recall* is defined as the ratio

$$\frac{\text{number of correctly classified instances of class } c}{\text{number of instances of class } c}$$

The trade-off between *precision* and *recall* is measured by the value of *F1-measure* defined as

$$\frac{2 * \mathit{precision} * \mathit{recall}}{\mathit{precision} + \mathit{recall}}.$$

All these measures analyse the performance for each class individually.

A true AF evaluation should compare between a reference (annotated at the feature level) and a hypothesized AF transcription. However, due to the cost and difficulty of corpus construction process, no feature annotated reference exists. In this paper, we directly convert reference annotations at the phone level into reference annotations at the articulatory feature level. These annotations lack some of the coarticulation information which would be typically found in references directly annotated at the articulatory feature level. However, this is the only resource available and it is widely accepted as the reference transcriptions for AF evaluation. Such transcription was done for the TIMIT corpus according to assignment of articulatory manner feature classes to phonemes presented in Table 1. Among the manner features we included both, closure and stop, where stop might be considered as a sequence of a closure and a burst. This allows us to see from performance of the respective classifiers if the simpler feature is more distinctive which is the case in our experiments.

Articulatory feature	Accuracy	Precision	Recall	F1-measure
−approximant	**93.12%**	0.947	0.98	0.963
+approximant		0.637	0.387	0.482
−closure	**94.41%**	0.957	0.982	0.970
+closure		0.763	0.567	0.651
−flap	**99.76%**	0.989	0.995	0.991
+flap		0.000	0.000	0.000
−fricative	**97.19%**	0.975	0.992	0.983
+fricative		0.956	0.865	0.908
−lateral approx.	**96.63%**	0.971	0.995	0.983
+lateral approx.		0.473	0.124	0.196
−nasal	**96.55%**	0.977	0.987	0.982
+nasal		0.636	0.504	0.562
−stop	**89.64%**	0.929	0.951	0.940
+stop		0.666	0.574	0.616
−vocalic	**89.22%**	0.907	0.907	0.906
+vocalic		0.873	0.874	0.873

Table 2: Accuracy Rates for Bagging with REP trees on TIMIT core test set for recognition of articulatory manner features

Articulatory feature	Accuracy	Precision	Recall	F1-measure
−approximant	**94.15%**	0.952	0.987	0.969
+approximant		0.569	0.251	0.348
−closure	**94.90%**	0.961	0.984	0.973
+closure		0.753	0.556	0.640
−flap	**99.76%**	0.998	1.000	0.999
+flap		0.000	0.000	0.000
−fricative	**93.84%**	0.952	0.974	0.963
+fricative		0.870	0.778	0.821
−lateral approx.	**96.77%**	0.968	1.000	0.984
+lateral approx.		0.000	0.000	0.000
−nasal	**96.30%**	0.973	0.988	0.981
+nasal		0.720	0.533	0.613
−stop	**89.86%**	0.903	0.990	0.945
+stop		0.795	0.276	0.410
−vocalic	**89.13%**	0.934	0.886	0.910
+vocalic		0.831	0.899	0.864

Table 3: Accuracy Rates for SVMs with linear kernel on TIMIT core test set for recognition of articulatory manner features

For reasons of further comparison we present in Table 2 the performance on the task of articulatory feature recognition for the method of bagging (Breiman, 1996) with reduced error pruned (REP) decision trees (Quinlan, 1987) that was reported to perform best among several machine learning techniques on the same data (Macek et al., 2005).

Articulatory feature	Accuracy	Precision	Recall	F1-measure
−approximant	**94.85%**	0.967	0.979	0.973
+approximant		0.606	0.497	0.546
−closure	**96.13%**	0.976	0.982	0.979
+closure		0.782	0.729	0.754
−flap	**99.76%**	0.998	1.000	0.999
+flap		0.000	0.000	0.000
−fricative	**95.10%**	0.958	0.984	0.970
+fricative		0.916	0.804	0.856
−lateral approx.	**97.44%**	0.977	0.997	0.987
+lateral approx.		0.772	0.294	0.426
−nasal	**97.94%**	0.985	0.993	0.989
+nasal		0.862	0.745	0.799
−stop	**92.35%**	0.949	0.965	0.957
+stop		0.726	0.642	0.682
−vocalic	**91.52%**	0.936	0.926	0.931
+vocalic		0.883	0.898	0.890

Table 4: Accuracy Rates for SVMs with polynomial kernel of order $d = 2$ on TIMIT core test set for recognition of articulatory manner features

Articulatory feature	Accuracy	Precision	Recall	F1-measure
−approximant	**94.96%**	0.969	0.977	0.973
+approximant		0.608	0.537	0.570
−closure	**96.30%**	0.977	0.982	0.980
+closure		0.789	0.745	0.766
−flap	**99.77%**	0.998	1.000	0.999
+flap		1.000	0.030	0.058
−fricative	**95.30%**	0.959	0.985	0.972
+fricative		0.922	0.810	0.862
−lateral approx.	**97.49%**	0.979	0.996	0.987
+lateral approx.		0.728	0.356	0.478
−nasal	**97.97%**	0.986	0.993	0.989
+nasal		0.856	0.757	0.803
−stop	**93.08%**	0.954	0.967	0.961
+stop		0.753	0.682	0.715
−vocalic	**91.73%**	0.935	0.931	0.933
+vocalic		0.890	0.895	0.893

Table 5: Accuracy Rates for SVMs with polynomial kernel of order $d = 3$ on TIMIT core test set for recognition of articulatory manner features

3. Support Vector Machines

Support Vector Machines learn separating hyperplanes to classify instances in the feature space that are mapped from the input space of the classified data. The mapping from input space to feature space is performed with application of a kernel on the feature space. The dimension of the feature space is typically much higher than that of the original input space. The term 'feature' in this context is of course distinct from articulatory feature.

For a binary classification task with data points $\mathbf{x}_i$ $(i = 1, \ldots, n)$ and labels y_i we have the decision function $f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$. If the dataset is separable we can find a $\mathbf{w}$ such that the decision function will assign value $f(\mathbf{x}_i) = y_i$ for every i. As the sign is invariant to positive scaling of the expression inside of the sign, we can define canonical hyperplanes such that $\mathbf{w} \cdot \mathbf{x} + b = 1$ for the closest points on one side and $\mathbf{w} \cdot \mathbf{x} + b = -1$ for the closest points on the other side. The separating hyperplane is then defined by $\mathbf{w} \cdot \mathbf{x} + b = 0$ and its normal is then $\mathbf{w}/\|\mathbf{w}\|_2$. The margin between the canonical hyperplanes can be found as a projection of distance between the two closest points on opposite sides ($\mathbf{x}_1$ and $\mathbf{x}_2$) on the normal of separating hyperplane. Since $\mathbf{w} \cdot \mathbf{x}_1 + b = 1$ and $\mathbf{w} \cdot \mathbf{x}_2 + b = -1$ the margin is $1/\|\mathbf{w}\|_2$.

The SVM approach to binary decision function learning is to maximize the margin $1/\|\mathbf{w}\|_2$ that is summarized in an optimization task formulation

$$\min g(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \textit{w.r.t.} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \text{ for all } i$$

and the learning task can be reduced to minimization of the primal lagrangian

$$L = \frac{1}{2}(\mathbf{w}^T \cdot \mathbf{w}) - \alpha_i(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1),$$

where α_i are Lagrangian multipliers.

3.1. Kernels

From the description of support vector machines it is apparent that for a nonlinear problem it is not suitable to use a linear classifier. To make use of the beneficial properties of a linear SVM we need to map nonlinearly separable data into a space of typically higher dimensionality where linear separation of the data is possible. Thus we define a map from the input space $\mathbf{X}$ into feature space $\mathbf{H}$, $\Phi : \mathbf{X} \rightarrow \mathbf{H}$.

Although there is an infinite number of such mappings only some are suitable for practical application for computational complexity reasons. The kernel trick (Schölkopf and Smola, 2002) relieves from often exponential explosion of computations by introducing kernel k that is equivalent to the map Φ in that it holds $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$, where $\langle ., . \rangle$ is dot product. This property holds for polynomial kernels that map input vector into the feature vector composed of ordered polynomial expansions, eg. for order $d = 2$ of the polynomial and 2-dimensional input space we have $\Phi : (x_1, x_2) \rightarrow (x_1^2, x_2^2, 2x_1x_2)$.

4. Experiments with SVMs for Articulatory Feature Recognition

We extracted 52 values for every frame of speech signal that were used as inputs for the SVM classifiers. From each frame we extracted 12 Mel-frequency Cepstral Coefficients together with first and second order differences, frequencies of formants (F1-F5) with first order differences, bandwidths of detected formants, and fundamental frequency. The length of the speech signal frames was set to 25 ms and step between two adjacent frames to 10 ms. The original speech signal was sampled at 16 kHz. The distributions of classes vary significantly for different types of features. While the distribution of classes is almost equal (the case of AF *vocalic*) for half of the articulatory features, in the rest of the cases the positive classes are rare in the data. This

has a strong influence on the recall of the positive classes while the overall accuracy remains high.

In Tables 3, 4 and 5 we present results for SVMs with linear kernel and with polynomial kernel of second and third order, respectively, for the recognition of manner features based on FER on TIMIT core test set. The values of recall, precision, and F1-measure are presented for positive and negative classes of a articualtory feature.

The comparison of individual kernels in the SVM classification leaves us with observation that the performance improves for all articulatory features with increasing order of used kernels. From comparison of the performances with bagging we see that all SVMs perform better in terms of F1-measure for all features except the feature *fricative*.

Interestingly, drop in the ratio of cases with positive class in the data need not necessarily lead to drop in performance if it is accompanied by increase of 'compactness' of the class. This can be seen from the better performance for the feature *closure* which is on the frame level a subset of the feature *stop*.

5. Conclusion

We presented support vector machines with three types of kernels as approaches to recognition of articulatory manner features that we use as a building block of a continuous speech recognizer. The comparison was made between a linear and two polynomial kernels of second and third order for isolated frame recognition approach. Our results show high dependence of the performance on positive/negative class balance in the data whereby with increasing unbalance of the class distributions the performance of recognizers degrades.

According to the frame based values of F1-measure the SVM with polynomial kernel of third order gave superior performance over SVMs with the remaining two types of kernel for all articulatory manner features. These superiority of the third order polynomial kernel is underlined by monotone increase of the F1-measure for all classes. The comparison of SVM with third order polynomial kernel with bagging gives very similar results except for the articulatory feature *fricative* where the performance is better for bagging.

Performance of the SVMs was dependent on the frequency of occurrence of classes in the data. It achieved better performance in terms of recall, accuracy and F1-measure in cases where the distribution of positive and negative classes was not too unbalanced. An especially interesting case of this influence is the feature *flap* for which the positive class is contained in less the one percent of the speech frames. Although this feature was virtually undetected with our methods, from the point of view of speech recognition its practical importance is obviously smaller than that of more frequent articulatory features.

6. Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 02/IN1/I100. We would also wish to express our gratitude to Dr. Anja Geumann for support of our work.

7. References

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Christopher J. C. Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Julie Carson-Berndsen. 1998. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*. Kluwer, Dodrecht, Netherlands.

Shuangyu Chang, Mirjam Wester, and Steven Greenberg. 2005. An Elitist Approach to Automatic Articulatory-acoustic Feature Classification for Phonetic Characterization of Spoken Language. *Speech Communication*, 47(3):290–311, November.

John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren. 1993. DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. NIST, USA.

Hynek Heřmanský. 1999. Mel cepstrum, deltas, double-deltas,.. - What else is new? In *Proc. of Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland.

Roman Jakobson, Gunnar Fant, and Morris Halle. 1952. *Preliminaries to Speech Analysis: The distinctive features and their correlates*. MIT Press.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods—Support Vector Learning*, pp. 169–184. MIT Press.

Supphanat Kanokphara, Jan Macek, and Julie Carson-Berndsen. 2006. Comparative Study: HMM & SVM for Automatic Articulatory Feature Extraction. In *Proc. of the 19th Int'l. Conference IEA/AIE*, Annecy, France.

Katrin Kirchhoff. 1999. *Robust Speech Recognition using Articulatory Information*. Ph.D. thesis, Bielefeld.

Benoît Launay, Olivier Siohan, Arun Surendran, and Chin-Hui Lee. 2002. Towards Knowledge-Based Features for HMM Based Large Vocabulary Automatic Speech Recognition. In *Proc. of IEEE ICASSP*, vol. 1, pp. 817–820, Orlando, FL, USA, May.

Jan Macek, Supphanat Kanokphara, and Anja Geumann. 2005. Articulatory-acoustic Feature Recognition: Comparison of Machine Learning and HMM methods. In *Proceedings of SPECOM 2005*, vol. 1, pp. 99–103.

John R. Quinlan. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27:221–234.

Bernhard Schölkopf and Alexander J. Smola. 2002. *Learning with Kernels*. MIT Press, Cambridge, MA, USA.

Kenneth N. Stevens. 1980. Acoustic correlates of some phonetic categories. *JASA*, 68(3):836–842.

Kenneth N. Stevens. 2000. *Acoustic Phonetics*. MIT Press, Cambridge, MA, USA.

Sebastian Stüker, Tanja Schulz, Florian Metze, and Alex Waibel. 2003. Multilingual Articulatory Features. In *Proc. of IEEE ICASSP*, vol. 1, pp. 144–147. IEEE.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, New York.

Statistical Language Modeling of SiBN Broadcast News Text Corpus

Grega Milharčič*, Janez Žibert†, France Mihelič†

*Department of Comparative and General Linguistics
Faculty of Arts
University of Ljubljana
Aškerčeva 2, SI-1000 Ljubljana
grega.milharcic@guest.arnes.si

†Laboratory of Artificial Perception, Systems and Cybernetics
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana
{janez.zibert, france.mihelic}@fe.uni-lj.si

Abstract

The paper presents acquisition of the Slovenian broadcast news text corpus and its application in statistical language modeling. The problems encountered during the acquisition are described as well as constructing of language models for speech recognition purposes. Three different types of language models are built: a word-based 4-gram model, a class-based model with statistically-derived class maps and an interpolated model, combining the previous two. The comparison of language models is represented in terms of estimated perplexities on one third and one tenth of overall data used in evaluation experiments.

Statistično jezikovno modeliranje tekstovnega korpusa informativnih oddaj SiBN

V članku je predstavljena gradnja slovenskega besedilnega korpusa televizijskih informativnih oddaj SiBN in njegova uporaba pri statističnem jezikovnem modeliranju, namenjenemu samodejnemu razpoznavanju govora. Opisana je priprava in zbiranje besedil za gradnjo korpusa ter preizkusi statističnega jezikovnega modeliranja z uporabo teh podatkov. Zgrajeni so bili trije različni tipi jezikovnih modelov: besedni 4-gramski model, razredni model s statistično pridobljenimi razredi in interpolacijski model, ki je kombinacija prejšnjih dveh. Jezikovne modele smo primerjali med seboj na podlagi ocenjenih perpleksnosti, ki smo jih pridobili na testnem besedilu ene tretjine in ene desetine celotne zbirke.

1. Introduction

Statistical language models estimate the probabilities of word sequences which are usually derived form large collections of text material. Statistical language models can be applied in several tasks of language technologies (Manning & Schütze, 1999), including automatic speech recognition, optical character and handwriting recognition, machine translation, spelling correction. In our case, a text corpus of Slovenian broadcast news was acquired.

The corpus currently consists of text transcriptions of daily TV news shows provided by the national broadcast company RTV Slovenija for a period of one year. The collected text material corresponds to the spoken language transcriptions and is therefore suitable for building proper language models for automatic speech recognition.

Properly annotated and documented text data was used for the evaluation of different statistical language models for speech recognition. They will be applied in a system for large vocabulary continuous speech recognition for automatic transcription of Slovenian broadcast news, which is being developed at the Laboratory of Artificial Perception, Systems and Cybernetics at the University of Ljubljana.

We derived three different types of statistical language models which were based on n–gram statistics. This approach is a standard way of building language models for speech recognition. However, it should be noted that other approaches can be also successfully applied. For example, (Chelba, 2000) makes use of syntactic structure in language modeling, (Beutler et al., 2005) integrates a non-probabilistic grammar into large vocabulary continuous speech recognition, and (Zheng et al., 2005) defines an ontology-based language model. Other possibilities, which were also tried for Slovenian language, include specially designed techniques for modeling highly inflected languages (Sepesy Maučec et al., 2004) and topic-sensitive language modeling (Sepesy Maučec & Kačič, 2000).

The paper is organized as follows. The next section describes text corpus acquisition, followed by a section on applied language models. In the evaluation section the constructed language models are compared in terms of estimated perplexities in different experiments. We end with the conclusion and possible future work.

2. Text Corpus Acquisition

Text material for the corpus was collected from the subtitling transcriptions of the daily broadcast news (BN) shows transmitted live via teletext from a TV station RTV Slovenija. This service is used for providing a subtitling information of BN shows and is intended to help deaf and partially deaf people to follow the BN shows.

The corpus currently consists of the teletext subtitling transcriptions from December 2003 to December 2004 of daily BN shows Poročila, Dnevnik and Odmevi at 7.00 AM, 8.00 AM, 9.00 AM, 1.00 PM, 4.30 PM, 7.00 PM and 10.00 PM. The text material cover different BN news stories, different topics and different kind of information. The main part of the text data represents international news and news from Slovenia, but there are also cultural, financial and sport news, weather reports, traffic information, etc. The basic information of the corpus is given in Table 1.

number of BN shows	1 358
number of sentences	139 251
number of word tokens	2 295 664
number of word types	102 895

Table 1: Basic information of the SiBN text corpus.

As could be seen from the Table 1, the corpus is currently comprised from 2 million word tokens with vocabulary size of 100k different words. Thus, we could consider this corpus as a relatively small corpus with big amount of different words. This is typical for such kind of data representing broadcast news, where there exist many different word types belonging to different proper names, geographical and geopolitical names, technological and scientific terms, sport's results, etc.

The similar data can be captured also from newspapers or internet sites, which provide many more resources for building such text corpora, but there exist one important difference. The text material, which is acquired directly from subtitling transcripts of BN shows, is closer to spoken language, and therefore is more suitable for deriving language models for speech recognition purposes.

Another issue of capturing the text data via teletext is concerning the annotation and organization of the data into the corpus. In next sections we describe the acquisition of the data, text segmentation and annotation, and text normalization.

2.1. Text Segmentation

In the text acquisition process we dealt with two groups of problems. The first was connected with teletext data transmissions and the second with subtitling information. The major problems encountered in the acquisition process were:

- unwanted teletext marks, which did not belong to text transcripts;
- unwanted text and characters can appear anywhere in the texts due to the signal disturbance;
- punctuation marks . ! ? do not necessarily mean the end of the sentence (abbreviations, ordinal numbers, one-word exclamations, etc.), sometimes they are absent or appear randomly in the text, ? sometimes replaces other characters;
- punctuation marks : ; – also sometimes designate the end of the sentence;
- one of punctuation marks that otherwise appear in pairs, like ' ' " " ( ) may be absent;
- direct speech were designated in many different ways;
- sometimes sentences did not start with a capital;
- some sentences were interrupted, unfinished, repeated or started more than once;
- abbreviations: some were standard and common (oz., itd., itn., npr., g., ga., dr., C., &), other arbitrary and thus unpredictable;
- acronyms: with full stops or without, with all capitals or not, different spellings when declined;
- multipart words and proper names: with hyphens or not, written together or separately;
- numbers: cardinal, ordinal, decimal, fractions, percents, sport results, dates, character and product codes;
- Internet addresses and other strings;

In the text segmentation process we had to organize and annotate the subtitling data in a text corpus, which could be used for building a language models. Thus, we had to provide several automatic tools and also make some manual checking to transform the erroneous subtitling data in a way we needed.

An example of the raw text data, which was captured during data acquisition, is shown in Figure 1. The raw text data (in the top window) was constructed from several teletext subtitling records. Each record was represented by a time stamp of capturing and with transmitted text. The time stamps were additionally provided by our captioning tool, but they were not used in extracting text material from subtitling data, which was our main concern in a text segmentation process. As it can be seen from Figure 1, during the text segmentation process the teletext data from subtitling records shown in the top window should be transformed into the well–organized text showed in the bottom window.

In the example in Figure 1 the subtitling data posses several incorrectness or inconsistencies, which we had to change or remove. Due to the teletext service each subtitling record could be transmitted several times or there could also exist empty records. This is also presented in Figure 1, where there exist several unwanted teletext marks for the sentence that has started several times and marked without punctuation. There are also some other mistakes and inconsistencies: an ordinal number and a multipart word *36 letni*, which could also be written elsewhere as *36-letni* or *36letni*, etc. Some of these problems were solved automatically, other manually during consistency checking. Apostrophes and quotation marks were deleted, parentheses were deleted with their contents included. Common abbreviations were expanded to approximate the spoken form. All other symbols, which were not expected in the text data, were additionally analyzed and text was corrected accordingly.

All the transformed text material was also automatically spell–checked during the text segmentation process. The

Figure 1: An example of text acquisition and segmentation.

spell–checking was performed based on tool *Aspell*[1] to find possible typing errors, to replace letters *c*, *s*, *z* with *č*, *š*, *ž*, where appropriate, and, in some cases, to standardize words that can be spelled in different ways.

[1]http://aspell.sourceforge.net

In the last phase of the text segmentation process we had to manually check all the translated texts. In this phase we had to solve problems that could not be processed automatically, especially unwanted text from other broadcasts and punctuation issues. This phase was the most time consuming part of the acquisition process, but it was essential for an arrangement of the corpus in a consistent way.

2.2. Text Normalization

In order to use this corpus for developing statistical language models for speech recognition the text material had to be further normalized.

The text normalization process included the following tasks and procedures: all punctuation marks were deleted, capitals were transformed into lower case, the character encoding was standardized and beginnings and ends of sentences were labeled as '<s>' and '</s>', respectively.

2.3. Manually–annotated Word Classes

Additionally, we manually derived some word classes, which we used them latter in the process of building language models. The word classes were defined manually and annotated automatically in the text corpora.

There were two main reasons for defining such word classes. The first was, that we wanted additionally standardize the text material in places where one could expect inconsistent annotations. The second reason was the fact that some words could be equally likely used in the text and could be therefore better modeled with word classes. Hence, we decided to derive classes which belongs to two major groups of words: proper names and numbers. The manually–annotated classes are the following:

- acronyms: included strings of two or more capitals with an optional hyphen and ending; this class consisted of 969 different words;
- proper names: in this class belong non–first words, starting with a capital and with more than one letter; this class consisted of 29 843 different words;
- cardinal numbers: are strings of digits with an optional full stop in between, or strings of digits in the end of the sentence (1 140 different words);
- ordinal numbers: are strings of digits with a full stop in the end, which do not mark the end of a sentence, or strings of digits with hyphens and an ending not longer than 4 letters (longer most possibly mean not an ending, but a word of its own where hyphen was meant as a dash); in this class 1 300 different words were included;
- decimal numbers: are strings of digits with a comma in between (268 different words);
- fraction: are strings of digits with a slash in between (22 different words);
- percents: are string of digits, followed by a percent sign or a string 'odstot' plus ending or a string 'procent' plus ending; it was found 419 different words of such kind;

- sport results: are strings of digits with one or more colons in between (420 different words).

Altogether these word classes covered 34 381 different words, which represented 33 % of all different words in the corpus.

An automatic procedure for applying these word classes on the text data was designed. This procedure just replaces the words in classes with their corresponding class labels. In a such way we obtained two kinds of texts: one with classes and another without classes. For building of all language models in our experiments we used the text data, where word classes were annotated.

3. Statistical Language Models

The acquired text data, as described in the previous section, were used in the very first experiments of constructing different language models (LM). Three different types of language models were built: a word-based 4-gram language model, a class-based model with statistically-derived class maps and an interpolated language model, combining the previous two.

For each of these types two kind of evaluation experiments were performed. We divided the text data from the corpus into thirds and tenths. In the first group of experiments we built all three language models each time on different two thirds of overall data serving as the training text and one third was used as the test set. In the same manner the second group of experiments were performed on ten language models where each time different nine tenths of overall data served as the training text and one tenth as the testing text. In a such way we performed several experiments on different test and train data, which guaranteed us more objective evaluation of the proposed language models.

In each experiment we had to build a new language model from different training texts and test it on corresponding test data. The average statistics of the training and testing texts, which were used in experiments, are shown in Table 2. The average statistics based on the vocabularies of each datasets reveal the expected proportion of sentences and words according to the proportion of data in each group of experiments (2/3 train, 9/10 train).

		2/3 train	9/10 train
sentences	test	46 436	13 931
	train	92 871	125 376
word tokens	test	766 488	229 947
	train	1 504 506	2 038 571
word types	test	51 019	28 275
	train	40 314	46 722
OOV rate	test	2.99%	2.44%
	train	1.75%	1.41%

Table 2: The average statistics of data using in test and train set texts.

Due to the different training data in each experiment, the special attention was needed to model out–of–vocabulary (OOV) words. The basic rule here was to drop out the words with frequency of 1 in all training texts and map them to the unknown word class. This class was then used for modeling OOV words in different types of language models. Average OOV rates for different train and test datasets are also shown in Table 2. All other words, which were not marked as OOV words, were then used as a basic vocabulary in the training of the language models.

All tested language models were built in a standard way using the HTK Toolkit (Young et al., 2005). So, in the following subsections the main ideas and approaches in constructing of all three language models will be presented.

3.1. Word-based Language Models

We built a standard word–based 4-gram language model, which was used as a baseline language model in all of our experiments.

The main idea in n–gram language modeling is, that the probabilities of word sequences are estimated based on frequencies of words and word sequences obtained from training text material. In a word 4-gram language model the probabilities of words sequences are approximated from the conditional probabilities based on sequences of last 4 words:

$$P(w_1, \ldots, w_n) \simeq \prod_{i=1}^{n} P(w_i \mid w_{i-3}, w_{i-2}, w_{i-1}). \quad (1)$$

In our case the probabilities of 4-gram language models were estimated using Good-Turing discounting with Katz back-off smoothing (Katz, 1987). This is a standard procedure in building a n–gram language model with HTK Toolkit (Young et al., 2005).

3.2. Class-based Language Models

Class–based language models work in the same manner than word–based models, but, instead of words, word classes are derived and used for estimation of probabilities. The probabilities of word sequences can be therefore estimated in the case of 4-gram as

$$P(w_1, \ldots, w_n) \simeq \prod_{i=1}^{n} P(w_i \mid G(w_i)) \times P(G(w_i) \mid G(w_{i-1}), G(w_{i-2}, G(w_{i-3})), \quad (2)$$

where w_i represents words and $G(w)$ word classes. The probability of word sequences $w_1, \ldots, w_n$ is here approximated by a product of conditional probability that a word w_i belong to a word class $G(w_i)$ and a conditional probability that a word class $G(w_i)$ followed a sequence of word classes $G(w_{i-1}), G(w_{i-2}), G(w_{i-3})$. The estimated probability is therefore a generalization of the form in equation (1), where probabilities are estimated from word–based sequences.

The main issue here is how to obtain word classes. General idea is that we map words with similar syntactic or semantic behavior into the same class or category. A member word of such class is considered as equally likely to appear in contexts of any other member of the same class. Classes can be obtained manually or statistically.

In our case we derived classes statistically. For deriving word classes from the training texts of the corpus a word

exchange algorithm (Young et al., 2005) was applied. The procedure is shown in Figure 2.

1. **Initialize**: $\forall w \in \mathbb{W} : G(w) = 1$
   Set up the class map so that all words are in the first class and all other classes are empty (*or* initialize using some other scheme)
2. **Iterate**: $\forall i \in \{1, \ldots, n\} \wedge \neg s$
   For a given number of iterations $1 \ldots n$ or until some stop criterion s is fulfilled

   Iterate: $\forall w \in \mathbb{W}$
   For each word w

   (a) **Iterate**: $\forall c \in \mathbb{G}$
   For each class c

   i. **Move** word w to class c, remembering its previous class

   ii. **Calculate** the change in F_{M_C} for this move (F_{M_C} is defined below)

   iii. **Move** word w back to its previous class

   (b) **Move** word w to the class which increased F_{M_C} by the most, or do not move it if no move increased F_{M_C}

$$F_{M_C} = \sum_{g,h \in \mathbb{G}} C(g,h) \cdot \log C(g,h) - 2 \sum_{g \in \mathbb{G}} C(g) \cdot \log C(g)$$

where (g, h) is a class sequence $'hg'$ and $C(g)$ is the number of times that the class g occurs in the training text list of words

Figure 2: The clustering procedure for obtaining class maps.

This clustering algorithm optimizes the classification function $G(w) = g$, which maps a word w from the set of all words $\mathbb{W}$ to a word class g from the set of all classes $\mathbb{G}$. The map is set on the basis of the differences in entropy of the bigram and unigram word class probabilities, defined with function F_{M_C}. The basic procedure is following. In every iteration each word is mapped into that class (the class maps are deterministic – each word can only be in one class), which increase the overall entropy in function F_{M_C}. This procedure actually performs an exhaustive searching of optimum mapping between words and word classes. Due to computational complexity of the algorithm one should carefully set the actual number of classes and the number of iterations for finding the optimal class maps. One should choose the number of classes significantly lower than there are words in a training-data vocabulary, otherwise the class–based language model approximates to the word–based model. Our language models were built using 2 iterations and 600 classes, as it was recommended for our vocabulary size in (Young et al., 2005).

Word classes reduce the number of parameters and give more reliable estimations of rare words, which is useful in situations, where train and test conditions do not match. Also note, that one should expect higher perplexities, when using such models on training data, in comparison to word–based models. This can be explained by the fact that word–based models better estimate the expected word sequences in training text, while class–based models generalize the estimation of the word sequences with word classes.

3.3. Interpolated Language Models

Interpolated language models are generated by combining word–based and class–based language models. There exist number of combining techniques how to join two language models together. In our case, word–based and class–based language models were joined together by linear interpolation (Young et al., 2005).

We had to additionally set the interpolation weights for both models in a way to favor one model against the other. The interpolation weights were chosen to maximize the overall perplexity of an interpolated model on test data. The results are shown in Figure 3.

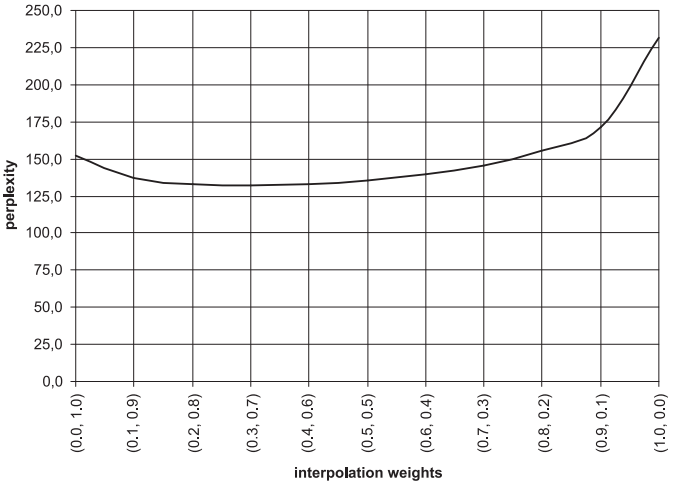


Figure 3: Setting interpolation weights for class–based and word–based language models in joined language models on test data. The first weight in a pair of weights in round brackets corresponds to class–based language model, the second weight belongs to word–based language model.

We tested different combinations of weights in interval from 0 to 1. As can be seen from the results in Figure 3, we obtained the best perplexity by applying linear interpolation of both models with interpolation weights of 0.7 and 0.3 for word–based and class–based models respectively.

4. Evaluation experiments

The perplexities of language models were estimated. In Table 3, the average perplexities of the three language models trained on two thirds of overall data and the ten language models trained on nine tenths of overall data for each type of language models built for 2-gram, 3-gram and 4-gram, evaluated on both testing and training texts parts, are given.

		word-based LM		class-based LM		interpolated LM	
		test	train	test	train	test	train
2-gram	2/3 train	233.6	84.5	322.7	272.3	213.2	93.1
	9/10 train	228.7	87.7	324.7	283.9	209.0	96.7
3-gram	2/3 train	171.2	20.0	255.4	117.5	150.7	23.0
	9/10 train	161.3	19.9	251.3	125.5	142.0	22.9
4-gram	2/3 train	163.8	13.1	239.7	59.3	142.8	14.7
	9/10 train	151.9	12.2	232.0	60.4	132.3	13.7

Table 3: The average perplexities of tested language models.

Results on the particular test an training text partition did not differ significantly between each other for the two thirds – one third and nine tenths – one tenth text partitions. Big differences between the estimated perplexities acquired from the training and test part of the corpus could be noticed. However estimated perplexities from the test parts are more descriptive. Interpolated n–grams models consistently gives better lower estimated perplexities results on the test sets compared to the corresponding n–grams word models. We also achieved as expected better results in the nine tenth train – one tenth test evaluation scenario. In this case larger amount of training data and smaller test part with consequently smaller OOV word rate was used (Table 2). As it was expected, the best model appeared to be the interpolated 4–gram model, trained on nine tenths of the overall data.

The results cannot be directly compared to the other recently reported results for Slovenian language since different text corpora with different vocabularies were used. However, a novel method for highly inflected languages used in language modeling of Slovenian should be mentioned: (Sepesy Maučec et al., 2004) reports of perplexity improvement from 360 to 248 and OOV rate improvement from 6.03% to 0.97% when cutting off the grammatical information from words in the so-called stem-ending language model, trained on a 59M-word corpus of newspaper Večer with vocabulary size of 64 000.

5. Conclusion

The acquisition of the Slovenian broadcast news text corpus was described. Using this corpus, three different types of statistical language models were built. We applied two different evaluation scenarios using two thirds and nine tenths of the corpus for training purposes. The language model with the lowest estimated perplexity on the test set was the interpolated 4-gram model, trained on nine tenths of overall data.

The statistical language models will be applied for automatic transcription of Slovenian broadcast news as one of the components of large vocabulary continuous speech recognizer. Future work might also include acquisition of a larger corpus, different extended annotations and application and experiments with other different approaches for language modeling.

6. Acknowledgement

The authors would like to thank the public broadcasting company RTV Slovenija for their permission to freely use the broadcast news data for scientific purposes.

7. References

Beutler, R., Kaufmann, T., & Pfister, B. (2005). Integrating a non-probabilistic grammar into large vocabulary continuous speech recognition. In *Proceedings of the IEEE ASRU 2005 Workshop*, pp. 104–109, San Juan, PR. IEEE.

Chelba, C. (2000). *Exploiting Syntactic Structure for Natural Language Modeling*. PhD thesis, Johns Hopkins University, Baltimore, MD.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401.

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.

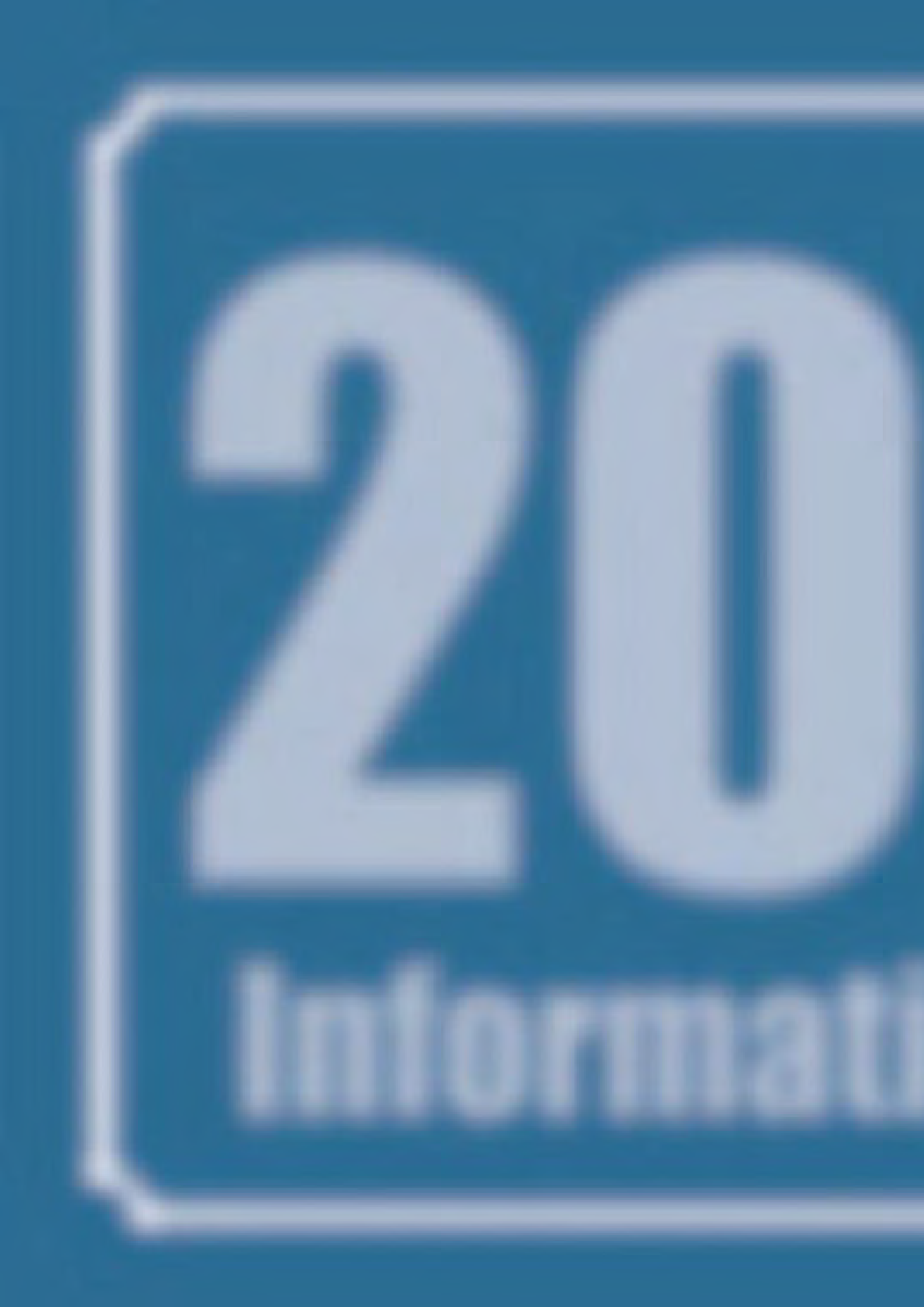
Sepesy Maučec, M. & Kačič, Z. (2000). Topic-sensitive language modelling. In *Proceedings of the Third International Workshop on Text, Speech and Dialogue*, pp. 235–258, Brno. Springer-Verlag.

Sepesy Maučec, M., Kačič, Z., & Horvat, B. (2004). Modelling highly inflected languages. *Information Sciences*, 166:249–269.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., & Woodland, P. (2005). *The HTK Book*. Cambridge University Engineering Department, Cambridge.

Zheng, D., Yu, H., Zhao, T., Li, S., & Peng, Y. (2005). Research on an ontology-based language model. In *Proceedings of the International Conference on Chinese Computing 2005*, Singapore. COLIPS.

Index avtorjev / Author index


20