Zbornik 15. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2012

Zvezek C

Proceedings of the 15th International Multiconference

INFORMATION SOCIETY – IS 2012

Volume C

Uredila / Edited by
Tomaž Erjavec, Jerneja Žganec Gros

8.–12. oktober 2012 / October 8th–12th, 2012
Ljubljana, Slovenia

Zbornik 15. mednarodne multikonference
INFORMACIJSKA DRUŽBA - IS 2012
Zvezek C

Proceedings of the 15th International Multiconference
INFORMATION SOCIETY - IS 2012
Volume C

Zbornik
Osme konference JEZIKOVNE TEHNOLOGIJE

Proceedings of the
Eighth LANGUAGE TECHNOLOGIES Conference

Uredila / Edited by

Tomaž Erjavec, Jerneja Žganec Gros

http://is.ijs.si

8. do 12. oktober 2012 / October 8th - 12th, 2012
Ljubljana, Slovenia

Uredniki:

Tomaž Erjavec
Odsek za tehnologije znanja
Institut »Jožef Stefan«, Ljubljana

Jerneja Žganec Gros
Alpineon d.o.o, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Tisk: Birografika BORI d.o.o.
Priprava zbornika: Mitja Lasič, Vedrana Vidulin, Vesna Lasič
Oblikovanje naslovnice: Vesna Lasič, Miran Krivec
Tiskano iz predloga avtorjev
Naklada: 50

Ljubljana, oktober 2012

Konferenco IS 2012 sofinancirajo
Ministrstvo za visoko šolstvo, znanost in tehnologijo
Javna agencija za raziskovalno dejavnost RS (ARRS)
Institut »Jožef Stefan«

Informacijska družba
ISSN 1581-9973

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

004.934(082)
81'322(082)

KONFERENCA Jezikovne tehnologije (8 ; 2012 ; Ljubljana)
Zbornik Osme konference Jezikovne tehnologije, 8. do 12. oktober 2012, [Ljubljana, Slovenia] : zbornik 15. mednarodne multikonference Informacijska družba - IS 2012, zvezek C = Proceedings of the Eighth Language Technologies Conference, October 8th-12th, 2012, Ljubljana, Slovenia : proceedings of the 15th International Multiconference Information Society - IS 2012, volume C / uredila, edited by Tomaž Erjavec, Jerneja Žganec Gros. - Ljubljana : Institut Jožef Stefan, 2012. - (Informacijska družba, ISSN 1581-9973)

ISBN 978-961-264-048-4
1. Dodat. nasl. 2. Erjavec, Tomaž, 1960- 3. Mednarodna multikonferenca Informacijska družba (15 ; 2012 ; Ljubljana)
263506944

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2012

V svojem petnajstem letu je multikonferenca Informacijska družba (http://is.ijs.si) med drugim z organizacijo konference ob **stoletnici Turingovega rojstva** še bolj utrdila mesto ene vodilnih srednjeevropskih konferenc, ki združuje znanstvenike z različnih raziskovalnih področij, povezanih z informacijsko družbo. Od leta 2012 dalje se bo nagrada za življenjske dosežke podeljevala v čast Donalda Michija in Alana Turinga. Letos smo v multikonferenco povezali deset odličnih neodvisnih konferenc, s čemer naša multikonferenca izstopa po širini in obsegu tem, ki jih obravnava, po akademski odprtosti in širini, ki spodbuja nove ideje, predvsem pa po tem, da ni tradicionalna konferenca, ampak se pogumno loteva vizionarskih tem, pogosto v interaktivni ali delavniški obliki.

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. Še bolj kot prejšnja leta smo prepričani, da sta stroka in vizija najpomembnejši za izhod iz stagnacije, v katero sta zašli Evropa in Slovenija.

Na vzporednih konferencah bo predstavljenih čez 210 referatov, vključevala pa bo tudi okrogle mize in razprave. Referati so objavljeni v zbornikih multikonference, izbrani prispevki pa bodo izšli tudi v posebnih številkah dveh znanstvenih revij, od katerih je ena Informatica, ki se ponaša s 35-letno tradicijo odlične znanstvene revije.

Multikonferenco Informacijska družba 2012 sestavljajo naslednje samostojne konference:

- 100 let Alana Turinga in 20 let SLAISa
- FORSEE - tehnološko predvidevanje na področju IKT
- Inteligentni sistemi
- Jezikovne tehnologije
- Kognitivne znanosti
- Robotika
- Rudarjenje podatkov in podatkovna skladišča (SiKDD 2011)
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Soočanje z demografskimi izzivi
- Vzgoja in izobraževanje v informacijski družbi.

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija in SLAIS. Zahvaljujemo se tudi Agenciji za raziskovalno dejavnost RS ter Ministrstvu za izobraževanje, znanost, kulturo in šport za sodelovanje in podporo. V imenu organizatorjev konference 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 2012 sta se programski in organizacijski odbor odločila, da bosta podelila posebno priznanje Slovencu ali Slovenki za izjemen življenjski prispevek k razvoju in promociji informacijske družbe v našem okolju. Z večino glasov je letošnje priznanje pripadlo dr. Francu Solini. Priznanje za dosežek leta je pripadlo dr. Juretu Leskovcu. V letu 2012 drugič podeljujemo nagrado »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobila ACTA, jagodo pa Urbana in Bicikelj. Čestitke nagrajencem!

Niko Zimic, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2012

In its 15th year, the Information Society Multiconference (http://is.ijs.si), among others with the conference devoted to **Alan Turing**, further established itself as one of the leading conferences in Central Europe gathering scientific community with a wide range of research interests in information society. For 2013 and further, the award for life-long outstanding contributions will be delivered in memory of Donald Michie and Alan Turing. This year, we organized ten independent conferences forming the Multiconference, delivering a broad range of topics and the open academic environment fostering new ideas makes which our event unique among similar conferences, promoting key visions in interactive, innovative ways.

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. We are more confident than ever that science and vision are the two most important issues to break the stagnation of Europe and Slovenia.

The Multiconference is running in parallel sessions with over 210 presentations of scientific papers. The papers are published in the conference proceedings, and in special issues of two journals. One of them is Informatica with its 35 years of tradition in excellent research publications.

The Information Society 2011 Multiconference consists of the following conferences:

- 100 years of Alan Turing and 20 years of SLAIS
- FORSEE - technological forecasting in ICT
- Intelligent Systems
- Language technologies
- Cognitive Sciences
- Robotics
- Data Mining and Data Warehouses (SiKDD 2011)
- Collaboration, Software and Services in Information Society
- Demographic Challenges in Europe
- Education in Information Society.

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

In 2012, the Programme and Organizing Committees decided to award one Slovenian for his/her life-long outstanding contribution to development and promotion of information society in our country. With the majority of votes, this honor went to Dr. Franc Solina. In addition, a reward for current achievements was pronounced to Dr. Jure Leskovec for his research on mining and modeling large social networks at Stanford. The information strawberry is pronounced to Urbana and Bicikelj, and the information lemon goes to ACTA. 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.

Niko Zimic, Programme Committee Chair
Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, Korea
Howie Firth, UK
Olga S. Fomichova, Russia
Vladimir A. 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

Matjaž Gams, chair
Vedrana Vidulin, co-chair
Mitja Luštrek
Robert Blatnik
Vesna Koricki Špetič
Mitja Lasič

Programme Committee

Nikolaj Zimic, chair
Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah

KAZALO / TABLE OF CONTENTS

Zbornik 15. mednarodne multikonference
INFORMACIJSKA DRUŽBA - IS 2012
Zvezek C

Proceedings of the 15th International Multiconference
INFORMATION SOCIETY - IS 2012
Volume C

Zbornik
Osme konference JEZIKOVNE TEHNOLOGIJE

Proceedings of the
Eighth LANGUAGE TECHNOLOGIES Conference

Uredila / Edited by

Tomaž Erjavec, Jerneja Žganec Gros

http://is.ijs.si

8. do 12. oktober 2012 / October 8th - 12th, 2012
Ljubljana, Slovenia

PREDGOVOR K ZBORNIKU OSME KONFERENCE »JEZIKOVNE TEHNOLOGIJE«

V pričujočem zborniku so objavljeni prispevki z osme konference "Jezikovne tehnologije", ki je potekala 8. in 9. oktobra 2012 v Ljubljani, v okviru multikonference "Informacijska družba" IS'2012. 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 konference o jezikovnih tehnologijah, organizirane v okviru IS. Zbornik vsebuje 38 prispevkov, ki obravnavajo široko paleto raziskav; posebej izstopa veliko število prispevkov o izdelavi korpusov in drugih jezikovnih virov ter jezikovnih orodij za slovenščino in hrvaščino, dobro zastopani pa so tudi prispevki s področja govornih tehnologij. Organizatorji bi se radi zahvalili vsem, ki so prispevali k uspehu konference: vabljenim predavateljem, avtorjem prispevkov, programskemu odboru za recenzentsko delo ter organizatorjem IS'2012.

PREFACE TO THE PROCEEDINGS OF THE EIGHTH LANGUAGE TECHNOLOGIES CONFERENCE

These proceedings contain the contributions for the Eighth Language Technologies Conference, which took place on October 8th and 9th 2012 in Ljubljana, in the scope of the Information Society multiconference, IS'2012. The conference was aimed at the members of the Slovenian Language Technology Society and others interested in the field, as a forum where they could present their work in the last two years, which have passed since the previous conference on Language Technologies organised in the scope of IS. The proceedings contain 38 contributions, which present a wide variety of research topics; especially numerous are contributions describing language technology tools and those dealing with creation and usage of corpora and other language resources for the Slovenian and Croatian languages, while papers about speech technologies are also well represented. 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 2012.

Tomaž Erjavec, Jerneja Žganec Gros

RECENZENTI

- doc. dr. Simon Dobrišek, Fakulteta za elektrotehniko, Univerza v Ljubljani
- doc. dr. Tomaž Erjavec (predsednik), Odsek za tehnologije znanja, Institut "Jožef Stefan"
- dr. Darja Fišer, Filozofska fakulteta, Univerza v Ljubljani
- doc. dr. Vojko Gorjanc, Filozofska fakulteta, Univerza v Ljubljani
- prof. dr. Ivo Ipsić, Faculty of Engineering, Univerza v Reki (Hrvaška)
- doc. dr. Primož Jakopin, Inštitut za slovenski jezik, ZRC SAZU
- prof. dr. Zdravko Kačič, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
- dr. Simon Krek, Amebis, d.o.o.
- doc. dr. Cvetana Krstev, Filozofska fakulteta, Univerza v Beogradu (Srbija in Črna gora)
- doc. dr. Nikola Ljubešić Odsek za informacijske in komunikacijske znanosti, Univerza v Zagrebu (Hrvaška)
- prof. dr. France Mihelič, Fakulteta za elektrotehniko, Univerza v Ljubljani
- doc. dr. Dunja Mladenić, Odsek za tehnologije znanja, Institut "Jožef Stefan"
- prof. dr. Marko Stabej, Filozofska fakulteta, Univerza v Ljubljani
- dr. Tomaž Šef, Odsek za inteligentne sisteme, Institut "Jožef Stefan"
- prof. dr. Rastislav Šuštaršič, Filozofska fakulteta, Univerza v Ljubljani
- prof. dr. Marko Tadić, Oddelek za jezikoslovje, Filozofska fakulteta, Univerza v Zagrebu (Hrvaška)
- doc. dr. Darinka Verdonik, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
- doc. dr. Špela Vintar, Filozofska fakulteta, Univerza v Ljubljani
- dr. Jerneja Žganec Gros (predsednica), Alpineon, d.o.o.
- doc. dr. Janez Žibert, Fakulteta za matematiko, naravoslovje in informacijske tehnologije, Univerza na Primorskem

Slovene-Croatian Treebank Transfer Using Bilingual Lexicon Improves Croatian Dependency Parsing

Željko Agić*, Danijela Merkler, Daša Berović****

*Department of Information and Communication Sciences, **Department of Linguistics
Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, 10000 Zagreb, Croatia
zagic@ffzg.hr, dmerkler@ffzg.hr, dberovic@ffzg.hr

Abstract

A method is presented for transferring dependency treebanks between similar languages by using a bilingual lexicon, aiming to improve dependency parsing accuracy on the target language. It is illustrated by transferring the Slovene Dependency Treebank to Croatian by using a GIZA++ bilingual lexicon constructed from the Croatian-Slovene 1984 parallel corpus from the Multext East project. The transferred treebank is merged with the Croatian Dependency Treebank and the merged treebank is used to train and test two graph-based dependency parsers. MSTParser and CroDep accuracy on parsing the 1984 fictional text shows a statistically significant increase and a similar decrease on parsing the Croatian Dependency Treebank newspaper text.

Slovensko-hrvaški prenos drevesnic z uporabo dvojezičnega leksikona izboljša odvisnostno razčlenjevanje hrvaščine

Prispevek predstavi metodo za prenos skladenjskih oznak korpusov med podobnimi jeziki z uporabo dvojezičnega leksikona, katere namen je izboljšati točnost odvisnostnega razčlenjevanja na ciljnem jeziku. Metodo ilustriramo s prenosom Slovenske odvisnostne drevesnice na hrvaški jezik z uporabo dvojezičnega leksikona, ki smo ga s programom GIZA++ izluščili iz vzporednega hrvaško-slovenskega korpusa 1984 projekta MULTEXT-East. Prenešena drevesnica je združena s Hrvaško odvisnostno drevesnico, združena drevesnica pa je nato uporabljena za učenje in testiranje dveh odvisnostnih razčlenjevalnikov, ki temeljita na teoriji grafov. Natančnost razčlenjevalnikov MSTParser in CroDep na leposlovnem delu 1984 pokaže statistično signifikantno izboljšanje in podobno zmanjšanje na razčlenjevanju Hrvaške odvisnostne drevesnice.

Keywords: treebank transfer, bilingual lexicon, dependency parsing

1. Introduction

Dependency treebanks are considered to be a sparse language resource. As stated in (Ambati and Chen, 2010), only a few languages in the world enjoy the status of resource-rich languages from the viewpoint of dependency treebanking, while tools and language resources supporting syntactic analysis in the framework of dependency syntax are unavailable or inadequate for many other languages. An illustration is given in (Zhao et al., 2009) that the treebanks of ten different languages from the CoNLL 2007 shared task on multilingual dependency parsing (Nivre et al., 2009) – restricted to a maximum of 500 thousand tokens per treebank – summed up to approximately 2 million tokens, 0.5 million of those being allocated by the Prague Dependency Treebank and certain treebanks accounting for not more than 30 thousand tokens or approximately 1.5 percent of the sum.

From this specific viewpoint, syntactically annotated corpora of Croatian and Slovene are considered to be small, while still sufficient to perform meaningful dependency parsing experiments from the viewpoint of the CoNLL 2006 and 2007 shared tasks. Croatian Dependency Treebank (Tadić, 2007) currently contains approximately 90 kw, while two dependency treebanks implementing two different models of dependency syntax are available for Slovene – the 30 kw Slovene Dependency Treebank (Džeroski et al., 2006) and the 100 kw JOS corpus (Erjavec et al., 2010). The first data-driven dependency parsing experiments for Slovene were conducted with the former one within the CoNLL 2006 shared task and the overall parsing accuracy score (LAS) of approximately 74% was observed, also showing that graph-based parsing methods significantly outperformed the transition-based parsing methods used in the experiments. Dependency parsing of Croatian texts by using the Croatian Dependency Treebank was thoroughly investigated just recently (Agić, 2012; Berović et al., 2012), showing a similar preference for graph-based over transition-based parsing (ca 74% vs. 71% LAS). Parsing accuracy within the data-driven graph-based dependency parsing framework was further increased by utilizing a k-best spanning tree parsing approach (Hall, 2007) with valency lexicon reranking (Agić, 2012), reaching an overall accuracy of approximately 78% LAS.

Croatian and Slovene are similar languages, i.e. they are both genetically and culturally close languages (Tadić, 2007) with small but usable dependency treebanks. Due to their similarity and resource availability, transferring a treebank from one language to another for purposes of improving dependency parsing accuracy is considered in this experiment. Treebank transfer is basically defined as "translating" a treebank from source language to target language while maintaining its syntactic annotation layer, effectively creating a syntactically annotated resource for the target language. Existing approaches mostly do not include

language similarity as a feature of significance for syntactic transfer and deal with generic approaches. These approaches include methods based on machine learning techniques (Jansche, 2005), word alignment and/or machine translation (Ambati and Chen, 2010) and parser delexicalization (Zeman and Resnik, 2008; Søgaard, 2011; McDonald et al., 2011). Lightweight machine-translation-related methods also exist and are mostly based on word-by-word transfer by using bilingual dictionaries of source and target languages (Zhao et al., 2009; Durrett et al., 2012). Relatedness of Croatian and Slovene in levels of linguistic description up to and including the syntactic level – both in terms of observed similarity and in terms of language resource compatibility – indicated that the computationaly inexpensive syntactic transfer method based on a Croatian-Slovene bilingual lexicon might improve dependency parsing scores. In the described experiment, Slovene was chosen as source language and Croatian as target language.
The following sections describe the resources and tools – parallel corpora, bilingual lexicon, treebanks and parsers – used in the experiment, the experiment preparation and its results in terms of observed dependency parsing accuracy in several test scenarios. Future work plans regarding treebank transfer and dependency parsing of Croatian are also briefly outlined.

2. Resources and tools

Treebank transfer from Slovene to Croatian by using a bilingual dictionary requires a dependency treebank of Slovene and a bilingual dictionary. Being that a dependency treebank for Croatian also exists, the transfer is implemented as a method for enlarging the Croatian treebank. To the best knowledge of the authors, a freely available Croatian-Slovene dictionary or bilingual lexicon is currently not available. Thus a bilingual lexicon was constructed for purposes of this experiment by using a freely available Croatian-Slovene parallel corpus. These resources are briefly described in the following section. Additionaly, the two graph-based parsers used in the experiment are also sketched.

2.1. Treebanks and other resources

The Croatian Dependency Treebank (HOBS) (Tadić, 2007) is a dependency treebank built along the principles of Functional Generative Description, as adapted in the Prague Dependency Treebank (Hajič et al., 2000). The ongoing construction of HOBS closely followed the guidelines set by the Prague Dependency Treebank, with their simultaneous adaptation to the specifics of the Croatian language. HOBS currently consists of 3,465 sentences in the form of dependency trees that were manually annotated with syntactic functions. These sentences, encompassing approximately 90,000 tokens, stem from the Croatia Weekly 100 kw corpus that is a part of the newspaper sub-corpus of the Croatian National Corpus. The Croatia Weekly sub-corpus was previously sentence-delimited, tokenized, lemmatized and MSD-annotated by linguists. Thus, each of the analyzed sentences contains the manually assigned information on part-of-speech, morphosyntactic category, lemma, dependency and syntactic function for each of the wordforms.

Sentences in HOBS are annotated according to the Prague Dependency Treebank syntactic annotation manual, with respect to differing properties of the Croatian language and consulting the Slovene Dependency Treebank (SDT) project (Džeroski et al., 2006). The syntactic functions utilized in HOBS are thus considered to be compatible with those used in SDT. The Slovene Dependency Treebank contains a part of the morphosyntactically annotated Slovene component of the parallel Multext East corpus (Erjavec, 2004), i.e. the first third of the Slovene translation of the novel 1984 by George Orwell, containing approximately 30,000 tokens in 2,000 sentences. Similar to HOBS, the SDT project was also based on the Prague Dependency Treebank – more precisely, development of HOBS stemmed from the experience of SDT in porting the Czech annotation rules to Slovene. With respect to this fact, the two treebanks can be considered to be highly compatible. The JOS syntactically annotated corpus (Erjavec et al., 2010) of aproximately 100,000 tokens in 6,100 sentences utilized a different syntactic annotation and was thus not used in this experiment. However, a mapping between the JOS annotation and the PDT-style annotation is possible.

As noted previously, no Croatian-Slovene dictionary-like resources were readily available and an approach with automatic construction of a bilingual lexicon from a parallel corpus was implemented here with respect to that fact. Two parallel corpora for the Croatian-Slovene language pair were usable when conducting the experiment – the fully completed 1984 parallel corpus from the Multext East project and the Croatian-Slovene Parallel Corpus in its early development state (Požgaj Hadži and Tadić, 2000). Being that the latter one is still in development, is not entirely document-, sentence- or word-aligned and differs in domain from the source treebank (i.e. SDT), the Croatian-Slovene subset from the 1984 parallel corpus was chosen for bilingual lexicon construction. The corpus was sentence-aligned using hunalign (Varga et al., 2005) in realignment mode, keeping only 1:1 sentence alignments. The resulting set of Croatian-Slovene sentences contained 6,337 sentence pairs and 210,948 tokens. Basic stats for this resource and the treebanks are given in table 1.

The dictionary was constructed from the sentence pairs using GIZA++ (Och and Ney, 2003). It contained 52,502 Slovene-Croatian word pairs for 16,432 different Slovene word forms that translated into 17,368 different Croatian word forms. The entries (i.e. word pairs) were sorted by translation probability obtained from the parallel corpus, respecting the GIZA++ format.

feature	HOBS	SDT	hr-1984	si-1984
sentences	3,465	1,997	6,337	6,337
tokens	88,045	36,554	101,774	102,837
MSD tags	828	789	802	1039
syntactic tags	69	69	N/A	N/A

Table 1: Basic stats for used corpora

2.2. Dependency parsers

Two graph-based dependency parsers were selected to be used in the experiment – MSTParser and CroDep. The selection was based on previous experiments with Croatian dependency parsing (Agić, 2012) that showed a strong preference towards graph-based, rather than transition-based dependency parsing of Croatian texts.

MSTParser (McDonald et al., 2006) is a state-of-the-art graph-based dependency parser generator with first and second order arc-factored language models, perceptron learning algorithm and both projective and non-projective parsing algorithms. It was used in this experiment to generate second order arc-factored non-projective parsers for Croatian. This MSTParser configuration was previously shown (Agić, 2012) to obtain the highest parsing accuracy on HOBS among all the tested standalone parsers (approximately 74.53% LAS).

CroDep is a novel k-best maximum spanning tree dependency parser with valency lexicon reranking (Agić, 2012), design specifically to increase the accuracy of parsing Croatian texts by utilizing a valency lexicon of Croatian verbs CROVALLEX (Mikelić Preradović, 2008; Mikelić Preradović et al., 2009). It is based on the k-MST parser (Hall, 2007) in that it produces a number of candidate dependency trees for an input sentence, sorted by confidence, and these candidate trees are then reranked by a rule-based reranking module that uses CROVALLEX as a knowledge base. Specifically, every dependency relation that attaches to a verb is assigned an additional weight, which is in turn decided by matching its properties with the constraints and requirements stated in CROVALLEX for that specific verb entry. Sums of these additional weights are assigned to the candidate trees and they are reranked by consulting both the ranking list of the k-best parser and the ranking list of the CROVALLEX-based module. The parser was tested on HOBS and achieved a parsing score of approximately 77.21% LAS, i.e. significantly better than the previously top-performing standalone MSTParser. As previously indicated, CroDep currently implements a first order arc-factored language model with perceptron learning, a k-best maximum spanning tree algorithm similar to the one implemented in k-MST and a valency lexicon reranker. It can be (made to be) used independently of the input language, as long as a verb valency lexicon is available for that language. The parser is currently a prototype and will be made publically available as soon as it leaves the early development stage and is tested on a certain number of languages meeting the stated requirements.

3. Experiment and results

3.1. Experiment setup

The experiment was basically envisioned in three stages. Firstly, SDT is "translated" to Croatian by simple word-to-word mapping with the Croatian-Slovene bilingual lexicon. Secondly, the translated resource (henceforth called hr-SDT) is assigned the Croatian metadata required for training the parsers, i.e. lemmas and morphosyntactic tags. Thirdly, parsers are trained and tested on manually dependency parsed Croatian texts. The translation of SDT was done at unigram level, i.e. by mapping each of the Slovene tokens to a respective Croatian token from the bilingual dictionary. Only the pairs with highest probabilities attached by GIZA++ were chosen from the dictionary. The resulting hr-SDT treebank therefore contained the same number of sentences and tokens as the original SDT (1,997 sentences and 36,554 tokens, see table 1) and the same syntactic features. A total of 29,344 token and 25,786 lemma replacements occured by consulting the lexicon. The transfer process is illustrated by Figure 1.

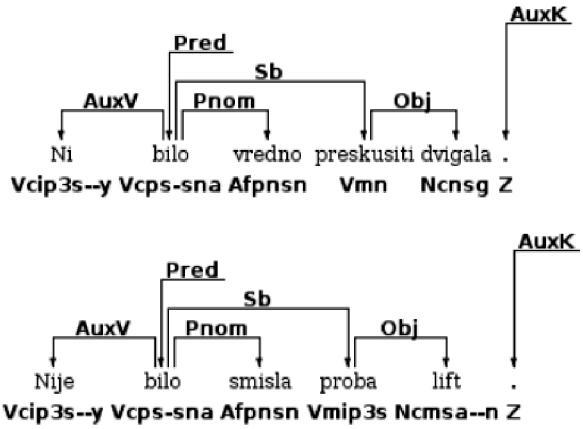


Figure 1: Dependency tree transfer example

For assessing the translation quality, 100 sentences were randomly selected from hr-SDT, paired with respective sentences from SDT and Croatian 1984 corpus and manually evaluated for adequacy and fluency on 1-5 scale. Translation adequacy was scored at approximately 3.64 and fluency at 2.99.

Two courses of action were taken with respect to metadata assignment. In the first one, Slovene lemmas were also translated to Croatian via bilingual lexicon and Slovene MSD tags were held. In the second one, hr-SDT was lemmatized and MSD-tagged using the CroTag HMM tagger and lemmatizer (Agić et al., 2008; Agić et al., 2009) trained on the Croatian 1984 corpus. Croatian MSD tags frequently differed from Slovene (14,216 occurences), but very infrequently in part-of-speech information. The tagging accuracy can be estimated at approximately 85% on basis of previous experiments (Agić et al., 2008).

Training sets for the parsers were created by attaching hr-SDT to training sets created from HOBS. As described in detail in (Agić, 2012) and according to CoNLL 2006 and 2007 shared task rules, 10 disjoint testing sets of approximately 5,000 tokens were extracted from HOBS, leaving 10 disjoint training sets for creating language models, approximately 83,000 tokens each. Each of these training sets was merged with both versions of hr-SDT, i.e. the one with Slovene MSDs and the one with lemmas and MSDs assigned by the CroTag tagger. This resulted in two batches of 10 training sets to be used in training MSTParser and CroDep. Tenfold cross-validated testing was to be done on both HOBS and the Croatian 1984 corpus. As the latter one is not syntactically annotated, we created a test set by man-

ually annotating 345 sentences and 5,226 tokens from the corpus respecting the HOBS and SDT standard. Results of a previous experiment (Agić, 2012) with parsing on HOBS were used here as a reference point. Labelled attachment score (LAS) was observed.

3.2. Results

The obtained results are displayed in table 2. They can be observed from several viewpoints.

Firstly and most importantly, the results indicate that the usefulness of treebank transfer is domain-dependent in this specific experiment. More specifically, introducing variants of hr-SDT – respecting the fact that SDT is a corpus of fictional text – to the HOBS training set decreases the overall parsing accuracy on parsing newspaper texts from HOBS by 0.53 and 0.57% LAS for MSTParser and by 0.32 and 0.44% LAS for CroDep, using MSD-tagged and untagged hr-SDT, respectively. The negative influence of introducing hr-SDT to HOBS changes to positive when parsing the hr-1984 test set of fictional text. The observed improvements over the baseline are 0.93 and 1.18% LAS for MSTParser and 0.89 and 1.11% LAS for CroDep. Parser language models obviously benefit from the quantity of data in HOBS when parsing fictional text, while the introduction of hr-SDT diverts the models from the properties of sentences in newspaper text.

At this point, it might be argued that using the 1984 Croatian-Slovene parallel corpus to construct a bilingual lexicon and to facilitate syntactic transfer introduces a bias with respect to the obtained results, being that the parsed text originates from the same source. However, the bias is here considered to be accounted for by the small size of the parallel corpus and the resulting bilingual lexicon and the resulting adequacy and fluency of translated text. Moreover, as discussed previously, other Croatian-Slovene parallel resources were not available at the time of conducting the experiment and thus using the 1984 parallel corpus was not a matter of choice.

Secondly, the observed decrease in parsing accuracy when shifting from newspaper to fictional text is substantial. Top scores for CroDep on these domains differ by 4.73% LAS in favor of newspaper text, while this difference amounts to 4.84% LAS for MSTParser. Treebank transfer benefits from lemmatization and MSD-tagging in all test scenarios. However, the observed difference between parsing accuracy when using tagger-assigned as opposed to transferred morphosyntactic tags is not shown to be statistically significant here.

Finally, The top-scoring parser on both fictional and newspaper text is CroDep. Its average difference over MSTParser is approximately 2.71% LAS across domains. It scored 72.48% LAS on hr-1984 and 77.21% LS on HOBS, topping MSTParser by 2.78% LAS and 2.68% LAS on these two test samples. UAS and LA metric were also used in the experiment and were shown to closely follow the pattern displayed by the LAS metric and were therefore excluded as they are less informative.

Test set	Model	MST	CroDep
hr-1984	HOBS	68.51	71.37
	HOBS + hr-SDT	69.44	72.26
	HOBS + hr-SDT tagged	69.69	72.48
HOBS	HOBS	74.53	77.21
	HOBS + hr-SDT	73.96	76.77
	HOBS + hr-SDT tagged	74.00	76.89

Table 2: LAS for MSTParser and CroDep hr-SDT language models on hr-1984 and HOBS

4. Conclusions and future work

Using Croatian Dependency Treebank, Slovene Dependency Treebank, Croatian-Slovene parallel resources and existing dependency parser generators, this experiment has shown that treebank transfer between similar languages by using a bilingual lexicon improves dependency parsing accuracy for the target language. It was also experimentally shown that the observed improvement is domain-dependent. Parsing accuracy peaked for the hybrid dependency parser CroDep at 72.48% LAS on fictional text and 77.21% LAS on newspaper text.

Future work in Slovene-Croatian treebank transfer will be targeted to several directions. Domain-specific bilingual lexica might introduce a positive bias for in-domain parsing. One such lexicon could be constructed from the Croatian-Slovene parallel corpus even in its early development stage, e.g. by using parallel sentence extractors such as LEXACC (Stefanescu et al., 2012), that operate on comparable corpora. The issue of bilingual lexica might also be addressed by using English as interlingua. Regarding bilingual lexica and transfer, a more elaborate approach to machine translation could be implemented along the lines of (Zhao et al., 2009) by using probabilistic word-by-word decoding to obtain translations of higher quality. The experiment presented here could also be repeated by setting Croatian as source and Slovene as target language. The syntactic annotation of the JOS corpus could also be mapped to SDT style and vice versa, as well as HOBS, providing an even larger resource for syntactic transfer. Moreover, the method could also be tested on other language pairs with compatible treebanks, e.g. Czech-Slovene and Czech-Croatian, even though syntactic transfer via bilingual lexica or statistical machine translation methods might pose a challenge with respect to availability of parallel corpora for these language pairs.

5. Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments.

The results presented here were partially obtained from research within project CESAR (ICT-PSP, grant 271022) funded by the European Commission, and partially from projects 130-1300646-0645 and 130-1300646-1776 funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

6. References

Agić Ž, Tadić M, Dovedan Z. 2008. Improving Part-of-Speech Tagging Accuracy for Croatian by Morphological Analysis. *Informatica*, 32(4), 2008.

Agić Ž, Tadić M, Dovedan Z. 2009. Evaluating Full Lemmatization of Croatian Texts. *Recent Advances in Intelligent Information Systems*, Warsaw, Academic Publishing House EXIT, 2009, pp. 175–184.

Agić Ž. 2012. *Pristupi ovisnosnom parsanju hrvatskih tekstova*. PhD thesis. University of Zagreb, Faculty of Humanities and Social Sciences, 2012.

Ambati V, Chen W. 2010. Cross Lingual Syntax Projection for Resource-Poor Languages.

Berović D, Agić Ž, Tadić M. 2012. Croatian Dependency Treebank: Recent Development and Initial Experiments. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, Istanbul, 2012.

Durrett G, Pauls A, Klein D. 2012. Syntactic Transfer Using a Bilingual Lexicon. In *Proceedings of the 2012 EMNLP-CoNLL*.

Džeroski S, Erjavec T, Ledinek N, Pajas P, Žabokrtský Z, Žele A. 2006. Towards a Slovene Dependency Treebank. In *Proceedings of Fifth International Conference on Language Resources and Evaluation*.

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*.

Erjavec T, Fišer D, Krek S, Ledinek N. 2010. The JOS Linguistically Tagged Corpus of Slovene. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*.

Hajič J, Böhmová A, Hajičová E, Vidová Hladká B. 2000. The Prague Dependency Treebank: A Three-Level Annotation Scenario. *Treebanks: Building and Using Parsed Corpora*, Kluwer, 2000.

Hall K. 2007. k-Best Spanning Tree Parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.

Jansche M. 2005. Treebank Transfer. In *Proceedings of the IWPT 2009*.

McDonald R, Lerman K, Pereira F. 2006. Multilingual Dependency Parsing with a Two-Stage Discriminative Parser. In *Proceedings of CoNLL-X*.

McDonald R, Petrov S, Hall K. 2011. Multi-Source Transfer of Delexicalized Dependency Parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Mikelić Preradović N. 2008. *Pristupi izradi strojnog tezaurusa za hrvatski jezik*. PhD thesis, University of Zagreb, Faculty of Humanities and Social Sciences, 2008.

Mikelić Preradović N, Boras D, Kišiček S. 2009. CROVALLEX: Croatian Verb Valence Lexicon. In *Proceedings of the ITI 2009 — 31st International Conference on Information Technology Interfaces*, SRCE, Zagreb, 2009, pp. 533–538.

Nivre J, Hall J, Kübler S, McDonald R, Nilsson J, Riedel S, Yuret D. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007*.

Och F J, Ney H. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1), pp. 19–51.

Požgaj Hadži V, Tadić M. 2000. Hrvatsko-slovenski paralelni korpus. In *Proceedings of the Language Technologies Conference*, Jožef Stefan Institute, Ljubljana, 2000.

Søgaard A. 2008. Data Point Selection for Cross-Language Adaptation of Dependency Parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Stefanescu D, Ion R, Hunsicker S. 2012. Hybrid Parallel Sentence Mining from Comparable Corpora. *Proceedings of the 16th EAMT Conference*, pp. 137–144.

Tadić M. 2007. Building the Croatian Dependency Treebank: the initial stages. *Suvremena lingvistika*, 63.

Varga D, Németh L, Halácsy P, Kornai A, Trón V, Nagy V. 2005. Parallel Corpora for Medium Density Languages. *Proceedings of the RANLP 2005*, pp. 590–596.

Zeman D, Resnik P. 2008. Cross-Language Parser Adaptation between Related Languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pp. 35–42.

Zhao H, Song Y, Kit C, Zhou G. 2009. Cross Language Dependency Parsing using a Bilingual Lexicon. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

Disambiguating vectors for bilingual lexicon extraction from comparable corpora

Marianna Apidianaki,* Nikola Ljubešić,† Darja Fišer[#]

* LIMSI-CNRS
BP 133, F-91403 ORSAY CEDEX, France
marianna.apidianaki@limsi.fr
† Department of Information Sciences, Faculty of Humanities and Social Sciences, University of Zagreb
Ivana Lučića 3, HR-10000 Zagreb, Croatia
nikola.ljubesic@ffzg.hr
[#] Department of Translation, Facult of Arts, University of Ljubljana
Aškerčeva 2, SI-1000 Ljubljana, Slovenia
darja.fiser@ff.uni-lj.si

Abstract

This paper presents an approach to enhance the extraction of translation equivalents from comparable corpora by plugging in bilingual lexico-semantic knowledge harvested from a parallel corpus. First, the bilingual lexicon obtained from word-aligning the parallel corpus replaces an external seed dictionary, making the approach knowledge-light and portable. Next, instead of using simple 1:1 mappings between the source and the target language, translation equivalents are clustered into sets of synonyms based on contextual similarities, enabling us to expand the translation of vector features with several translation variants. And last but not least, the vector features are disambiguated and translated only with the translation variants from the most appropriate cluster, thus producing less noisy vectors that allow for a more successful cross-lingual comparison of the vectors compared to simpler methods.

Razdvoumljanje vektorjev za izboljšanje luščenja dvojezičnih leksikonov iz primerljivih korpusov

V prispevku predstavljamo pristop za izboljšanje luščenja prevodnih ustreznic iz primerljivih korpusov z dodatnim virom leksiko-semantičnega znanja, izluščenega iz vzporednega korpusa. Za razliko od večine sorodnih pristopov dvojezični leksikon, potreben za prevajanje kontekstnih vektorjev, izdelamo avtomatsko iz vzporednega korpusa. Tako pristop ni več odvisen od slovarja, potrebnega za prevajanje kontekstnih vektorjev in je tako prenosljiv na številne jezikovne pare in strokovna področja. V naslednjem koraku prevodne ustreznice v dvojezičnem leksikonu razvrstimo v gruče, kar nam omogoča, da lastnosti v kontekstnih vektorjev, izdelanih iz primerljivih korpusov, prevajamo z več kot eno prevodno različico. To nam olajša primerjavo kontekstnih vektorjev v izvornem in ciljnem jeziku. Tretja izboljšava, ki jo v prispevku predstavljamo, pa je razdvoumljanje večpomenski lastnosti kontekstnih vektorjev iz primerljivega korpusa z gručami, generiranimi iz dvojezičnega leksikona, ki omogoča natančnejše prevajanje vektorjev in izboljša njihovo primerjavo z vektorji v ciljnem jeziku.

1. Motivation and related work

Due to the lack of general language parallel corpora, finding translations in comparable corpora has become a very active area of research. The main idea behind this approach is the assumption that a source word and its translation appear in similar contexts, so that in order to identify them their contexts are compared via a seed dictionary (Fung, 1998; Rapp, 1999). The biggest advantage of the approach is that it offers low-resourced language pairs and domains a fast and affordable way to construct bilingual lexica. However, it also presupposes the availability of a bilingual dictionary to translate vector features, which is not the case for many language pairs or domains. In addition, the original approach and most of its extensions (Shao and Ng, 2004; Otero, 2007; Yu and Tsujii, 2009; Marsi and Krahmer, 2010) neglect polysemy and consider a translation candidate as correct if it is an appropriate translation for at least one possible sense of the source word, which will often be the most frequent sense of the word due to the way context vectors are built.

The goal of this paper is twofold: (1) we eliminate the need for an external knowledge source by automatically extracting a bilingual lexicon from a parallel corpus, and (2) we propose a way of disambiguating polysemous features in the context vectors, as these features may be translated differently according to the sense in which they are used in a given context.

The need to bypass pre-existing dictionaries has been addressed by Koehn and Knight (2002) who built the initial seed dictionary automatically, based on identical spelling features. Cognate detection has also been used by Saralegi et al. (2008). Both approaches have been successfully combined by Fišer and Ljubešić (2011) who showed that the results with an automatically created seed lexicon, based on the similarity between the languages, can be as good as with a pre-existing dictionary.

But all these approaches cannot be used as successfully for language pairs with little lexical overlap, such as English (EN) and Slovene (SL), which is the case in this experiment. We believe we can produce less noisy vectors and improve their comparison across languages by using contextual information to disambiguate their features. A similar idea has been implemented by Kaji (2003) who clustered synonymous Japanese translations of English words in comparable corpora using pre-defined bilingual dictionaries. In addition, instead of providing one translation for each disambiguated feature, we translate it with all translation equivalents that belong to the assigned cluster similar to Déjean et al. (2005) who use a bilingual thesaurus instead of a lexicon. The contribution of this paper is a language independent and fully automated corpus-based approach to bilingual lexicon extraction from comparable corpora that does not rely on any external knowledge sources to determine word senses or translation equivalents.

The rest of the paper is organized as follows: In the next section we present the resources that were used in our experiments. In Section 3, we describe the approach and the experimental setup in detail. Evaluation and discussion of the obtained results are given in Section 4, after which the paper is wrapped up with some concluding remarks and ideas for future work.

2. Resources used

2.1. Comparable corpus

The custom-built English-Slovene comparable corpus that we use for bilingual lexicon extraction is a collection of popular health and lifestyle articles found in healthy-living magazines and on the Internet. The core part of the corpus was collected manually from the Slovene reference corpus FidaPLUS (Arhar et al. 2007), already part-of-speech (PoS) tagged and lemmatized. All articles from the Zdravje magazine published between 2003 and 2005 have been included, amounting to 1 million words. For English, an equivalent amount of articles from Health Magazine has been included. We PoS-tagged and lemmatized the English part of the corpus with TreeTagger (Schmid, 1994).

We then extended the initial corpus automatically from the 2 billion-word ukWaC (Ferraresi et al., 2008) and the 380 million-word slWaC (Ljubešić and Erjavec, 2011). We took into account all the documents that pass a document similarity threshold with respect to the core corpus that was experimentally set in Fišer et al. (2011).

2.2. Parallel corpus

2.2.1. Data

In this work, we enhance bilingual lexicon extraction from comparable corpora by applying a data-driven approach to the translation of source vectors. More precisely, we replace the external seed lexicon, used in previous work on lexicon extraction from comparable corpora, with the output of a cross-lingual WSD method (Apidianaki, 2009). The method exploits the results of a cross-lingual Word Sense Induction (CL-WSI) method that identifies word senses by clustering their translations in a parallel corpus. In the current setting, the English translations of Slovene in a parallel corpus are clustered and the obtained sense-clusters describe the senses of source words. The corpus used for sense induction is composed of the Slovene-English part of Europarl (release v6) (Koehn, 2005) and the Slovene-English part of the JRC-Acquis corpus (Steinberger et al., 2006), amounting to approximately 35M words per language.

2.2.2. Pre-processing

Prior to being used for sense induction, the training corpus is subject to several pre-processing steps, such as elimination of sentence pairs with a great difference in length, lemmatization and PoS-tagging with TreeTagger (for English) and ToTaLe (for Slovene) (Erjavec et al., 2010). Next, the corpus is word-aligned with GIZA++ (Och and Ney, 2003) and two bilingual lexicons are extracted from the alignment results, one for each translation direction (EN–SL/SL–EN).

The lexicons are cleaned by applying a set of filters, in order to retain only intersecting alignments of the same PoS. The filtered EN-SL lexicon contains entries for 6,384 nouns, 2,447 adjectives and 1,814 verbs having more than three translations in the training corpus. The translations used for clustering are the ones with a minimum frequency of 10 in the training corpus and a minimum alignment certainty of 0.01. The resulting lexicon is then used for word sense induction (cf. Section 3).

2.3. Gold standard

We evaluate the results of different experimental settings by comparing them to a gold standard lexicon, which was collected from the corpus and manually inspected. It contains 187 domain terms (nouns) that are present in the source language corpus with a minimum frequency of 50. 23 of these terms have two attested translations in the corpus (e.g. Eng. *rectum* → Slo. *danka, rektum)* while the rest have just one (e.g. Eng. *breast* → Slo. *dojka).*

3. Experimental setup

3.1. Cross-lingual sense clustering

The translations of each English content word (*w*) in the parallel corpus are clustered on the basis of source language distributional information. Each Slovene translation (*t*) of *w* is characterized by a vector built from the co-occurrences of *w* in the aligned sentences where it is translated by *t.* The vectors contain lemmas of content words that co-occur with *w* and their frequency counts. Using these vectors, pairwise similarities between the translations of *w* are calculated by a variation of the Weighted Jaccard measure (Grefenstette, 1994; Apidianaki, 2008). This measure assigns weights to the features that reflect their relevance for calculating the similarity of the vectors. The score assigned to a pair of vectors and the corresponding translations indicates their degree of similarity. Translation pairs with a score above a threshold defined locally for each *w* and dependent on the similarity scores assigned to its pairs of translations are considered as semantically related.[1]

The clustering algorithm groups Slovene translations into clusters that describe the senses of the corresponding English words. More precisely, the algorithm takes as input the list of translations of an English word, their similarity scores and the computed similarity threshold, and it outputs clusters that contain semantically related translations. Table 1 gives examples of clusters for words of different PoS with clear sense distinctions. For each English word, we provide its clusters of Slovene translations that were obtained and include a description of the sense described by each cluster. For instance, the clusters of the word *sphere*: *{krogla}* and *{sfera, področje}*, describe the two senses of *sphere* observed in the corpus: "*geometrical shape*" and "*area*". The obtained cluster inventory contains 13,352 clusters for 8,892 words. 2,585 of the words (1518 nouns, 554 verbs and 513 adjectives) have more then one cluster.

[1] The threshold is set following the method proposed in Apidianaki and He (2010).

Language	POS	Source word	Slovene sense clusters
EN-SL	Nouns	sphere	{krogla} (*geometrical shape)* {sfera, področje} (*area*)
		address	{obravnava, reševanje, obravnavanje} (*dealing with*) {naslov} (*postal address*)
		portion	{kos} (*piece*) {obrok, porcija} (*serving*) {delež} (*share*)
		figure	{številka, podatek, znesek} (*amount*) {slika} (*image*) {osebnost} (*person*)
	Verbs	seal	{tesniti} (*to be water-/airtight)* {zapreti, zapečatiti} (*to close an envelope or other container)*
		weigh	{pretehtati} (*consider possibilities)* {tehtati, stehtati} (*check weight)*
		educate	{poučiti} (*give information)* {izobraževati, izobraziti} (*give education)*
		consume	{potrošiti} (*spend money/goods)* {uživati, zaužiti} (*eat/drink*)
	Adjs	mature	{zrel, odrasel} (*adult*) {zorjen, zrel} (*ripe*)
		minor	{nepomemben} (*not very important)* {mladoleten, majhen} (*under 18 years old)*
		juvenile	{nedorasel} (not *adult/biologically mature yet)* {mladoleten, mladoletniški} (*not 18/legally adult yet)*
		remote	{odmaknjen, odročen} (*far away and not easily accessible)* {oddaljen, daljinski} (*controlled from a distance)*

Table 1: Entries from the English-Slovene sense cluster inventory.

3.2. Vector building from the comparable corpus

Context vectors in both the source and the target language are built for nouns occurring at least 50 times in the comparable corpus. As features, we use three content words to the left and to the right of the retained nouns, stopping at the sentence boundary. The position of each content word is not taken into account. Feature weights are calculated by the TF-IDF measure, where IDF weights are calculated on the whole ukWaC and slWaC. The feature weights serve to filter out the *weak* features that were shown not to be useful for the lexicon extraction task. The threshold is experimentally set at 0.01.

3.3. Vector disambiguation

3.3.1. A data-driven approach

For extracting bilingual lexicons from comparable corpora, the vectors built in the two languages must be compared. This comparison serves to quantify the similarity of the source and target language words represented by the vectors, and the highest ranked pairs are proposed as entries for the lexicon. For that, source language vectors must first be translated into the target language. In most previous work, the vectors were translated with external dictionaries: the first translation in the dictionary was used to translate all the instances of the word in the vectors irrespective of their sense, and no disambiguation was performed.

The use of external resources ensures the quality of the translations used for translating the source vectors. Moreover, the selection of the most frequent translation often results in good translations because of the skewed distribution of the translations corresponding to different senses of the words. Nevertheless, this technique limits the usability of the proposed lexicon extraction methods to languages and domains where such resources are available.

In this work, we translate the source language vectors using a data-driven cross-lingual WSD method (CL-WSD) (Apidianaki, 2009). The method exploits the sense clusters acquired from parallel corpora (see Section 3.1). This property extends the applicability of the method to languages lacking large-scale lexical resources but for which parallel corpora are available.

3.3.2. Cross-lingual WSD

The sense clusters of translations obtained during WSI represent the candidate senses of the English words in the parallel corpus. We exploit this sense inventory for disambiguating the features in the English vectors extracted from the comparable corpus. More precisely, the CL-WSD method has to select among the available clusters the one that correctly translates in Slovene the sense of the English features contained in the vectors built from the comparable corpus.

In the current setting, the selection is performed by comparing information from the context of the features to the distributional information that served to estimate the semantic similarity of the clustered translations. The context of a feature to be disambiguated corresponds to the

rest of the vector where it appears. Inside the vectors, the features are ranked and filtered according to their score (calculated as explained in Section 3.2). The retained features are considered as a bag of words. On the clusters side, the information used for disambiguation is found in the source language vectors that revealed the semantic similarity of the clustered translations.

If common features (CFs) are found between the context of a feature and just one cluster, this cluster is selected to describe the feature's sense. Otherwise, if there exist CFs with more than one cluster, then a score is assigned to each 'cluster-feature' association. This weight corresponds to the mean of the weights of the *CFs* relative to the clustered translations (weights assigned to each feature during WSI). In the following formula, CF_j is the set of *CFs* found between the cluster and the new context and N_{CF} is the number of translations T_i in the cluster characterized by a *CF*:

$$assoc\_score = \frac{\sum_{i=1}^{N_{CF}} \sum_j w(T_i, CF_j)}{N_{CF} \cdot |CF_j|}$$

The highest scored cluster is selected and assigned to the feature as a sense tag. The features are also tagged with the most frequent translation of the word in the training corpus, which sometimes already exists in the cluster selected during WSD. In Table 2, we present some examples of disambiguated vector features of different PoS. For each case, we provide the headword entry to which the vector corresponds, a feature from the vector that has been disambiguated and the context that was used for disambiguation, which consists of the other strong features found in the same vector (i.e. features with a weight above a threshold). From the candidate clusters available for the feature (column 4), the WSD method selects the most appropriate one (in boldface) to describe the feature's sense in this context. In the last column of the table, we provide the most frequent sense/translation (MF) for the feature. We observe that the MF translation may already exist in the cluster selected by the WSD method, like in the first example where *obravnava* is already in the selected cluster. The inverse, i.e. that the MF is not found in the proposed cluster, is also possible as is the case with the *zapečatiti* translation of the verb *seal.*

The disambiguation of source language features using cross-lingual sense clusters constitutes the main contribution of this work and presents several advantages. First, the method performs disambiguation by using sense descriptions derived from the data, which extends its applicability to resource-poor languages. This procedure clearly differentiates our method from previous approaches where the first translation in a dictionary – which is often the most frequent one – was selected for translating each vector feature. An additional advantage is that the sense clusters assigned to features may contain more than one translation. This property is important in this setting as it provides supplementary material for the comparison of the vectors in the target language.

3.4. Cross-lingual vector comparison

For context vectors to be comparable between languages, the same vector space has to be produced. This is done by translating the source language features to the target language. We translated the features in three ways:

1. by keeping the translation a feature was most frequently aligned to in the parallel corpus (MF);
2. by keeping the most frequent translation from the cluster assigned to the feature during disambiguation (CLMF); and
3. by using the same cluster as in the second approach, but producing features for all translations in the cluster with the same weight (CL).

The first approach is used as a baseline since instead of the sense clustering and WSD results, it just uses the "most frequent sense/alignment" heuristic. Since in the first batch of the experiments we noticed that the results of the CL approach heavily depend on the part-of-speech of the features, we divided the CL approach into three sub-approaches:

1. translate only nouns with the clusters and other features with the MF approach (CL-n);
2. translate nouns and adjectives with the clusters and verbs with the MF approach (CL-na); and
3. translate all PoS with the clusters (CL-nav).

Once the source language vectors are built, the distance between the translated source and the target-language vectors is computed by the Dice metric which has proven to be very efficient when combined with the TF-IDF weighting (Ljubešić et al., 2011). We also experiment with a minimum feature weight threshold since, during our experiments, we observed the phenomenon where discarding the weakest features from the context vectors in the source language significantly improves the results. We call this parameter the 'minimum feature weight threshold' (mfwt). By comparing the translated source vectors to the target language ones, we obtain a ranked list of candidate translations for each gold standard entry.

Headword	Feature (POS)	Context	Candidate clusters	MF alignment
infertility (n)	treatment (n)	*doctor, diabetes, health, emergency, check, ...*	- {**zdravljenje, obdelava, obravnavanje, obravnava, ravnanje**} (*treat an illness)* - {čiščenje} (*treat a person/animal)* - {raba} (*usage*)	obravnava
clot (n)	seal (v)	*block, heart, vessel, pressure, infection, ...*	- {**tesniti**} (*to be waterproof or airtight)* - {zapreti, zapečatiti} (*to close)*	zapečatiti
arrhythmia (n)	irregular (a)	*heart, abnormal, monitor, failure, risk, ...*	- {**nepravilen, nereden**} (*not regular)* - {ilegalen} (*illegal*)	nepravilen

Table 2: Examples of disambiguated vector features.

4. Evaluation and discussion of the results

4.1. Evaluation procedure

We evaluate the final result of our method, i.e. the ranked lists of translation candidates for gold standard entries by the mean reciprocal rank (MRR) which takes into account the rank of the first good translation found for each entry. Formally, MRR is defined as

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where $|Q|$ is the length of the query, i.e. the number of gold standard entries we compute translation candidates for, and $rank_i$ is the position of the first correct translation in the candidate list. Since most of the entries of our gold standard contain just one translation, we did not consider using some more advanced evaluation measure, like mean average precision.

4.2. Results & discussion

The results of our final experiment are shown in Figure 1. The x axis shows the minimum feature weight threshold (mfwt) while on the y axis the evaluation measure MRR is plotted. The phenomenon that is first observed on the graphs is the one for which we have introduced the minimum feature weight threshold parameter: the best results are obtained when discarding all features that have a TF-IDF weight score lower than 0.01. This is something we had not noticed before and we will look into this phenomenon more thoroughly in a new set of experiments, by measuring its consistency when different weight measures, distance measures, seed lexicons, language pairs and comparable corpora are used.

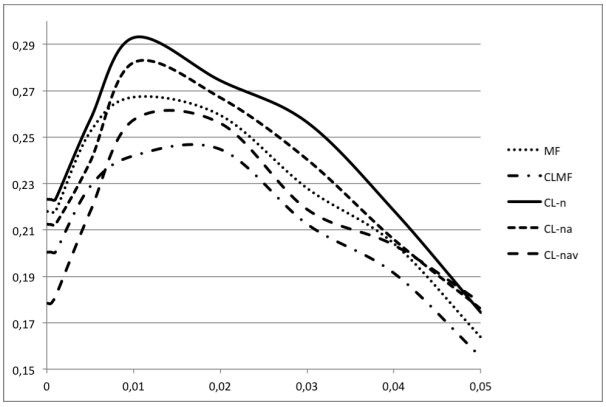


Figure 1: Evaluation of the different translation approaches regarding the minimum feature weight threshold.

Overall, the worst results are obtained when using the CLMF approach, i.e. using only the most frequent translation from the cluster chosen through the WSD procedure. A possible reason for this is the fact that alignment frequencies used for finding the most frequent alignment in the cluster were calculated on a corpus of a different domain than our comparable corpus (Europarl vs. health corpus). The baseline that always uses the most frequent translation of the feature from the parallel corpus, without any sense clustering and WSD, achieves a medium result, being outperformed by the CL-n and the CL-na approach but outperforming the CL-nav approach.

The CL sub-approaches yield somewhat expected results. The biggest gain is obtained from clustering and WSD information calculated on nouns, nouns and adjectives scored second and the lowest results are obtained when verbs are added to the mix. This is probably due to the fact that the verbal clusters are noisier than the nominal and adjectival ones. We intend to explore this issue in future work.

Since our gold standard is quite small, we checked the statistical significance of the difference in the results of the baseline MF approach and the winning CL-n approach. We used the approximate randomization procedure with R = 1000 (i.e. 1000 random assignments were done without replacement of the two sets of results). The resulting *p-value* is 0.091, which is higher than the commonly used 0.05 threshold. These results show that, in our future experiments, we will need a larger gold standard to draw safer conclusions on the statistical significance of the results. However, since the p-value is below 0.1 and is accompanied by a consistent increase in performance throughout a large number of experiments, we are rather confident that this increase is not the result of random variation.

The main conclusions that can be drawn from the results demonstrated here are that:

- extending the feature set with multiple translations obtained by sense clustering and word sense disambiguation of features is beneficial to the lexicon extraction procedure;
- the most valuable information obtained from the clustering and WSD approach comes from nouns;
- using just the most frequent translation inside the cluster selected during WSD does not yield good results, and
- further investigation of the phenomenon where discarding the weak features improves the result is needed.

5. Conclusions and future work

We presented an approach that allows to use lexico-semantic knowledge acquired from parallel corpora in order to improve the extraction of translation equivalents from comparable corpora. A parallel corpus served as the source of the seed dictionary, so that the translation of features in context vectors no longer relies on an external knowledge source. In addition, the seed dictionary was enhanced with clusters of translation variants obtained from the parallel corpus in an unsupervised way. The cross-lingual clusters were used to disambiguate the features in the context vectors, thus reducing noise, and allowed for a more accurate comparison of source and target vectors. Furthermore, the tagging of the vector features with clusters during disambiguation increased the translation information available for each feature and, therefore, facilitated the comparison of context vectors across languages.

The results show that lexico-semantic knowledge derived from a parallel corpus can help to circumvent the need for an external seed dictionary, traditionally

considered as a prerequisite for bilingual lexicon extraction from parallel corpora. Moreover, it is clear that disambiguating the vectors improves the quality of the extracted lexicons and manages to beat the simpler, but yet powerful, most frequent sense/alignment heuristic.

These encouraging results pave the way towards pure data-driven methods for bilingual lexicon extraction from comparable corpora. This knowledge-light approach can be applied to languages and domains that do not dispose of large-scale seed dictionaries but for which parallel corpora are available. Moreover, the use of a data-driven cross-lingual WSD method, such as the one proposed in this paper, can contribute to obtain less noisy translated vectors, which is important especially when lexicon extraction is performed from general language comparable corpora.

The experiments carried out till now focus on a health comparable corpus. Although this is not a very specialized corpus but a rather popular one, cases of true polysemy are still less frequent than in a general corpus. We would thus like to extend this work by applying the method to a more general comparable corpus, for instance a corpus built from Wikipedia texts. We expect that the effect of applying the WSD method on a general corpus will be highly beneficial, as ambiguity problems will be more prevalent.

Another avenue that we want to explore is to use second order co-occurrences for disambiguation. For the moment, the context used to disambiguate vector features consists of the other features that appear in the same vector. However, these features are direct co-occurrences of the headword, which does not necessarily mean that the features themselves co-occur with each other in the corpus. We consider that it would be preferable to replace this context with the co-occurrences of the features in the corpus for disambiguation, which would correspond to the second order co-occurrences of the English nouns, and investigate the effect of using this type of context on lexicon extraction. Last but not least, we would like to apply the method to the opposite direction (i.e. from Slovene to English) and compare the results obtained in both directions.

6. References

M. Apidianaki. 2008. Translation-oriented Word Sense Induction based on Parallel Corpora. *Proc. of LREC*, Marrakech, Morocco.

M. Apidianaki, Y. He. 2010. An algorithm for cross-lingual sense clustering tested in a MT evaluation setting. *Proc. of the 7th IWSLT,* 219–226, Paris, France.

M. Apidianaki. 2009. Data-driven Semantic Analysis for Multilingual WSD and Lexical Selection in Translation. *Proc. of the 12th EACL-09,* 77–85, Athens, Greece.

Š. Arhar, V. Gorjanc, S. Krek 2007. FidaPLUS corpus of Slovenian: the new generation of the Slovenian reference corpus: its design and tools, *Proc. of the Corpus Linguistics conference*, Birmingham, UK.

T. Brants. 2000. Tnt a statistical part-of-speech tagger. In *Proceedings of the 6th ANLP,* Seattle, WA.

H. Déjean, E. Gaussier, J. Renders, F. Sadat. 2005. Automatic processing of multilingual medical terminology: applications to thesaurus enrichment and cross-language information retrieval. *Artif. Intell. Med.,* 33(2):111–124.

T. Erjavec, D. Fišer, S. Krek, N. Ledinek. 2010. The JOS linguistically tagged corpus of Slovene. *Proc. of 7th LREC,* Valletta, Malta.

A. Ferraresi, E. Zanchetta, M. Baroni, S. Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. *Proc. of the 4th WAC: Can we beat Google,* 47–54.

D. Fišer, N. Ljubešić. 2011. Bilingual lexicon extraction from comparable corpora for closely related languages. *Proc. of RANLP,* 125–131, Hissar, Bulgaria.

D. Fišer, N. Ljubešić, Š. Vintar, S. Pollak. 2011. Building and using comparable corpora for domain-specific bilingual lexicon extraction. *Proc. of the 4th BUCC: Comparable Corpora and the Web,* 19–26, Portland, Oregon, USA.

P. Fung. 1998. Machine translation and the information soup, third conference of the association for machine translation in the Americas, *AMTA, Vol. 1529 of LNCS.* Springer.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Publishers, Norwell, MA.

H. Kaji. 2003. Word sense acquisition from bilingual comparable corpora. *Proc. of HLT-NAACL.*

P. Koehn, K. Knight. 2002. Learning a translation lexicon from monolingual corpora. *Proc. of ACL Workshop on Unsupervised Lexical Acquisition,* 9–16.

P. Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *Proc. of MT Summit X,* 79–86, Phuket, Thailand.

N. Ljubešić, T. Erjavec. 2011. hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. *TSD Vol. 6836 of LNCS,* 395–402. Springer.

N. Ljubešić, D. Fišer, Š. Vintar, S. Pollak. 2011. Bilingual lexicon extraction from comparable corpora: A comparative study. *Proc. of WOLER*, Ljubljana, Slovenia August 1-5, 2011.

E. Marsi, E. Krahmer. 2010. Automatic analysis of semantic similarity in comparable text through syntactic tree matching. *Proc. of COLING,* 752–760, Beijing, China, August.

F.J. Och, H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics,* 29(1):19–51.

P. Gamallo Otero. 2007. Learning bilingual lexicons from comparable English and Spanish corpora. *Proc. of MT Summit XI,* 191–198.

R. Rapp. 1999. Automatic identification of word translations from unrelated English and German corpora. *Proc. of the 37th ACL,* 519–526, College Park, Maryland, USA.

Z. Saralegi, I. San Vicente, A. Gurrutxaga. 2008. Automatic extraction of bilingual terms from comparable corpora in a popular science domain. *Proc. of the 1st Building and Using Comparable Corpora (BUCC) workshop,* Marrakech, Morocco.

H. Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proc. of the International Conference on New Methods in Language Processing,* 44–49, Manchester, UK.

L. Shao, H. Tou Ng. 2004. Mining new word translations from comparable corpora. *Proc. of COLING,* 618–624, Geneva, Switzerland, Aug 23–Aug 27.

R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufiş. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proc. of the 5th LREC,* 2142–2147.

K. Yu, J. Tsujii. 2009. Extracting bilingual dictionary from comparable corpora with dependency heterogeneity. *Proc. of ACL,* 121–124, Boulder, Colorado, USA.

Izdelava korpusa Gigafida in njegovega spletnega vmesnika

Špela Arhar Holdt*, Iztok Kosem, Nataša Logar Berginc*****

* Trojina, zavod za uporabno slovenistiko
Partizanska cesta 5, SI-4220 Škofja Loka
spela.arhar@trojina.si
** Trojina, zavod za uporabno slovenistiko
Partizanska cesta 5, SI-4220 Škofja Loka
iztok.kosem@trojina.si
*** Fakulteta za družbene vede, Univerza v Ljubljani
Kardeljeva ploščad 5, SI-1000 Ljubljana
natasa.logar@fdv.uni-lj.si

Povzetek

V projektu *Sporazumevanje v slovenskem jeziku* smo izdelali nov referenčni korpus slovenščine Gigafida. Gre za javno dostopni lematizirani in oblikoskladenjsko označeni korpus pisnih besedil. Vzporedno z izdelavo korpusa je potekala priprava novih korpusnih orodij z intuitivno zasnovanim spletnim vmesnikom, ki bi tudi nejezikoslovnim uporabnikom omogočal enostavno korpusno iskanje. V prispevku tako prikazujemo izhodišča in rešitve v zvezi z dvojim: (a) z zbiranjem gradiva in vsebino korpusa ter njegovo končno podobo glede na deleže besed po taksonomskih kategorijah in (b) z novostmi na ravni korpusnega vmesnika, kot so: uporabniško prijazne iskalne možnosti, samodejna lematizacija iskalnega pogoja, uvedba podatkovnih filtrov itd.

The development of the Gigafida corpus and its online interface

The paper describes the building of a new reference corpus of Slovene, called Gigafida, as part of the *Communication in Slovene* project. The Gigafida corpus is freely available corpus of written texts, which has been lemmatized and POS-tagged. In addition to building the corpus, we have developed new corpus tools with intuitive online interface, which enables easy corpus searches to language users. The paper describes the design of the corpus, and decisions made in terms of a) collection of texts, corpus contents, and its final structure according to taxonomic categories, and b) new features in interface functionality such as: user-friendly search options, automatic lemmatization of query, introduction of filters etc.

1. Uvod

V sklopu projekta *Sporazumevanje v slovenskem jeziku* (http://www.slovenscina.eu/; SSJ)[1] je med drugim potekala izdelava oz. posodobitev in nadgradnja več besedilnih korpusov za slovenščino. Nastalo jih je šest, med njimi korpusi pisne slovenščine Gigafida, KRES, ccGigafida in ccKRES.[2] V prvem delu prispevka se bomo posvetili predvsem zbiranju gradiva in vsebini referenčnega korpusa Gigafida, ki je uporabnikom od leta 2011 v različici beta prosto dostopen na spletu (http://demo.gigafida.net/); prosta dostopnost v istem konkordančniku in spletnem vmesniku bo kmalu veljala tudi za uravnoteženi 100-milijonski podkorpus KRES (več v Logar Berginc et al., 2012), medtem ko sta drugi dve izvedenki, ccGigafida in ccKRES, prosto dostopni kot podatkovna baza za prenos pod licenco Creative Commons: "priznanje avtorstva" + "nekomercialno" (več o cc-korpusih v Erjavec, Logar Berginc, 2012).

Izhajajoč iz dejstva, da bo novi korpus v marsičem nadaljevanje svojih dveh predhodnikov, korpusov FIDA in FidaPLUS (konec koncev bo vseboval vsa njuna besedila), smo si v projektu zastavili dva glavna cilja: na eni strani izgradnjo javno in prosto dostopnega lematiziranega ter oblikoskladenjsko označenega pisnega korpusa sodobne slovenščine z obsegom milijarde besed in na drugi strani pripravo zmogljivih prosto dostopnih korpusnih orodij, ki bodo omogočala enostavno uporabo korpusa tako strokovni kot širši zainteresirani javnosti. V nadaljevanju predstavljamo del rezultatov teh dveh projektnih aktivnosti in nekatere odločitve, ki smo jih pri delu sprejeli.

2. Gigafida: zbiranje gradiva, vsebina

Ob pripravi na izdelavo novega korpusa, ki smo ga nekaj časa imenovali Korpus SSJ (prim. Logar Berginc, Šuster, 2009), smo se zavezali k čim celovitejši dokumentiranosti zbiranja besedil in popisu končne vsebine. Tako lahko o zaključeni Gigafidi predstavimo ne le običajne ključne sumarne podatke, ampak tudi vse glavne sezname in merila, ki so nas vodili pri zbiranju; dalje popoln seznam del, ki so v korpusu; seznam založnikov, ki so ta besedila izdali, oz. spletnih strani, s katerih smo jih pridobili; ter do besede natančen obseg vseh vključenih besedil, skupaj s taksonomsko kategorijo, ki smo jim jo pripisali (zlasti podrobno smo tak prikaz pripravili za korpus KRES). V Gigafidi je poleg velike večine besedil iz FidePLUS gradivo, ki smo ga zbrali do 29. maja 2010 (tisk) oz. do 11. aprila 2012 (internet). Končno število besed v Gigafidi je 1.187.002.502, kar

[1] Operacijo, v okviru katere je nastala raziskava, delno financira Evropska unija iz Evropskega socialnega sklada ter Ministrstvo za izobraževanje, znanost, kulturo in šport Republike Slovenije. Operacija se izvaja v okviru Operativnega programa razvoja človeških virov za obdobje 2007–2013, razvojne prioritete: razvoj človeških virov in vseživljenjskega učenja; prednostne usmeritve: izboljšanje kakovosti in učinkovitosti sistemov izobraževanja in usposabljanja 2007–2013.

[2] Preostala dva korpusa sta še korpus govorjene slovenščine GOS (http://www.korpus-gos.net/; Verdonik, Zwitter Vitez, 2011) in korpus Šolar (Rozman et al., 2010).

pomeni, da je novi referenčni korpus skoraj enkrat večji od predhodne različice.

V nadaljevanju tega razdelka osvetljujemo izhodišča in rezultate izdelave Gigafide z naslednjih nekaj vidikov: ustavili se bomo pri seznamih želenih in pridobljenih knjig ter periodike, dalje pri izboru spletnih strani in načinu pridobitve besedil z njih, nato pa bomo pojasnili še taksonomijo ter pokazali končni delež besed po njenih kategorijah. Sledila bo primerjava dveh podkorpusov Gigafide, in sicer podkorpusa besedil iz obdobja 1990–2006 (pretežno besedila iz FidePLUS) in podkorpusa besedil iz obdobja 2007–2011 (na novo zbrana besedila), pri čemer smo primerjali vrhnji del frekvenčnega seznama lem (absolutna in relativna frekvenca) ter vrhnji del seznama ključnih besed.

2.1. Leposlovje, stvarna besedila

Pri zbiranju knjižnih del so bila naša glavna izhodišča naslednja: pregledali smo Cobissove sezname najbolj izposojanih in največkrat rezerviranih knjig ter sezname slovenskih avtorjev, katerih dela so največkrat izposojana in so zato upravičeni do knjižničnega nadomestila. Na seznam želenih knjig smo dali dela, ki so v zadnjih letih prejela katero od knjižnih nagrad. Po evidenci AJPES-a smo v kombinaciji s podatki iz Cobissa zbrali založbe, ki so bile v zadnjih letih dejavne pri izdajanju knjig, pridružili pa smo jim še tiste, ki so se predstavljale na Knjižnem sejmu 2008 v Ljubljani. Poleg tega smo za besedila zopet prosili vse besedilodajalce, ki so sodelovali že pri gradnji korpusa FidaPLUS.

V celoti je tako v korpusu Gigafida 534 različnih leposlovnih del (če navedeno le tri, ki so tudi med najbolj izposojanimi: D. Brown: *Da Vincijeva šifra*, V. Pečjak: *Drejček in trije Marsovčki*, N. Sparks: *Viharna noč*), ki jih je izdalo 55 različnih založnikov. Med najbolj izposojanimi slovenskimi avtorji so v Gigafidi npr. dela P. Suhadolčana, I. Sivca, B. Novaka, J. Vidmar, M. Podgoršek, T. Pavčka, K. Koviča, R. Berni itd.

Oznako stvarno besedilo ima v korpusu 1.082 različnih del, ki jih je izdalo 89 različnih založnikov, med njimi so naslovi *Aromaterapija*, *Astronomija*, *Bivalni vrt*, *Branje z dlani*, *Do zdravja z zdravo hrano*, *Geografija za 7. razred*, *Mali družinski katekizem*, *Najboljši recepti* itd.

2.2. Časopisi, revije

Že pri gradnji prvega referenčnega korpusa slovenščine, korpusa FIDA, so sestavljalci pri (i)zbiranju periodičnega gradiva izhajali iz podatkov raziskave branosti tiskanih medijev, ki jo je pred dobrim desetletjem izvajala Mediana. Pri najnovejšem zbiranju je izhodiščni seznam želenih časopisov in revij nastal na podlagi podatkov Nacionalne raziskave branosti (NRB, http://www.nrb.info/). Najbolj natančno smo si ogledali lestvico, ki zajema valutno obdobje 2. polletje 2009 in 1. polletje 2010 (NRB 2010). Na njej je 53 naslovov časopisov in 93 naslovov revij. Od teh smo jih za korpus uspeli dobiti 20 oz. 54, npr. *Žurnal*, *Nedeljski dnevnik*, *Dobro jutro*, *Slovenske novice*, *Kmečki glas*, *Delo*, *Večer*, *Dnevnik*, *Družina, Finance*; *Lady*, *Ognjišče*, *Motorevija*, *Zdravje*, *Jana*, *Cosmopolitan*, *Rože & vrt*, *Avto magazin*, *Smrklja*, *Mladina* itd., nismo pa npr. dobili *Goriške*, *Moje Gorenjske*, *Mariborskega utripa*; *Razvedrila*, *Vzajemnosti*, *Lovca* ipd. Kljub približno polovičnemu uspehu glede na NRB 2010 pa je treba poudariti, da je v Gigafidi še kar nekaj naslovov periodike, ki se sicer na lestvico NRB 2010 niso uvrstili – natančneje: gre za še dodatnih 31 časopisov (od tega dva časopisa Slovencev v Italiji: *Novi Matajur* in *Novi glas*) ter 73 revij.

2.3. Spletna besedila

Glede na predhodni korpus FidaPLUS (Arhar Holdt, Gorjanc, 2007) smo znatno povečali obseg besedil s spleta (v FidiPLUS je bil tak le 1,24-odstotni delež) – oz. bolje rečeno: k vključitvi besedil s spleta smo tokrat pristopili drugače. Dogovorili smo se, da bo končni delež besed s spleta v korpusu vsaj 10-odstotni, nato pa smo izbrali 11 novičarskih spletnih strani (med njimi *24ur.com*, *siol.net*, *rtvslo.si*; pri tem smo upoštevali tudi njihovo obiskanost, zlasti merjenje MOSS, http://www.moss-soz.si/), 28 strani največjih slovenskih podjetij (*krka.si*, *mobitel.si*, *mercator.si* itd.) ter 62 strani pomembnih državnih, izobraževalnih, raziskovalnih in kulturnih ustanov (*gov.si*, *sazu.si*, *drama.si* itd.), s katerih smo ta besedila želeli pridobiti. Sodelavec Miha Grčar (Inštitut Jožef Stefan) je razvil spletnega pajka, ki je z omenjenega vnaprej določenega seznama začetnih naslovov dnevno, mesečno ali enkratno (odvisno od dinamičnosti spletnega mesta) zbiral tekstovne dokumente, iz katerih so bila nato odstranjena spremna in vnaprej pripravljena besedila ter dvojniki in približni dvojniki.[3]

Detekcijo (približnih) dvojnikov je M. Grčar izvedel na naslednji način: sprva je bil uporabljen algoritem, ki temelji na podobnostni razpršilni funkciji Simhash (občutljivost: $k = 3$ biti ali manj), ki pa ni bila zadostna. Ker je tekstovne segmente določil že odstranjevalnik spremnih in vnaprej pripravljenih besedil, je bil na koncu cevovoda vsak tak segment po enostavni normalizaciji s postopkom MD5 (http://en.wikipedia.org/wiki/MD5) pretvorjen v razpršilno kodo. Kode se hranijo v razpršilni tabeli, kar omogoča preverbo, ali smo neki segment že zapisali v korpusa ali ne. Odstranitev podvojenih segmentov je prvotno število znakov zmanjšala za več kot polovico (več v Logar Berginc et al., 2012). V Gigafido je na koncu s spleta prišlo več kot 180 milijonov besed oz. skoraj 16 % korpusa.

2.4. Taksonomija z deleži besed

Medtem ko je bila taksonomija FidePLUS tridelna (prenosnik, zvrst, lektoriranost) in tudi dalje notranje dokaj podrobno členjena, smo taksonomijo korpusa Gigafida poenostavili v enodelno ter členjeno do tretje podravnine (gl. Tabela 1, prvi stolpec).

Izhajali smo iz naslednjega besedilnozvrstnega izhodišča, seveda primerjalno s FidoPLUS: (a) členitev na umetnostna in neumetnostna besedila je samodejno mogoča le pri knjigah; ter (b) izraz "neumetnostni" z izločitvijo periodičnega postane preširok in nepoveden. Posledično smo knjige ločili na leposlovje in stvarna besedila (tj. besedila z nefikcijsko vsebino) ter pri prvem opustili delitev na pesniško, prozno in dramsko, saj med drugim ni bilo niti približno upravičeno pričakovati, da bomo poezije dobili v zadostnem obsegu, da bi bil obstoj te kategorije upravičen. Z mislijo na izdelavo 100-milijonskega uravnoteženega podkorpusa smo namreč kot spodnjo mejo za obstoj ločene kategorije določili 5 %

[3] Detekcijo dvojnikov in približnih dvojnikov smo izvedli le pri internetnih besedilih, ne pa tudi pri tiskanih.

vseh besed, se pravi, da bi – če bi želeli ohraniti taksonomsko kategorijo poezije – morali zbrati vsaj 5 milijonov besed iz te besedilne vrste.[4]

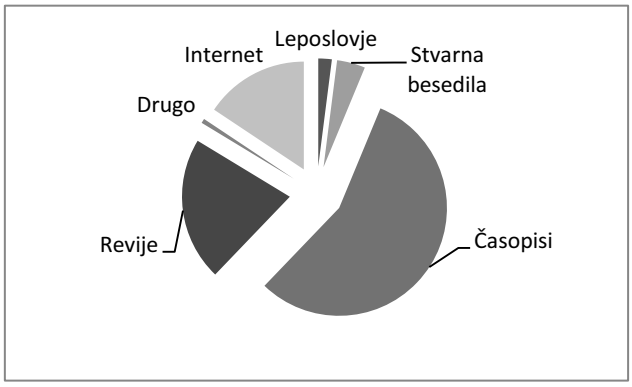
Hkrati smo ocenili členitev periodike, npr. revijalnega tiska na tedensko, štirinajstdnevno, mesečno, redkeje kot na mesec in občasno, ki je veljala pri korpusih FIDA in FidaPLUS, za preveč podrobno, pa tudi raziskave slovenskega poročevalstva je kot stilotvorno ali jezikovnorazlikovalno relevantne (še) niso potrdile (Kalin Golob, 2003), zato smo ločili le med časopisi in revijami.

Za poenostavitev taksonomije smo se odločili tudi na podlagi podatkov o načinu podkorpusnega iskanja po FidiPLUS; po statistikah iz novembra 2008 so bile npr. podkategorije pri revijah in časopisih glede na pogostost izhajanja izbrane v manj kot enem odstotku razširjenih iskanj.

Edina kategorija, ki tako še ostane, je kategorija Drugo. Zanjo smo na novo zbrali podnapise filmov, nadaljevank in dokumentarnih oddaj ter t. i. postprodukcijska besedila, ki smo jih vse dobili na RTV Slovenija, iz obeh predhodnih korpusov pa smo sem umestili še zapise sej Državnega zbora RS in besedila z več manjkajočimi bibliografskimi podatki, pri katerih nismo mogli določiti druge kategorije, tudi t. i. besedilni drobiž (skupaj ta, "neznani" del v korpus prinaša 2,6 milijona besed).

Taksonomija	**Delež v %**
tisk	84,35
knjižno	6,26
leposlovje	2,02
stvarna besedila	4,24
periodično	77,42
časopisi	55,91
revije	21,51
drugo	0,67
internet	15,65
Skupaj	**100,00**

Tabela 1: Delež besed po taksonomiji v Gigafidi (v %).



Slika 1: Delež besed po taksonomiji v Gigafidi.

Tabela 1 in Slika 1 kažeta rezultat našega dela še z vidika števila oz. deleža besed: v Gigafidi imajo več kot polovični delež časopisna besedila, tem nato sledijo z 21 % revije. Periodika ima v Gigafidi skupaj 77-odstotni obseg oz. vanjo prinaša 918.936.054 besed. Knjige imajo v Gigafidi 6-odstotni delež ali 74.356.531 besed, od tega sta 2 % besed iz leposlovja, 4 % besed pa prihajajo iz stvarnih besedil. Kategorija Drugo je majhna, manj kot enoodstotna, vsebuje pa 7.951.450 besed. Celotni tisk ima 84-odstotni delež, preostalih skoraj 16 % pripada internetnim besedilom.

Glede na podobno, po obsegu prvenstveno vlogo periodike v FidiPLUS in glede na to, da je tudi sicer dnevna, tedenska ali mesečna produkcija časopisov ter revij veliko večja (merjeno v številu besed) kot produkcija knjig, so razmerja pričakovana, poleg tega so tudi rezultat večje zadržanosti pred brezplačno oddajo elektronskega izvoda, na katero smo naleteli pri knjižnih založbah. Bolj uravnotežena razmerja med besedilnimi zvrstmi smo zato že predhodno načrtovali in jih tudi uresničili v korpusu KRES (Logar Berginc et al., 2012).

2.5. Staro proti novemu

Polovico korpusa Gigafida (52 %) predstavljajo besedila korpusa FidaPLUS in poraja se vprašanje, ali Gigafida v leksikalnem smislu prinaša kaj bistveno novega. Zaradi različnega načina oblikoskladenjskega označevanja – za označevanje Gigafide smo uporabili statistični označevalnik Obeliks (Krek, Grčar, Dobrovoljc, 2012), medtem ko je FidaPLUS označena z označevalnikom, ki deluje na podlagi pravil – bi bile primerjave besednovrstnih statističnih podatkov obeh korpusov dokaj neustrezne. Poleg razlik v sami metodologiji označevalnikov namreč obstajajo tudi razlike v oznakah, ki jih uporabljata. Kot primer lahko navedemo pogostost veznika *kot*, ki je v korpusu Gigafida 17. najpogostejša lema (4.556,8 pojavitev na milijon besed), medtem ko je v FidiPLUS šele na 227. mestu (338,3 pojavitve na milijon besed) – do takšne razlike pride predvsem zato, ker ima označevalnik, ki deluje na podlagi pravil, tri besednovrstne oznake za *kot* (samostalnik, predlog in veznik), statistični označevalnik pa samo dve (veznik in samostalnik). Zaradi takšnih razlik med označevalnikoma lahko torej kvečjemu primerjamo metodi oblikoskladenjskega označevanja pri obeh korpusih, ne pa tudi leksike.

Zato smo ubrali drugačno pot in Gigafido razdelili na dva podkorpusa: podkorpus 1990–2006 (682.902.105 besed), v katerem je velika večina besedil iz FidePLUS,[5] in podkorpus 2007–2011 (504.100.397 besed), v katerem so samo na novo dodana besedila. Nato smo opravili dve primerjavi: med 25 najpogostejšimi lemami v podkorpusu 1990–2006 in v podkorpusu 2007–2011 ter 30 ključnimi besedami[6] v obeh podkorpusih (Tabela 3).

Primerjava 25 najpogostejših lem in njihovih relativnih frekvenc (Tabela 2) na prvi pogled ne kaže večjih razlik med podkorpusoma, vendar pa lahko pri določenih lemah v podkorpusu 2007–2011 opazimo občutno povečanje v frekvenci na milijon besed (npr. zaimka *ta* in *se*, veznika

[4] V FidoPLUS so pesniška besedila prinesla 366.215 besed, dramska 480.957 besed, prozna 20.178.021 besed, dodatno pa nadkategorija "umetnostna" še 543.750 besed (Erjavec, 2008).

[5] Približno 10 % besedil oz. besed tega podkorpusa je bilo pridobljenih pri novem zbiranju in jih v FidiPLUS še ni.

[6] Seznam ključnih besed, torej besed, ki se v enem (pod)korpusu pojavljajo precej bolj pogosto kot v drugem, smo izdelali v orodju Sketch Engine s funkcijo *Keywords* (Ključne besede) v zavihku *Word list* (Seznami).

da ter *pa*, členek *ne*), pri drugih pa opazno zmanjšanje (npr. zaimek *on*).

1990–2006		2007–2011	
Lema	Frekv. na milijon besed	Lema	Frekv. na milijon besed
biti	73.280,9	biti	73.493,1
v	26.388,5	v	26.167,8
in	25.629,3	in	25.432,9
na	15.683,6	se	16.194,2
se	15.613,4	na	16.020,1
z	13.336,8	za	13.299,6
za	12.889,1	z	13.143,9
da	11.595,0	da	13.116,7
ki	10.068,3	ta	11.095,8
on	9.981,3	ki	10.408,2
ta	9.388,2	pa	9.872,1
pa	9.097,4	on	9.469,3
tudi	6.902,0	tudi	7.022,7
ne	5.876,3	ne	6.787,2
po	5.164,9	po	5.093,5
še	4.725,5	še	4.809,6
kot	4.462,4	kot	4.684,7
leto	3.933,9	leto	3.985,9
iz	3.631,8	imeti	3.834,9
imeti	3.458,9	jaz	3.590,0
pri	3.458,8	o	3.529,8
ves	3.407,2	ves	3.455,1
od	3.339,5	iz	3.309,1
o	3.274,4	od	3.263,6
do	3.129,1	pri	3.246,5

Tabela 2: 25 najpogostejših lem v podkorpusih Gigafide 1990–2006 in 2007–2011.

Ključne besede 1990–2006	**Ključne besede 2007–2011**
tolar	EUR
SIT	evro
LJUBLJANA	člen
ponovitev	odstavek
murski	NP
marka	Pahor
TV	določba
dnevnik	organ
poročilo	kriza
VPS	hvala
vreme	ja
magazin	sgam
serija	postopek
vestnik	odločba
val	javen
tel.	št.
oglas	SD
kronika	zakon
kmetijski	nadaljevanje
toda	sodišče
nan.	Janša
novomeški	amandma
slika	NLB
amer.	RS
am.	tožen
dolenjski	pooblaščenec
show	tukaj
novica	praven
milijarda	naveden
nad.	podlaga

Tabela 3: 30 ključnih besed podkorpusov Gigafide 1990–2006 in 2007–2011.

Precej bolj pomenljive razlike med podkorpusoma pokaže primerjava ključnih besed oz. lem (Tabela 3). Ker smo pri izdelavi ključnih besed primerjali oba podkorpusa – to dejansko pomeni, da bodo ključne besede za en podkorpus tiste, ki so v drugem podkorpusu redkejše – sta seveda seznama v vrhnjem delu povsem različna.

Na prvih mestih ključnih besed obeh podkorpusov sta poimenovanji oz. kratici za staro in novo slovensko valuto, kar je lep odraz (zunaj)jezikovne spremembe, ki se je zgodila prav v prelomnem letu, ki smo ga izbrali tudi za ločitev obeh podkorpusov; sem sodi tudi *marka*, ki je v podkorpusu 2007–2009 izredno redka, pretežno na denarni svet pa je vezana tudi *milijarda*, ki ji tipično sledijo *dolarji*, *evri*, *tolarji* itd.

Podobnosti med obema podkorpusoma pa se tu končajo. Pri podkorpusu 1990–2006 je namreč na vrhu seznama ključnih besed veliko lem, ki so značilno vezane na časopise in revije, natančneje na televizijske ter radijske sporede, v katerih se nenehno ponavljajo izrazi kot *ponovitev*, *VPS*, *serija*, *am.*(*eriški*), *nan.*(*izanka*) in/ali naslovi (delov) oddaj, programov, rubrik, časopisov ipd. tipa *TV-dnevnik*, *Poročila*, *Vreme*, (*Oprah*/*Cosby*/*Muppet*) *show*, (*Gospodarski*) *vestnik*, *Val* (*202*), *Dolenjski* (*list*) itd. V tem okviru še najbolj izstopata pridevnika *kmetijski* in *novomeški*. Hiter vpogled v okolico pokaže, da so bile v tem času v časopisih in revijah aktualne teme v zvezi s *kmetijskimi zemljišči*, *Kmetijsko svetovalno službo*, *kmetijskim ministrstvom* in *ministrom* itd. ter *novomeško občino*, *Krko*, *porodnišnico*, *županom*, *košarkarji* ipd.

Po drugi strani med ključnimi besedami podkorpusa 2007–2011 prevladujejo leme, vezane na pravno-upravno vsebino: *člen*, *odstavek*, *določba* itd. (prim. Erjavec, Logar Berginc, 2012); zgolj v manjšini so tu še leme, povezane z aktualno politiko in gospodarstvom (*Pahor*, *Janša*, *SD*, *kriza*, *NLB*). Izstopajo *NP*, ki je kratica z več pomeni, npr. *nepovezani poslanci*, *notranja politika*, *ni podatka* ipd. in v tem delu Gigafide skoraj v celoti prihaja iz Dnevnika, dalje *sgam*, ki je ime sklada, *hvala*, ki v veliki večini prihaja iz internetnih besedil, natančneje s strani *dz-rs.si* (tudi tu prim. Erjavec, Logar Berginc, 2012), in *ja*, katere glavni vir je prav tako internetni del korpusa. Izpostaviti velja še dve lemi: veznik *toda* v podkorpusu 1990–2006, ki je glede na splošno pogostost funkcijskih besed nekoliko presenetljiva ključna beseda, in prislov *tukaj*, ki ga v podkorpusu 2007–2011 spet "krepijo" internetna besedila.

Nakazali smo le nekaj analiz, s katerimi bi lahko izvedeli več o podobnostih in razlikah med korpusom Gigafida in njegovim predhodnikom, korpusom FidaPLUS. Za podrobnejše razumevanje odnosa med korpusoma pa bo potrebna obsežnejša študija, v zvezi z Gigafido vsekakor dopolnjena tudi s primerjavami s tujimi referenčnimi korpusi.

3. Spletni vmesnik za dostop do korpusov

Že predhodni referenčni korpus FidaPLUS je bil uporabnikom prosto dostopen na spletu (http://www.fidaplus.net), in sicer skupaj s Konkordančnikom ASP32, ki ga je razvilo podjetje Amebis, d. o. o., Kamnik. Ker se je predvidevalo, da bo vsak uporabnik za delo s korpusom ustrezno strokovno izobražen, pa tudi zato, ker v predhodnih projektih to časovno-finančno preprosto ni bilo izvedljivo, uporabniška prijaznost korpusnega vmesnika ni bila med razvojnimi prioritetami. Glavnina pozornosti je bila usmerjena k razvoju iskalnih postopkov, ki na različne načine upoštevajo raznovrstne korpusne oznake.

Z rastjo količine gradiva, zajetega v besedilne korpuse, in novimi vrstami korpusnih podatkov oz. novimi možnostmi obdelave slednjih se je pojavila potreba po celoviti prenovi konkordančnika za slovenske korpuse. Ta prenova je ponudila možnost, da vmesniški del konkordančnika zasnujemo povsem na novo – z upoštevanjem tujih dobrih praks, ugotovitev korpusnih jezikoslovcev ter nenazadnje mnenj in želja dosedanjih korpusnih uporabnikov.

3.1. FidaPLUS: uporaba, uporabnik

Prvi korak pri snovanju novega korpusnega vmesnika je bil izvedba uporabniške evalvacije korpusa FidaPLUS. Rezultati evalvacije so na voljo na spletu (Arhar Holdt, 2010), na tem mestu zato le povzemamo bistveno.

Velika večina uporabnikov korpusa FidaPLUS se študijsko ali poklicno ukvarja z jezikom. Korpus se uporablja primarno kot pripomoček pri lektoriranju, prevajanju in pisanju besedil in pri pripravi jezikoslovnih raziskav. Večina uporabnikov se je dela s korpusom naučila samih, tipično pa ga uporabljajo nekajkrat tedensko do nekajkrat mesečno.

Obenem se je pokazalo, da uporabniki številnih programskih možnosti ne uporabljajo oz. zanje sploh še niso slišali. Pri vsem trudu, vloženem v označevanje korpusnih besedil, nas je resnično presenetilo dejstvo, da več kot četrtina vprašanih ni vedela za možnost iskanja z uporabo besedne leme, dobra tretjina pa ne za možnost iskanja s pomočjo oblikoskladenjskih oznak. Rezultati vprašalnika so pokazali tudi, da velik del vprašanih – tudi tistih, ki korpus redno uporabljajo – v resnici nima pravega znanja za učinkovito ter ustrezno izrabo možnosti, ki jih ta vir ponuja.

Ugotovili smo torej: če želimo omogočiti (redno in napredno) širšo rabo referenčnega korpusa, je v prvi vrsti nujno do skrajnosti poenostaviti iskalne postopke, pregledovanje in nadaljnjo obdelavo korpusnih podatkov pa zasnovati tako, da bo uporabnikom intuitivna in enostavno razumljiva. Cilj je izčiščen, premišljeno strukturiran vmesnik, ki se v osnovnih funkcionalnostih približuje drugim programom, s katerimi se uporabniki srečujejo na vsakodnevni ravni, obenem pa ohranja vse potrebne specifike in zmogljivosti korpusnega orodja.

3.2. Gigafida: spremembe v vmesniški zasnovi

Natančen opis vseh nadgradenj in sprememb vmesnika je na voljo v Logar Berginc et al. (2012). V tem prispevku predstavljamo le odločitve, ki prinašajo najpomembnejše novosti na področju slovenskih korpusnih vmesnikov.

Osnovna struktura vmesnika je s stališča, kako so jezikovni podatki na spletni strani razporejeni in kako se uporabnik po podatkovnih seznamih premika, sorodna spletnim iskalnikom. Korpus ne zahteva registracije in prijavljanja za delo, pred pričetkom rabe se prav tako ni potrebno prebijati skozi navodila za uporabo ali druge informacije o korpusu. Izdelava iskalnega pogoja je prva aktivnost, ki se od uporabnika pričakuje, zato je na osnovni, vhodni strani vmesnika v ospredju predvsem iskalno okence.

Iskanje po korpusu je v želji po intuitivnosti rabe primerljivo uporabi spletnih iskalnikov. Pri izdelavi iskalnega pogoja ni nujno poznavanje regularnih izrazov, oblikoskladenjskih oznak, posebnih simbolov oz. postopkov, ampak v iskalno okence preprosto vnesemo znakovni niz, ki ga v korpusu želimo poiskati. Iščemo lahko posamezne besede (npr. *medved*), besedne zveze (npr. *polarni medved*) oz. besedne nize, ki lahko vsebujejo tudi ločila (npr. *kljub temu, da*).

Pri naprednem iskanju lahko uporabnik dodatno pogojuje, katere zadetke želi pridobiti, recimo glede na oblikoskladenjske lastnosti iskane besede ali glede na druge besede v besedilni okolici. Tudi tukaj ni zahtevano poznavanje posebnih postopkov, ampak uporabnik pogoje enega za drugim enostavno poklika v predpripravljenih vmesniških tabelah.

Velika razlika glede na prejšnje slovenske konkordančnike je vpeljava samodejne lematizacije iskalnega pogoja.[7] Če je pri korpusu FidaPLUS uporabnik moral v iskalnem okencu opredeliti, da ga zanimajo *vse* oblike vnesene besede, mora po novem (tako, da iskano obliko postavi v narekovaje) opredeliti, kadar ga zanima *ena sama*, določena oblika. S stališča tipične uporabe slovenskih besedilnih korpusov je ta pot izdelave iskalnega pogoja precej bolj smiselna.

Pomembna novost je tudi uvedba t. i. podatkovnih filtrov. Filtri, ki se ob vsakem korpusnem iskanju avtomatsko pripravijo na osnovi korpusnih oznak (tako podatkov v glavah korpusnih besedil kot tudi lem, oblikoskladenjskih in korpusnih pojavnic samih), uporabniku na pregleden način pokažejo razpršenost iskanega jezikovnega pojava po besedilih. Uporabnik lahko v filtrih denimo vidi, kako pogosto se iskana beseda pojavlja glede na leto izida, vir besedila itd. Filtri obenem omogočajo, da uporabnik z enim samim klikom loči določen nabor podatkov iz celotne množice, npr. iz celotnega konkordančnega niza izbere samo tiste zadetke, ki izvirajo z interneta, ali iz celotnega nabora kolokatorjev izbere le tiste, ki so besednovrstno označeni kot glagoli.

Preglednost vmesnika smo dosegli tako, da smo uporabniku vedno ponudili samo tiste programske funkcije in povezave, ki jih pri določenem koraku svojega dela dejansko potrebuje. Na temeljni ravni se to odraža v delitvi vmesnika na tri dele (zavihke), vsak omogoča iskanje in pregledovanje druge vrste korpusnih podatkov: konkordančnih nizov, seznamov konkordanc ali besednih seznamov.[8] Vsak od treh delov vmesnika prinaša enako osnovno strukturo – možnost izvoza in tiskanja podatkov,

[7] Kadar je iskalni pogoj na ravni lem dvoumen, program poišče vse ustrezajoče zadetke in obenem ponudi možnost, da uporabnik v naslednjem koraku (z enim samim klikom) zadetke selekcionira sam.

[8] Besedni seznam je seznam besed, ki so v določenem delu enake, v drugem delu pa se razlikujejo. Za primer, v vrhu seznama lem, ki vsebujejo *-pisati*, so primeri: *napisati*, *zapisati*, *podpisati*, *vpisati*, *opisati*, *pripisati* itd.

podatkovne filtre, iskalno okence, zgodovino iskanj ipd. uporabnik najde vedno na istem mestu – obenem pa je v določenih točkah prilagojen specifikam obravnave jezikovnih podatkov, ki jih prinaša.

V primerjavi z večino korpusnih orodij vmesniku Gigafida manjka nekaj funkcionalnosti, ki so bile opuščene ravno z namenom zagotavljanja uporabniške prijaznosti. Tako je bil npr. izbor vzorca naključnih zadetkov opuščen, ker po eni strani vzorec lahko ponudi napačno sliko o rabi besede, po drugi strani pa smo menili, da ustrezno vzorčenje omogočajo že podatkovni filtri. Bolj tehnične narave je bila odločitev o opustitvi možnosti abecednega razvrščanja zadetkov, npr. po prvi besedi pred iskano besedo, saj je tako razvrščanje pri (zelo) pogostih besedah nadvse dolgotrajno in posledično uporabniku neprijazno, z uvedbo zmogljive izdelave seznama kolokatorjev pa postane vprašljiva tudi njegova smiselnost. Problem razvrščanja velikega števila zadetkov je vsekakor splošen, saj tudi najbolj napredna korpusna orodja, npr. Sketch Engine, ne ponujajo hitrejših rešitev.[9]

4. Sklep

V prispevku smo predstavili nekaj novosti, ki jih na področje referenčnega besedilnega korpusa za slovenščino prinaša korpus Gigafida. Slednji je skoraj enkrat večji od predhodnega korpusa FidaPLUS, sodobnejši na ravni zajetih besedil, preglednejši na ravni taksonomije, v katero besedila uvrščamo,[10] kvalitetneje lematiziran in oblikoskladenjsko označen (o čemer sicer v tem prispevku nismo govorili), korpusni konkordančnik je znatno hitrejši, na voljo pa je s povsem novim, izrazito uporabniško naravnanim vmesnikom.

Korpus Gigafida sledi ideji, da je potrebno referenčni besedilni korpus neprestano nadgrajevati in izboljševati, saj je to edina možnost, da imamo na voljo kvalitetne in sodobne jezikovne podatke, na podlagi katerih je mogoče izvajati verodostojne jezikoslovne raziskave. Pri korpusu Gigafida pa smo želeli narediti še korak dlje in korpusno rabo dejansko – ne le v teoriji – odpreti za širšo javnost.

Čeprav smo z razvojem vmesnika, ki neposredno izhaja iz uporabniških želja in potreb,[11] brez dvoma naredili pomemben korak naprej, se zavedamo, da so pojmi *uporabniška prijaznost, intuitivnost, preglednost* ipd. precej subjektivni. Zato imamo v načrtu tudi za novi vmesnik izvesti enako evalvacijo, kot smo jo izvedli za korpus FidaPLUS in primerjati rezulate. Na tej podlagi bo mogoče o izboljšavah novega vmesnika govoriti z večjo objektivnostjo.

Novi konkordančnik se je oblikoval pri grajenju korpusa Gigafida, vendar je uporaben tudi za druge vrste korpusov; kot smo že zapisali, bomo v projektu SSJ vanj vključili tudi korpus KRES, s pomembnimi prilagoditvami pa je bil konkordančnik uporabljen tudi pri govornem korpusu GOS (Verdonik, Zwitter Vitez, 2011). Na daljši rok želimo namreč za vse širše rabljene korpusne vire omogočiti dostopnost v enotnem programskem okolju, kar bi dodatno poenostavilo korpusno uporabo tako za specializirane kot laične uporabnike.

Da bi bili novi korpus res na voljo širokemu spektru uporabnikov, pa bo treba poskrbeti tudi za organizirano opismenjevanje uporabnikov za delo z njim, predvsem na ravni interpretacije korpusnih podatkov, s čimer naša ciljna publika trenutno še nima prav dosti izkušenj.

[9] Npr. v po velikosti primerljivem korpusu ukWac smo v orodju Sketch Engine razvrstili po prvi besedi na levi zadetke za samostalnik *relationship* ('odnos'), ki je po pogostosti (225.868 zadetkov) primerljiv z glagolom *kupiti* (231.509 zadetkov), tj. 579. besedo (lemo z besednovrstno oznako) na frekvenčnem seznamu korpusa Gigafida. Razvrščanje zadetkov je trajalo dolgih 42 sekund.

[10] Pri čemer nov prihodnji izziv vidimo v dodatni taksonomski kategorizaciji besedil glede na njihovo tematiko, kakršno je npr. uresničil *Oxford English Corpus* (prim. Atkins, Rundell, 2008: 72–73, 89).

[11] Slednje smo identificirali ob evalvaciji korpusa FidaPLUS (Arhar Holdt, 2010).

5. Literatura

Arhar Holdt, Š., Gorjanc, V., 2007. Korpus FidaPLUS: nova generacija slovenskega referenčnega korpusa. *Jezik in slovstvo*, 52(2): 95–110.

Arhar Holdt, Š., 2010. *Poročilo o evalvaciji korpusa FidaPLUS: Analiza odgovorov na anketni vprašalnik*. Dostopno na: http://www.slovenscina.eu/Media/Kazalniki/Kazalnik11/Evalvacija_FidaPLUS.pdf.

Atkins, B. T. S., Rundell, M., 2008: *The Oxford Guide to Practical Lexicography.* New York: Oxford University Press.

Erjavec, T., 2008. *Analiza metapodatkov korpusa FidaPLUS*. Interno gradivo.

Erjavec, T., Logar Berginc, N., 2012. Referenčni korpusi slovenskega jezika (cc)Gigafida in (cc)KRES. V T. Erjavec, J. Žganec Gros (ur.), *Zbornik Osme konference Jezikovne tehnologije*. Ljubljana: Institut Jožef Stefan.

Kalin Golob, M., 2003. *H koreninam slovenskega poročevalnega stila*. Ljubljana: Jutro.

Krek, S., Grčar, M., Dobrovoljc, K., 2012. Označevalnik za slovenski jezik Obeliks. V T. Erjavec, J. Žganec Gros (ur.), *Zbornik Osme konference Jezikovne tehnologije*. Ljubljana: Institut Jožef Stefan.

Logar Berginc, N., Šuster, S., 2009. Gradnja novega korpusa slovenščine. *Jezik in slovstvo*, 54(3–4): 57–68.

Logar Berginc, N., Grčar, M., Brakus, M., Erjavec, T., Arhar Holdt, Š., Krek, S., 2012. *Korpusi slovenskega jezika* Gigafida*,* KRES*,* ccGigafida *in* ccKRES*: gradnja, vsebina, uporaba.* Ljubljana: Trojina, zavod za uporabno slovenistiko; Fakulteta za družbene vede.

Rozman, T., Stritar, M., Krapš Vodopivec, I., Kosem, I. Krek, S., 2010. *Nova didaktika poučevanja slovenskega jezika.* Dostopno na: http://www.slovenscina.eu/Media/Kazalniki/Kazalnik15/Nova_didaktika_Sporazumevanje.pdf.

Verdonik, D., Zwitter Vitez, A., 2011. *Slovenski govorni korpus Gos*. Ljubljana: Trojina, zavod za uporabno slovenistiko.

6. Spletne strani

Korpus FidaPLUS: http://www.fidaplus.net.

Korpus Gigafida (beta): http://demo.gigafida.net/.

Korpus GOS: http://www.korpus-gos.net/.

MD5: http://en.wikipedia.org/wiki/MD5.

Merjenje obiskanosti spletnih strani: http://www.moss-soz.si/.

Nacionalna raziskava branosti: http://www.nrb.info/.

Sporazumevanje v slovenskem jeziku: http://www.slovenscina.eu/.

SPOOK.Sem: semantično označevanje vzporednega prevodoslovnega korpusa

Kristina Bizjak, Darja Fišer

Oddelek za prevajalstvo, Filozofska fakulteta
Aškerčeva 2, 1000 Ljubljana
darja.fiser@ff.uni-lj.si
kristabizjak@gmail.com

Povzetek

V prispevku predstavljamo prvi poskus ročnega semantičnega označevanja vzporednega prevodoslovnega korpusa SPOOK. Pri označevanju smo uporabljali pomene iz semantičnega leksikona sloWNet, ki temelji na Princeton WordNetu in je bil izdelan avtomatsko s pomočjo prosto dostopnih korpusnih in leksikalnih virov. Glavni namen raziskave je bil s primerjavo oznak v angleškem in slovenskem delu korpusa ugotoviti, v kolikšni meri se pojmi med jezikoma prekrivajo in ali je na tujem viru temelječ sloWNet primeren za označevanje slovenskih besedil. Z raziskavo pa smo želeli tudi zasnovati in preizkusiti označevalni sistem, ki bi bil uporaben za označevanje obsežnejšega korpusa, ter preučiti možnosti avtomatskega označevanja vzporednih korpusov na semantični ravni.

SPOOK.Sem: semantic annotation of a parallel translational corpus

This paper presents the first attempt to semantically annotate the parallel translational corpus SPOOK. The sense inventory used for annotation was the Slovene semantic lexicon sloWNet which is based on the Princeton WordNet and was developed automatically from a number of freely available corpus and lexical resources. The main goal of the study was a comparison of the annotations assigned in both languages in order to determine to what extent the concepts overlap in the two languages and whether the English-based sloWNet is suitable for annotating Slovene texts. In addition, we also wanted to develop and test an annotation scheme that would be suitable for the annotation of a larger corpus, and look into the possibilities of automatizing the annotation of parallel corpora at the semantic level.

1. Uvod

Semantično označevanje je ena od ravni označevanja korpusov, pri kateri besedam v korpusu pripisujemo pomenske lastnosti, ki jih izkazujejo glede na sobesedilo. Ne glede na to, ali semantično označevanje izvajamo ročno ali avtomatsko, velja semantično označevanje za eno najtežjih vrst označevanja korpusa. Za razliko od oblikoskladenjskega označevanja, kjer vse enote označujemo z istim naborom kategorij, moramo pri označevanju pomena besed za vsako besedo uporabiti drugačne kategorije. Osnovni problem pri semantičnem označevanju je v tem, da je pomen besed zelo izmuzljiva kategorija. Meje med posameznimi pomeni so pogosto zabrisane, razlikovanje med njimi pa je vsaj do neke mere subjektivno (Lakoff 1987). Kritiki kategorizacije besednih pomenov opozarjajo, da so le-ti izpeljani, prilagojeni ali celo ustvarjeni s konkretnim kontekstom, v katerem je beseda uporabljena, zaradi česar jih ni mogoče vnaprej našteti v leksikonu (Kilgarriff 1997, Hanks 2000). Poleg tega se pod predpostavko, da imajo besede določljivo število ločenih pomenov in podpomenov, takoj pojavi tudi vprašanje, kako to število določiti in kako pomene klasificirati, kar je ena od osrednjih tem v leksikografiji in leksikalni semantiki.

Ne glede na vse težave, na katere označevalci naletijo med označevanjem korpusa, pa se je potrebno zavedati, da so semantično označeni korpusi nepogrešljiv vir za razvoj sodobnih jezikovnih tehnologij, kot so avtomatsko razreševanje večpomenskosti, iskanje informacij po obsežnih zbirkah dokumentov in strojno prevajanje, prav tako koristijo tudi v uporabnem jezikoslovju na področju leksikografije in jezikovne pedagogike ter v splošnem jezikoslovju za proučevanje pogostosti in sopojavljanja posameznih pomenov.

Glede na to, da semantično označevanje korpusov še precej zaostaja za označevanjem na oblikoslovni in skladenjski ravni, ne preseneča dejstvo, da se s semantičnim označevanjem vzporednega prevodoslovnega korpusa srečujemo prvič, saj je za slovenščino zaenkrat na voljo le manjši enojezični semantično označeni korpus (Fišer 2010), ki je nastal v okviru projekta Jezikovno označevanje slovenščine (Erjavec et. al. 2010).

V prispevku najprej na kratko predstavimo uporabljene vire in sorodne raziskave. Nato natančno opišemo postopek in rezultate označevanja ter predstavimo tipične težave, na katere smo med delom naleteli, ter podamo predloge za spopadanje z njimi. Prispevek sklenemo z zaključnimi mislimi in predlogi za nadaljnje delo.

2. Uporabljeni viri

2.1. Princeton WordNet

Princeton WordNet (PWN) je obsežna leksikalna zbirka za angleški jezik, ki je začela nastajati v 80. letih prejšnjega stoletja na Univerzi v Princetonu in je kmalu postala zelo priljubljen vir pri najrazličnejših nalogah računalniške obdelave naravnega jezika. V njej so samostalniki, glagoli, pridevniki in prislovi razvrščeni v t.i. sinsete, nize kognitivnih sinonimov oz. literalov, ki se uporabljajo za izražanje istega pojma (npr. *sick, ill*, slo. *bolan*). Sinsetom je dodana razlaga, pogosto tudi primer rabe in domenska oznaka, posamezni sinseti pa so s semantičnimi in leksikalnimi relacijami (npr. *antonym*, slo. *protipomenka*) povezani v pojmovno mrežo. Wordnet vsebuje tako enobesedne kot večbesedne nize, pri čemer je upoštevana tudi metaforična in idiomatska raba (Fellbaum 1998: 3-17).

V tej raziskavi uporabljamo različico 3.0, ki vsebuje 155.327 različnih besed. Te so razvrščene v 117.597 sinsetov, od katerih je slabih 70 % samostalniških. Enopomenskih besed v PWN je 128.321, večpomenskih pa 27.006, povprečna stopnja večpomenskosti je tako 1,23 za samostalnike, 2,16 za glagole, 1,41 za pridevnike in 1,24 za prislove[1].

2.2. sloWNet

Po vzoru PWN je nastal tudi slovenski wordnet (sloWNet), ki je bil izdelan avtomatsko in je pri tem ohranil strukturo ter pojme iz PWN. Gradnja sloWNeta, ki je še vedno v razvoju, je doslej potekala v treh fazah: avtomatska indukcija slovenskih sinsetov na podlagi različnih že obstoječih dvo- in večjezičnih jezikovnih virov (Fišer in Sagot 2008), širjenje sloWNeta s pomočjo metod strojnega učenja (Sagot in Fišer 2011) in identifikacija nezanesljivih literalov z uporabo referenčnega korpusa in temeljnih načel distribucijske semantike (Sagot in Fišer 2012).

V najnovejši različici sloWNeta je 82.721 literalov, ki so razvrščeni v 42.919 sinsetov, kar predstavlja 36 % vseh sinsetov v Princeton WordNetu. Zaradi virov in metod, uporabljenih pri izdelavi wordneta, je v sloWNetu daleč največ samostalnikov (70 % vseh sinsetov). 66 % literalov je enopomenskih, povprečna stopnja večpomenskosti je 2,07, kar je nekoliko več kot v angleškem wordnetu, iz česar pa ne gre sklepati, da je slovensko besedišče bolj večpomensko, ampak je treba vzeti v obzir dejstvo, da je bil sloWNet izdelan avtomatsko, zato še vedno vsebuje mnogo napak, ki jih je v prihodnje potrebno odpraviti.

2.3. Korpus SPOOK

Slovenski prevodoslovni korpus SPOOK (Vintar 2009), ki je nastal v okviru projekta Slovensko prevodoslovje – viri in raziskave (2009!2012), je petjezični primerljivo-vzporedni korpus, ki v vzporednem delu vsebuje literarna besedila v angleščini, nemščini, francoščini in italijanščini ter njihove prevode v slovenščino, v primerljivem delu pa izvirna besedila v slovenskem jeziku. Korpus, ki obsega 8 milijonov besed, je stavčno poravnan, tokeniziran, oblikoskladenjsko označen, lematiziran in zapisan v XML v skladu z načeli TEI P5 (Erjavec 2012).[2]

Podkorpus, ki smo ga iz korpusa izbrali za semantično označevanje, obsega pet angleških romanov, pri čemer smo v želji po čim bogatejšem besedišču in čim večji raznolikosti zastopanih pomenov pazili na to, da smo izbrali dela petih različnih avtorjev in različnih žanrov. Izdelani podkorpus tako vsebuje znanstveno-fantastični roman *The Supernaturalist* avtorja Eoina Colferja, ki je bogat s tehničnimi opisi izmišljenega sveta iz bližnje prihodnosti, kriminalko *The Way through the Woods* Colina Dexterja, *Harry Potter and the Deathly Hallows* pisateljice J.K.Rowling, *White Teeth* britanske pisateljice Zadie Smith, ki v svojem romanu veliko uporablja pogovorni jezik ter Tolkienov *The Two Towers,* drugi del epske fantazijske pripovedi *Lord of the Rings.* Izbrana besedila je v slovenščino prevedlo pet različnih slovenskih prevajalcev, v tabeli 1 pa je mogoče videti velikost posameznih del, ki smo jih označevali.

[1] http://wordnet.princeton.edu/man/wnstats.7WN [15.5.2012]
[2] http://lojze.lugos.si/spook/korpus.html [18.9.2012]

Naslov in avtor dela	Št. besed - izvirnik	Št. besed - prevod
The Supernaturalist (Eoin Colfer)	62.235	58.775
The Way through the Woods (Colin Dexter)	87.024	76.270
Harry Potter and the Deathly Hallows (J. K. Rowling)	56.078	58.778
Lord of the Rings: The Two Towers (J. R. R. Tolkien)	146.771	150.367
White Teeth (Zadie Smith)	169.099	171.548
Skupaj	521.207	515.738

Tabela 1: Seznam in velikost del, zajetih v raziskavo

3. Sorodne raziskave

Ker se pri tej raziskavi prvič srečujemo s semantičnim označevanjem vzporednega korpusa za slovenščino, smo se pri označevanju naslanjali na izkušnje, ki so jih pri semantičnem označevanju korpusov pri sorodnih projektih pridobili tuji kolegi. Prvi vzporedni korpus, označen s pomeni iz wordneta, je MultiSemCor (Bentivogli, Forner in Pianta 2004), ki temelji na predpostavki, da se med prevajanjem izvirnika semantične funkcije v veliki meri ohranijo, zato uporabijo angleški korpus SemCor (Miller idr. 1994), ki že vsebuje semantične oznake iz Princeton WordNeta (Fellbaum 1998), ga prevedejo v italijanščino, avtomatsko poravnajo na besedni ravni, nato pa semantične oznake iz izvirnika prenesejo še v prevod. Recikliranje semantičnih oznak je mogoče, ker italijanski wordnet (Artale, Magnini in Strapparava 1997) vsebuje identične kode za sinsete.

Podoben pristop smo v pričujoči raziskavi uporabili tudi sami, tako da smo se skušali izogniti najbolj perečim problemom iz omenjene raziskave. Pri MultiSemCoru so uporabili angleški korpus, ki so ga prevedli v italijanščino, vendar so prevajalcem naročili, naj izvirne stavke prevajajo čim bolj dobesedno, da bi korpus čim lažje poravnali na besedni ravni. S tem so bistveno vplivali na podobo italijanščine v prevodu in leksikalno-semantični inventar, čemur smo se v naši raziskavi izognili, saj smo označevali vzporedni korpus, ki vsebuje prevode profesionalnih prevajalcev, namenjene knjižni objavi.

Po prevajanju so italijanski kolegi izvirnike in prevode avtomatsko poravnali na besedni ravni, kar jim je omogočilo prenos originalnih semantičnih oznak v italijanščino. Na podlagi lastnih izkušenj z uporabo orodij za besedno poravnavo vzporednih besedil lahko trdimo, da takšne poravnave zaenkrat vsebujejo toliko napak, da ta pristop brez temeljitega ročnega popravljanja za slovenščino še ni uporaben. To je za naš korpus še toliko relevantnejše, ker vsebuje literarna besedila, za katera je tipično, da so njihovi prevodi ohlapnejši in se manj držijo izvirnika, s čimer je avtomatska poravnava stavkov na besedni ravni še veliko težja.

Avtorji, ki pri svojem delu potrebujejo semantično označene korpuse, pogosto posegajo po semantičnih označevalnikih, kot sta na primer UKB (Agirre in Soroa 2009) in SenseRelate (Pedrsen in Kolhatkar 2009), ki korpus avtomatsko označujeta tako, da večpomenskim besedam glede na sobesedilo pripišeta najverjetnejši pomen iz wordneta. Za naše potrebe so omejitve pri uporabi tovrstnih orodij naslednje: večina orodij zaenkrat še ni prilagojenih za slovenščino, zato bi jih lahko

uporabili samo za angleški del korpusa. Edino orodje, ki ga je po naših informacijah mogoče uporabiti za kateri koli jezik, za katerega obstaja wordnet, je UKB. Vendar zanj avtorji poročajo, da pri označevanju dosega približno 70-odstotno natančnost, kar je za naše potrebe premalo, saj to pomeni, da bi med 100 označenimi pojavitvami nekega večpomenskega samostalnika v povprečju imeli 30 napačno pripisanih oznak, pri čemer bi bila ta številka najverjetneje še bistveno višja za samostalnike z visoko stopnjo večpomenskosti.

Tretja pomembna ovira pri uporabi avtomatskega označevalnika pomenov pa je ta, da je za označevalnike, ki temeljijo na wordnetu, zelo pomembno, da je wordnet, ki ga pri odločanju za izbiro najustreznejšega pomena uporabljajo, čim obsežnejši in kvalitetnejši, saj bi sicer šum iz wordneta negativno vplival na označevanje. Ker je bil sloWNet izdelan avtomatsko in še ni bil ročno pregledan, opažamo, da je za učinkovito rabo v te namene zaenkrat še premalo obsežen (vsebuje številne prazne sinsete), vsebuje pa tudi precej napak (v sinsetih se pojavljajo besede, ki niso ustrezne leksikalizacije tega pojma), zato menimo, da ga je pred uspešno uporabo v označevalnikih potrebno še izpopolniti.

Iz naštetih razlogov smo se za pilotno študijo označevanja lotili ročno, ki ga bomo na podlagi pridobljenih izkušenj v najkrajšem možnem času tudi avtomatizirali. Na izbranem vzorcu, izluščenem iz korpusa, označevanje najprej preizkusimo v smeri angleščina-slovenščina, nato pa še v obratni smeri. Čeprav se zavedamo, da bi za celovit preizkus semantičnega inventarja v sloWNetu potrebovali vzporedni korpus s slovenskimi izvirniki in angleškimi prevodi, smo v pričujoči raziskavi omejeni z naborom besedil v slovensko-angleškem delu korpusa SPOOK, ki zaenkrat vsebuje zgolj znanstvene prispevke s področja jezikoslovja, ta pa za proučevanje večpomenskosti splošnega besedišča niso primerna. Zato v tej raziskavi za označevanje v smeri slovenščina-angleščina uporabljamo kar obrnjen angleško-slovenski korpus. Kljub temu da se zavedamo, da bomo pri tem izgubili določeno število kulturno-specifičnih pomenov, ki se pojavljajo v slovenskem izvirnem leposlovju, pa ocenjujemo, da so izbrani prevodi delo najboljših slovenskih prevajalcev in so zato izrazno in slogovno na zelo visoki ravni, tako da nam bo tudi ta korpus omogočal zanimiv vpogled v zastopanost, distribucijo in prekrivnost pomenov med jezikoma. Vsekakor pa raziskavo nameravamo razširiti na avtentična slovenska besedila in njihove prevode takoj, ko bodo na voljo.

4. Označevanje korpusa

4.1. Izbor besed za označevanje

Pri izboru besed za semantično označevanje smo se omejili na večpomenske občne samostalnike, saj je razdvoumljanje osrednji problem semantičnega označevanja. Iz istega razloga smo uvedli tudi dva dodatna pogoja, in sicer, da se besede v vseh petih knjigah podkorpusa pojavijo vsaj desetkrat in se hkrati pojavljajo tudi v angleškem ter slovenskem wordnetu. Iskanje konceptualnih razlik med jezikoma in kulturno-specifičnih pomenov bi bilo verjetno bolj zanimivo na redkejšem besedišču, vendar je le-to v sloWNetu zaenkrat še razmeroma slabo pokrito in s stališča večpomenskosti manj problematično, poleg tega pa naš osnovni namen raziskave ni identifikacija izjem in redkih pojavov, temveč predvsem medjezikovno proučevanje pomenskega inventarja osnovnega besedišča z dolgoročnejšim ciljem razvoja avtomatskega pomenskega označevalnika.

V angleščini omenjenim pogojem ustreza 39 besed, v slovenščini pa 35. Večpomenskost med posameznimi besedami je močno variirala, najbolj večpomenska angleška beseda glede na wordnet je beseda *head* s 33 enobesednimi pomeni, najmanj pomenov pa imajo besede *child, hour, people* in *year,* in sicer po 4. Od izbranih slovenskih samostalnikov je imela najvišjo stopnjo večpomenskosti beseda *vrsta* (35 pomenov), najmanj pa *misel, oči, postelja* in *roka* (po 4 pomene). Ker slovenski wordnet vsebuje precej šuma, smo vse pomene izbranih 35 besed pred začetkom označevanja ročno pregledali in popravili napake.

Za vsako izbrano besedo smo iz vsake knjige izluščili po pet naključnih stavkov, ki so vsebovali izbrano besedo, so obsegali med 5 in 50 besed, označevana beseda pa ni ne prva ne zadnja v stavku. Za vsako besedo smo tako zbrali 25 oznak, kar skupno pomeni 975 angleško in 875 slovensko semantično označenih stavkov.

4.2. Označevanje korpusa

Semantično označevanje korpusa je potekalo ročno. Vsaki izmed 975 angleških besed smo glede na sobesedilo v korpusu in razlago ter semantične relacije v wordnetu pripisali pomen (sinset ID) iz semantičnega leksikona PWN. Nato smo v slovenskem prevodu istega stavka preverili, ali prevodna ustreznica, ki jo je izbral prevajalec, v sloWNetu sodi v isti koncept ter še njej pripisali ustrezen sinset ID. V nasprotnem primeru smo zanjo poiskali najustreznejši pojem in ji pripisali njegov ID. Po zaključenem označevanju angleško-slovenskih stavkov smo postopek ponovili še v obrnjeni smeri, slovensko-angleški, kjer smo pomen pripisovali besedam v 875 stavkih.

4.3. Rezultati označevanja

Izkazalo se je, da je bilo vse angleške izluščene besede podkorpusa mogoče označiti z enim izmed pomenov v wordnetu, prav tako to velja tudi za njihove prevodne ustreznice v slovenščini. Precej težje je bilo označevanje v slovensko-angleški smeri, ker so se pojavljali pomeni, ki jih na prvi pogled ni bilo lahko ločiti, saj so bile razlike med njimi ponekod minimalne ali celo nejasne. Na zahtevnost pripisovanja pomenov je vplivala tudi stopnja večpomenskosti posamezne besede v sloWNetu. To pomeni, da bo sloWNet v prihodnosti potrebno še bolj natančno pregledati in te nejasnosti odpraviti.

Po končanem označevanju smo imeli za izbrane besede in njihove prevodne ustreznice v obeh jezikovnih smereh skupaj pripisanih 3.700 pomenov iz wordneta. Zaradi boljše preglednosti rezultate angleško-slovenskega in slovensko angleškega podkorpusa navajamo v ločenih razdelkih.

4.3.1. Rezultati angleško-slovenskega korpusa

Med označevanjem angleško-slovenskega korpusa smo od 372 možnih pomenov izbranih besed v wordnetu uporabili zgolj 205 (55 %). Med posameznimi besedami so velika odstopanja, saj so za besedi *child* in *people* uporabljeni vsi pomeni, ki so bili na voljo (4), najmanjši

odstotek pomenov pa je bil uporabljen za besedo *head* (7 od 33). Ko smo za označene besede v vzporednih stavkih iskali prevodne ustreznice, smo našteli 163 različnih. Poleg označevanja pomena posameznih besed smo v korpusu pri 14 besedah od skupno 39 izbrali sinset, kjer označena beseda tvori del večbesedne zveze in tako uporabili 22 različnih pomenov za večbesedne literale, ki so bili uporabljeni v 50 stavkih (npr. *arm rest*). Nadaljnjih 11 besed oz. 40 stavkov je bilo označenih s pomenom nesamostalniškega literala (npr. *by heart*)

4.3.2. Rezultati slovensko-angleškega korpusa

Med označevanjem slovensko-angleškega korpusa je bil delež uporabljenih pomenov za označevanje nekoliko večji (58 %), število možnih pomenov za enobesedne literale v wordnetu pa nekoliko nižje kot za angleški del korpusa (262). Samostalnik *glava* se je pojavil v 2 od 13 pomenov iz sloWNeta, *miza* pa v vseh svojih 4 pomenih. Označene besede imajo v korpusu 125 različnih angleških ustreznic, ena sama je uporabljena za *življenje*, kar devet pa za *vrsto*. Pri 12 od 35 besed smo poleg pomenov za enobesedne uporabili tudi 25 pomenov za večbesedne literale (npr. *rojstni dan*), pri petih pa še sedem pomenov nesamostalniških literalov (npr. *v redu*).

4.3.3. Analiza ujemanja pojmov med jezikoma

Med semantičnim označevanjem nas je zanimalo tudi, kakšno je (ne)ujemanje izbranega sinset id-ja za angleške besede in sinset id-ja, ki je bil pripisan njeni prevodni ustreznici v slovenščini. Izkazalo se je, da ujemanje v angleško-slovenskem podkorpusu povprečno znaša 75 %, v slovensko-angleškem pa je nekoliko višje, in sicer 78 %.

Razlog za neujemanje	**ANG-SLO (št. primerov)**	**SLO-ANG (št. primerov)**
Parafraza	162	84
Spec./generalizacija	44	84
Idiomatska raba	21	5
Večbesedna zveza	7	21
Konceptualna razlika	5	0
Skupaj	239	194

Tabela 2: Pregled razlogov za neujemanje sinset ID-jev

Na tem mestu bi bila zanimiva primerjava medjezikovnega ujemanja pojmov v kakšnem od sorodnih projektov semantičnega označevanja vzporednih korpusov, vendar, kolikor nam je znano, tuji kolegi o tovrstnih analizah ne poročajo, saj so se posvečali predvsem proučevanju (ne)ujemanja uporabljenih oznak med različnimi anotatorji. Analiza, ki jo predstavljamo v tem razdelku, tako podaja pogled na označeni korpus z novega zornega kota. Posebno pozornost smo namenili primerom, pri katerih ni bilo ujemanja med pripisanimi sinset id-ji v obeh jezikih. Razloge za neujemanje smo razvrstili v pet kategorij. Število posameznih primerov je za oba podkorpusa razvidno iz tabele 2.

Ker smo označevali korpus literarnih besedil, prevodi katerih so pogosto svobodnejši, neujemanje ne pomeni nujno, da pojmi med jezikoma niso prekrivni, saj so številni avtorji pri proučevanju prevodoslovnih pojavov (Baker 1993) ugotovili, da prevajalci tovrstnih besedil radi posegajo po parafrazah in izpustih ter natančnejših oz. ohlapnejših prevodih glede na izvirnik. To se je izkazalo tudi za naša podkorpusa. Ugotovili smo namreč, da po pogostosti najbolj izstopa parafraziranje, ko prevajalec iz slogovnih ali individualnih razlogov izvirno besedo nadomesti z drugačnimi jezikovnimi sredstvi, čeprav bi bil neposredni prevod, v katerem je uporabljen literal iz istega sinseta kot v izvirniku, jezikovno povsem ustrezen.

Primer 1:

*Ang.: I'd rather go to **bed** than get into this.*
*Slo.: Rajši bi šla malo **spat** kot pa tole.*
(čeprav bi bilo ustrezno tudi *šla v posteljo).*

V slovensko-angleškem delu je število primerov spec./generalizacije enako številu parafraz, v angleško-slovenskem delu pa smo na prevode, ki so pod- ali nadpomenke oz. mero- oz. holonimi izvirnikov, naleteli 44-krat.

Primer 2:

Ang.: *The sniper in the rafters transferred the laser dot to Stefan's **head.***
*Slo.: Ostrostrelec je laserski žarek nameril v Štefanovo **čelo.***
(čeprav bi bilo ustrezno tudi *Štefanovo glavo)*

V obeh podkorpusih so se pojavljali tudi primeri, ko je bila idiomatska raba nekega izraza v izvirniku prevedena razlagalno oz. nadomeščena s slovenskim idiomom s podobno funkcijo oz. obratno.

Primer 3:

*Ang.: I had some **part** in that: for I sat in a high place, and I strove with the Dark Tower; and the Shadow passed.*
*Slo.: Nekaj **prstov** sem imel pri tem zraven jaz: kajti sedel sem na visokem kraju in se kosal s Temnim stolpom; in Senca je prešla.*

Do neujemanja sinset ID-jev v enem in drugem jeziku je prišlo tudi zato, ker so nekateri koncepti v enem jeziku izraženi z večbesedno zvezo, v drugem pa z enobesednim leksemom. Do razlike lahko prihaja, ker večbesedna zveza v enem od jezikov v wordnetu ne obstaja (glej primer 4) ali pa se ji spremeni besedna vrsta (glej primer 5).

Primer 4:

*Slo.: No, ampak ravno sem te hotela vprašati, če imaš kakšno posebno željo v zvezi s praznovanjem **rojstnega dne.***
*Ang.: Actually, I've been wanting to ask you how you want to celebrate your **birthday**, Harry.*

Primer 5:

*Slo.: Vrečka je bil **v redu**, samo govoril je preveč.*
*Ang.: Ziplock was **OK**, except that he talked too much.*

Vsekakor pa nas je pri označevanju presenetilo dejstvo, da smo naleteli na zelo majhno število konceptualnih razlik. V angleško-slovenskem delu je bilo takšnih 7, v slovensko-angleškem delu pa nobene, kar je za metodo, s katero je bil izdelan sloWNet zelo spodbudna ugotovitev, saj kaže na to, da prevzemanje konceptualne strukture iz tujega jezika za uporabo leksikona v praksi nima velikega negativnega vpliva. Vse konceptualne razlike izhajajo iz kulturno-specifičnih razlik, kot je npr. sistem merskih enot.

Primer 6:

> Ang.: *You know it only rises about two **feet** off the ground but he nearly killed the cat and he smashed a horrible vase Petunia sent me for Christmas (no complaints there).*
> *Slo.: Kot veš, se metla dvigne največ pol **metra** visoko, a skoraj bi ubil mačka in razbil je grozljivo vazo, ki mi jo je Petunija poslala za božič.*

4.4. Opis tipičnih težav in strategij za spopadanje z njimi

Pri označevanju korpusa z ustreznim sinset ID-jem iz semantičnega leksikona se je izkazalo, da je bilo veliko lažje označevati angleško-slovenski korpus kot slovensko-angleškega. Eden izmed razlogov za to tiči v tem, da smo pri označevanju slovensko-angleškega dela morali najprej počistiti napake, ki so nastale pri avtomatski gradnji sloWNeta. Če si ogledamo samo 4 od 35 označevanih besed (*vrsta, mesto, glava* in *konec),* lahko vidimo, da smo omenjenim besedam s popravljanjem napak v sloWNetu število pomenov več kot prepolovili. Pri besedi *vrsta* smo tako število pomenov zmanjšali s 54 na 21, pri *mestu* s 54 na 14, pri *glavi* s 53 na 13, pri besedi *konec* pa z 42 na 12.

Kot smo že omenili, večjih težav s konceptualnimi razlikami pri označevanju izbranega korpusa nismo imeli. Veliko več je bilo primerov, pri katerih ni bilo ujemanja med besednima vrstama izbranega pojma v enem in drugem jeziku. Čeprav smo se zaradi največje zastopanosti samostalnikov v sloWNetu odločili za označevanje slednjih, smo med označevanjem stavkov mnogokrat ugotovili, da je izbrana beseda nastopala v vlogi prislova ipd., kar je posledica napak pri avtomatskem označevanju korpusa na oblikoskladenjski ravni. Tako se je npr. izkazalo, da se beseda *strah* v 4 od 25 stavkov pojavi v povedno-prislovni obliki *biti strah*, za katero v wordnetu nismo našli primernega sinseta.

Primer 7:

> *Slo.: Harry, vem, kako te vleče v Godricov Dol, ampak mene **je strah**...*
> Ang.: *Harry, I know you really want to go to Godric's Hollow, but I'm **scared**.*

Problematična je bila tudi večbesedna zveza *pri roki*. Kot kaže primer 8, je njena angleška ustreznica ***at hand*** v Princeton WordNetu uvrščena med pridevniške sinsete, medtem ko v slovenščini *pri roki* obravnavamo kot prislov.

Primer 8:

> *Slo.: Nemogoče, da bi bili z opremo, ki jo imamo **pri roki**, kos takšni okvari.*
> *Ang.: There's no way we could deal with this kind of damage with the equipment we have **at hand.***

Kar se tiče težav pri pripisovanju pomenov, pa se je pri označevanju besed v obeh delih podkorpusa izkazalo, da je najtežje določiti ustrezen sinset ID abstraktnim samostalnikom. Na angleški strani so takšni bili: *thing, life, part, time,* na slovenski strani pa *vrsta, življenje, čas, prostor,* ki so sicer izkazovali nekoliko višjo stopnjo večpomenskosti, vendar je bilo veliko bolj kot število pomenov problematično to, da so bili posamezni pojmi v wordnetu opredeljeni z ohlapnimi in nejasnimi razlagami ter primeri rabe. Veliko hitreje je označevanje potekalo pri stvarnih samostalnikih, še posebej tistih, ki v enem svojih pomenov označujejo del človeškega telesa (*arm, back*, *head, hand, eye, foot, mouth* oz. *glava, noga, obraz, prst, roka)*, saj se je pri označevanju kot najpogostejši koncept izkazalo prav poimenovanje za del človeškega telesa. Čeprav ima tudi beseda *head* oz. *glava* razmeroma visoko stopnjo večpomenskosti, 33 pomenov v angleščini in 13 v slovenščini, pa to ni povzročalo težav pri označevanju, saj so razlike med izbranimi pomeni jasne.

Večjo zanesljivost oznak bi bilo mogoče doseči z uporabo več kot enega anotatorja za vsak stavek, kot je to v navadi pri večini projektov, ki se ukvarjajo z jezikoslovnim označevanjem korpusov. To bi bilo še posebej koristno pri težjih primerih, ki bi jih z večjim številom anotatorjev veliko lažje identificirati, prav tako pa bi na podlagi več glasov lahko izbrali splošno najbolj sprejemljivo rešitev. Ker je dvojno ali večkratno označevanje korpusa presegalo zmožnosti in okvire te pilotne študije, se bomo primerjavi in analizi odločitev različnih anotatorjev posvetili v prihodnje.

5. Zaključek

V prispevku smo predstavili uporabo medjezično poravnanih wordnetov za semantično označevanje vzporednega korpusa literarnih besedil, ki je bil izluščen iz prevodoslovnega korpusa SPOOK. Analiza pripisanih oznak je pokazala, da večjih težav z neujemanjem zaradi jezikovnih in kulturnih razlik med angleščino in slovenščino ni bilo, nad čimer smo bili vsekakor pozitivno presenečeni. Ker smo za označevanje uporabili samo angleško-slovenski korpus, bi bilo stopnjo ujemanja na konceptualni ravni med tema dvema jezikoma v prihodnosti nujno potrebno preveriti tudi na korpusu avtentičnih slovenskih besedil z angleškimi prevodi, kjer pričakujemo večja odstopanja. Kljub vsemu pa izkušnje, pridobljene v predstavljeni raziskavi, ne kažejo bistvenih konceptualnih razlik, ki bi izdelavo slovenskega wordneta s prevzemanjem semantičnega inventarja iz Princeton WordNeta postavljala pod vprašaj, kar je za nadaljnji razvoj vira zelo spodbudno.

Vendar je pri tem treba poudariti, da kljub spodbudnim rezultatom, pridobljenih v pričujoči raziskavi, še vedno obstaja precejšen razkorak med angleškim in slovenskim wordnetom na ravni pokrivanja leksikalnega inventarja, ki se pojavlja v korpusu. Medtem ko Princeton WordNet vsebuje večino splošnega in specializiranega besedišča, zaenkrat sloWNet zadovoljivo pokritost zagotavlja le za

najpogostejše besedišče, medtem ko je v njem srednjepogostih in redkejših izrazov bistveno manj, kar mu močno zmanjšuje uporabno vrednost, zato bi ga bilo potrebno čim prej razširiti tudi z manj pogostim besediščem.

Poleg analize semantično označenega korpusa ima opravljena raziskava tudi povsem oprijemljiv rezultat, ki se kaže v obliki prvega vzorčnega vzporednega korpusa za slovenščino, ki je označen na semantični ravni. Na podlagi izkušenj, pridobljenih v prvem poskusu tovrstnega označevanja, bomo v prihodnje definirali smernice za obsežnejše označevanje korpusa SPOOK na semantični ravni, s čimer bomo omogočili najrazličnejše leksikološke in komparativne študije, ter razvili vir, ki bo uporaben tudi v večjezičnih jezikovnotehnoloških aplikacijah. Poleg razširitve ročnega označevanja načrtujemo s sistematičnim popravljanjem sloWNeta in prilagajanjem avtomatskih orodij, potrebnih za delo z jezikovnim parom angleščina-slovenščina, omogočiti tudi avtomatsko označevanje celotnega vzporednega korpusa.

Čeprav se opravljena raziskava ukvarja z eno najpomembnejših posledic prevzemanja tujejezičnega vira, t.j. testiranjem nabora in distribucije pomenov slovenskih besed glede na jezikovno realnost, izpričano v korpusu, v njej nismo preverjali, v kolikšni meri zasnova semantičnega leksikona na obstoječem viru vpliva tudi na strukturo dobljene semantične mreže in na katerih mestih bi bilo zaradi konceptualizacijskih razlik ter jezikovnih posebnosti med angleščino in slovenščino potrebno omogočiti odstopanja od nje. Diagnostični testi, ki jih v sorodnih raziskavah uporabljajo za potrjevanje semantičnih relacij med dvema pojmoma, so namreč zanesljivi le na zelo velikih količinah podatkov, bistveno večjih od označenega korpusa, čemur se nameravamo posvetiti v nadaljnjem raziskovalnem delu.

6. Literatura

Agirre, E. in Soroa, A., 2009: Personalizing PageRank for Word Sense Disambiguation. *Proceedings of the 12th Conference of the European chapter of the Association for Computational Linguistics (EACL'09)*.

Artale, A., Magnini, B., in Strapparava, C., 1997: WordNet for Italian and Its use for Lexical Discrimination. *Proceeedings of the 5th Congress of the Italian Association for Artificial Intelligence*, Rome, Italy, 16-19 September 1997, Springer Verlag.

Baker, M., 1993: Corpus Linguistics and Translation Studies: Implications and Applications, *Text and Technology: In Honour of John Sinclair*, Baker, Francis and Tognini-Bonelli (Eds), Amsterdam/ Philadelphia, John Benjamins, pp. 233-250.

Bentivogli, L., Forner, P., in Pianta, E., 2004: Evaluating cross-language annotation transfer in the MultiSemCor corpus. *Proceedings of the 20th international Conference on Computational Linguistics*.

Erjavec, T. (v tisku): Vzporedni korpus SPOOK: označevanje, zapis in iskanje.

Fellbaum, C., 1998: *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

Fišer, D., in Sagot, B., 2008: Combining Multiple Resources to Build Reliable Wordnets. *Proceedings of the 11th Text, Speech and Dialogue Conference*.

Fišer, Darja. Pristopi za avtomatizirano gradnjo semantičnih zbirk. *Terminologija in sodobna terminografija*. Ljubljana: Založba ZRC, ZRC SAZU, 2009, pp. 357-370.

Fišer, Darja. Semantično označevanje korpusov. *Slovenske korpusne raziskave*. Ljubljana: Znanstvena založba Filozofske fakultete, 2010, pp. 110-130.

Hanks, P., 2000: Do word meanings exist? *Computers in the Humanities*, 34 (1–2).

Kilgarrif, A., 1997: I don't believe in word senses. *Computers in the Humanities*, 31 (2), 91–113.

Lakoff, G., 1987: *Women, fire, and dangerous things: what categories reveal about the mind*. Chicago: University of Chicago Press.

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., in Thomas, R. G., 1994: Using a semantic concordance for sense identification. *Proceedings of the workshop on Human Language Technology*.

Pedersen, T., in Kolhatkar, M. 2009: WordNet::SenseRelate::AllWords - A broad coverage word sense tagger that maximimizes semantic relatedness. *Proceedings of NAACL '09*, pp. 17-20.

Sagot, B., in Fišer, D., 2011: Extending wordnets by learning from multiple resources. *Proceedings of LTC 2011*, Poznan, Poland

Sagot, B., in Fišer, D., 2012: Cleaning noisy wordnets. *In Proceedings of LREC 2012*, Istanbul, Turkey.

Vintar, Š., 2009: Slovenski prevodoslovni korpus. In M. Stabej (ur.), *Infrastruktura slovenščine in slovenistike*., Ljubljana: Znanstvena založba Filozofske fakultete: 385-391

Sistem vsebinskega priporočanja dokumentov kot izboljšava funkcionalnosti v digitalni knjižnici Univerze v Mariboru

Mladen Borovič, Milan Ojsteršek

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Smetanova 17, 2000 Maribor
{mladen.borovic, ojstersek}@uni-mb.si

Povzetek

Prispevek predstavlja sistem vsebinskega priporočanja dokumentov v digitalni knjižnici Univerze v Mariboru. S pomočjo takšnega sistema lahko uporabnikom ponudimo več vsebine, hkrati pa izboljšamo nekatere že obstoječe funkcionalnosti v digitalni knjižnici. Sistem priporočanja obdeluje vsebino dokumentov in pri razvrščanju zadetkov upošteva tudi uporabniške aktivnosti. V prispevku podrobneje opišemo kako sistem obdela besedila in razvrsti zadetke po podobnosti. Podamo rezultate meritev vplivov različnih obdelav besedila na kvaliteto priporočil. Na koncu podamo nekaj idej za izboljšave funkcionalnosti v digitalni knjižnici.

A content-based recommender system and its role in functionality improvements of the Digital Library of University of Maribor

This article presents a content-based document recommender system in the digital library of University of Maribor. By using this kind of a system, we can offer users more content while simultaneously enhancing some established features of the digital library. In the system workflow, the documents are processed and user activities are also taken into account in the document similarity ranking process. The article describes in detail, the document processing, as well as the document similarity ranking process. We provide the measurement results which show the effect of different document processing techniques on recommendation quality. Some ideas where to use the functionality of this system in the digital library, apart from recommending documents, are also provided.

1. Uvod

Sistemi priporočanja so del vsakdanje izkušnje na spletu, predvsem pri spletnih trgovinah in spletnih iskalnikih. Glavni cilj teh sistemov je ponuditi uporabniku vsebine, ki bi ga najbolj zanimale. Motivacija za to je lahko različna. Spletne trgovine uporabljajo takšne sisteme, da bi povečale predstavitev ponudbe. Pri spletnih iskalnikih iskalnikih je cilj nuditi poosebljeno iskanje in tako uporabniku dostaviti najbolj ustrezne zadetke. Sistem priporočanja mora torej na inteligentni način razpoznati uporabnikovo zanimanje zgolj na podlagi njegovih aktivnosti na spletni strani.

Pristopi priporočanja se delijo na dve skupini. Prva skupina pristopov deluje nad uporabniškimi aktivnostmi. Vanjo spadajo sodelujoče filtriranje *(angl. collaborative filtering)*, pristopi z binarnimi vektorji (Melville in Sindhwani, 2010) in algoritmi Slope One (Lemire in Maclachlan, 2005). Druga skupina pristopov priporočanja deluje zgolj nad vsebino, uporabniške aktivnosti pa nimajo poglavitne teže in se tako navadno uporabljajo za dodatne uteži pri razvrščanju rezultatov. Te metode se uporabljajo tudi na področju pridobivanja informacij *(angl. information retrieval)*. Med bolj znane spadajo metoda BM25 (Garcia, 2011), latentna semantična analiza (LSA) (Deerwester et al., 1990) in druge, ki so bile izpeljane iz podobnih predpostavk.

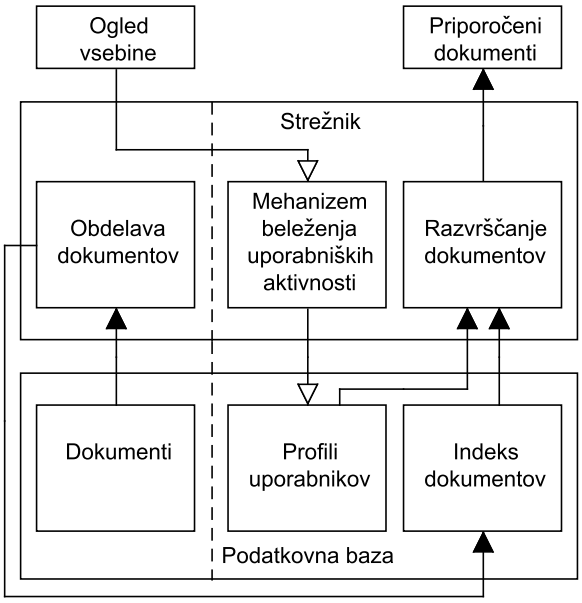
V tem članku predstavimo sistem vsebinskega priporočanja dokumentov v digitalni knjižnici Univerze v Mariboru (v nadaljevanju DKUM) in pokažemo kako posredno in neposredno izboljšuje njene trenutne funkcionalnosti. Pokažemo tudi, da se lahko sistemi priporočanja uporabijo tudi v druge namene. Izhajamo iz dejstva, da je delovanje sistema priporočanja podobno delovanju spletnega iskalnika. Obema je skupno, da vračata najbolj ustrezne rezultate glede na vhodni vnos. Razlikujeta se le v načinu podajanja vhodnega vnosa. Iskalni pojem kot vhod v spletni iskalnik največkrat vnese uporabnik, medtem ko se pri sistemu priporočanja kot vhod ovrednoti uporabnikova aktivnost. Kadar izvajamo vsebinsko priporočanje, pravzaprav tvorimo značilke iz besedila. Slednje počnejo tudi spletni iskalniki. Čeprav imajo uporabnikove aktivnosti pri tem stransko vlogo pa jih lahko še vedno uporabimo kot dodatne uteži, da bi še bolje filtrirali ustrezne vsebine in se s tem približali ideji poosebljenega iskanja.

V prispevku najprej opišemo sistem vsebinskega priporočanja dokumentov v DKUM. Podrobneje opišemo strukturo sistema, uporabljene obdelave besedila in postopek razvrščanja zadetkov. Sledi analiza vpliva različnih obdelav besedila na kvaliteto priporočil. Zatem opišemo možne uporabe sistema priporočanja v druge namene - natančneje pri iskanju potencialnih plagiatov in svetovanju pri izbiri mentorjev zaključnih del. V zaključku podamo smernice za nadaljnje delo.

2. Sistem vsebinskega priporočanja

Naš sistem tvorijo strežnik in spletne storitve za dostop do strežnika. Strežnik je sestavljen iz treh ključnih komponent, ki omogočajo priporočanje (slika 1). Z modulom za

beleženje uporabniških aktivnosti pridobimo informacije o uporabniških aktivnostih kot so število ogledov, število prenosov in ocene gradiv. Modul obdelave dokumentov zagotavlja poenoteno predstavitev vseh dokumentov v obliki naslova, ključnih besed in povzetka. Hkrati se v tem modulu izvaja izračun ocen BM25, ki tvori indeks dokumentov. Slednjega lahko predstavimo kot matriko podobnosti vseh dokumentov. Strežnik izvaja obdelavo dokumentov periodično, saj se v zbirko dokumentov dodajajo novi dokumenti. Na tak način posodabljamo indeks dokumentov. Podatki o uporabnikih in izračunane podobnosti med dokumenti so vhod v modul razvrščanja dokumentov, kjer izberemo podobne dokumente glede na dokument, ki si ga uporabnik trenutno ogleduje.



Slika 1: Struktura sistema priporočanja dokumentov.

2.1. Obdelava besedila

Vhodne dokumente smo predstavili z naslovi, ključnimi besedami in povzetki. Iz besedila vhodnih dokumentov smo najprej odstranili pogoste besede, ki ne vsebujejo informacije o njegovem pomenu. To so slovnične besedne zveze med katere spadajo vezniki, predlogi, členki in medmeti. Seznam pogostih besed smo tvorili sami s štetjem besed v dokumentih, ki so trenutno v DKUM. Nadalje smo uporabljali dve vrsti obdelave - lematizacijo in avtomatsko pridobivanje ključnih besed s pomenskim označevanjem. Tako lematizator kot tudi pomenski označevalnik sta plod dela naše razvojne skupine in ju uporabljamo po principu črne škatle.

2.1.1. Lematizacija

Lematizacija (*angl. lemmatisation*) ali geslenje je postopek določanja osnovne slovarske oblike (leme) besedam v besedilu (Brezovnik, 2009). Lematizaciji zelo podoben postopek je krnjenje. Glavna razlika med lematizacijo in krnjenjem je v tem, da krnjenje besede ne pretvori v slovarsko obliko, ampak preprosto odreže končnico besede tako, da ostane le krn. Pri besedilnem rudarjenju se lematizacija lahko uporablja pri odkrivanju kontekstov besedil. Tako si z lematizacijo olajšamo delo, saj sklepamo, da bo lema obdržala pomen besede.

2.1.2. Avtomatsko pridobivanje ključnih besed s pomenskim označevanjem

Za opis tematike dokumenta ponavadi vzamemo ključne besede, podane s strani avtorja ali knjižničarja. S slednjimi lahko uvrstimo dokument v ustrezne kategorije. Nekatere ključne besede pa niso vedno primerne, saj so preveč splošne ali pa preveč specifične. V tem primeru poznamo rešitev v obliki avtomatskega pridobivanja ključnih besed. Postopek deluje tako, da kot vhod podamo celoten dokument ali pa le en odsek, na izhodu pa dobimo besede, ki opisujejo tematiko vhodnega besedila. Uporabili smo postopek, ki uporablja pomensko označevanje s pomočjo Wikipedije (Burjek, 2011). Pridobivanje ključnih besed poteka v naslednjih korakih:

I. Z algoritmom za iskanje besed najdi primerne besede za pomensko označevanje.

II. Razvrsti besede glede na podatek o temah, na katere je posamezna beseda kazala v Wikipediji.

III. S pomočjo algoritma za izračun sorodnosti, razločevalnika in klasifikatorja naučenega z algoritmom C4.5, izračunaj splošnosti besed.

IV. Za vsako besedo izračunaj vrednost za verjetnost, da bi bila beseda povezava na temo, če bi se pojavila v Wikipediji.

V. Vrni seznam besed, urejenih po verjetnostih izračunanih v koraku IV.

Tabela 1: Primer rezultata pridobivanja ključnih besed s pomenskim označevanjem.

Vhod
V diplomskem delu obravnavamo problematiko pisanja tehničnih vsebin in objavljanj le-teh na spletu. V teoretičnem delu predstavimo slovnico označevalnih jezikov LaTeX in MathML. V praktičnem delu smo izdelali spletni urejevalnik WYSIWYG, ki omogoča vnos matematičnih in kemijskih formul, zapisanih v označevalnem jeziku MathML.
Izhod
MathML, WYSIWYG, LaTeX, Jezik, Slovnica, Splet, Tehnika

V koraku I pred izbiranjem primernih besed za pomensko označevanje uporabimo parameter p_{min}, ki predstavlja minimalno zahtevano verjetnost, da je beseda označena kot povezava v Wikipediji. Ta parameter lahko nastavljamo

poljubno, avtor pa svetuje vrednost 0.85, kar pomeni 85% verjetnost, da gre za povezavo v Wikipediji. Verjetnosti se izračunajo z enačbo 1, kjer $n_l(w)$ pomeni število pojavitev kot beseda, $n_W(w)$ pa število vseh pojavitev v Wikipediji.

$$p_l(w) = \frac{n_l(w)}{n_W(w)} \tag{1}$$

Po koraku I so znane besede, ki so se pojavile v besedilu, ne pa tudi njihov pomen. Zato v koraku II ugotavljamo pomen besed na podlagi tematik, v katerih so se pojavile. Besede lahko imajo več pomenov, zato je treba ugotoviti katere so enopomenske in katere večpomenske. To storimo tako, da ovrednotimo povezavo med besedami in tematikami glede na pogostost uporabe. Tukaj igrajo glavno vlogo verjetnosti, da so besede povezave v Wikipediji. V kolikor je verjetnost besede večja od minimalne zahtevane verjetnosti, jo smatramo kot enopomensko in tako določimo pomen besede.

V koraku III s pomočjo razločevalnika izračunamo verjetnosti mišljenih pomenov pri ostalih besedah. Razločevalnik izračuna verjetnost s pomočjo splošnosti *angl. commonness*, sorodnosti in podatkov s konteksta. Pri kontekstu se upošteva kvaliteta konteksta; če so enopomenske besede v istem dokumentu sorodne, je pri določanju pomena večpomenskih besed bolj pomembna sorodnost, v primeru slabe sorodnosti pa je bolj pomembna splošnost. Splošnost je dana z enačbo 2 in predstavlja inverz verjetnosti, da je beseda povezava v Wikipediji.

$$c(w) = \frac{n_W(w)}{n_l(w)} \tag{2}$$

Sledi še korak IV, kjer se za vsako besedo izračuna vrednost za verjetnost, da bi bila beseda povezava na temo, če bi se pojavila v Wikipediji. Izračun poteka s pomočjo algoritma C4.5, ki tvori odločitveno drevo s katerim klasificiramo primerne besede. Uporabljen pomenski označevalnik je zasnovan na odprtokodni rešitvi Wikipedia Miner, kjer se prav tako uporablja algoritem C4.5, zanimivo pa bi bilo preizkusiti tudi kakšen drug algoritem za gradnjo odločitvenih dreves. Vhod v ta algoritem so torej pojavitve besed, izhod razločevalnika (verjetnost, da je pomen pravilen), sorodnost z ostalimi tematikami, verjetnost povezave, globina tematike v hierarhiji in pozicija besede v besedilu (pojavitev na enem mestu ali razpršena pojavitev). Vse verjetnosti za besede, ki so manjše od p_{min}, se odstranijo iz seznama. S korakom V se seznam preostalih besed uredi po verjetnostih.

2.2. Razvrščanje zadetkov

Za razvrščanje zadetkov smo uporabili metodo razvrščanja BM25 skupaj z dodatnimi utežmi, ki so pridobljene z metapodatki dokumentov in opazovanjem aktivnosti uporabnikov.

2.2.1. BM25

BM25 (*Best Match 25*) je metoda razvrščanja, ki omogoča razvrščanje dokumentov po podobnosti na podlagi besed, ki se pojavljajo v dokumentih. BM25 v bistvu ni samo ena funkcija, temveč družina več funkcij, ki se razlikujejo po utežnih shemah in vrednostih parametrov pomembnosti za uteži. Največkrat se uporabljata uteži tf in idf. Utež tf *(angl. term frequency)* predstavlja frekvenco določene besede v dokumentu, utež idf *(angl. inverse document frequency)* pa pomembnost besede glede na celotno zbirko dokumentov.

$$tf(t,d) = ||n : t \in d|| \tag{3}$$

$$n(t) = ||d \in D : t \in d|| \tag{4}$$

$$idf(t) = \log \frac{||D|| - n(t) + 0.5}{n(t) + 0.5} \tag{5}$$

$||D||$ v enačbi 5 predstavlja kardinalnost (velikost) zbirke D, $n(t)$ dan z enačbo 4 pa število dokumentov, ki vsebujejo besedo t. Ocena BM25 ($s(d,q)$) je odvisna od uteži tf in idf ter parametrov k_1 in b. Splošna enačba izračuna ocene BM25 za dokument d glede na zahtevo q z besedami q_i je podana z enačbo 6.

$$s(d,q) = \sum_{i=1}^{||q||} idf(q_i) \cdot \frac{tf(q_i,d) \cdot (k_1 + 1)}{tf(q_i,d) + k_1 \cdot B}, \quad q_i \in q \tag{6}$$

2.2.2. Parametra k_1 in b

Ključno vlogo imata parametra k_1 in b, ki uravnavata težo uteži tf in težo dolžine dokumentov v končnem izračunu. Dolžina dokumentov se meri s številom besed. Parametra sta bila utemeljena zaradi dveh predpostavk o značilnostih, ki se pojavljajo pri pisanju dokumentov. Predpostavka o širini vsebine dokumenta (*angl. verbosity hypothesis)* govori o tem, da je lahko dokument daljši zaradi uporabe nepomembnih ali redundantnih besed, medtem ko predpostavka o obsegu dokumenta (*angl. scope hypothesis*) govori o daljših dokumentih zaradi dejanske koristne vsebine. V praksi gre za kombinacijo teh dveh predpostavk, zato potrebujemo ustrezno normalizacijo. Dolžino vsakega dokumenta lahko normaliziramo s povprečno dolžino dokumentov (enačba 8). Nadalje lahko to normalizacijo reguliramo s parametrom b kot kaže enačba 9.

$$||d|| = \sum_{i=1}^{n} tf(t_i,d), \quad t_i \in d \tag{7}$$

$$A = \frac{\sum_j^{||D||} ||d_j||}{||D||} \tag{8}$$

$$B = 1 - b + b \cdot \frac{||d_j||}{A} \tag{9}$$

Parameter k_1 uravnava pomembnost uteži tf, parameter b pa pomembnost dolžine dokumentov. Vrednosti parametrov k_1 in b lahko dobimo z optimizacijskimi postopki, navadno pa se uporabijo vrednosti $k_1 \in [1.2, 2.0]$ in $b = 0.75$.

2.2.3. Dodatne uteži

Vsak dokument je ob vsebini predstavljen tudi z metapodatki. Ti vsebujejo podatke o organizaciji, mentorju, študijskem programu, letu izida, primarnem jeziku ipd. Skupaj s temi podatki smo uporabili tudi uporabniške aktivnosti, ki se odražajo v številu ogledov, številu prenosov in

Tabela 2: Obdelava besed in tokenizacija besedne zveze ob lematizaciji

obdelava besed	porazdeljenega → porazdeljen računalniškega → računalniški sistema → sistem
obdelava besednih zvez	porazdeljenega računalniškega sistema → porazdeljen_računalniški_sistem

povprečni oceni dokumenta. Tako uporabljamo več kriterijev za razvrščanje, ki jih združimo v strategijo razvrščanja, v kateri ima vsak kriterij svojo prioriteto. Slednja je predstavljena s tabelo 3.

Tabela 3: Strategija razvrščanja rezultatov priporočanja.

pomembnost	**utež**
1	ocena BM25
2	leto izida
3	število prenosov
4	število ogledov
5	povprečna ocena
6	fakulteta
7	mentor
8	študijski program

2.2.4. Odstranjevanje neustreznih zadetkov z dinamično mejo

Včasih se lahko zgodi, da za rezultat ne bomo dobili dobrih priporočil. Razlog je v raznolikosti gradiv, ki nam ne zagotavlja, da obstaja toliko podobnih dokumentov, kolikor je naše minimalno želeno število priporočil. V tem primeru je potrebno razviti naknadno filtriranje z dinamično mejo. Če je v seznamu priporočil zaznana velika razlika med vrednostmi ocen BM25, lahko iz seznama odstranimo vse zadetke, ki se pojavijo za zaznano veliko razliko. S tem zagotovimo le pomensko ustrezne zadetke. Primer filtriranja z dinamično mejo je podan v tabeli 5, kjer je vrednost minimalnega želenega števila zadetkov enaka 5. Dinamična meja je postavljena na polovico največje ocene, kar lahko interpretiramo kot upoštevanje vseh zadetkov, ki so za več kot 50% pomensko povezani z najustreznejšim zadetkom. Nato izračunamo razlike Δ_i med največjo oceno in vsako naslednjo oceno. Iz seznama odstranimo tiste zadetke, kjer je razlika ocen večja od dinamične meje.

3. Vpliv različnih obdelav besedila

Pri našem delu smo preučevali obnašanje sistema pri različnih obdelavah vhodnih besedil. Preverjali smo kvaliteto rezultatov ob obdelavi na nivoju besed in besednih zvez. Ker so metode, ki smo jih uporabljali, namenjene obdelavi besed, smo upoštevanje besednih zvez omogočili tako, da smo besedno zvezo prepoznali in jo tokenizirali. Tako smo dobili tokenizirano besedno zvezo, ki se je v metodi za izračun podobnosti obnašala kot beseda. Primerjavo obdelave besed in tokenizacije besednih zvez prikazuje tabela 2.

Hkrati smo preučevali vpliv lematizacije. Podobno smo preučevali tudi vpliv avtomatskega pridobivanja ključnih besed s pomenskim označevanjem. Uspešnost je bila zmerjena na 80 naključno izbranih dokumentih v DKUM, kjer smo se omejili na gradiva iz Fakultete za elektrotehniko, računalništvo in informatiko Univerze v Mariboru (v nadaljevanju UM-FERI). Za vsako gradivo smo vrnili 5 priporočil in ugotavljali ustreznost priporočila za človeka. Ustreznost smo merili tako, da je skupina izbranih ljudi ocenjevala seznam priporočil z oceno med 0 in 5. Ta ocena pomeni število za človeka ustreznih zadetkov v seznamu priporočil. Ocene smo za vsako kombinacijo obdelav sešteli in izračunali odstotek ustreznosti. Med ocenjevalci so bili študenti zaključnih letnikov študijskih programov 1. in 2. bolonjske stopnje, kot tudi nekateri zaposleni na UM-FERI. Te ljudi smo smatrali kot domenske strokovnjake. Ocenjevalci so za vsak dokument dobili le informacije o naslovu in ključnih besedah.

Tabela 4: Rezultati kvalitete rezultata pri različnih obdelavah vhodnih dokumentov.

	brez lem.	**z lem.**
besede	51.80%	**55.70%**
besedne zveze	52.60%	**59.70%**
besede + pom. ozn.	47.40%	46.20%
besedne zveze + pom. ozn.	48.60%	**55.50%**

Dobljeni rezultati so pokazali, da se z vidika kvalitete priporočanja bolje obnese obdelava besedil na nivoju besednih zvez, kot tudi obdelava z lematizacijo. Slednja je bistveno pripomogla k boljšemu odstotku ustreznosti. Avtomatsko pridobivanje ključnih besed se je v večini primerov izkazalo kot dober način bogatenja konteksta v opisu dokumenta. Ob podrobnejši analizi spreminjanja vrednosti parametra p_{min} pri pomenskem označevalniku menimo, da bi lahko še povečali faktor vpliva na končen rezultat.

Tabela 5: Filtriranje z dinamično mejo; neustrezen zadetek izpade iz seznama zadetkov, saj ima meja vrednost 6.32.

Naslov vhodnega dokumenta			
Izdelava WYSIWYG urejevalnika s podporo matematičnim in kemijskim znakom			
Seznam zadetkov			
i	**Naslov dokumenta**	**Ocena BM25**	Δ_i
1	Izdelava e-dokumentov s programskim jezikom LaTeX	12.64	0
2	Objava znanstvenih dokumentov na spletnih straneh	10.91	1.73
3	Urejanje in obdelava besedil	7.05	5.59
4	Analiza uporabnosti urejevalnikov besedil v poslovnih sistemih	6.85	5.79
5	Virtualna radijska novinarska redakcija	5.93	**6.71**

4. Uporaba priporočanja v druge namene

Priporočanje dokumentov lahko uporabimo tudi za druge namene. Zaradi narave rezultata (to je seznam podobnih dokumentov) lahko priporočanje dokumentov uvrstimo v delovni tok kot korak obsežnejšega procesa. Primer takšnega procesa sta recimo odkrivanje potencialnih plagiatov in svetovanje pri izbiri mentorjev zaključnih del.

4.1. Izboljšava iskalnika

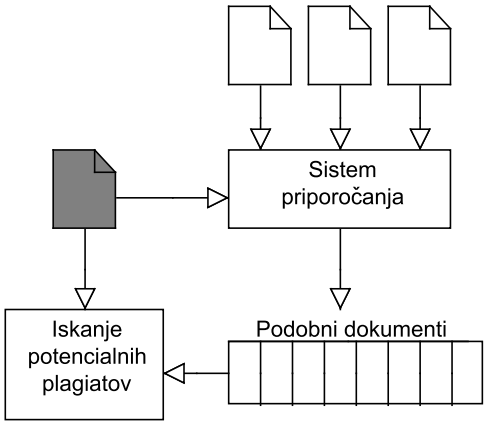
Trenutni iskalnik na DKUM deluje na podlagi ujemanja iskalnega niza z naslovi, ključnimi besedami in avtorji gradiv. Pri tem nekatere ustrezne zadetke uvršča nižje, saj ne upošteva sinonimov in drugih pomenskih informacij. Trenutni iskalnik prav tako ne smatra sklanjatev vhodnih besed kot ene besede, temveč kot različne. Posledično vrača le tista gradiva, ki vsebujejo točno podano sklanjatev ali pa jo imajo vsebovano v naslovu in ključnih besedah kot podniz. Iskalnik bi lahko izboljšali z upoštevanjem zadetkov, ki jih najde sistem priporočanja. Slednji je odporen na problem sklanjatev zaradi uporabe lematizacije in upošteva pomen dokumenta, ne le ujemanja nizov.

4.2. Iskanje potencialnih plagiatov

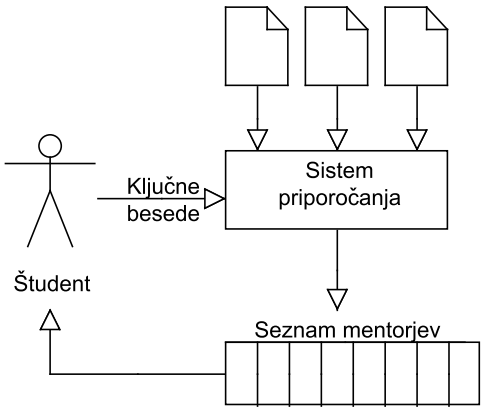
Pri iskanju potencialnih plagiatov navadno delamo primerjave med vsemi dokumenti. Z vključitvijo delovanja priporočanja dokumentov bi lahko eliminirali dokumente, ki si niso podobni, kar bi zmanjšalo obdelovano množico. S tem bi ustvarili seznam filtriranih dokumentov, katere naj sistem za iskanje potencialnih plagiatov upošteva pri nadaljnji obdelavi. Priporočanje dokumentov služi kot predhodna obdelava dokumentov, ki vrne le najbolj podobne dokumente in s tem pohitri dejanski postopek odkrivanja potencialnih plagiatov nad njimi (slika 2).

4.3. Svetovanje pri izbiri mentorja zaključnih del

Še ena aplikacija priporočanja bi lahko bila v obliki orodja, ki študentom svetuje izbiro mentorja pri zaključnih delih. Študent bi v aplikacijo vpisal ključne besede iz področja, ki ga zanima, aplikacija pa bi med diplomskimi, magistrskimi in doktorskimi nalogami našla najbolj ustreznega mentorja (slika 3). To bi potekalo tako, da bi najprej poiskali dokumente, ki so podobni vhodnim ključnim besedam, nato pa bi prešteli, kolikokrat se mentorji pojavljajo v seznamu. V primeru izenačenja bi lahko uporabili še dodatno utež glede na to, kateri dokument ima največjo podobnost z vhodnimi ključnimi besedami.



Slika 2: Vloga sistema priporočanja znotraj procesa iskanja potencialnih plagiatov.



Slika 3: Uporaba sistema priporočanja pri svetovanju mentorja.

5. Zaključek

Prispevek predstavlja sistem vsebinskega priporočanja dokumentov v digitalni knjižnici Univerze v Mariboru. S pomočjo takšnega sistema lahko uporabnikom ponudimo več vsebine, hkrati pa izboljšamo nekatere že obstoječe funkcionalnosti v digitalni knjižnici. Sistem priporočanja obdeluje vsebino dokumentov in pri razvrščanju zadetkov upošteva tudi uporabniške aktivnosti. Predstavili smo tudi notranje funkcije sistema priporočanja v smislu obdelave besedil, pomenskega označevanja in razvrščanja zadetkov, katerih delovanje smo tudi podrobneje opisali. Pokazali smo, da se je najbolje obnesla obdelava besedil na nivoju besed v navezi s pomenskim označevanjem in lematizacijo. V splošnem je obdelava z lematizacijo doprinesla k boljši ustreznosti priporočil. Opisali smo kako lahko predstavljeni sistem priporočanja izboljša trenutne funkcionalnosti DKUM v smislu izpopolnjenega iskanja in potencialnega iskanja plagiatov ter omogoča izdelavo aplikacije za svetovanje mentorjev pri zaključnih delih. Razvit sistem priporočanja se od junija 2012 aktivno uporablja na straneh DKUM.

Nadaljnje delo obsega preučevanje drugih metod za iskanje vsebinske podobnosti. Med te metode spada latentna semantična analiza, katero bi v prihodnosti radi vključili v delovanje sistema, saj omogoča boljšo razpoznavo sinonimov. Rezultat bi tako bila hibridna dvofazna funkcija razvrščanja, ki bi v prvi fazi grobo filtrirala zadetke s pomočjo BM25, v drugi fazi pa fino filtrirala z latentno semantično analizo in upoštevanjem uporabniških aktivnosti. Prav tako bi želeli preizkusiti kvaliteto delovanja sistema, če bi dokumente predhodno gručili in klasificirali glede na tematiko. Trenutno delovanje bi morda lahko še izboljšali z optimizacijo parametrov BM25 glede na korpus besedil. Želeli bi preveriti vpliv parametrov BM25 na rezultat priporočanja in izvesti uglaševanje teh parametrov s pomočjo nekaterih optimizacijskih postopkov. Predvsem želimo zasnovati še objektivno metriko za ocenjevanje priporočanja, na podlagi števila obiskanih priporočil.

6. Literatura

J. Brezovnik in M. Ojsteršek. 2011. Textproc - a natural language processing framework and its use as plagiarism detection system. *International Journal of Education and Information Technologies*, 1(5):293–300.

J. Brezovnik. 2009. Programsko orodje za procesiranje besedil v naravnem jeziku. Magistrsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Univerza v Mariboru.

M. Burjek. 2011. Wikifikacija vsebin v digitalni knjižnici UM. Diplomsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Univerza v Mariboru.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, in R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

S. T. Dumais. 2004. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38:188–230.

E. Garcia. 2011. A tutorial on Okapi BM25. http://www.miislita.com/information-retrieval-tutorial/okapi-bm25-tutorial.pdf.

G. Hrovat. 2010. Izdelava oblikoslovnega označevalnika za slovenski jezik in primerjava z drugimi rešitvami. Diplomsko delo, Fakulteta za elektrotehniko, računalništvo in informatiko Maribor, Univerza v Mariboru.

D. Lemire in A. Maclachlan. 2005. Slope one predictors for online rating-based collaborative filtering. V: *Proceedings of SIAM Data Mining (SDM'05)*, str. 471–475.

Y. Lv in C. Zhai. 2012. A log-logistic model-based interpretation of TF normalization of BM25. V: *Proceedings of the 34th European Conference on Information Retrieval*.

P. Melville in V. Sindhwani. 2010. Recommender systems. V: *Encyclopedia of Machine Learning*, str. 829–838.

F. Ricci, L. Rokach, in B. Shapira. 2011. Introduction to recommender systems. V: *Recommender Systems Handbook*, str. 1–35.

A Survey of Chabot Systems through a Loebner Prize Competition

Luka Bradeško*, Dunja Mladenić*

* Artificial Intelligence laboratory, Jozef Stefan Institute,
Ljubljana Slovenia
{luka.bradesko, dunja.mladenic}@ijs.si

Abstract

Starting in 1966 with the introduction of the ELIZA chatbot, a great deal of effort has been devoted towards the goal of developing a chatbot system that would be able to pass the Turing Test. These efforts have resulted in the creation of a variety of technologies and have taken a variety of approaches. In this paper we compare and discuss the different technologies used in the chatbots which have won the Loebner Prize Competition, the first formal instantiation of the Turing Test. Although there was no game changing breakthrough in the chatbot technologies, it is obvious they evolved from the very simple pattern matching systems towards complicated patterns combined with ontologies and knowledge bases enabling computer reasoning.

Pregled najboljših programov za klepetanje z računalnikom iz Loebnerjevega tekomovanja

Vse od leta 1966 naprej, ko se je pojavil prvi program za klepet ELIZA, se poskuša razviti program ki bi bil sposoben opraviti Turingov test. V tem času je bilo razvitih veliko različnih rešitev in pristopov k reševanju tega kompleksnega problema. V tem članku primerjamo in opišemo pristope pri programih za klepetanje ki so zmagali na Loebnerjevem tekmovanju. Loebnerjevo tekmovanje je prva formalna izvedba Turingovega testa in se izvaja že vse od leta 1991. Kljub temu, da do danes še nobenemu programu ni uspelo prestati testa, so močno napredovali. Iz zelo enostavnih algoritmov z iskanjem vzorcev besedila, so se z leti razvili v kompleksne sisteme prepoznavanja vzorcev, ki vključujejo tudi ontologije in tako do neke mere že omogočajo tudi računalniško sklepanje.

1. Introduction

There are numerous approaches to human-computer interaction. One of them is via natural language (NL), which again has more sub-approaches and goals. In this paper we focus on chatbots, which are gaining popularity again due to success of virtual assistants such as Siri, Evi, S-Voice, Jeannie, CallMom and others.

The main purpose and idea of the so called chat-bots is that the computer is performing a natural language conversation with human clients which should be as human-like as possible. Based on the task bot was made for, the conversations then usually serves some specific purpose such as searching the web, organizing files on the computer, setting up appointments, etc.

Currently the biggest challenge that existing chat-bots have is maintaining of the context and understanding the human inputs and its responses. Most of the existing bots still work just on the pattern matching of inputs and then trying to find a scripted response which matches the input. This approach however cannot result in a fully satisfying conversation or lead a conversation with some specific purpose.

Due to the obvious drawbacks of scripted responses, developers and researchers kept adding new functionalities to the existing ways how chatbots works, converging mostly to the use some sort of ontologies and remembering facts from the conversation. While these improvements made chatbots much more successful, at the same time introduced a number of different approaches, systems and solutions to the same problem.

The goal of this paper is to make a survey of chatbot technologies and approaches and thus make it easier for a developer and/or a researcher on to which technology to use for the research or further development of the chatbot system.

2. Early chatbots

There were numerous chatbots and chatbot technologies already before the first Loebner competition, mostly in games and focused domain expert systems. It is not known how well they performed and they were never compared against each other.

The very first known chatbot was Eliza, which was developed in 1966. Its goal was to behave as a Rogerian psychologist. It used simple pattern matching and mostly returned users sentences in a form of questions. Its conversational ability was not very good, but it was enough to confuse people at a time when they were not used to interact with computers and to start the development of other chatbot systems. The very first online implementation of Eliza was done by the researches at Jozef Stefan Institute in Ljubljana, Slovenia and is still available[1] for testing.

The first such a system that was actually evaluated using some sort of Turing Test was PARRY (Colby, 1975). Parry was designed to talk as a paranoid person. Its transcripts were given to psychiatrists together with transcripts from real paranoia patients for comparison. The psychiatrists were able to make the correct identification only 48% of the time.

3. Loebner Prize Competition

Loebner Prize Competition[2] (Loebner prize for artificial intelligence) is an annual competition for conversational agents (chatbots), where they are being

[1] http://www-ai.ijs.si/eliza-cgi-bin/eliza_script
[2] http://www.loebner.net

Year	Chatbot	Technology	Language Tricks
1991	PC Therapist III (Weintraub, 1986)*	parsing, pattern matching, word vocabulary, remembers sentences	non sequitur, canned responses
1992	PC Professor (Weintraub, 1986)*		
1993	PC Politician (Weintraub, 1986)*		
1994	TIPS (Whalen, 1994; Hutchens, 1997)	Pattern matching, database like system	Model of personal history
1995	PC Therapist (Weintraub, 1986)*	Same as in 1991	Same as in 1991
1996	HeX (Hutchens, 1997)	Pattern matching, Markov chain models to construct some replies	database of trick sentences, Model of personal history, not repeating itself
1997	CONVERSE (Levy, 1997)	Statistical parser, pattern matching, modular with weighted modules, WordNet synonyms, list of proper names, ontology, database for storing facts	Proactivity
1998	Albert One (Garner, 1995)	Pattern matching, hierarchical composition of other chat bots (Eliza, Fred, Sextalk)	Proactive monologues
1999	Albert One (Garner, 1995)		
2000	A.L.I.C.E (Wallace, 2003)	AIML (Artificial Intelligence Mark-up Language – advanced pattern matching)	
2001	A.L.I.C.E (Wallace, 2003)		
2002	Ella (Copple, 2009)	Pattern matching, phrase normalization, abbreviation expansion, WordNet	Monologues, not repeating itself, identify gibberish, play knock-knock jokes
2003	Jabberwock (Pirner, 2003)	Parser, pattern matching – simpler than AIML, Markov Chains, Context free grammar (CFG)	
2004	A.L.I.C.E (Wallace, 2003)	Same as in 2000	
2005	George (Carpenter, 2006)	Based on Jabberwacky chatbot. No pattern matching or scripts. Huge database of people's responses.	
2006	Joan (Carpenter, 2006)		
2007	UlrtraHAL by Robert Medeksza*	Combination of VB code and pattern matching scripts	
2008	Elbot (Roberts, 2007)*	Commercial NLI system	
2009	Do-Much-More (Levy, 2009)*	Commercial property of Intelligent Toys Ltd.	
2010	Suzette (Wilcox, 2011)	ChatScript (AIML successor. Concepts, triples, variables)	
2011	Rosette (Wilcox, 2011)		
2012	Chip Vivant (Embar, 2011)	Not publicly disclosed. Common ontology and AI, responses taken from ChatScript (but not in ChatScript format or engine).	

Table 1: List of Loebner winners with appropriate technologies

tested via a method called Turing Test (Turing 1950). Loebner Competition is known to be the first formal instantiation of a Turing Test.

There is a controversy whether this competition is really contributing to the development of AI, or it is blocking it (Shieber, 2006; Maughan, 2002; Hutchens, 1997; French, 1990). The doubt is because the competition is forcing chatbots to pretend to be a human which causes bots to simply pretend they are thinking without real intelligence. Some of the chatbots even fake the spelling mistakes and corrections. Another stated flaw is that the competition causes people to work apart instead to collaborate and thus lead to many incompatible chatbot technologies.

However this competition still methodologically compares chatbot technologies, rates them in a conversational sense and thus gives some sort of a general feedback over the used technologies.

3.1. Turing Test

Turing Test is also known as the Imitation Game. In this test, the goal for the chatbot is to maintain a conversation which is indistinguishable from a human conversation. The usual way to apply the test is that there is a human observer (judge), who is asking questions or having a conversation with someone over the computer link. That someone can be a computer (chatbot) or a person. If on the other side there is a chatbot and the judge would think it is a person, then the chatbot would pass the test.

3.2. Winning chatbots

Regardless of the fact that none of the existing chatbots were able to pass the Turing Test, each there is a winner of the Loebner Prize Competition that appears most human from all the competing chatbots. The list of each year's winners together with the used technologies can be seen on the Table 1. We tried to separate the technology part into the technical approaches and algorithms and the language and approach tricks used to confuse judges on the Turing Test. The technologies are explained in the following chapters in more detail. The winners marked with asterisk (*) are commercial programs and thus their technologies and internal structure is not publicly available.

4. Technical approaches and algorithms

4.1. Pattern Matching

This is by far the most common approach and technique used in chatbots. Variations of some pattern matching algorithm exist in every existing chatbot system.

The pattern matching approaches can vary in their complexity, but the basic idea is the same. The simplest patterns were used in earlier chatbots such as ELIZA and PC Therapist. For example:

Pattern: "I need a ?X"

Response: "What would it mean to you if you got a ?X?"

4.2. Parsing

Textual Parsing is a method which takes the original text and converts it into a set of words (lexical parsing) with features, mostly to determine its grammatical structure. On top of that, the lexical structure can be then checked if it forms allowable expression (syntactical parsing).

The earlier parsers were very simple, looking for recognizable keywords in allowed order. Example of such parsing would be that sentences "please take the gold" and "can you get the gold" would be both parsed into "take gold". With this approach the chatbot with a limited set of patterns can cover multiple input sentences.

The more complicated parsers used in latter chatbots do the complete grammatical parsing of the natural language sentences.

4.3. Markov Chain Models

The Idea behind Markov Chain Models is that each occurrence of a letter or a word in some textual dataset occurs with a fixed probability. The order of a model means how many consecutive occurrences the model takes into the account. For example if an input text is "agggcagcgggcg", then the Markov model of order 0 predicts that letter 'a' occurs with a probability 2/13. The model with order 1 would state that each letter still occurs with a fixed probability, but that probability depends on the letter before.

In chatbots the Markov Chain Models were being used to construct responses which are probabilistically more viable and thus more correct. In some cases (HeX) these models were even used to generate a nonsense sentence that sounds right, as a failback method.

4.4. Ontologies (semantic nets)

Ontology or semantic network as it is called in some chatbot systems is a set of hierarchically and relationally interconnected concepts. These concepts can have natural language names and can be used directly in chatbots, to figure out hyponyms, synonyms and other relations between the concepts. Example of such an ontology which is often used or at least tried to be used in chatbots is OpenCyc[3] (Lenat, 1995). The advantage of the ontologies is that the concepts are interconnected into a graph, which enables computers to search through and using special reasoning rules even imply new statements (reasoning).

[3] http://www.opencyc.org/

4.5. AIML

AIML's syntax is XML based and consists mostly of input rules (categories) with appropriate output. The pattern must cover the entire input and is case insensitive. It is possible to use a wildcard (*) which binds to one or more words. The simplest example of it can be written like seen on Figure 1. Due to simple and effective explanation, this and as well the other examples were taken from the paper Beyond Façade: Pattern Matching for Natural Language Applications (Wilcox, 2011).

```
<Category>
<pattern> I NEED HELP * </pattern>
<template>Can you ask for help in the form of a question?
</template>
</category>
```

Figure 1: Simple AIML rule (pattern).

The real power of AIML lies in its ability to recursively call itself (Wallace, 2003; Wilcox, 2011). It can submit input to itself using the <*srai*> tag and the contents of * using <*star/*>. Example of such recursion can be seen on Figure 2, where the AIML engine forwards everything before the phrase "right now" to another pattern. The second pattern then forwards everything after the phrase "can you please".

```
=> Can you please tell me what LINUX is right now?
<category>
<pattern> * RIGHT NOW <\pattern>
<template> <srai><star/></srai></template>
</category>
=> CAN YOU PLEASE TELL ME WHAT LINUX IS
<category>
<pattern> CAN YOU PLEASE * <\pattern>
<template> <srai> Please <star/></srai></template>
</category>
```

Figure 2: AIML recursion.

AIML allows chatbots to have topics which give it a way to prioritize the patterns. It has the <*that*> pattern as well, which if it matches the output of the previous sentence it has priority over the other rules.

4.6. ChatScript

ChatScript is successor of the AIML language. It focuses on the better syntax which makes it easier to maintain. It fixes the zero word matching problems and introduces a bunch of additional functionalities such as concepts, continuations, logical and/or, variables, fact triples and functions. With these functionalities it tries to cover the need for ontologies inside the scrip itself. Example of a script defining a concept of meat and one pattern can be seen on Figure 3.

```
concept: ~meat ( bacon ham beef meat flesh veal lamb
chicken pork steak cow pig )
s: ( I love ~meat ) Do you really? I am a vegan.
```

Figure 3: Chatbot concept definition and simple pattern.

5. Language approaches and tricks

5.1. Non Sequitur

Non sequitur (Latin) is an argument that has conclusions which does not imply from its premises. Example from everyday speech would be: "Life is life and fun is fun, but it's all so quiet when the goldfish die."

5.2. Simulating keystrokes and typing errors

The chat protocol that is used in Loebner Competitions works in a way that the judges see the sentences as they are being typed. This forces the chatbots to "pretend" they are tying word by word. Some of the bots even fake the spelling mistakes and backspacing.

5.3. Canned responses

Canned responses are predefined (hard coded) responses to questions. To some extent all of the chatbots patterns could be counted as canned responses if bot only uses these. This would vastly increase the number of patterns and would make them even more unmanageable, so these responses are usually used only for things which cannot be covered with the main chatbot technology.

5.4. Model of personal history

With the goal for a chatbot to appear more convincing, developers are inserting a personal story (imaginary or based on a real person) into chatbot responses. This includes memories from the past, childhood stories, parents, interests, political and religious views etc.

6. Conclusions

Through the years of Loebner Prize Competitions we can see how the chat technologies evolved from the very simple pattern matching systems, over the statistical models of chats, towards complicated patterns in combination with ontologies and knowledge bases. It can be argued that even the newest approaches (ChatScript, AIML) are still just a small improvement over the ELIZA pattern matching idea and that the biggest improvement is the amount of scripts written for it. We agree that there is some truth in it; however it is notable that the recent developments, especially with ChatScript the chatbots are moving out of the scripted era.

It is obvious that there is a trend towards semantics, which can lead to a conclusion that future chat bots will evolve from the pattern matching, towards more semantic approaches and will probably start to incorporate more and more computer reasoning systems.

Independently of Loebner competitions and other chat bot systems, IBM in 2004 started developing a question answering system (Watson), which won the show in 2011. Technically it is not a chatbot, since it is only able to answer questions, but their research currently leads into that direction as well. On top of more than 100 different text processing approaches, they used ontologies such as DBPedia, WordNet, and Yago for support to other techniques and to enable reasoning, which goes in hand with other newer chatbot approaches.

7. References

Colby K. M., 1975, Artificial Paranoia: A computer program for the study of natural language communication between man and machine, Communications of the ACM, vol. 9, pp. 36-45.

Copple K. L., 2009. Bringing AI to Life: Putting Today's Tools and Re-sources to Work.

Embar M., 2011. Chip Vivant, http://www.chipvivant.com/motivations-and-functionality/

French R., 1990. Sub cognition and the limits of the Turing test, Mind, vol. 99, pp. 53-65.

Fryer L., Carpenter R., 2006. Emerging technologies: Bots as language learning tools. Language Learning and Technology, 10(3). Pp. 8-14, available at http://llt.msu.edu/vol10num3/pdf/emerging.pdf

Garner R., 2005. Multifaceted Conversational Systems, Colloquium on Conversational Systems, University of Surrey, available at http://www.robitron.com/Robby/Multifaceted.ppt

Hutchens L. J., 1997. How to pass the Turing Test by Cheating, University of Western Australia, available at http://www.nyu.edu/gsas/dept/philo/courses/mindsandmachines/Papers/hutchens96how.pdf

Lenat, D. B., 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM, 38(22).

Levy D., Catizone R., Battacharia B., Krotov A., Wilks Y., 1997. CONVERSE: A Conversational Companion, In Proceedings of 1st International Workshop on Human-Computer Conversation, available at http://staffwww.dcs.shef.ac.uk/people/Y.Wilks/papers/converse.pdf

Levy D., 2009. DO-MUCH-MORE Chatbot Wins 2009 Loebner Prize for Computer Conversation, News from the Benelux Association for Artificial Intelligence, Vol. 26, No. 4, p. 67-78, available at http://www.unimaas.nl/bnvki/archive/2009/26.4.pdf

Maughan R., 2002. Conversational Agents, School of Computer Science and Information Technology, University of Nottingham

Pirner J., 2003. About Jabberwock, available at http://www.abenteuermedien.de/jabberwock

Roberts F., Gülsdorff B., 2007. Techniques of Dialogue Simulation, Lecture Notes in Computer Science p. 420-421.

Shieber S. M., 2006, Does the Turing Test Demonstrate Intelligence or Not?, AAAI-06, Boston, Ma.

Turing A.M., 1950. Computing Machinery and Intelligence, MIND the Journal of the Mind Association, vol. LIX, pp. 433-460.

Wallace R., 2003. The elements of AIML style. ALICE AI Foundation, available at http://www.alicebot.org/style.pdf

Weintraub J., 1986, History of the PC Therapist, available at http://www.loebner.net/Prizef/weintraub-bio.html.

Whalen T., 1994, My experience at Loebner prize, available at http://thomwhalen.com/ThomLoebner1994.html

Wilcox, B., 2011. Beyond Façade: Pattern Matching for Natural Language Applications, available at http://www.gamasutra.com/view/feature/134675/beyond_fa%C3%A7ade_pattern_matching_.php

Tehnologije govorjenega jezika v pametnih nadzornih sistemih

Simon Dobrišek*, Boštjan Vesnicer†, France Mihelič*

*Univerza v Ljubljani, Fakulteta za elektrotehniko, LUKS
Tržaška 25, Ljubljana, Slovenija
{simon.dobrisek, france.mihelic}@fe.uni-lj.si

†Alpineon d.o.o.
Ulica Iga Grudna 15, Ljubljana, Slovenija
bostjan.vesnicer@alpineon.si

Povzetek

V članku je podan pregled obstoječe in možne uporabe tehnologij govorjenega jezika v t.i pametnih (inteligentnih) nadzornih sistemih. Tehnologije, ki omogočajo samodejno razpoznavanja govora, govorcev in njihovega psihofizičnega stanja s pomočjo računalniške analize govornega zvočnega signala, odpirajo povsem nove dimenzije razvoja pametnih nadzornih sistemov. Predstavljeno je trenutno stanje razvoja teh tehnologij ter različni scenariji uporabe in primeri tovrstnih sistemov. Naslovljena so tudi širša pravna in etična vprašanja, ki jih odpira razvoj in uporaba tovrstnih tehnologij.

Spoken language technologies in smart surveillance systems

The paper provides an overview of existing and potential use of spoken language technologies in so-called smart (intelligent) surveillance systems. Technologies that enable automatic speech and speaker recognition as well as their psychophysical state by computer analysis of acoustic speech signals provide an entirely new dimension to the development of smart surveillance systems. The paper investigates the current state of development of these technologies and different application scenarios as well as existing examples of such systems. It also addresses the broader legal and ethical issues raised by the development and use of such technologies.

1. Uvod

Izjemen tehnološki napredek v zadnjih desetletjih je omogočil razvoj vedno bolj kompleksnih in vseprisotnih nadzornih tehnologij, katerih glavni namen je izboljšanje učinkovitosti pri zaznavanju in preprečevanju kriminala in terorizma. Pri sodobni varnostni paradigmi se pojavlja potreba po prehodu iz retroaktivnega forenzičnega preiskovanja preteklih varnostnih incidentov v proaktivno sprotno odzivanje na samodejno zaznane varnostne incidente in grožnje s pomočjo t.i. inteligentnih oziroma pametnih nazornih tehnologij.

Pametne nadzorne tehnologije so integrirani računalniški sistemi, ki vključujejo tehnologije za zajem raznih senzorskih in drugih nadzornih podatkov ter računalniške postopke za njihovo samodejno obdelavo, ovrednotenje in analizo, kakor tudi postopke za samodejno odločanje oziroma podporo odločanju varnostnih operaterjev na podlagi rezultatov samodejne analize zbranih podatkov. Ti sistemi predstavljajo razvojni napredek v primerjavi s tradicionalnimi nadzornimi sistemi, ki navadno vključujejo le osnovno infrastrukturo za zajemanje, shranjevanje in distribucijo nadzornih podatkov, nalogo zaznavanja oziroma preiskovanja varnostnih incidentov in groženj pa v glavnem še vedno prepuščajo razmeroma neučinkovitim človeškim operaterjem. Tipični tovrstni tradicionalni sistemi so t.i CCTV video-nadzorni sistemi.

Pri novejših CCTV sistemih se danes z metodami samodejne računalniške analize video vsebin že poskuša doseči zmožnost samodejnega zaznavanja in razpoznavanja varnostno sumljivih dogodkov, okoliščin ali obnašanja ljudi (Piciarelli et al., 2011). Mnogih varnostnih incidentov pa ni mogoče zaznati zgolj z analizo videa. Povsem novo dimenzijo inteligentnega nadzora ponuja integracija video-nadzornih sistemov z inteligentnimi avdio-nadzornimi sistemi, ki omogočajo samodejno varnostno analizo zajetega zvočnega signala (Onut et al., 2011).

Avdio-nadzorni sistemi z zmožnostjo tristošestdeset stopinjskega pokrivanja prostora omogočajo razširitev nadzorovanega prostora preko vidnega polja navadnih nadzornih kamer. S samodejnim zaznavanjem in razpoznavanjem varnostno sumljivih zvokov, kot je kričanje, klicanje na pomoč, glasno izgovarjanje groženj ali pa zvoki razbijanja predmetov, korakov, odpiranje vrat, poka pištole ipd., lahko nadzornemu sistemu dodamo zmožnost samodejnega osredotočanja pozornosti v smeri izvorov teh sumljivih zvokov. Samodejno razpoznani varnostni incidenti bi lahko sprožili ustrezen odziv sistema, kot je samodejni klic policije in reševalnih služb ali umetno govorno opozarjanje in obveščanje prisotnih ljudi o zaznanem varnostnem incidentu.

Tehnologija	Varnostno-nadzorna uporaba
Razpoznavanje govora	Razpoznavanje izgovorjenih groženj in drugih varnostno sumljivih izjav ter neposrednih in prikritih klicev na pomoč
Razpoznavanje govorcev	Razpoznavanje znanih kriminalcev in varnostno sumljivih posameznikov
Razpoznavanje psihofizičnega stanja govorca	Razpoznavanje agresivnega in drugače varnostno sumljivega obnašanja ali prestrašenosti ljudi
Umetno tvorjenje govora	Govorno obveščanje prisotnih ljudi o zaznanem varnostnem incidentu

Tabela 1: Pregled možnih varnostno-nadzornih uporab tehnologij govorjenega jezika.

Med razvijajočimi se tehnologijami govorjenega jezika je precej takšnih, ki jih je mogoče neposredno uporabiti v inteligentnih avdio-nadzornih sistemih. Osnovni pregled teh tehnologij in njenih možnih uporab v različnih varnostno-nadzornih scenarijih je podan v tabeli **Tabela 1**, v

naslednjem poglavju pa bolj podrobno obravnavamo nekaj različnih možnih varnostno-nadzornih scenarijev, pri katerih pridejo v poštev navedene tehnologije.

Uporaba teh tehnologij odpira precej pravnih in etičnih vprašanj, ki jih naslavljamo v zadnjem delu članka. Dejstvo pa je, da se v Evropi in po svetu te tehnologije razvija in uporablja tudi v vedno bolj integriranih nadzornih sistemih.

2. Varnostno-nadzorni scenariji

Pri razvoju novih tehnologij navadno izvedemo študijo in ovrednotenje scenarijev njihove smiselne uporabe. Pri pametnih nadzornih sistemih, ki vključujejo tehnologije govorjenega jezika, pridejo v poštev varnostno-nadzorni scenariji, ki se kakor koli nanašajo na samodejno računalniško analizo zajetih zvočnih govornih signalov.

2.1. Avdio nadzor komunikacijskih kanalov

Uporaba tehnologij govorjenega jezika za nadzor avdio-komunikacijskih in informacijskih kanalov je med vsemi obravnavanimi varnostno-nadzornimi scenariji še najbolj znana in tudi razvita. Zaradi nacionalnih varnostnih interesov razvoj teh tehnologij v največji meri neposredno podpirajo kar vlade različnih najbolj razvitih držav.

Na tem področju se za proaktivne nadzorne sistem štejejo predvsem sistemi za samodejno zaznavanje in razpoznavanje (identifikacijo) govorcev, ki jih varnostno-obveščevalne službe obravnavajo in spremljajo zaradi utemeljenih sumov storitve kaznivih dejanj in katerih govor se pojavi v avdio-komunikacijskih kanalih. S tehnologijo razpoznavanja govora pa se poskuša samodejno zaznati in razpoznati izgovorjena sporočila, ki so varnostno sumljiva (denimo, napeljevanje in napovedovanje kriminalnih ali terorističnih dejanj ipd) in so potrebna proaktivne in preventivne obravnave varnostno-obveščevalnih služb.

Poleg navedenih tehnologij je za tovrstne varnostno-nadzorne scenarije uporabna tudi tehnologija razpoznavanja govorjenega jezika, s katero se lahko doseže, da nadzorni sistem samodejno zazna in razpozna jezik govorca ali celo njegov materni jezik, ko ta govori tuj jezik.

2.2. Integrirani avdio-vizualni nadzor prostorov

Varnostni nadzor prostorov se danes izvaja predvsem z video-nadzornimi sistemi. Večino tovrstnih varnostnih scenarijev je mogoče razširiti z dodajanjem funkcije pametnega avdio nadzora. Ta razširitev predvideva obstoj možnosti namestitve mikrofonov za zajemanje zvočnih signalov. Najsodobnejše motorizirane mrežne nadzorne kamere imajo pogosto že vgrajen mikrofonski vhod (slika 1) in z njihovo primerno namestitvijo lahko postavimo nadzorno polje mikrofonov. S sodobnimi postopki obdelave zvočnih signalov iz polja mikrofonov lahko izvedemo časovno in prostorsko lokalizacijo zvočnih virov, ki se pojavljajo v nadzorovanem prostoru (Keyrouz, 2007). Po lokalizaciji zvočnih virov lahko izvedemo postopke samodejnega razpoznavanja varnostno sumljivih zvokov, med katerimi je lahko tudi govor in drugi človeški glasovi, kot so kričanje, izgovarjanje groženj, klici na pomoč ipd. Po lokalizaciji in razpoznavanju varnostno sumljivih zvokov ali glasov bi takšen sistem lahko preusmeril pozornost in vidno polje motoriziranih video-nadzornih kamer v smeri teh zvočnih virov in pritegnil pozornost morebitnih človeških varnostnih operatorjev in služb.

Slika 1: Primer mrežne motorizirane video-nadzorne kamere firme Axis z mikrofonskim vhodom.

Tipični varnostno-nadzorni scenariji, ki bi vključevali takšne razširjene sisteme, so danes že skoraj običajni varnostni nadzori javnih prostorov, kot so mestne ulice, potniške postaje, podhodi in dvigala ter javna parkirišča, garaže, igrišča in tudi javna prevozna sredstva.

V primeru zaprtih varovanih javnih prostorov, kjer je večja možnost poskusov ropa (to so na primer zlatarne, pošte, banke ipd), bi samodejnemu razpoznavanju izgovorjenih groženj, kričanja in klicev na pomoč lahko dodali tudi funkcijo za samodejno govorno proženje alarma ter klic policije in reševalnih služb z izgovarjanjem vnaprej predvidenih prikritih prožilnih govornih izjav. S tehnologijami samodejnega razpoznavanja psihofizičnega stanja govorca pa bi bilo sistem možno usposobiti tudi tako, da bi samodejno zaznal izrazito agresivnost ali prestrašenost govorečih prisotnih ljudi. Tovrstni sistemi bi tako lahko reševali življenja, saj so znani primeri (ropi na poštah in v zlatarnah), ko v ropu ranjeni ljudje niso uspeli sprožiti klasičnega alarma ali pravočasno priklicati pomoči.

2.3. Samostojni avdio-nadzor prostorov

V primerih, ko video nadzor še ni ali ne more/sme biti vzpostavljen (slaba vidljivost ali varovanje zasebnosti), je mogoče razmišljati o uporabi samostojnih pametnih avdio-nadzornih sistemov. Takšni sistemi bi prišli denimo v poštev v javnih prostorih, kjer je večja možnost kriminalnih dejanj v nočnem času (spolno nadlegovanje, poskusi ropa ipd). Primeri takšnih prostorov so odprta ali pokrita slabo osvetljena parkirišča, garažni koridorji, cestni podhodi in prehodi, javna dvigala, javna stranišča ipd. V vseh teh primerih bi prišel v poštev samostojni avdio-nadzorni sistem, ki bi imel poleg samodejnega razpoznavanja varnostno sumljivih zvokov, kot je razbijanje in drug neobičajen hrup, tudi zmožnost samodejnega razpoznavanja kričanja, izgovorjenih groženj, klicev na pomoč ipd. V primeru, ko so v prostor nameščeni tudi zvočniki (tipično je to v dvigalih), bi lahko sistem tudi govorno opozoril prisotne na zaznano neobičajno obnašanje, kar bi lahko prisotne obvarovalo ali odvrnilo od kriminalnih dejanj.

Avdio-nadzorne sisteme je mogoče uporabiti tudi za druge namene, kot je le preprečevanje kriminala in terori-

zma. Možno jih je namreč uporabiti tudi v zasebnih varovanih stanovanjih, v katerih bivajo ostareli, bolni ali onemogli ljudje, ki jih zaradi zasebnosti moti video nadzor. V teh primerih bi avdio-nadzorni sistem lahko uporabili za samodejno razpoznavanje klica na pomoč ali poslabšanega psihofizičnega stanja oziroma stiske varovancev. Tak sistem bi lahko samodejno razpoznal tudi druge neobičajne zvoke v prostoru, kot je hrup padajočih predmetov ipd. Tovrstni sistemi bi lahko preprečili pogoste neprijetne dogodke, ko osamljeni onemogli starejši varovanci po padcu na svojem domu ležijo tudi po nekaj dni na tleh in ne uspejo priklicati pomoči.

Vsi navedeni varnostno-nadzorni scenariji ponujajo precej možnosti bolj intenzivnega razvoja tehnologij govorjenega jezika, ki presega uporabo v klasičnih uporabniških vmesnikih za mobilne in druge informacijsko-komunikacijske platforme.

3. Varnostno-nadzorne govorne tehnologije

Pri razvoju in uporabi obstoječih tehnologij govorjenega jezika v pametnih nadzornih sistemih se izkaže, da jedro teh tehnologij navadno ni potrebno posebej prilagajati varnostno-nadzornemu področju uporabe. Še največ težav se pojavlja pri pridobivanju primernih zbirk zvočnih govornih posnetkov, ki ustrezajo izbranim varnostno-nadzornim scenarijem in so nujno potrebne za izvedbo raznih učnih postopkov in ovrednotenje zanesljivosti delovanja sistemov.

3.1. Varnostno-nadzorne zbirke govornih posnetkov

Pri pridobivanju varnostno-nadzornih zbirk govornih posnetkov je možnih več pristopov in vsak ima svoje slabosti in prednosti. Pridobivanje govornih posnetkov, ki verodostojno odražajo obravnavan varnostno-nadzorni scenarij, je razmeroma zahtevno in pri tem se navadno poslužujemo ene od treh metodologij.

Pri prvi se zanašamo na snemanje govora v igranih razmerah, ki jih izvedemo posebej za te potrebe, pogosto pa uporabimo kar posnetke igranih ali dokumentarnih filmov, ki vsebujejo primerne filmske sekvence. Druga možnost je, da pri pridobivanju govorne zbirke sodelujejo prostovoljci, ki jih z različnimi psihološkimi tehnikami spodbudimo k pričakovanemu obnašanju. Najtežje pa je pridobiti posnetke resničnih varnostno-nadzornih razmer, kot so posnetki že nameščenih avdio-vizualnih nadzornih sistemov, posnetki klicev ljudi v komunikacijski center policije ali reševalnih služb ipd. Te zbirke najbolj verno odražajo resnične varnostno-nadzorne razmere, vendar jih je zelo težko pridobiti zaradi pravnih in drugih ovir.

S to razmeroma zahtevno problematiko se ubadajo predvsem raziskovalci na področju čustvenega računalništva (angl. affective computing), ki v okviru različnih projektov in mrež odličnosti (denimo, http://emotion-research.net) pridobivajo tovrstne zbirke avdio-vizualnih posnetkov (denimo, korpus SAFE – Clavel, 2006).

3.2. Razpoznavanje govora

Obstoječo razmeroma razvito tehnologijo samodejnega razpoznavanja govora je mogoče neposredno uporabiti za različne varnostno-nadzorne scenarije (razpoznavanje izgovorjenih groženj, klicev na pomoč in varnostno sumljivih izjav ter prikrito govorno proženje alarma ipd). Ta tehnologija danes v glavnem temelji na računalniškem izvajanju postopka dekodiranja govornega signala z uporabo govornega modela, ki je predstavljen kot hierarhična struktura verjetnostnih končnih avtomatov. Najvišji nivo te hierarhične strukture modelira dani govorjeni jezik, vmesni nivo slovar izgovarjav danih besed in najnižji nivo akustične uresničitve posameznih glasov danega govorjenega jezika.

Celotno strukturo govornega modela se lahko obravnava kot strukturirani končni pretvornik (angl. Finite State Transducer), ki v najnižjem nivoju vsebuje stanja prikritih Markovovih modelov (Jelinek, 1998; Mohri et al.,2008). Takšen govorni model se je izkazal kot zelo prilagodljiv različnim področjem uporabe. Za sisteme avdio-nadzora prostora je potrebnega še nekaj razvojnega in raziskovalnega dela za zagotavljanje večje robustnosti delovanja v zahtevnih akustičnih okoljih (ulični hrup ipd), ter za boljšo izrabo metod časovne in prostorske lokalizacije govornih zvočnih virov v prostoru.

3.3. Razpoznavanje govorcev

Postopke razpoznavanja govorcev bi lahko v grobem razdelili v dve skupini. V prvo uvrščamo postopke, ki se uporabljajo za besedilno-odvisno razpoznavanje, v drugo pa postopke, ki se uporabljajo za besedilno neodvisno razpoznavanje. Za avdio-nadzorne sisteme (razpoznavanje poljubnega govora znanih kriminalcev in osumljencev) pridejo v poštev predvsem besedilno-neodvisni sistemi.

Tehnologija besedilno-neodvisnega razpoznavanja govorcev je v zadnjem poldrugem desetletju doživela precejšen napredek. V veliki meri gre zasluge za to pripisati ameriški organizaciji NIST, ki je z rednim prirejanjem dogodkov, na katerih se med seboj »pomerijo« najboljši raziskovalci s tega področja, uspela privabiti ugledne raziskovalne ustanove s celega sveta.

Eden izmed ključnih prebojev na tem področju je bil dosežen z uvedbo t.i. splošnega modela govorcev (UBM), ki iz več sto ur posnetkov govora velikega števila različnih govorcev strne večino pomembnih akustičnih lastnosti govorcev v relativno majhno množico parametrov statističnega modela mešanice Gaussovih porazdelitev (GMM) (Reynolds et.al., 2000). Uvedba modela UBM pa je poleg številnih drugih prednosti preko postopka največje posteriorne verjetnosti (MAP) uspela zagotoviti visoko zanesljivost razpoznavanja.

Pristop z uporabo modela UBM je zelo učinkovit v primeru, ko so akustične razmere (šum, lastnosti mikrofona in prenosnih poti itd.) v govornih posnetkih enake, a se uspešnost razpoznavanja precej poslabša, kadar temu ni tako. Raziskovalci so predlagali številne rešitve, s katerimi so poskušali izničiti ali vsaj zmanjšati vpliv sejne spremenljivosti. Med vsemi predlaganimi rešitvami je bila največje pozornosti deležna analiza vezanih faktorjev (JFA) (Kenny et al., 2007; Dehak et al. 2011), s katero se govorne posnetke danega govorca da pretvoriti v nizko-razsežen vektor značilk (poimenovan i-vektor), ki ohrani večino diskriminatorne informacije, ki jo potrebujemo za ločevanje med različnimi govorci.

3.4. Razpoznavanje psihofizičnega stanja govorcev

Tehnologija razpoznavanja psihofizičnega stanja govorcev je v osnovi precej podobna tehnologiji razpoznavanja govorcev (uporaba UBM-MAP modela itd), pri čemer se govorne posnetke razvršča namesto v razrede govorcev v nekaj razredov obravnavanih psihofizičnih stanj (Gajšek et al., 2012). Za razločljiva psihofizična stanja govorcev, ki bi jih naj bilo mogoče razpoznati na podlagi akustične analize govornega signala, se navadno obravnava nekaj izbranih emocionalnih stanj (strah, jeza, presenečenje, ipd) ter psihofizična stanja, ki so posledica alkoholiziranosti ali vpliva mamil.

Precej spodbude razvoju na tem področju so dala tekmovanja, ki so organizirana v okviru serije največjih mednarodnih konferenc na področju tehnologij govorjenega jezika Interspeech (Schuller et al., 2009). Tovrstne sisteme bi se dalo neposredno uporabiti za razpoznavanje agresivnega obnašanja ali prestrašenosti posameznikov, ki so prisotni v varovanem prostoru.

4. Pravna in etična vprašanja

Predstavljene tehnologije in njihova uporaba v različnih varnostno-nadzornih scenarijih odpirajo precej pravnih in etičnih vprašanj, še posebej v Evropi, kjer se daje precej poudarka človekovim pravicam do varovanja osebnih podatkov in pravici do zasebnosti. Veljavna zakonodaja v Evropi je precej nedorečena glede vprašanja varovanja teh pravic pri samodejni obdelavi in izmenjavi varnostno-nadzornih podatkov (Cannataci, 2010). Veljavna zakonodaja tako praktično onemogoča uporabo pametnih avdio-nadzornih sistemov za varnostni nadzor javnih prostorov in to kljub temu, da bi ti sistemi lahko v precej primerih reševali življenja. Obstaja namreč upravičen strah, da bi tovrstni sistemi omogočili varnostno-obveščevalnim službam neupravičeno prisluškovanje pogovorom ljudi v nadziranih prostorih. To skrb je potrebno upoštevati, zato bi se moralo v obravnavane tehnologije vgraditi sistemske varovalke, ki bi onemogočile zlorabo v druge namene, kot je bilo prvotno zamišljeno (denimo, onemogočanje nepotrebnega shranjevanja zajetih govornih posnetkov ipd).

Sodelavci našega laboratorija sodelujemo pri evropskih projektih, ki naslavljajo ta vprašanja in katerih cilj je podpora modernizaciji in izboljšanju učinkovitosti sredstev in delovanja organov kazenskega pregona ter izmenjavi informacij na tem področju za izmenjavo dobrih praks ter pripravo smernic in modelnih zakonov, ki bi vsebovali primerne zaščitne ukrepe za državljane pri implementaciji pametnih nadzornih tehnologij. Pridobljeno znanje v okviru teh projektov v laboratoriju upoštevamo tudi pri svojem razvojno-raziskovalnem delu, ki ga izvajamo na tem področju.

5. Sklep

V članku je obravnavana problematika uporabe tehnologih govorjenega jezika v pametnih nadzornih sistemih. To področje ponuja precej priložnosti za bolj intenzivno razvojno in raziskovalno delo tudi pri nas, saj so te tehnologije v veliki meri odvisne od govorjenega jezika in ni pričakovati, da bodo tuji razvijalci v kratkem razvili tovrstne sisteme, ki bodo uspešno delovali tudi za slovensko govorno področje. V laboratoriju imamo dolgoletne izkušnje z razvojem vseh omenjenih tehnologij za slovenski govorjeni jezik, vključno z razvojem emocionalnega sintetizatorja govora. V prihodnosti se imamo zato namen bolj posvetiti razvoju teh tehnologij tudi za uporabo v izbranih primernih varnostno-nadzornih scenarijev.

6. Zahvala

Delo, predstavljeno v tem prispevku, je bilo deloma podprto s financiranjem iz Sedmega okvirnega programa Evropske unije (FP7-SEC-2010-1) na podlagi sporazuma o financiranju številka 261727. Raziskovalno delo drugega avtorja je delno financirala Evropska unija iz Evropskega sklada za regionalni razvoj v okviru Operativnega programa krepitve regionalnih razvojnih potencialov za obdobje 2007-2013 po pogodbi št. 3211-10-000468 KC OpComm.

7. Literatura

Clavel, C., Vasilescu, I., Devillers, L., Ehrette, T. in Richard, G., 2006, The SAFE Corpus: fear-type emotions detection for surveillance applications. In Proc. LREC'06, Genoa, Italy, pp. 1099–1104.

Cannataci, J. A., 2010, Squaring the circle of smart surveillance and privacy, Fourth Inter. Conf. on Digital Society, DOI 10.1109/ICDS.2010.55, pp. 323-328.

Dehak, N., Kenny, P., Dehak, R. in Dumouchel, P., 2011, »Front-End Factor Analysis for Speaker Verification«. IEEE Trans. Audio, Speech, Lang. Process., zv. 19, št. 4, pp. 788-798.

Gajšek, R., Mihelič, F., Dobrišek, S., 2012, "Speaker state recognition using an HMM-based feature extraction method", Computer Speech & Language, DOI 10.1016/j.csl.2012.01.007.

Jelinek, F., 1998. Statistical Methods for Speech Recognition. MIT Press, Cambridge, MA, USA.

Kenny, P., Boulianne, G., Ouellet, P. in Dumouchel, P., 2007, »Speaker and Session Variability in GMM-Based Speaker Verification«. IEEE Trans. on Audio, Speech, and Language Processing, 15 (4), pp. 1448–1460.

Keyrouz, F., Diepold, K., Keyrouz, S., 2007, »High performance 3D sound localization for surveillance applications«, In Proc. IEEE AVSS '07, London, UK, pp. 563-566.

Mohri, M., Pereira, F. C. N. in Riley, M., 2008. Speech recognition with weighted finite-state transducers, pogl. Part E: Speech recognition. Springer-Verlag, Germany.

Onut, I.V., Aldridge, D., Mondel, M. in Perelgut, S., 2011. 2nd Workshop on Smart Surveillance System Applications. In Proc. CASCON '11, IBM Corp., Riverton, NJ, USA, pp. 382-384.

Piciarelli, C. in Foresti, G. L., 2011, "Surveillance-oriented event detection in video streams", IEEE Intelligent Systems, vol. 26, issue 3, pp. 32–41.

Reynolds, D. A., Quatieri, T. F. in Dunn, R. B., 2000, »Speaker Verification using Adapted Gaussian Mixture Models«. Digital Signal Processing, 10, pp. 19–41.

Schuller, B., Steidl, S., Batliner, A., 2009. The Interspeech 2009 emotion challenge. In: Proc. Interspeech 2009, ISCA, Brighton, UK, pp. 627: 312–315

Skladenjski razčlenjevalnik za slovenščino

Kaja Dobrovoljc,[1] Simon Krek,[2] Jan Rupnik[3]

[1]Trojina, zavod za uporabno slovenistiko
Partizanska cesta 5, 4220 Škofja Loka
kaja.dobrovoljc@trojina.si
[2] Laboratorij za umetno inteligenco, Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
simon.krek@ijs.si
[3] Odsek za tehnologije znanja, Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
jan.rupnik@ijs.si

Povzetek

V prispevku opišemo skladenjski razčlenjevalnik za slovenščino, ki je osnovan na razčlenjevalniku MSTParser (Minimum-Spanning Tree Parser) in je bil izdelan v okviru projekta Sporazumevanje v slovenskem jeziku. Naučen je na učnem korpusu ssj500k, ki vsebuje 11.411 ročno preverjenih povedi, razčlenjenih po sistemu odvisnostne drevesnice JOS. Pri sistemu JOS je natančnost razčlenjevalnika 90,43 % za napovedane povezave in 87,52 % za napovedane in označene povezave. Prispevek predstavi zasnovo razčlenjevalnika, učni korpus ssj500k ter sistem odvisnostne drevesnice JOS. Jedro prispevka je podrobna analiza natančnosti razčlenjevanja po skladenjskih oznakah, na koncu prispevka pa predstavimo možnosti za izboljšanje razčlenjevalnika na podlagi analize. Razčlenjevalnik je prosto dostopen pod licenco Apache License V2.0.

Dependency Parser for Slovene

This paper introduces the dependency parser for Slovene based on the MSTParser (Minimum-Spanning Tree Parser), which was developed as part of the Communication in Slovene project. It was trained on the ssj500k training corpus, containing 11.411 manually annotated sentences, parsed in accordance with the JOS Dependency Treebank system. For the JOS system, the parser's accuracy measures 90.43% for unlabelled dependencies and 87.52% for labelled dependencies. The paper presents the design of the parser, the ssj500k training corpus and the JOS Dependency Treebank system. The core of this paper consists of a detailed analysis of the parser's accuracy in relation to dependency labels, which also serves as the basis for the final part of the paper, in which we present possibilities for further improvement. The parser is freely available under the Apache License v2.0.

1. Uvod

Skladenjsko razčlenjevanje predstavlja enega od temeljnih jezikovnotehnoloških postopkov obdelave besedil, ki omogoča in podpira kompleksnejše jezikovne tehnologije, kot so strojno prevajanje, luščenje informacij, govorno komuniciranje, avtomatsko povzemanje, odgovarjanje na vprašanja itd.

Pri obravnavi skladenjskega razčlenjevanja je treba ločiti med uporabljenim (teoretskim) jeziko(slo)vnim modelom in metodo, ki jo uporablja razčlenjevalnik. Jezikoslovni modeli, uporabljeni za različne jezike, so zelo raznoliki, načeloma jih lahko razdelimo v dve večji skupini: sistem odvisnostnih drevesnic ter frazna gramatika, pri čemer velja, da je prva uporabnejša za jezike, ki imajo prost besedni red (npr. slovanski jeziki), druga pa za jezike s staln(ejš)im besednim redom (npr. angleščina). Obstajajo tudi hibridni modeli, ki beležijo informacije tako o odvisnostnih razmerjih kot o notranji sestavi zvez.

Po drugi strani so strojne metode, uporabljene pri razčlenjevanju, tipično treh vrst: razčlenjevanje na podlagi vnaprej pripravljenih pravil[1], statistično razčlenjevanje ter razčlenjevanje s hibridnimi postopki, pri katerih je uporabljena kombinacija obeh metod. Razčlenjevalniki, ki delujejo na podlagi pravil, za svoje delovanje potrebujejo množico ročno napisanih pravil, s katerimi interpretirajo slovničnost ali neslovničnost analiziranih struktur. Na nasprotni strani so statistični razčlenjevalniki, ki ne potrebujejo vnaprej pripravljenih informacij o jeziku, potrebujejo pa učno množico oz. učni korpus, iz katerega izdelajo statistični model, s katerim interpretirajo nova, neznana besedila. Obe metodi imata svoje prednosti in slabosti. Kot prednost statistične metode je mogoče omeniti predvsem manjši časovni in finančni vložek, ki je potreben za hitro doseganje razmeroma dobrih rezultatov, kar je spodbudilo nastanek množice statističnih razčlenjevalnikov v 90-ih letih prejšnjega stoletja.

V prispevku opisujemo statistični razčlenjevalnik, ki je nastal na podlagi razčlenjevalnika MSTParser, ki kot statistično metodo uporablja iskanje minimalnega vpetega drevesa v usmerjenih grafih in ga podrobneje opišemo v nadaljevanju.

2. Odvisnostni model

Pri obravnavanem razčlenjevalniku je bil uporabljen odvisnostni model, razvit v okviru projekta Jezikoslovno označevanje slovenščine (JOS) (Ledinek in Erjavec, 2009; Erjavec et al., 2010). Sistem obsega 10 oznak, ki jih glede na strukturnoskladenjsko raven delimo na tri skupine.

Povezave prvega nivoja (oznake *del, dol, vez, skup, prir*) označujejo razmerja znotraj besednih zvez – med jedrnim in nejedrnim delom povedka, predlogom in jedrom besedne zveze, (stavčnimi oz. nestavčnimi) prilastki in odnosnico, modalnim glagolom in dopolnilom ter med povedkom in povedkovim določilom.

[1] Tak razčlenjevalnik za slovenščino je denimo integriran v slovnični pregledovalnik BesAna podjetja Amebis: http://besana.amebis.si/preverjanje/.

Povezave drugega nivoja (oznake *ena, dve, tri, štiri*) sovpadajo s tradicionalnim pojmovanjem osebka, predmeta in prislovnih določil ter se uporabljajo tako za označevanje stavčnočlenskih vlog znotraj stavka kot za označevanje odvisnikov.

Povezava tretjega nivoja je pravzaprav ena sama (*modra*), uporablja pa se za povezovanje hierarhično najvišjih pojavnic (najpogosteje jedrni del povedka glavnega stavka in priredij) ter skladenjsko manj predvidljivih ali oddaljenih struktur, ki bi sicer ostale nepovezane, pa tudi za vsa ločila, ki niso povezana s katero izmed ostalih povezav.

Posamezne oznake in njihova razmerja do skladenjskih kategorij, kakršne poznamo iz tradicionalnih opisov slovenskega jezikoslovja in zaradi robustne narave avtomatskega označevanja niso neposredno prenosljive v predstavljeni model, so podrobneje opisani v specifikacijah za označevalce učnega korpusa[2].

3. Uporabljeni viri in orodja

3.1. MSTParser

3.1.1. Razčlenjevalnik

V pričujočem razdelku bomo povzeli delovanje skladenjskega razčlenjevalnika Minimum-Spanning-Tree Parser (MSTP) (McDonald, Lerman in Pereira, 2006). Najprej uvedimo nekaj osnovnih pojmov. Naj bo besedilo $B = (x_1, \dots, x_m)$ zaporedje povedi, kjer vsaka poved $x_i = (t_{i,1}, \dots, t_{i,n_i})$ predstavlja zaporedje pojavnic $t_{i,j}$, kjer n_i označuje število pojavnic v i-ti povedi. Odvisnostno drevo y za dano poved $x = (t_1, \dots, t_k)$ je definirano kot usmerjen povezan acikličen označen graf $y = (V_x, A_y, L_y)$, kjer $V_x = (v_1, \dots, v_k)$ predstavlja množico označenih vozlišč (prirejenih pojavnicam), $A_x = ((izvor_1, cilj_1), \dots, (izvor_{|A_y|}, \dots, cilj_{|A_y|}))$ predstavlja vektor usmerjenih povezav med pari vozlišč, kjer sta $izvor$ in $cilj$ zaporedji indeksov. $L_y = (l_1, \dots, l_{|A_y|})$ predstavlja množico oznak, prirejenih usmerjenim povezavam. Vsak l_i je element množice možnih oznak povezav: {"dol", "del", "vez", "prir", "skup", "ena", "dve", "tri", "štiri", "modra"}. Zaradi pogojev acikličnosti in povezanosti sledi, da je y drevo. S povezavo $(izvor_i, cilj_i)$ predstavimo skladenjsko odvisnost pojavnice, prirejene vozlišču $cilj_i$, od pojavnice, prirejene vozlišču $izvor_i$. Zaradi tehničnih razlogov dodamo grafu še tehnično vozlišče v_0, ki v množici povezav lahko nastopa samo kot izvor (metaelement). Naloga odvisnostnega razčlenjevanja je za dano poved x poiskati najustreznejše odvisnostno drevo y. Najprej bomo povzeli, kako se v modelu MSTP izračuna ustreznost drevesa y za dano poved x. Predstavili bomo rešitev za iskanje neoznačenih odvisnostnih dreves in kasneje omenili razširitev z označenimi povezavami. S pomočjo funkcije ustreznosti lahko razčlenjevanje prevedemo na problem iskanja najustreznejšega drevesa med vsemi možnimi odvisnostnimi drevesi, ki jih lahko priredimo dani povedi. Sedaj poenostavimo oznake in povezavo $(izvor_i, cilj_i)$ označimo kot (i, j).

Model MSTP je predstavljen kot N-dimenzionalni vektor uteži $w = (w_1, \dots, w_N)$, skupaj z N funkcijami lastnosti $(f_1(\cdot), \dots, f_N(\cdot))$. Vsaka funkcija lastnosti preslika par povedi in usmerjene povezave, $(x, (v_i, v_j))$, v množico $\{0,1\}$. Naloga funkcij lastnosti je zaznava različnih značilnosti dane povezave med parom pojavnic. Naj bo $tag(x) = (pos_1, \dots pos_k)$ zaporedje oblikoskladenjskih oznak, prirejeno povedi x. Navedimo nekaj primerov funkcij lastnosti:

$$f_1(x = (t_1, \dots, t_k), (i,j)) = \begin{cases} 1; \text{če: } t_i = "strgan", t_j = "čevelj" \\ 0; sicer \end{cases},$$

$$f_2(x = (t_1, \dots, t_k), (i,j)) = \begin{cases} 1; \text{če: } t_i = "je",\ pos_i = "Gp-ste-n" \\ 0; sicer \end{cases},$$

$$f_3(x = (t_1, \dots, t_k), (i,j)) = \begin{cases} 1; \text{če: } pos_i = "Rsr", pos_{i-1} = "Kav", pos_j = "Dr", pos_{j+1} = "Somer" \\ 0; sicer \end{cases}.$$

Kvaliteto odvisnostnega drevesa v MSTP izrazimo kot vsoto kvalitet povezav: $score(x, y) = \sum_{i=1}^{|y_A|} s(x, A_y(i))$, kjer kvaliteto i-te povezave izračunamo kot uteženo vsoto vektorja lastnosti: $s\left(x, A_y(i)\right) = \sum_{\ell=1}^{N} w_\ell \cdot f_\ell(x, A_y(i))$. Uteži w_ℓ so rezultat strojnega učenja in predstavljajo pomembnost funkcij lastnosti f_ℓ za merjenje kvalitete povezav. Tehnično poročilo (Mcdonald, Crammer in Pereira, 2005) vsebuje podroben opis vseh tipov funkcij lastnosti, ki so uporabljene v sistemu MSTP. Če na kratko povzamemo, avtorji uporabijo indikatorje različnih kombinacij: beseda izvora (f_1, f_2), beseda cilja (f_1), oblikoslovna oznaka izvora (f_2, f_3), cilja (f_3) in besed poleg izvora ali cilja (f_3).

3.1.2. Razčlenjevanje

Razčlenjevanje nove povedi poteka v dveh fazah, kjer privzamemo, da imamo na voljo naučen model MSTP, tj. vektor uteži $w = (w_1, \dots, w_N)$. Vhodni povedi $x = (t_1, \dots, t_k)$ priredimo poln usmerjen graf $G_x = (V = \{0,1, \dots, k\}, A)$, kjer V vsebuje $k+1$ vozlišč (vključno s tehničnim vozliščem) , in A vsebuje naslednje povezave: $A = \{(i,j) | i > 0, j > 0, i \neq j\} \cup \{(0,j) | j > 0\}$. Vsaki usmerjeni povezavi $(i,j) \in A$ priredimo vektor lastnosti: $F_{i,j} \rightarrow (f_1(x, (i,j)), \dots, f_N(x, (i,j)))$ in nato kvaliteto povezave: $S_{i,j} = \sum_{k=1}^{N} F_{i,j}(k) \cdot w_k$. Graf G_x skupaj s koeficienti $S = \{S_{i,j} | (i,j) \in A\}$ predstavlja utežen usmerjen graf. Zaradi linearnosti funkcije $score(x, y)$ je optimalno odvisnostno drevo natanko maksimalno vpeto drevo uteženega grafa (G_x, S). Za iskanje maksimalnega vpetega drevesa obstaja učinkovita rešitev s časovno zahtevnostjo $O(k^2)$ (Chu in Liu, 1965; Tarjan, 1977), kjer je k število pojavnic v povedi x.

3.1.3. Učenje

Naj bo $U = \{(x_1, y_1), \dots, (x_m, y_m)\}$ množica parov povedi in odvisnostnih dreves. Učenje odvisnostnega razčlenjevalnika predstavlja reševanje naslednjega optimizacijskega problema:

$$\min_{w \in R^N} \|w\|$$
$$p.p.$$
$$score(x_i, y_i) - score(x_i, y') \geq L(y_i, y'),\ \ \forall y' \in dt(x_i),$$
$$\forall i = 1{:}m,$$

kjer $dt(x)$ predstavlja množico vseh dopustnih neoznačenih odvisnostnih dreves in $L(y, y')$ prešteje število napak drevesa y' glede na dano pravilno drevo y. Minimizacija norme vektorja uteži sili učenje k preprostejšim modelom, s čimer preprečuje pretirano prileganje modela k učni množici. Optimizacijski pogoji omejujejo prostor iskanja modelov na modele, ki konsistentno ocenjujejo kvaliteto odvisnostnih dreves na učni množici. Avtorji (McDonald, Lerman in Pereira, 2006) za iskanje rešitve uporabijo iterativno metodo k-best MIRA (Margin Infused Relaxed Algorithm).

[2] http://www.slovenscina.eu/Media/Kazalniki/Kazalnik2/SSJ_Kazalnik_2_Specifikacije-ucni-korpus_v1.pdf

3.1.4. Označena odvisnostna drevesa

Iskanje označenih odvisnostnih dreves poteka v dveh fazah: v prvi fazi se povedi priredi neoznačeno odvisnostno drevo, za tem pa se za to drevo uporabi markovski model prvega reda za iskanje najverjetnejšega zaporedja oznak povezav. Razčlenjevanje celotnega učnega korpusa (11.411 oblikoslovno označenih stavkov) na procesorju Intel Core i7 3.07GHz CPU traja 31 minut.

3.2. Učni korpus ssj500k

Učni korpus ssj500k je bil izdelan v okviru projekta Sporazumevanje v slovenskem jeziku[3] (SSJ) in temelji na obeh učnih korpusih, izdelanih v okviru projekta JOS. Sestavljen je iz celotnega korpusa jos100k ter dodatnih 400.000 besed iz enomilijonskega korpusa jos1M. Vsi jezikoslovni metapodatki (oznake, leme, tokenizacija) so bili v korpusu ssj500k še enkrat ročno pregledani, povečana je bila množica skladenjsko označenih in ročno pregledanih povedi. V delu, ki ga zajema korpus jos100k, so bile dodane informacije o lastnih imenih za potrebe strojnih prepoznavalnikov imenskih entitet. Za razliko od korpusov jos100k in jos1M je bila v korpusu ssj500k v celoti ročno pregledana in popravljena tudi stavčna segmentacija in tokenizacija, kar omogoča tudi preverjanje uspešnosti označevalnikov in razčlenjevalnikov pri teh dveh postopkih. Številčni podatki o elementih v korpusu ssj500k so v Tabeli 1.

oznaka	opis	ssj500k
	besedilo	1.677
<p>	odstavek	8.137
<s>	stavek oz. poved	27.829
<w>	beseda	500.295
<c>	ločilo/simbol	85.953
<w> + <c>	pojavnica	586.248
<links>	element s skladenjskimi povezavami	11.411
<link>	skladenjska povezava	235.865
<chunks>	element s povezavami na imenske entitete	2.178
<chunk>	imenska entiteta	4.398

Tabela 1: Število elementov v učnem korpusu ssj500k

Učni korpus ssj500k je prosto dostopen na spletnih straneh projekta SSJ[4] pod licenco Creative Commons Priznanje avtorstva-Nekomercialno 3.0.[5]

4. Skladenjski korpus in natančnost razčlenjevanja

Skladenjsko označeni del ssj500k je izdelan po sistemu odvisnostne drevesnice JOS (Ledinek, 2010; Ledinek in Erjavec, 2009; Erjavec et al., 2010).[6] Učna množica za skladenjski razčlenjevalnik obsega 11.411 ročno preverjenih povedi ali 200.320 besed, kar predstavlja približno dve petini učnega korpusa ssj500k.

Skladenjski del korpusa ssj500k je nastajal v treh fazah. V prvi je bil v projektu JOS vzporedno s pripravo specifikacij za skladenjsko označevanje oblikovan testni nabor 500 povedi. Ta je služil kot učna množica za učenje prvega statističnega modela, s katerim je bil označen korpus jos100k. Avtomatsko pripisane skladenjske povezave so nato v programu Označevalnik stavkov[7] ročno pregledali in popravili označevalci, po dva za vsako poved oziroma trije, kadar je med prvima dvema označevalcema prihajalo do razlik. V tretji fazi je bila v okviru projekta Sporazumevanje v slovenskem jeziku učna množica povečana in celoten krog polavtomatskega razčlenjevanja ponovljen za novih 100.000 besed, vključno z analizo in popravki celotnega skladenjskega dela korpusa ssj500k.

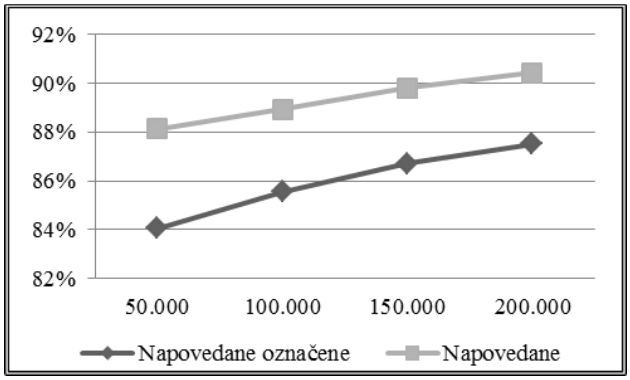
Tabela 1 prikazuje pogostost posameznih tipov skladenjskih razmerij in njihovo povprečno dolžino, tj. absolutno razdaljo med položajem izvorne in ciljne pojavnice. Navidezno je najdaljša korenska (*modra*) povezava, saj korenski element zavzema mesto ničte pojavnice.

št.	Oznaka	Frekvenca	povprečna dolžina
1	modra	66.155	16,2
2	del	16.134	2,4
3	dol	79.627	1,8
4	ena	11.690	3,5
5	dve	15.639	2,9
6	tri	5.779	2,7
7	štiri	14.246	4,0
8	prir	6.482	5,0
9	vez	19.288	2,6
10	skup	825	1,2

Tabela 2: Tipi skladenjskih povezav glede na pogostost v učnem korpusu ssj500k in njihova povprečna dolžina

Z večanjem učne množice je natančnost označevanja postopoma naraščala (slika 1), pri čemer najvišja natančnost skladenjskega razčlenjevanja z učenjem na učnem korpusu ssj500k (~200.000 besed), merjena z 10-kratnim prečnim preverjanjem, meri 90,43 % za natančnost napovedanih povezav (tj. pravilno določeno mesto povezave) oziroma 87,52 % za natančnost napovedanih označenih povezav (tj. pravilno določena mesto in tip povezave).

[3] http://www.slovenscina.eu/tehnologije/ucni-korpus
[4] http://www.slovenscina.eu/tehnologije/ucni-korpus, http://razclenjevalnik.slovenscina.eu/
[5] http://creativecommons.org/licenses/by-nc/3.0/deed.sl
[6] Sistem uporablja deset oznak, za razliko od sistemov SDT (Erjavec in Ledinek, 2006) in PDT (http://ufal.mff.cuni.cz/pdt2.0/), ki uporabljata 28 oznak v precej kompleksnejšem sistemu.
[7] Program je dostopen na strani projekta SSJ: http://www.slovenscina.eu/Vsebine/Sl/Kazalniki/K10.aspx



Slika 1: Natančnost glede na velikost učne množice

V okviru analize je bila izvedena tudi primerjava glede na stopnjo uporabe oblikoskladenjskih informacij pri učenju razčlenjevalnika. Natančnost razčlenjevanja z upoštevanjem celotnih oblikoskladenjskih oznak, upoštevanjem zgolj vrhnjih kategorij (podatka o besedni vrsti) oz. neupoštevanjem oblikoskladenjskih oznak je predstavljena v Tabeli 3.

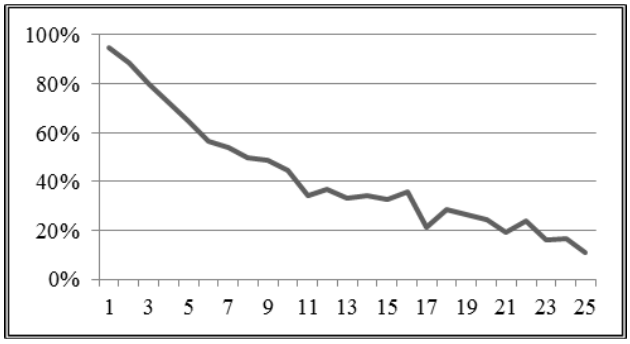
	Celotne oznake	Besedne vrste	Brez oznak
Napovedane povezave	90,43 %	88,86%	73,71 %
Napovedane označene povezave	87,52 %	84,41%	65,34 %

Tabela 3: Natančnost skladenjskega razčlenjevalnika z upoštevanjem oblikoskladenjskih oznak.

Kot izhodišče za nadaljnjo podrobnejšo analizo rezultatov skladenjskega razčlenjevanja je bila upoštevana prva množica, torej rezultat označevanja korpusa z ročno pregledanimi celotnimi oblikoskladenjskimi oznakami, v njej pa smo se osredotočili na opazovanje natančnosti razčlenjevanja glede na zgradbo povedi in posebnosti posamezne skupine skladenjskih razmerij.

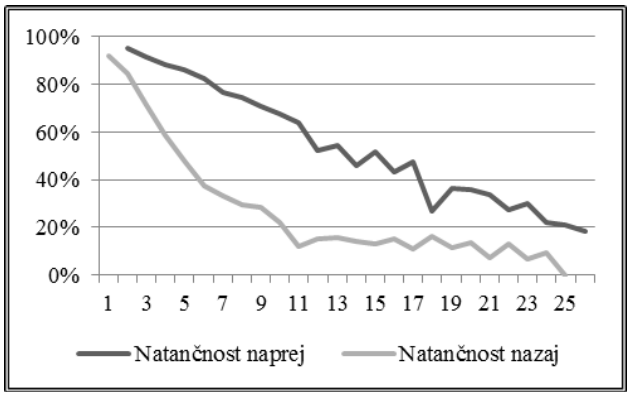
Natančnost označevanja z dolžino povedi pričakovano upada, od povprečno 95,7-odstotne natančnosti za povedi z 1–5 pojavnicami do 84,4-odstotne natančnosti za povedi z več kot 50 pojavnicami. Pri podrobnejšem pregledu povedi z manj kot 30 pojavnicami vidimo, da natančnost strmo pada do dolžine 10, nato je krivulja položnejša. Povprečna poved skladenjskega korpusa vsebuje 18 besed oz. 21 pojavnic (vključno z ločili).

V teoriji bi moral biti model, ki temelji na iskanju globalno optimalnih rešitev na grafu, kakršen je model MSTP, enako uspešen ne glede na dolžino povezave med izvorno in ciljno pojavnico, toda realno natančnost tudi z dolžino povezav upada (slika 2), saj so daljše povezave običajno tudi bolj dvoumne (McDonald in Nivre, 2007), denimo pri nizih prislovnih določil, ki so glede na pomen lahko povezani v eno besedno zvezo ali pa cilj več različnih (razmeroma dolgih) povezav.



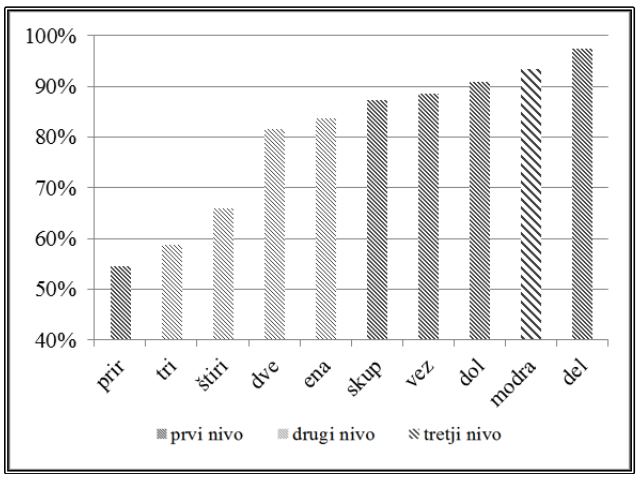
Slika 2: Natančnost glede na razdaljo med izvorno in ciljno pojavnico

Precejšna razlika se pojavlja v natančnosti povezovanja glede na smer povezave, torej glede na to, ali ciljna pojavnica stoji za izvorno ali pred njo (slika 3), natančnost označevanja pri povezavah naprej je namreč v povprečju kar za 16,2 odstotne točke višja kot pri povezavah nazaj.



Slika 3: Natančnost glede na smer in dolžino povezave med izvorno in ciljno pojavnico

5. Analiza po oznakah



Slika 4: Natančnost označevanja glede na tip povezave.

Na sliki 4 prikazujemo natančnost napovedanih označenih povezav glede na tip povezave. Natančnost je z izjemo povezave prir najvišja pri povezavah prvega nivoja

(*del*, *dol*, *vez*, *skup*) in pri povezavi tretjega nivoja (*modra*), nižja pa je pri povezavah drugega nivoja (*ena*, *dve*, *tri*, *štiri*) in povezavi *prir*.

5.1. Povezave prvega nivoja

Kot je razvidno iz slike 4, je za to skupino (besednozveznih) povezav natančnost razčlenjevanja najvišja, še posebej pri povezovanju delov zloženega povedka (povezava *del*, 97,5 %). Nekoliko nižja je natančnost pri povezovanju jedra in določila drugih besednih zvez (povezava *dol*; 90,8 %), ki pa se zelo razlikuje glede na posamezen tip besedne zveze: razčlenjevalnik je izredno natančen pri povezovanju predložnih zvez (98,3 %; X-*dol*-D[8], tj. povezava *dol* s katerekoli pojavnice na predlog) ter pri povezovanju (pridevniškega, zaimkovnega ali številskega) prilastka s samostalniškim jedrom (97,1 %; S-*dol*-[PZK]), nekoliko manj pa pri povezovanju drugih tipov nestavčnih modifikacij. Povprečne rezultate dosega pri označevanju samostalniških zvez s samostalniškim prilastkom (88,5 %; S-*dol*-S), pridevniških povedkovih določil (89,7 %; G-*dol*-P), prislovnih zvez s prislovnim prilastkom (87,9 %; R-*dol*-R) ter pri povezovanju modalnih glagolov in nedoločnika oz. namenilnika (89,2 %; G-*dol*-G), medtem ko je najmanj natančen pri povezovanju prilastkovih odvisnikov (55,7 %; [^G]-*dol*-G). Podobno nizko stopnjo natančnosti dosega pri povezovanju nepridevniških povedkovih določil (52,4 %; G-*dol*-[^GP]), toda za razliko od vseh ostalih povezav prvega nivoja je pri slednjih vendarle razmeroma uspešen z vidika pravilno napovedanih (a ne tudi označenih) povezav (natančnost naraste na 92,0 %; v 94 % napak razčlenjevalnik namreč namesto povedkovega določila oz. povezave *dol* pripiše osebek oz. povezavo *ena*). Zanimivo je, da so povezave prvega nivoja tudi tiste, kjer so rezultati označevanja besedila brez oblikoskladenjskih oznak najboljši, denimo 81,7 % pri povezavi *del*, 75,5 % pri povezavi *skup*, 68,9 % pri povezavi *dol* in 65,4 % pri povezavi *vez* oziroma skupaj v povprečju 21 odstotnih točk slabši od razčlenjevanja oblikoskladenjsko označenega besedila.

Ne glede na stopnjo označenosti korpusa je povezava *prir* tista, ki med vsemi tipi povezav dosega daleč najslabše rezultate (54,6% natančnost). Razčlenjevalnik uspešno prepozna priredni veznik ter jedro drugega dela priredja (81,0 % oz. 89,4 % za vezniške besede brez upoštevanja ločil; [^G]-*vez*-G), ne prepozna pa povezave med jedrom prvega in drugega dela priredja. Označevanje vezniških besed je še bolj natančno na stavčni ravni (93,6 %; G-*vez*-X), še posebej pri povezavah na veznike (95,4 %; G-*vez*-V).

5.2. Povezave drugega nivoja

Povezave drugega nivoja skupno dosegajo razmeroma nizko, 74,6-odstotno, natančnost označenih napovedanih povezav, toda v slabi polovici primerov so napake posledica napačno pripisane oznake in ne napačno napovedane povezave. Natančnost napovedanih povezav tako meri 85,6 %. Ker povezave drugega nivoja vsebujejo podatke o glagolski vezljivosti, omenjena podatka pri analizi med spadata med pomembnejše.

Med povezavami drugega nivoja razčlenjevalnik najbolj natančno označuje povezavo *ena* (83,7 %), ki z vidika tradicionalne slovnice ustreza povezavi med povedkom in osebkom. Razčlenjevalnik je najbolj uspešen pri določanju samostalniškega (87,5 %; G-*ena*-S) oziroma zaimkovnega osebka (82,1 %; G-*ena*-Z), manj pa pri prepoznavanju osebkov drugih besednih vrst in pa pri povezovanju osebkovnega odvisnika (60,0 %; G-*ena*-G). V povprečju se v približno dveh tretjinah napak moti v samem mestu povezave, v preostali tretjini pa sicer pravilno povezanemu paru pojavnic pripiše napačen tip povezave (namesto povezave *ena* največkrat zmotno pripiše povezavo za povedkovo določilo [*dol*] ali predmet [*dve*]).

Razčlenjevalnik je podobno uspešen tudi pri označevanju povezave *dve* med povedkom in predmetom stavka (81,5 %), bolj pri povezavah na samostalnik (84,1 %; G-*dve*-S) in zaimek (90,8 %; G-*dve*-Z), manj pa pri povezovanju predmetov drugih besednih vrst in predmetnega odvisnika (65,4 %; G-*dve*-G) ter pri predmetnih povezavah s pridevniškega povedkovega določila (51,6 %; P-*dve*-X). Za razliko od povezave *ena* je pri povezavi *dve* razmerje med napakami izvora in napakami tipa precej enakovredno, pri čemer razčlenjevalnik pri predmetnih povezavah z glagola na samostalnik, zaimek, številnik in prislov pri več kot polovici napak določi pravilno mesto povezave, a napačen tip. V dveh tretjinah takih primerov izbere oznako *štiri*, kar je glede na pomanjkanje leksikonskih informacij o vezljivosti glagolov pričakovano, zlasti pri predložnih besednih zvezah.

Poleg povezave *prir* sta med povezavami z najnižjo stopnjo natančnosti povezavi *tri* in *štiri* (58,6 % oz. 66,0 %), ki ustrezata povezavi s povedka na prislovno določilo. Označevanje prislovnih povezav je razmeroma uspešno le pri povezavah z glagola na prislov (81,8 % za povezavo *tri* in 79,5 % za *štiri*), pri vseh ostalih besednih vrstah in pri prislovnih povezavah s povedkovega določila pa je natančnost nižja. Daleč najnižja je natančnost pri prislovnih odvisnikih (0,6 % oz. 16 %; G-*tri*/*štiri*-G), kar je bistveno slabši rezultat kot pri drugih vrstah odvisnikov.

Ker imata povezavi *tri* in *štiri* razmeroma visok delež napak tipa in je ločevanje med njima z vidika avtomatskega razdvoumljanja precej zahtevno, smo ju v eni od analiz združili v en sam tip povezave. Število napak zaradi napačno pripisane oznake pravilno povezanima pojavnicama se je sicer zmanjšalo, natančnost napovedanih označenih povezav pa je ostala nespremenjena. Za 6,3 odstotnih točk se je povečala zgolj natančnost označenih povezav *tri* in *štiri* (s 63,8 % na 70,2 %), označevanje ostalih dveh povezav drugega nivoja (*ena* in *dve*) pa se je celo poslabšalo (za 0,9 oz. 0,5 odstotne točke). Tako se z združitvijo obeh prislovnih povezav natančnost napovedanih povezav na celotni množici ni bistveno izboljšala (s 87,52 % na 87,98 %), natančnost napovedanih označenih povezav pa se je zaradi drugih sprememb celo rahlo poslabšala (z 90,43 % na 90,41 %),

5.3. Povezave tretjega nivoja

Skupna natančnost označevanja povezave *modra* je razmeroma visoka (93,4 %), pri čemer je treba upoštevati, da ta vrednost združuje tako natančnost korenskih povezav za besedne pojavnice (87,8 %) kot povezav na

[8] Pri poimenovanju besednih vrst uporabljano vrhnje kategorije po sistemu JOS (http://nl.ijs.si/jos/msd/html-sl/index.html).

ločila (99,0 %). Pri povezavah na besede je natančnost določanja pravilnega mesta povezave nadpovprečno visoka pri členku, okrajšavi, glagolu, vezniku in številniku, precej pod povprečjem pa pri prislovu, samostalniku, predlogu, pridevniku in zaimku. Razumljivo se pri tej povezavi pojavljajo samo napake izvora, ne pa tudi tipa (vedno *modra*). Spodbudna je natančnost pri prepoznavanju hierarhično najvišje vloge povedka v glavnih stavkih in priredjih (93,4 %; metaelement-*modra*-G).

6. Zaključek

Kot vsi statistični modeli tudi statistični razčlenjevalnik daje dobre rezultate pri označevanju tipičnega v jeziku, slabše pa pri označevanju manj pogostih struktur. Natančnost pada tudi z dolžino povedi in povezav. Podrobnejša analiza rezultatov za posamezne skupine povezav kaže, da je razčlenjevalnik nadpovprečno uspešen pri povezovanju besednih zvez (še posebej glagolskih, samostalniških in predložnih), pri določanju vezniških besed v stavkih in pri povezovanju pojavnic na korenski element. Povprečne rezultate glede na skupno natančnost dosega pri povezovanju povedkovih določil, modalnih glagolskih zvez ter pri prepoznavanju stavčnega osebka in predmeta, podpovprečen pa je pri povezovanju besednozveznih priredij in prislovnih določil. Podpovprečna je tudi natančnost označevanja prilastkovih, osebkovih in predmetnih odvisnikov, še posebej nizka pa je natančnost označevanja prislovnih odvisnikov.

Vsekakor obstaja še veliko možnosti za izboljšavo rezultatov. Na podlagi podatkov na sliki 1 nadaljnje širjenje učne množice v tej fazi razvoja razčlenjevalnika ni smotrno, saj natančnost narašča razmeroma počasi v primerjavi s potrebno količino vloženega ročnega dela. Pred vsebinskimi izboljšavami razčlenjevalnika je smiselno temeljito preveriti morebitne napake in neskladja pri ročnem označevanju, bodisi s še enim krogom usmerjenega ročnega pregledovanja bodisi s pomočjo specializiranih programov za avtomatsko odkrivanje napak pri odvisnostnem razčlenjevanju (Boyd, Dickinson in Meurers, 2008; Hall in Novák, 2011).

Z vidika izboljšav v delovanju razčlenjevalnika je perspektivno predvsem aktivno učenje (*active learning*) posameznih tipov povedi, besednih zvez ali povezav (Mirroshandel in Nasr, 2011), poleg tega je smiselno razmišljati v smeri hibridnega razčlenjevalnika, ki bi poleg statističnih modelov vseboval tudi ročno napisana skladenjska pravila oz. kombinacijo dveh samostojnih razčlenjevalnikov. Uspešnost takih hibridnih modelov se v tujih raziskavah že potrjuje (Foth in Menzel, 2006; Sennrich et al., 2009). V kontekst hevrističnega usmerjanja statističnih modelov se umeščajo tudi podatki o vezljivosti, denimo iz leksikalne baze za slovenščino (Gantar in Krek, 2011), s katerim bi lahko bistveno zmanjšali delež napak tipa predvsem pri povezavah drugega nivoja, kjer ima razčlenjevalnik težave pri razdvoumljanju predmetnih oz. prislovnodoločilnih povezav.

7. Literatura

Boyd, A., Dickinson, M., Meurers, D. (2008). On Detecting Errors in Dependency Treebanks. Research on Language and Computation, 6(2), 113-137.

Chu, Y., Liu, T. (1965). On the shortest arborescence of a directed graph. *Science Sinica*, 14, 1396-1400.

Erjavec, T., Ledinek, N. (2006). Slovenska odvisnostna drevesnica: prvi rezultati. V *Zbornik 5. slovenske in 1. mednarodne konference Jezikovne tehnologije IS-LTC* (str. 162-167). Ljubljana: Institut Jožef Stefan.

Erjavec, T., Fišer, D., Krek, S., Ledinek, N. (2010). The JOS Linguistically Tagged Corpus of Slovene. V *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)* (str. 1806-1809). Pariz: ELRA.

Foth, K.A., Menzel, W. (2006). Hybrid parsing: using probabilistic models as predictors for a symbolic parser. V *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL* (str. 321-328). Sydney: ACL.

Gantar, P., Krek, S. (2011) Slovene lexical database. V *Natural language processing, multilinguality: sixth international conference* (str. 72-80). Modra, Slovaška: Slovenská akadémia vied.

Hall, K.B., Novak, V. (2011). Corrective Dependency Parsing. *Text, Speech and Language Technology: Trends in Parsing Technologies*, 1(43), 151-167.

Ledinek, N. (2010). Slovenska skladnja v skladenjsko označenih korpusih slovenščine. Doktorska disertacija.

Ledinek, N., Erjavec, T. (2009). Odvisnostno površinskoskladenjsko označevanje slovenščine: specifikacije in označeni korpusi. V *Zbornik Simpozija Obdobja: Infrastruktura slovenščine in slovenistike* (str. 2019-224). Ljubljana: Znanstvena založba Filozofske fakultete.

McDonald, R., Crammer, K., Pereira, F. (2005). Spanning Tree Methods for Discriminative Training of Dependency Parsers, UPenn CIS Technical Report: MS-CIS-05-11.

McDonald, R., Nivre, J. (2007). Characterizing the Errors of Data-Driven Dependency Parsing Models. V *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (str. 168-170). Praga: ACL.

McDonald, R., Lerman, K., Pereira, F. (2006). Multilingual Dependency Parsing with a Two-Stage Discriminative Parser. V *Tenth Conference on Computational Natural Language Learning (CoNLL-X)*.

Mirroshandel, S. A., Nasr, A. (2011). Active Learning for Dependency Parsing Using Partially Annotated Sentences. V *Proceedings of International Conference on Parsing Technologies (IWPT)*.

Sennrich, R., Schneider, G., Volk, M., Warin, M. (2009). A New Hybrid Dependency Parser for German. V *Proceedings of the Biennial GSCL Conference*.

Tarjan, R.E. (1977) Finding Optimum Branchings. *Networks*, 7, 25-35.

Širjenje slovarja in dvoprehodni algoritem v razpoznavalniku tekočega govora UMB Broadcast News

Gregor Donaj, Zdravko Kačič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova ulica 17, 2000 Maribor
gregor.donaj@uni-mb.si, kacic@uni-mb.si

Povzetek

V članku bomo predstavili nekatere najnovejše poskuse na razpoznavalniku slovenskega govora z velikim slovarjem UMB Broadcast News. Različico sistema, ki je bila predstavljena leta 2010, smo nadgradili z jezikovnimi modeli z večjimi slovarji in dvoprehodnim algoritmom, ki uporablja različne jezikovne modele v prvem in drugem prehodu. V članku primerjamo delež napačno razpoznanih besed in faktor realnega časa razpoznavanja pri različnih velikostih slovarja in različnih redih jezikovnih modelov. S predstavljeno različico sistema smo z enoprehodnim algoritmom dosegli najmanjši delež napačno razpoznanih besed 25,65%. Pokazali smo tudi, da lahko z uporabo dvoprehodnega algoritma dosežemo primerljivo uspešnost razpoznavanja v bistveno krajšem času. Prav tako nam predstavljeni rezultati služijo tudi kot smernice za nadaljevanje dela na tem področju.

Vocabulary enlargement and a two-pass algorithm for the UMB Broadcast News continuous speech recognizer

In this paper we present some recent experiments on the UMB Broadcast News large vocabulary continuous speech recognition system. We took the 2010 version and added new language models with larger vocabularies and a two-pass recognition algorithm that uses different language model in its two passes. We compare word error rates and real time factors on different vocabulary sizes and model orders. We achieved a minimum word error rate of 25.65% on a one-pass algorithm. We also show that comparable results can be achieved with significantly less time effort using a two-pass algorithm. The presented results also serve as guidelines for further work in this area.

1. Uvod

Razpoznavanje tekočega govora z velikim slovarjem je kljub napredkom tehnologije še vedno ena najzahtevnejših nalog na področju procesiranja govora. To velja tako za uspešnost kot tudi za hitrost razpoznavanja. V tem članku[1] bomo predstavili nekatere najnovejše pristope za izboljšanje uspešnosti v sistemu za razpoznavanje tekočega govora UMB Broadcast News, ki je bil razvit na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru.

Prva različica sistema UMB BN je bila predstavljena leta 2006 (Žgank & Sepesy Maučec, 2006). Zadnja predstavljena različica je iz leta 2010 (Žgank & Sepesy Maučec, 2010), v kateri so bili izboljšani akustični in jezikovni modeli. Dosežena je bila pravilnost razpoznanih besed 71,3% oz. delež napačno razpoznanih besed (NRB) 28,7%. Iz te različice smo prevzeli akustične modele, izdelali pa smo nove jezikovne modele. Namen našega dela je bil izboljšati uspešnost razpoznavanja ob čim manjšem povečanju časovne zahtevnosti.

1.1. Faktor realnega časa

Časovno zahtevnost razpoznavanja merimo s faktorjem realnega časa (RTF). To je razmerje med procesorskim časom, ki ga iskalni algoritem potrebuje, da iz nekega zvočnega segmenta razpozna izgovorjene besede, in dolžino tega segmenta. RTF je odvisen tako od uporabljene strojne in programske opreme, kot tudi od uporabljenih modelov. Nas je zanimal vpliv uporabe različnih jezikovnih modelov na RTF. Da smo lahko primerjali rezultate za naše modele, smo vse poskuse izvajali v enakih pogojih na istem strežniku.

1.2. Besede OOV

Pri povečevanju slovarja smo se osredotočili na zmanjšanje napak, ki nastanejo zaradi besed izven slovarja (OOV). Pri uporabi dvoprehodnega algoritma pa na možnost uporabe jezikovnih modelov ločeno od akustičnega razpoznavanja, ki je časovno zelo zahtevno. Slovenščina je pregibni jezik, zaradi česar vsebuje veliko več besednih oblik, kot pa drugi jeziki, kot je na primer angleščina (Rotovnik et al., 2003). Posledično se ob enaki velikosti slovarja v slovenskem besedilu pojavi več besed OOV. Kadar se v besedilu pojavi beseda, ki se ne nahaja v slovarju razpoznavalnika, je ta tudi ne more pravilno prepoznati. Posledično se pojavi napaka. Zmanjšanje deleža OOV dosegamo z večanjem slovarja. Ocenjujemo, da pri slovenščini in drugih slovanskih jezikih, kot sta češki in ruski, potrebujemo približno sedem do desetkrat večje slovarje za enako pokritost korpusa (Rotovnik et al., 2003; Zablotskiy et al., 2010; Nouza et al., 2010).

1.3. Iskalni algoritem

Iskalni algoritem (Aubert, 2002) je ključnega pomena za uspešnost razpoznavalnika. Njegova naloga je s pomočjo slovarja in akustičnih ter jezikovnih modelov najti besedne hipoteze, ki najbolje ustrezajo nekemu zvočnemu posnetku. Poznamo različne vrste iskalnih algoritmov. V tem delu smo uporabljali dvoprehodni algoritem.

V prvem prehodu uporabljamo tako akustične kot tudi bigramske in trigramske jezikovne modele. V drugem pre-

[1]Delo je bilo delno sofinancirano s štipendijo ARRS mladega raziskovalca po pogodbi 1000-10-310131

hodu pa uporabljamo le štirigramske jezikovne modele. V drugem prehodu smo delali le s tekstovnimi datotekami. Ker je obdelovanje teksta časovno veliko manj zahtevno, kot pa razpoznavanje iz zvočnega posnetka, smo upali s tem pristopom v razmeroma kratkem času doseči izboljšanje rezultatov, ki jih dobimo po prvem prehodu.

1.4. Struktura članka

V nadaljevanju bomo najprej predstavili uporabljene govorne in tekstovne vire za izdelavo modelov. V tretjem poglavju članka bomo predstavili način gradnje slovarjev in jezikovnih modelov. V četrtem poglavju bo predstavljeno delovanje uporabljenega dvoprehodnega razpoznavalnega algoritma. Rezultati razpoznavanja bodo predstavljeni v petem poglavju. Najpomembnejši zaključki in nekatere smernice za možnost nadaljevanja dela bomo predstavili v šestem poglavju.

2. Uporabljeni viri

Za gradnjo modelov sta bila uporabljena slovenska govorna baza BNSI Broadcast News (Žgank et al., 2008/2) in slovenski referenčni jezikovni korpus FidaPLUS (Arhar & Gorjanc, 2007).

Baza BNSI je sestavljena iz 36 ur govornega materiala, ki je bil zbran iz različnih informativnih televizijskih oddaj RTV Slovenije. Največji del baze so učni podatki, na katerih so naučeni akustični modeli. Za učenje modelov smo uporabili orodje HTK (HTK, 2010). Razvojna in testna množica obsegata po slabe 3 ure materiala. Vse poskuse razpoznavanja smo izvajali na testni množici. Razvojna množica je namenjena optimiziranju parametrov modela, ki smo jih prevzeli iz prejšnje verzije. Baza vsebuje tudi tekstovni del. Ta obsega približno 11 milijonov besed, vendar pa ga pri izdelavi modelov nismo uporabljali.

Za gradnjo slovarjev in jezikovnih modelov smo uporabljali korpus FidaPLUS, ki vsebuje različna slovenska besedila s skupno 621 milijoni besed. Korpus je lematiziran in vsebuje morfosintaktične oznake, ki pa jih nismo uporabljali.

3. Širitev slovarja in gradnja jezikovnih modelov

Korpus FidaPLUS smo najprej obdelali tako, da smo iz njega izluščili le besede. Števila, okrajšave in druge besede, ki so vsebovale številke ali pa posebne znake, smo nadomestili s posebnimi oznakami. Odpravili smo vsa naglasna znamenja in podobne oznake ob črkah.

Slovarje smo zgradili tako, da smo v njih dodajali besede korpusa v vrstnem redu glede na njihovo pogostost. Kadar smo dosegli želeno velikost slovarja smo dodali še vse besede, ki se pojavijo enako pogosto, kot zadnja dodana beseda. Pri tem smo izbrali različne želene velikosti slovarjev od 60.000 besed do 300.000 besed. V slovarjih sta zraven besed le še oznaki za začetek in konec stavka. Vse ostale oznake smo izključili iz slovarja. Končni slovarji so zaradi načina njihove gradnje bili nekoliko večji od želenih velikosti.

Modele bomo kasneje vrednotili na testni množici BNSI. Za vsak slovar smo izračunali delež besed v testni množici, ki se ne pojavijo v slovarju. Tabela 1 prikazuje velikosti slovarjev in deleže OOV. Ti so primerljivi z rezultati na češkem jeziku (Nouza et al., 2010), kjer je bilo ugotovljeno tudi izboljšanje uspešnost razpoznavanja primerljivo z zmanjšanjem deleža besed OOV. Podobna izboljšanja pričakujemo tudi za slovenščino.

Tabela 1: Velikosti slovarjev in deleži besed OOV na testni množici.

Slovar	Velikost	OOV [%]
60k	60.022	6,94
100k	100.189	3,44
150k	150.285	2,24
200k	201.034	1,64
250k	251.352	1,29
300k	301.357	1,02

Za vse velikosti slovarjev smo zgradili bigramske (2g), trigramske (3g) in štirigramske (4g) modele. Uporabljali smo Good-Turingovo glajenje in sestopanje po Katz-u. Posebne oznake, ki smo jih uporabljali pri obdelavi korpusa nismo vključili v modele, saj se uporabljajo le pri pisanih besedilih in zato pri modeliranju govorjenega besedila niso uporabne. Med modeliranjem se obnašajo kot neznane besede.

4. Dvoprehodni algoritem

V izločanju značilk iz zvočnih posnetkov smo uporabljali mel kepstralne koeficiente (MFCC)(Biem et al., 2005) in energijo signala ter prve in druge odvode. Uporabljali smo 26 mel filtrov. Značilke smo izločali v oknih dolžine 32 ms in v presledkih 10 ms. Uporabljeni akustični modeli so medbesedni trigrafemski zvezni HMM s 16 Gaussovimi porazdelitvami in vezanimi stanji.

4.1. Prvi prehod

Prvič smo na razpoznavalniku UMB Broadcast News uporabili dvoprehodni algoritem. V prvem prehodu smo uporabljali orodje HDecode (HTK, 2010). Med razpoznavanjem se uporablja sinhroni iskalni algoritem z Viterbijevo aproksimacijo in snopovnim omejevanjem. Prvi prehod smo izvajali z vsemi predstavljenimi velikostmi slovarja ter bigramskimi in trigramskimi modeli.

4.2. N-najboljših seznami

V prvem prehodu algoritma smo dobili za vsak segment govora dva rezultata. Prvi je bil najboljša hipoteza, ki se lahko neposredno ovrednoti (izračunamo število napačno razpoznanih besed). Iz najboljših hipotez smo kasneje izračunali uspešnost razpoznavanja prvega prehoda.

Istočasno pa nam je iskalni algoritem vrnil tudi besedni graf (besedno mrežo), v katerem je predstavljen iskalnih prostor algoritma ob zaključku segmenta. Iz nje lahko razberemo več različnih besednih hipotez, ki jih je iskalni algoritem vrednotil med delovanjem. Za vsako besedo v grafu imamo podan čas začetka in konca ter verjetnosti akustičnega in jezikovnega modela. Iz besednih grafov lahko razberemo več različnih hipotez za celotne segmente.

Uporabili smo orodje s katerim smo iz vseh besednih mrež izpisali sezname 1000 najboljših hipotez. Te sezname smo kasneje uporabljali v drugem prehodu algoritma. V nekaterih primerih je bil iskalni prostor ob delovanju algoritma tako majhnem, da ni bilo možno tvoriti 1000 hipotez. V teh primerih smo dobili manjše sezname. Za vsako hipotezo imamo v seznamih podane vse podatke s katerimi se hipoteze ocenjujemo: število besed ter verjetnosti akustičnega in jezikovnega modela.

4.3. Drugi prehod

Osnovna ideja predstavljenega dvoprehodnega algoritma je ta, da s kompleksnejšimi modeli ponovno ovrednotimo vse hipoteze. Uporabljali smo orodje SRI Language Modeling Toolkit (Stolcke, 2002).

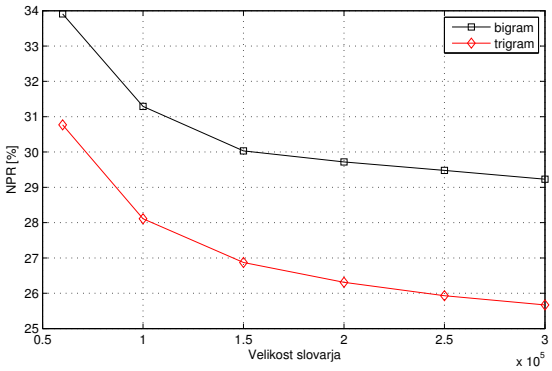
V drugem prehodu smo uporabljali le jezikovne modele. Verjetnost akustičnega modela smo prevzeli iz prvega prehoda. Nove verjetnosti jezikovnih modelov pa smo dobili z uporabo štirigramskih modelov. Ko smo ocenjevali hipoteze iz prvega prehoda smo vedno uporabljali model, ki je bil zgrajen na istem slovarju kot model iz prvega prehoda. Potem, ko se vse hipoteze ponovno ocenijo, se izbere nova najboljša hipoteza, ki se uporablja v vrednotenju uspešnosti razpoznavanja.

5. Rezultati eksperimentov

Uspešnost izdelanih modelov in algoritma smo vrednotili na ročno segmentirani testni množici baze BNSI. Vse rezultate podajamo v deležu napačno razpoznanih besed. To je razmerje med vsoto zamenjanih, vrinjenih in izbrisanih besed ter številom vseh besed v testni množici.

Najprej smo ovrednotili uspešnost prvega prehoda algoritma za različne velikosti slovarja. Rezultati so podani na sliki 1. Iz grafa je razvidno manjšanje števila napak pri večanju slovarja tako pri uporabi bigramskega kot trigramskega modela. Pri primerjavi rezultatov med slovarjema 60k in 300k ugotovimo, da je bilo z bigramskim modelom doseženo zmanjšanje deleža napak za 4,67%, pri trigramskem modelu pa 5,09%. Ta izboljšanja so primerljiva z zmanjšanjem deleža besed izven slovarja, ki znaša 5,92%. Ta primerljivost je bila pričakovana, saj smo predvidevali, da bomo z večanjem slovarja izločili napake, ki nastanejo zaradi besed izven slovarja. Majhen delež preostalih besed izven slovarja in potek grafov na sliki pa kažeta na možnost, da z dodatnim večanjem slovarja nad 300k ne bomo več dosegli bistvenih izboljšav.

V nadaljevanju smo se osredotočili na modele s slovarjema 60k in 300k. V tabeli 2 so predstavljeni rezultati napačno razpoznanih besed in faktorja realnega časa. Iz podatkov vidimo, da povečanje slovarja iz 60k na 300k prinese 4,67% oz. 5,09% zmanjšanje deleža napačno razpoznanih besed. Izboljšanje pri zamenjavi bigramskega modela s trigramskim pa prinese 3,15% oz. 3,57% zmanjšanje deleža napačno razpoznanih besed. Iz podatkov o faktorju realnega časa lahko vidimo, da se algoritem ob uporabi večjega slovarja upočasni približno za faktor 2, ob zamenjavi bigramskega modela s trigramskim pa za faktor 3. Iz teh ugotovitev lahko povzamemo, da ima uporaba večjega slovarja tako večji doprinos k uspešnosti razpoznavanja kot



Slika 1: Uspešnost prvega prehoda.

tudi manjše povečanje časovne zahtevnosti kot pa zamenjava bigramskih modelov s trigramskimi.

Dobljeni rezultati za deleže besed OOV in rezultate razpoznavanja so primerljivi z sistemom, opisanem v (Nouza et al., 2010), za razpoznavanje češkega jezika, ki je po svojih lastnostih podoben slovenskemu.

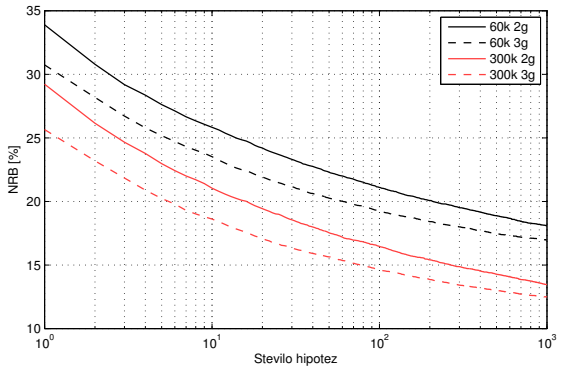
V naših poskusih smo uporabljali le ročno segmentirano testno množico. Izkušnje iz preteklih različic sistem pa kažejo, da z uporabo avtomatske segmentacije izgubimo približno 2 do 3 odstotke uspešnosti razpoznavanja (Žgank et al., 2008/1).

Tabela 2: Rezultati prvega prehoda.

Model	NRB [%]	RTF
60k 2g	33,89	6,29
60k 3g	30,74	18,46
300k 2g	29,22	12,66
300k 3g	25,65	37,09

Preden smo vrednotili razpoznavanje v drugem prehodu, smo si podrobneje pogledali hipoteze v N-najboljših seznamih. Na sliki 2 so prikazani deleži napak v N-najboljših seznamih, ki jih dobimo tako, da vzamemo najboljših n hipotez, pogledamo število napak v vsaki hipotezi in kot rezultat podamo najmanjše število napak. Tako dobimo pričakovano izboljšanje rezultata, če bi v naslednjem prehodu algoritma imeli na voljo idealni razpoznavalnik, ki bi vedno znal izbrati najboljšo hipotezo. Iz slike 2 je razvidno padanje števila napak z večanjem števila hipotez. Pri 1000 hipotezah se število napak razpolovi. Iz grafa je tudi razvidno, da je padanje števila napak hitrejše pri rezultatih, ki so bili dobljeni z uporabo bigramskega modela.

Zadnji rezultati, ki smo jih vrednotili se nanašajo na dvoprehodni algoritem. Podani so v tabeli 3. Rezultate prvega prehoda iskalnega algoritma pri modelih iz tabele 2 (prvi stolpec) smo ponovno vrednotili s štirigramskimi modeli. Izračunali smo nov delež napačno razpoznanih besed (tretji stolpec) in razliko glede na rezultat prvega prehoda (četrti stolpec). Izboljšanje rezultatov se kaže v primerih, kjer smo v prvem prehodu uporabljali bigramski jezikovni model. Pri teh primerih smo poskuse ponovili tudi s trigramskimi jezikovnimi modeli v drugem prehodu, ven-



Slika 2: Napake v N-najboljših seznamih.

dar ni opaznih razlik do podanih rezultatov uspešnosti in faktorja realnega časa pri uporabi štirigramskega modela. Iz rezultatov lahko predvidevamo, da dodatno višanje reda običajnih jezikovih modelov ne bo prineslo dodatnih izboljšav.

Tabela 3: Rezultati drugega prehoda.

1. prehod	2. prehod	NRB [%]	ΔNRB [%]	RTF
60k 2g	60k 4g	31,01	-2,88	0,02
60k 3g	60k 4g	30,73	-0,01	0,02
300k 2g	300k 4g	25,85	-3,37	0,03
300k 3g	300k 4g	25,64	-0,01	0,03

V zadnjem stolpcu so podani še faktorji realnega časa drugega prehoda. Ker v drugem prehodu delamo le z omejenim številom hipotez in uporabljamo le jezikovne modele, so ti faktorji zelo majhni v primerjavi s faktorji prvega prehoda. Skupni faktor realnega časa za dvoprehodni algoritem dobimo tako, da faktorja obeh prehodov seštejemo. Čas, ki je potreben za pretvorbo besednih grafov v sezname N-najboljših hipotez je v primerjavi s časom delovanja algoritma zanemarljiv. Za hitrost delovanja algoritma je torej pomemben predvsem prvi prehod. Vidimo, da smo z uporabo bigramskega modela v prvem prehodu in štirigramskega v drugem prehodu dosegli uspešnost razpoznavanja primerljivo (razlika 0,12% oz. 0,2%) z uspešnostjo enoprehodnega algoritma s trigramskim modelom, ki pa zahteva trikrat več časa za razpoznavanje.

6. Zaključek

V članku smo predstavili nekatere rezultata iz trenutnega razvoja razpoznavalnika UMB Broadcast News. Z uporabo razširjenega slovarja smo uspeli zmanjšati napako razpoznavanja na 25,65%. Uporaba dvoprehodnega algoritma pa nam je omogočila doseganje primerljivih rezultatov s trikrat krajšim časom razpoznavanja.

Predstavljeni rezultati nakazujejo, da z dodatnim večanjem slovarja in vpeljavo običajnih jezikovnih modelov višjih redov, kot pa smo jih sedaj uporabljali, ni več mogoče pričakovati bistvenih izboljšanj v uspešnosti razpoznavalnika.

Izboljšanja lahko pričakujemo s kombiniranjem predstavljenih pristopov in nekaterih drugih že pripravljenih izboljšav razpoznavalnika, kot so interpolirani jezikovni in izboljšani akustični modeli. Predstavljen sistem dvoprehodnega algoritma pa nam omogoča začetek raziskovanja na zapletenejših jezikovnih modelih, ki bi jih zgradili z uporabo morfosintaktičnih oznak iz korpusa FidaPLUS.

7. Literatura

Arhar Š., Gorjanc, V. 2007. Korpus FidaPLUS: nova generacija slovenskega referenčnega korpusa. *Jezik in slovstvo*, 52(2):95–110.

Aubert., X.L. 2002. An overview of decoding techniques for large vocabulary continuous speech recognition. *Computer Speech and Language*, 16(1):89–114.

Biem A., Mcdermott A., Katagiri, S. 2005. A discriminative filter bank model for speech recognition. In *ESCA Eurospeech 2005*, Lizbona, Portugalska.

HTK domača stran, http://htk.eng.cam.ac.uk

Nouza, J., Zdansky, J., Cerva, P., Silovsky, J. 2010. *Development of Multimodal Interfaces: Active Listening and Synchrony*, pogl. Challenges in speech processing of slavic languages (case studies in speech recognition of czech and slovak) Springer Berlin / Heidelberg

Rotovnik, T., Sepesy Maučec, M., Kačič Z. 2003. Large vocabulary continuous speech recognition of an inflected language using stems and endings. *Speech Communication*, 49(6):437–452.

Stolcke, A. 2002. SRILM - an extensible language modeling toolkit. In *International Conference on Speech and Language Processing*, Denver, Colorado.

Zablotskiy, S., Zablotskaya K., Minker W. 2010. Some approaches for russian speech recognition. In *Sixth Internation Conference on Intelligent Environments*, Kuala Lumpur, Malezija.

Žgank, A., Sepesy Maučec, M. 2006. Osnovna zgradba razpoznavalnika slovenskega tekočega govora UMB Broadcast News. *Jezikovne tehnologije 2006*, Ljubljana, Slovenija

Žgank, A., Kos, M., Kotnik, B., Sepesy Maučec, M., Rotovnik, T., Kačič, Z. 2008. Nadgradnja sistema za razpoznavanje slovenskega tekočega govora UMB Broadcast News *Jezikovne tehnologije 2008*, Ljubljana, Slovenija

Žgank, A., Verdonik, D., Kačič, Z. 2008. Slovenska baza BNSI Broadcas News za razpoznavanje tekočega govora. *Elektrotehniški vestnik*, 75(3):85–90.

Žgank, A., Sepesy Maučec, M. 2010. Razpoznavalnik tekočega govora UMB Broadcast News 2010: nadgradnja akustičnih in jezikovnih modelov. *Jezikovne tehnologije 2010*, Ljubljana, Slovenija

Jezikovni viri starejše slovenščine IMP: zbirka besedil, korpus, slovar

Tomaž Erjavec

Odsek za tehnologije znanja, Institut »Jožef Stefan«
Jamova cesta 39, Ljubljana
tomaz.erjavec@ijs.si

Povzetek

V prispevku predstavimo tri jezikovne vire starejšega slovenskega jezika: zbirko besedil oz. digitalno knjižnico, referenčni jezikoslovno označeni korpus in slovar. Zbirka besedil vsebuje večinoma knjige, z redigirano transkripcijo besedila in faksimili, korpus sestavlja besedilo strani, vzorčenih iz zbirke, kjer je vsaki besedni pojavnici pripisana ročno pregledana sodobna ustreznica, njena lema in leksikalna oblikoskladenjska oznaka, slovar pa je bil zajet iz razširjenega ročno pregledanega korpusa. Vsi trije viri so zapisani skladno s smernicami za zapis besedil TEI (Text Encoding Initiative Guidelines) in dostopni na spletu za pregledovanje in preiskovanje, kot tudi za prenos pod licenco Creative Commons – priznanje avtorstva. Ti viri po eni strani predstavljajo podatkovno infrastrukturo za razvoj jezikovnih tehnologij za starejšo slovenščino, po drugi pa omogočajo empirično podprte diahrone jezikoslovne raziskave in približajo starejša besedila in leksiko sodobnemu bralcu. Viri IMP so dostopni na http://nl.ijs.si/imp/.

Language resources for historical Slovene

The paper presents three language resources for historical Slovene: the text collection or digital library, a linguistically annotated reference corpus and a dictionary or lexicon. The text collections contains mostly books with hand corrected transcriptions and facsimile, the corpus comprises sampled pages from this text collections, where each word token has been manually annotated with its contemporary equivalent, its lemma and part-of-speech tag, while the lexicon was automatically extracted from the (extended) annotated corpus. The three resources are encoded according to the Text Encoding Initiative Guidelines and are available on the web for browsing and searching, as well as for download under the Creative Commons – Attribution licence. The resources, on the one hand, constitute a data infrastructure for the development of language technologies for historical Slovene, and, on the other, enable corpus based diachronic studies of Slovene and bring old texts and vocabulary closed to today's readers. The IMP language resources are available from http://nl.ijs.si/imp/.

1. Uvod

V prispevku predstavimo tri medsebojno povezane digitalne vire starejšega slovenskega jezika, ki predstavljajo referenčne podatkovne vire za razvoj jezikovnih tehnologij za obravnavo starejše slovenščine, imajo pa tudi namen razširiti ponudbo spletnih priročnikov in postaviti osnovo za korpusno podprte diahrone jezikoslovne študije. Če se osredotočimo samo na jezikovnotehnološke vidike uporabe, imajo viri naslednji namen:

- **zbirka besedil** vsebuje čistopise besedil, s tem pa tudi besedišče starejše slovenščine, ki je uporabno za izboljšanje avtomatskega prepoznavanja besedil iz slik (OCR); zbirka služi tudi kot osnovna podatkovna množica, ki jo lahko izkoristimo za povečanje jezikoslovno označenega korpusa in leksikona;
- **jezikoslovno označeni korpus** omogoča šolanje oblikoskladenjskih označevalnikov za starejši jezik in razvoj programov za avtomatsko posodabljanje starih besedil, s čimer jih naredijo bolj razumljiva sodobnemu bralcu;
- **slovar** omogoča šolanje modulov za posodabljanje in lematizacijo besed in je uporaben za podporo programom za iskanje po celotnem besedilu digitalnih knjižnic.

S formalno validacijo in upoštevanjem standardov poskrbimo, da so jezikovni viri brez tehničnih napak in da je njihova struktura dobro dokumentirana. Viri IMP so skladno zapisani in medsebojno povezani, pri čemer upoštevajo dvoje specifikacij:

- **shema XML IMP** definira strukturo XML posameznih, sicer strukturno zelo raznovrstnih virov;
- **oblikoskladenjske specifikacije IMP** definirajo nabor oblikoskladenjskih oznak, ki se uporabljajo v korpusu oz. slovarju.

Izdelava predstavljenih virov je bil razmeroma dolgotrajen in drag proces, zato je smiselno poskrbeti, da so ne samo čim širše uporabni, temveč tudi uporabljani. Vsi viri so dostopni neposredno na spletu skozi izvedene oblike (HTML), obenem pa tudi za prenos, pod licenco Creative Commons – Priznanje avtorstva.

2. Računalniški zapis

Glede na možnost prenosa celotnih virov je format, v katerem so zapisani, pomemben tudi za druge uporabnike. Viri IMP so zapisani v skladu s smernicami za zapis besedil TEI »Text Encoding Initiative Guidelines«, (TEI, 2007). Smernice temeljijo na standardu XML (Extensible Markup Language) in so namenjene za zapis besedil v znanstvene namene. Uporabljajo se za večino kompleksnejših izdaj v digitalnih knjižnicah kot tudi za zapis jezikoslovno označenih korpusov in slovarjev. Smernice TEI definirajo in dokumentirajo nabor oznak (elemente in atribute XML) za zapis strukturnih in konceptualnih lastnosti besedil. Smernice so izražene kot modularna in razširljiva shema XML, ki ji je pridružena podrobna dokumentacija, dostopne pa so pod odprtokodno licenco. Prva je iz leta 1994, zadnja izdaja smernic, ki se sicer sproti dopolnjuje in popravlja, pa je TEI P5 iz leta 2007 in je usklajena z ustreznimi smernicami W3C in ISO, ki jih s tem upoštevajo tudi naše izdaje.

V kontekstu starejše slovenščine velja posebej opozoriti na kodo, ki jo uporabljamo za besedila, zapisana v bohoričici, in sicer sl-bohoric, ki smo jo tudi prijavili na IANA (Internet Assigned Numbers Authority), skupaj s kodama za metelčico (sl-metelko) in dajnčico (sl-dajnko), čeprav slednjih dveh pisav vir IMP zaenkrat ne vsebujejo.

Kanonična oblika posameznega vira IMP je torej dokument XML, ki je veljaven (validiran) glede na shemo XML IMP, ta pa je narejena v skladu s smernicami TEI. S tem je zagotovljeno, da neodvisno od naših datotek obstaja podrobna dokumentacija, v kateri je struktura virov dokumentirana tako s proznim opisom (smernice) kakor s shemo XML, s pomočjo katere je mogoče pravilnost strukture teh virov tudi formalno preveriti.

Zapis TEI je namenjen izmenjavi, primeren je za raznovrstne uporabe, neodvisen od računalniške platforme, in, kolikor je to glede na hiter razvoj računalniških tehnologij sploh mogoče, odporen na zastaranje. Ta format s skriptami XSLT nato pretvorimo v formate za uporabo v konkretnem orodju (npr. konkordančniku) ali za prikaz na spletu.

3. Zbirka besedil

Zbirka besedil IMP je zasnovana kot digitalna knjižnica in vsebuje večinoma celotna dela, v obliki faksimilov in pregledanih transkripcij besedil. Trenutno vsebuje 158 del (13.000 strani oz. 2 milijona besed), sestavljena pa je iz zbirk AHLib (Erjavec, 2011) in zbirk besedil, označenih v NUK ter na ZRC SAZU (Erjavec in dr., 2011). Dela so večinoma celotne knjige, 38 enot pa so izdaje časopisa *Kmetijske in rokodelske novice*.

Stopnja označevanja se razlikuje glede na izvor posameznega dela, v vseh primerih pa vsebuje prelome strani s kazalci na faksimile, naslove, odstavke in oznake za posebne dele besedila, kot so številke strani, tiskarska znamenja itd., kot je razvidno iz primera na sliki 1.

Na spletu je zbirka predstavljena kot digitalna knjižnica, kjer je vsaka enota stavljena kot svoj HTML, ti pa so povezani s statičnimi kazali glede na različne metapodatke: naslov, avtor, leto, signatura.

4. Jezikoslovno označeni korpus

Ročno pregledani referenčni korpus goo300k (Erjavec, 2012) obsega 1.000 vzorčenih strani iz besedilne zbirke IMP oz. nekaj manj kot 300.000 besednih pojavnic. Postopek vzorčenja je potekal v dveh fazah. Najprej smo iz zbirke IMP izbrali besedila izdana pred letom 1900, ki so čim bolj raznovrstna, obenem pa ne preveč izstopajoča po zapisu, s čimer smo dobili 81 enot. Iz teh besedil smo nato naključno izbrali posamezne strani, z nastavljenimi maksimumi za število strani po besedilu in časovnem obdobju. Prednost smo dali besedilom, izdanim med letoma1850 in 1875, saj smo po eni strani imeli največ besedil iz tega obdobja, po drugi pa je jezik tu že zadosti različen od sodobne slovenščine, da bi bila jezikovnotehnološka podpora že upravičena, vendar hkrati še ni preveč različen, da bi bilo posodabljanje posameznih besed neuporabno. Vseeno pa korpus vsebuje tudi besedila, ki segajo do leta 1750 in celo par starejših vzorcev.

Korpus je bil najprej avtomatsko označen, nato pa ročno pregledan s strani skupine študentov, ki so pri svojem delu uporabljali urejevalnik CoBaLT (Kenter in dr. 2012). Poleg pregledovanja oznak je bilo popravljeno tudi besedilo, saj čistopis ni bil brez napak.

Korpus je zapisan v 1.001 datoteki, ena krovna, ostale pa za posamezne strani. Krovna datoteka vsebuje element teiCorpus, ki je sestavljen iz kolofona TEI (teiHeader) in serije elementov TEI, od katerih vsak vsebuje po eno enoto korpusa. Element TEI ima nato svoj kolofon, temu sledijo podatki o faksimilu (za vsako stran njen identifikator, URL-je grafičnih datotek na strežniku in ustrezne strani v digitalni knjižnici) in XInclude kazalce na posamezne strani, ki so vključene v korpus.

Kot je ilustrirano na sliki 2, ima posamezna stran kazalko na svoj faksimile in označeno besedilo, ki je najprej razdeljeno na bloke (naslove, odstavke itd.), ti na stavke oz. povedi, ti pa na besedne pojavnice, ločila in presledke.

```
<ab type="p" corresp="NUKR10214-1790/00422751.xml#r49">Goſpa. O ! jeſt niſim vezh tvoja <lb n="3"/>
Rosalka , katęro ſi ſizer lubil ! jeſt ſim<lb/>
ena oboga shéna, ena ſirota — <lb n="3"/>
nimam moshá!<lb/>
</ab>
<ab type="fw" subtype="catch" corresp="NUKR10214-1790/00422751.xml#r4">Ba-</ab>
<ab type="fw" subtype="sig" corresp="NUKR10214-1790/00422751.xml#r5">c 5</ab>
<pb xml:id="pb.095" n="95" facs="#NUKR10214-1790-00422752"/>
<ab type="fw" subtype="pageNum" corresp="NUKR10214-1790/00422752.xml#r1">58</ab>
<ab type="p" corresp="NUKR10214-1790/00422752.xml#r6">Baron. Vſmili ſe!<lb/></ab>
```

Slika 1. Zapis zbirke besedil.
Element ab zaznamuje »anonimni blok«, ki je opredeljen z vrednosti atributa @type: odstavek, naslov, tiskarsko znamenje, itd. Element lb poda prelom vrstice, pb pa strani. Slednji ima tudi kazalko na faksimile.

```
<?xml version="1.0" encoding="utf-8"?>
<div xmlns="http://www.tei-c.org/ns/1.0" type="pb" xml:lang="sl-bohoric" xml:id="goo18B-NUKR10214-1790.pb.095"
   n="NUKR10214-1790.pb.095_Shupanova_Mizka" facs="../goo300k.xml#NUKR10214-1790-00422752">
  <ab type="p" corresp="NUKR10214-1790/00422752.xml#r6" part="F">
    <s>
      <w nform="baron" mform="baron" lemma="baron" ctag="Ncm">Baron</w><pc ctag=".">.</pc>
    </s>
    <c> </c>
    <s>
      <w nform="vſmili" mform="usmili" lemma="usmiliti" ctag="Vme">Vſmili</w><c> </c>
      <w nform="ſe" mform="se" lemma="se" ctag="P">ſe</w><pc ctag="!">!</pc>
    </s>
  </ab>
  …
</div>
```

Slika 2. Zapis označenega korpusa.

Element div/@type="pb" vsebuje eno stran označenega besedila. Atributi določijo, da element pripada imenskemu prostoru TEI, da je »jezik« na tej strani bohoričica, mu podajo identifikator in labelo, ter kazalko na faksimile. Bloki so razdeljeni na stavke, ti pa na besede, ločila in presledke. Besede imajo pripisano normalizirano obliko, posodobljeno obliko, lemo in oblikoskladenjsko oznako.

4.1. Jezikoslovne oznake

Vsaki besedi v korpusu je pripisana normalizirana oblika, sodobna oblika, lema in oblikoskladenjska oznaka IMP (gl. sliko 2). Normalizirane oblike so zapisane z malimi črkami, poleg tega pa so odstranjena naglasna znamenja nad samoglasniki, saj se ta v sodobni slovenščini ne uporabljajo več.

Sodobne oblike besed so najbolj zanimive, saj takih oznak ne najdemo v korpusih sodobnega jezika. Razdelimo jih lahko v štiri skupine:

1. besedna oblika iz korpusa je enaka sodobni obliki, kot je to pri prvi besedi na sliki 2;
2. razlika je samo v zapisu posamezne besede, kot je to pri drugi besedi na sliki 2;
3. razlika je pri pisanju skupaj – narazen (npr. *nar bolj*, sedaj *najbolj* ali obratno *namoresh*, sedaj *ne moreš*); take besede so problematične s stališča zapisa jezikoslovnega označevanja, saj so posodobljene oblike sicer enostavno pripisane kot atribut posamezni besedi, tu pa je potrebno vzpostaviti relacijo med več besednimi pojavnicami in eno analizo oz. eno pojavnico in nizom analiz;
4. zastarele besede, torej tiste, ki nimajo sodobne ustreznice, ali pa so se jim spremenile skladenjske lastnosti, kot npr. spol; v takih primerih je kot posodobljena oblika vzeta kar zastarela beseda, vendar napisana v skladu s sodobnim pravopisom (npr. *ajfram* posodobimo v *ajfrom*), je pa takim besedam pripisana tudi najbližja sodobna ustreznica oz. ustreznice (v tem primeru *gorečnost*).

Pri zastarelih besedah velja še opomba, da kot zastarelih zaenkrat nismo šteli tistih, ki se pojavljajo v SSKJ, četudi imajo pripisano oznako *zastarelo* oz. *starinsko*. Glavni razlog je bil, da so te besede že obdelane v SSKJ in je sodobne ustreznice oz. razlage možno zajeti iz tega vira.

Besedam je nadalje pripisana lema oz. osnovna oblika besede, ki je tudi posodobljena in izhaja iz posodobljene besedne oblike, npr. *sonce*, *ne moči*, *ajfer*.

Tretji jezikoslovni podatek, ki je pripisan besednim pojavnicam, je njihova kontekstno razdvoumljena oblikoskladenjska oznaka – za namene projekta smo razvili nov nabor oznak oz. specifikacije zanje, kar je podrobneje opisano v naslednjem razdelku.

Korpus je dostopen za iskanje preko spletnega vmesnika, ki omogoča izpis konkordanc in frekvenčnih leksikonov, tudi po regularnih izrazih, iskanje in prikaz vseh oznak po besedah, filtriranje in prikaz bibliografskih podatkov, izračun kolokacij, itd.

4.2. Oblikoskladenjske specifikacije

V korpusih sodobnega jezika, kot sta FidaPLUS (Arhar in Gorjanc, 2007) in JOS[1] (Erjavec in Krek, 2008), uporabljamo oblikoskladenjske oznake, ki zajemajo tako leksikalne (npr. obči samostalnik srednjega spola) kot pregibne lastnosti (npr. rodilnik ednine) posameznih besed. V korpusu in slovarju smo ta sistem, ki zajema skoraj 2.000 različnih oznak, poenostavili in besedam pripisali samo leksikalne lastnosti, tako da število oznak pade na 32.

Oznake IMP so definirane, tako kot oblikoskladenjske oznake JOS, v dokumentu TEI, kjer so definirane besedne vrste, vsaki pripisane njene oblikoskladenjske lastnosti, množice teh pa pripisane posameznim oblikoskladenjskim oznakam, pri čemer so imena lastnosti in oznak definirana tako v slovenščini kot angleščini. V korpusu uporabljamo oznake v angleščini, vendar je te preko izvedenih tabel enostavno prevesti v slovenščino in jih tudi razstaviti v posamezne lastnosti. Tako imamo npr. korpusno oznako

[1] http://nl.ijs.si/jos/

Vmp, ki pomeni Verb Type = main, Aspect = progressive, ki je ekvivalentna slovenski oznaki Ggn oz. glagol vrsta = glavni, vid = nedovršni.

Razlog za uvedbo poenostavljenega nabora oznak je predvsem v tem, da je bil poudarek pri ročnem označevanju na posodobljenih oblikah besed, pri tem pa je natančno označevanje oblikoskladnje zelo zamudno – zato smo raje označili več besedila, vendar z bolj grobimi oznakami. Oznake so vseeno koristne, saj nam omogočijo, da v korpusu npr. iščemo vse kombinacije pridevnikov z neko besedo, po drugi strani pa jih lahko uporabimo za učenje modelov avtomatskega oblikoskladenjskega označevanja starejših besedil.

5. Slovar IMP

Tretji jezikovni vir starejše slovenščine je slovar oz. besedišče. Slovar je bil avtomatsko izluščen iz korpusa goo300k, poleg tega pa so mu bile dodane tiste besedne oblike iz zbirke besedil IMP, ki se ne pojavljajo v goo300k, v zbirki besedil pa vsaj dvakrat. Tudi tem besednim oblikam smo oznake najprej pripisali avtomatsko, nato pa ročno pregledali v CoBaLTu, ravno tako pa je bil naknadno pregledan še celoten slovar. Iz takšnega postopka izdelave seveda sledi, da slovar vsebuje samo korpusno izpričane oblike.

Celoten slovar vsebuje preko 25.000 lem, 50.000 besednih in 70.000 zgodovinskih besednih oblik, vendar to zajema vse pregledane besedne pojavnice iz korpusa oz. zbirke besedil, torej tudi številke, simbole, tujejezične in zatipkane besede in besedne oblike, ki so enake sodobnim. Če se omejimo samo na »prave« besede, pade število lem na nekaj pod 20.000, če samo na tiste leme, ki imajo vsaj eno besedno obliko drugačno, kot je sodobna, na 11.000, če samo na zastarele besede, pa na 2.000.

Slovar, kot vsak dokument TEI, vsebuje najprej kolofon, ki mu sledijo geselski članki. Kot je ilustrirano na sliki 3, vsebuje vsak zaglavje in korpusno izpričane besedne oblike. Zaglavje je sestavljeno iz geselske iztočnice, torej leme, njene oblikoskladenjske oznake oz. lastnosti in, za zastarele besede, sodobne ustreznice kot tudi vira, na osnovi katerega so bile te ustreznice določene. Posamezen geselski sestavek tako definira njegovo zaglavje: enake leme se kot homonimi pojavljajo v več geselskih člankih, če se ti razlikujejo glede na oblikoskladenjsko oznako ali sodobne ustreznice.

Zaglavju sledi seznam vseh sodobnih besednih oblik, vsaka od teh pa ima seznam svojih zgodovinskih različic. Vsaka zgodovinska različica je pospremljena s primeri uporabe iz korpusa, ki so jim pripisani bibliografski podatki. V slovarju je vključenih samo nekaj primerov za vsako zgodovinsko obliko, saj bi bilo vseh, posebej za visokofrekvenčne funkcijske besede, preveč.

Tako kot zbirka besedil je tudi slovar dostopen na spletu za pregledovanje, v več različicah, od polnega slovarja do tistega, ki vsebuje samo zastarele besede. Vsake različica slovarja je razdeljena na večje število strani v HTML, ki so med sabo povezane preko kazala po geselskih iztočnicah. Vsako geslo je opremljeno s kazalci v SSKJ in Pleteršnikov slovar na ZRC SAZU, v konkordančnik in v digitalno knjižnico.

```
<entry xml:id="lex.19e01750cca5a43b17fb31078b279905" n="anati-Vmp_izogibati_se">
  <form type="lemma">
    <orth type="hypothetical">anati</orth>
    <gramGrp norm="Vmp">
      <gram type="msd">Ggn</gram>
      <pos>glagol</pos> <gram type="vrsta">glavni</gram> <gram type="vid">nedovršni</gram>
    </gramGrp>
    <gloss>izogibati se</gloss> <bibl>Pleteršnik</bibl>
  </form>
  <form type="wordform">
    <orth type="hypothetical">anaj</orth>
    <form type="historical">
      <orth type="normalised">anej</orth>
      <cit>
        <quote xml:lang="sl-bohoric">drugemi pohujshanje, ke je njim ſavol myrnoſte tega serza
        navoshliv. <oVar>Anej</oVar> se slehernega kraja, kjer myru ni, inu vogibej se tajſteh </quote>
```

Slika 3. Zapis slovarja.

Geslo (entry) vsebuje podatke o lemi (form/@type="lemma") in njenih posodobljenih besednih oblikah (form/@type="wordform"), ti pa vsebujejo vse atestirane besedne oblike (form/@type="historical") skupaj s primeri uporabe.

6. Dostopnost virov

Če je ena plat dostopnosti virov njihov zapis, je druga dejanska možnost dostopa do njih. Dostopnost jezikoslovnih virov je v Sloveniji vse prevečkrat omejena na spletno pregledovanje, kar sicer v večini primerov zadošča za njihovo neposredno (jezikoslovno) uporabo, ne omogoča pa uporabe za razvoj jezikovnih tehnologij ali za bolj poglobljene, celostne jezikoslovne študije, kjer potrebujemo možnost prenosa celotnega vira na lastni računalnik. Razlogi za takšno zapiranje so v nekaterih primerih sicer legitimni (npr. avtorska zaščita izvornih besedil), v večini primerov pa so bolj želja institucije, ki je vir razvila, da ohrani monopol nad njim, in to kljub temu, da je bil izdelan z javnimi sredstvi (Erjavec, 2009) – najbolj znan primer zapiranja nacionalno pomembnega jezikovnega vira je seveda SSKJ.

Za vse vire IMP velja, da so dostopni ne samo za pregledovanje preko spleta, temveč tudi za prenos v kanonični obliki TEI po licenci *Creative Commons, priznanje avtorstva*. Licenca omogoča prenos virov za uporabo v raziskovalne namene ali komercialno, kot tudi predelavo virov in nadaljnjo distribucijo virov. Edini pogoj, ki je postavljen, je, da se vir IMP, ki se ga uporablja, tudi primerno citira.

7. Zaključki

V članku smo predstavili tri uniformno zapisane, medsebojno povezane in prosto dostopne referenčne vire starejšega slovenskega jezika, dosegljive na http://nl.ijs.si/imp/.

V nadaljnjem delu bi želeli razširiti ročno označeni korpus, predvsem z besedili iz druge polovice 18. in prve polovice 19. stoletja, ravno tako pa zbrati še dodatna pregledana besedila in s pomočjo teh povečati tudi slovar.

Zaželeno bi bilo tudi obogatiti metapodatke virov IMP. Tako ima npr. trenutno vsaka publikacija en naslov, kot smo ga dobili s strani izdelovalcev digitalne predloge. Vendar pa so, posebej pri starejših besedilih, naslovi v različnih jezikih ali pisavah, zato bi bilo koristno imeti naslov v več variantah, tudi v sodobni slovenščini.

Glavna bodoča naloga pa je uporaba razvitih virov, tako v lastnih raziskavah, ki se bodo osredotočile na posodabljanje starejših besedil, kot tudi spodbujanje drugih, da vire uporabijo pri svojih raziskavah in razvoju. Trenutno se slovar že uporablja pri podpori iskanja po starejših besedil v digitalni knjižnici dLib.si, imajo pa izdelani viri še mnogo širše potenciale.

Zahvala

Avtor se zahvaljuje anonimnima recenzentoma za koristne pripombe in nasvete. Pri delu, ki je opisano v prispevku, so sodelovali Kozma Ahačič, Tina Benčina, Katja Cingerle, Metod Čepar, Darja Fišer, Alenka Jelovšek, Urška Kamenšek, Alenka Kavčič Ćolić, Maša Kodrič, Nina Mikulin, Matija Ogrin, Daša Pokorn, Erich Prunč, Zala Šmid, Ines Vodopivec in Maja Žorga Dulmin. Delo sta podprla projekt EU IP IMPACT *Improving Access to Text* in nagrada Google *Developing Language Models of Historical Slovene*

Literatura

Špela Arhar, Vojko Gorjanc. 2007. Korpus FidaPLUS: nova generacija slovenskega referenčnega korpusa. Jezik in slovstvo, 52 (2). URL: http://www.jezikinslovstvo.com/pdf/2007-02-Razprave-SpelaArharInVojkoGorjanc.pdf

Tomaž Erjavec, Simon Krek. 2008. Oblikoskladenjske specifikacije in označeni korpusi JOS. Zbornik Šeste konference Jezikovne tehnologije, Ljubljana. URL: http://nl.ijs.si/jos/bib/jos_isltc08.pdf

Tomaž Erjavec. 2009. Odprtost jezikovnih virov za slovenščino. V zborniku Obdobja, Simpozij »Infrastruktura slovenščine in slovenistike«. str. 115–121. URL: http://www.centerslo.net/files/file/simpozij/simp28/Erjavec.pdf

Tomaž Erjavec. 2011. Slovenska prevodna književnost 1848–1918 : digitalna knjižnica in korpus AHLIB. V zborniku Meddisciplinarnost v slovenistiki, (Obdobja, Simpozij, = Symposium, 30). Ljubljana: Znanstvena založba Filozofske fakultete, str. 33–40. URL: http://www.centerslo.net/files/file/simpozij/simp30/Zbornik/Erjavec.pdf

Tomaž Erjavec, Ines Jerele, Maša Kodrič. 2011. Izdelava korpusa starejših slovenskih besedil v okviru projekta IMPACT. V zborniku Obdobja, Simpozij »Meddisciplinarnost v slovenistiki«. Ljubljana: Znanstvena založba Filozofske fakultete, str. 41–47. URL: http://www.centerslo.net/files/file/simpozij/simp30/Zbornik/Erjavec_Jerel_Kodric.pdf

Tomaž Erjavec. 2012. The goo300k corpus of historical Slovene. V zborniku Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul. European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2012/summaries/445.html

Tom Kenter, Tomaž Erjavec, Maja Žorga Dulmin, Darja Fišer. 2012. Lexicon construction and corpus annotation of historical language with the CoBaLT editor. V zborniku EACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, Avignon, France, April. ACL.

TEI Consortium (2007). TEI P5: Guidelines for Electronic Text Encoding and Interchange. URL: http://www.tei-c.org/Guidelines/P5/

Referenčni korpusi slovenskega jezika (cc)Gigafida in (cc)KRES

Tomaž Erjavec*, Nataša Logar Berginc**

* Odsek za tehnologije znanja, Institut "Jožef Stefan"
Jamova cesta 39, SI-1000 Ljubljana
tomaz.erjavec@ijs.si
** Fakulteta za družbene vede, Univerza v Ljubljani
Kardeljeva ploščad 5, SI-1000 Ljubljana
natasa.logar@fdv.uni-lj.si

Povzetek

V prispevku predstavimo nova referenčna korpusa slovenskega jezika Gigafido in KRES, s poudarkom na njunima prostodostopnima različicama ccGigafidi in ccKRES-u. Gigafida je nadgradnja korpusa FidaPLUS, vsebuje novejša besedila, tudi besedila s spleta, ima dobro milijardo besed in je na novo oblikoskladenjsko označena. KRES je uravnoteženi del Gigafide, vsebuje pa 100 milijonov besed. Z metodo vzorčenja odstavkov sta bila narejena korpusa ccGigafida in ccKRES, ki sta desetkrat manjša od izvirnikov, zato pa dostopna za prenos po licenci Creative Commons Priznanje avtorstva-Nekomercialno in sta do sedaj največja prosto dostopna korpusa slovenskega jezika. V zadnjem delu prispevka podajamo ugotovitve o značilnih elementih vsake od šestih taksonomskih kategorij ccGigafide, tako po najbolj zastopanih metapodatkih, kot so založba, naslov in avtor, kot leksikalno, s pomočjo metode frekvenčnega profila.

The (cc)Gigafida and (cc)KRES Slovene reference corpora

The paper introduces the new reference corpora of Slovene Gigafida and KRES, with a focus on their freely available derivatives ccGigafida and ccKRES. Gigafida is an upgrade of the FidaPLUS corpus which includes newer texts, also from the internet, contains about one billion words and has an improved morphosyntactic tagging. KRES is a balanced sample of Gigafida containing 100 million words. The ccGigafida and ccKRES corpora are ten times smaller than their base corpora and were made by random paragraph selection. Both are available in their source form under the Creative Commons — Attribution-NonCommercial licence and are currently the largest freely available corpora of Slovene. The paper concludes with an analysis of the six taxonomic categories of ccGigafida obtained through inspection of their most frequent metadata categories, such as publisher, author and title, and through their lexical profiles, obtained with the method of frequency profiling.

1. Uvod

Za slovenski jezik obstaja sedaj že večje število korpusov, tako referenčnih kot specializiranih, vendar dostop do njih pri veliki večini poteka zgolj prek spletnih konkordančnikov. Uporaba korpusa prek konkordančnika je nujno omejena, saj jo določa zmogljivost orodja, pa tudi obseg rezultatov poizvedb ima dostikrat vnaprej določene meje. Kljub temu je za jezikoslovne študije tak dostop v večini primerov zadosten. Tega pa ne moremo reči za uporabo korpusov v namene razvoja jezikovnih tehnologij, ker tam potrebujemo dostop do celotnega korpusa kot podatkovne baze, saj ga šele tako lahko uporabimo za učenje oz. testiranje različnih programov za obdelavo jezika.

Za slovenski jezik že obstaja nekaj korpusov, ki so dostopni za prenos, npr. slovenski deli večjezičnih korpusov MULTEXT-East (Erjavec, 2004) in JRC-ACQUIS (Steinberger et al., 2006), vendar ta dva vsebujeta zelo ozko zvrst besedil: v prvem primeru je to samo roman »1984« G. Orwella, v drugem pa besedila pravnega reda Evropske unije. Bolj zanimiva sta korpusa JOS[1] (Erjavec in Krek, 2008), ki sta vzorčena iz korpusa FidaPLUS: jos100k ima 100.000 besed in vsebuje ročno jezikoslovno označena besedila, jos1M pa je delno ročno označen in ima milijon besed. Čeprav velikost slednjega ni zanemarljiva, je za današnji čas še vedno razmeroma majhen, poleg tega pa vsebuje samo besedila FidePLUS, torej besedila, izdana do leta 2006, ravno tako pa nobenih besedil s spleta (oz. le zanemarljiv, nesistematično zbran 1,24-odstotni delež).

V prispevku predstavljamo dva nova prostodostopna korpusa slovenskega jezika, nastala v sklopu projekta *Sporazumevanje v slovenskem jeziku*,[2] ki sta veliko večja od korpusa jos1M, vsebujeta tudi novejša besedila, sta pa, kar je glede na velikost tudi edino možno, samo avtomatsko jezikoslovno označena. V nadaljevanju prispevka tako najprej na kratko predstavljamo korpusa Gigafida in KRES, v razdelku 3 postopek vzorčenja, s katerim je bil najprej narejen KRES, nato pa še ccGigafida in ccKRES, v razdelku 4 razpravljamo o dostopnosti korpusov, sledi analiza značilnih elementov vsake od taksonomskih kategorij ccGigafide, nato pa še zaključki ter načrti za nadaljnje delo.

2. Gigafida in KRES

2.1. Gigafida

Gigafida je nadgradnja referenčnega korpusa slovenskega jezika FidaPLUS (Arhar Holdt in Gorjanc, 2007) s 621 milijoni besed, ki je bil povečan za 560 milijonov, tako da je dosegel želeno velikost milijarde in 180 milijonov besed (oz. natančno: 1.187.002.502). Nova besedila so bila izbrana po razmeroma kompleksni mreži meril: glede na besedilne zvrsti, letnice izida, ocene branosti itd. (Logar Berginc in Šuster, 2009; Kazalnik 1, 2009; Arhar Holdt, Kosem in Logar Berginc, 2012). Največja nova pridobitev so besedila s spleta, ki v korpus

[1] http://nl.ijs.si/jos/

[2] http://www.slovenscina.eu/

prinašajo 185 milijonov besed oz. dobrih 15 %,[3] sicer pa so v Gigafidi besedila iz obdobja 1990–2011.[4]

Že FidaPLUS je vsebovala avtomatsko pripisane jezikoslovne oznake na besednih pojavnicah, in sicer leme in oblikoskladenjske oznake. Za korpus Gigafida je bil izboljšan postopek avtomatskega označevanja (Krek, Grčar in Dobrovoljc, 2012), popravljen pa je bil tudi sistem oblikoskladenjskih oznak: FidaPLUS se je ravnala po določilih za slovenski jezik MULTEXT-East različice 3.0[5], medtem ko Gigafida upošteva oznake, razvite v okviru projekta JOS (Erjavec in Krek, 2008), ki so definirane v določilih za slovenski jezik MULTEXT-East različice 4.0[6] (Erjavec, 2010).

Struktura Gigafide je enaka kot struktura FidePLUS, tj. vsakemu besedilu ustreza ena datoteka, ki je obenem tudi dokument XML. Gigafida je bila prečiščena glede na uporabo znakov Unikod in zapisana v skladu s priporočili za zapis besedil TEI P5 (TEI Consortium, 2007), medtem ko je bila pri FidiPLUS uporabljena še starejša različica TEI P3. Za namene korpusa Gigafida in pridruženih korpusov smo naredili parametrizacijo TEI, na osnovi katere je nato mogoče narediti shemo XML, ki je neposredno uporabna za validacijo dokumentov korpusa Gigafida. Glede na FidoPLUS so bili spremenjeni tudi določeni podatki v kolofonu TEI, predvsem je bila dodana nova, enostavnejša taksonomija besedilnih zvrsti (gl. prvi stolpec v Tabeli 1).

Taksonomija	**Delež besed v %**
Tisk	80
Knjižno	35
Leposlovje	17
Stvarna besedila	18
Periodično	40
Časopisi	20
Revije	20
Drugo	5
Internet	20
Novičarski portali	8
Podjetja in ustanove	12
SKUPAJ	**100**

Tabela 1: Delež besed po taksonomiji v KRES-u.

2.2. KRES

Za korpuse, ki predstavljajo celovito podobo nekega jezika, je ključno, da so veliki in besedilnozvrstno pestri. Gigafida je tak, referenčni korpus, težko pa bi mu pripisali uravnoteženost, saj je v njem – kot posledica tega, da smo v Gigafido vključili vse, kar smo dobili in je avtorskopravno urejeno s pogodbo – 77 % besed iz periodike (časopisi, revije) in le dobrih 6 % besed iz knjig (leposlovje, stvarna besedila). Kot Gigafidin uravnoteženi podkorpus smo zato že predhodno načrtovali 100-milijonski KRES, katerega sestava je podana v Tabeli 1.

Izbiro besedil za KRES v smislu *kaj in koliko* sta poleg vnaprej dogovorjenih deležev po taksonomiji usmerjala dva vira podatkov: *Nacionalna raziskava branosti*[7] (NRB), v kateri so podatki o recepciji časopisov in revij, ter *Merjenje obiskanosti spletnih strani MOSS*[8], na podlagi katerega smo določili obseg besed s treh najbolj obiskanih novičarskih portalov (*24ur.com*, *rtvslo.si*, *siol.net*). Pri vseh drugih taksonomskih kategorijah smo sledili razmerjem v Gigafidi: iz leposlovja smo v KRES zajeli 71 % celote, iz stvarnih besedil 36 %, iz kategorije drugo 96 % zapisov sej Državnega zbora RS in besedil z RTV Slovenija, v okviru spletnih besedil pa še 12,5 % besed s strani podjetij ter 87,5 % besed s strani ustanov.[9] Natančni opredelitvi besedil in količine je sledilo vzorčenje.

3. Postopek vzorčenja

Osnova za vzorčenje besedil za KRES je bila tabela, v kateri posamezna vrstica vsebuje bibliografske podatke besedila oz. besedil in zahtevano število besed zanje. Bibliografski podatki v vzorčni tabeli vsebujejo naslov, letnico izida, založbo, umestitev v taksonomijo Gigafide ter vir dela, pri čemer ni nujno, da so v posamezni vrstici navedeni vsi podatki. Tako so npr. knjižna dela polno opisana in eni vrstici vzorčne tabele tipično ustreza ena datoteka Gigafide, medtem ko imajo internetna besedila podano število besed samo glede na domeno (vir) in eni vrstici ustreza večje število datotek, kar velja tudi za revije ter časopise. V prvi fazi vzorčenja smo zato identificirali besedila, ki ustrezajo eni bibliografski postavki, pri čemer smo izpustili datoteke, ki imajo manj kot 20 besed.

Postopek vzorčenje je bil podoben tistemu, ki smo ga razvili za izdelavo korpusov jos100k in jos1M, ki sta bila vzorčena iz korpusa FidaPLUS (Erjavec in Krek, 2008). Enota vzorčenja ni posamezno besedilo, pač pa odstavek, s čimer omogočamo čim boljšo zastopanost posameznih del. Če bi v korpus dodajali celotna besedila, bi neko besedilo ali v celoti izpadlo ali pa bi bilo – posebej pri obsežnejših besedilih, kot so knjige ali celotni letniki časopisov, združenih v eno datoteko – v korpusu preveč prevladujoče. Seveda pa ta način vzorčenja pomeni, da v korpusu niso več zajeta celotna besedila.

Iz Gigafide smo vzeli vse identifikatorje posameznih odstavkov skupaj s številom besed, ki jih vsebujejo, in ta seznam premešali, tako da je postalo zaporedje odstavkov v njem naključno. Program za vzorčenje je nato iz seznama odstavkov zaporedoma jemal njihove identifikatorje in njihovo število besed prištel vsoti glede na posamezno vrstico vzorčne tabele. Če je bila skupna vsota besed za vrstico manjša, kot je zahtevano število besed, se je odstavek dodal v vzorčeni korpus, sicer pa ne. Na ta način smo dobili množico naključno izbranih odstavkov, ki skupaj zadoščajo zahtevam, ki jih izraža

[3] Pajkanje spletnih besedil je izvedel sodelavec Miha Grčar (Institut "Jožef Stefan"), ki je celotni postopek opisal v Logar Berginc et al. (2012).

[4] Več o zbiranju gradiva in zgradbi Gigafide gl. v Arhar Holdt, Kosem, Logar Berginc (2012) in Logar Berginc et al. (2012)

[5] http://nl.ijs.si/ME/V3/

[6] http://nl.ijs.si/ME/V4/

[7] http://www.nrb.info/

[8] http://www.moss-soz.si/

[9] Ker je šlo tokrat v metodološkem smislu za prvi večji poskus pridobivanja spletnih besedil za referenčni korpus pri nas, ki bi lahko oblikoval smernice za prihodnjo gradnjo takih korpusov slovenščine ter nakazal nekatere zanimive (besedilnozvrstno primerjalne) jezikoslovne analize, smo prvotno načrtovali zelo okviren oz. širok obseg internetnega dela korpusa: od 10 do 50 % besed. Pri izbiri spletnih besedil za Gigafido smo se – dokaj poskusno – omejili na strani z informativnimi vsebinami (novičarski portali, ustanove, podjetja; o načinu izbora gl. Logar Berginc et al., 2012), nadaljnjih omejitev glede dolžine besedila, vsebine, formata zapisa ipd. pa za proces pajkanja nismo podali.

vzorčna tabela. V zadnjem koraku vzorčenja je program uporabil izbran seznam identifikatorjev odstavkov in te odstavke nato vzel iz Gigafide – ostale podatke o besedilu, predvsem metapodatke, pa prepisal ter določene dele priredil dejstvu, da je vzorec besedila sedaj del vzorčenega korpusa in ima manjši obseg kot izvirnik.

Enak postopek kot za vzorčenje korpusa KRES je bil izveden tudi za korpusa ccGigafida in ccKRES, a s to razliko, da sta bili vzorčni tabeli izdelani avtomatsko: v tabeli za ccGigafido je vsaki vrstici ustrezala natanko ena datoteka Gigafide, število zahtevanih besed zanjo pa je bilo nastavljeno na 9 % celotnega števila besed v datoteki, enako pa tudi za ccKRES, samo da je bil tu izvorni korpus KRES, in ne Gigafida.

4. Dostopnost

Tako Gigafida kot KRES sta prosto dostopna prek konkordančnika, vendar pa smo se že na začetku gradnje odločili, da bomo pri obeh omogočili tudi dostop do celote kot podatkovne baze ter tako omogočili izvedbo kvantitativnih raziskav, ki so omejene samo z domišljijo in znanjem programskih orodij. Prenos celotnega korpusa omogoča njegovo uporabo tudi za razvoj jezikovnih tehnologij, kot npr. razvoj modelov oblikoskladenjskega označevanje in lematizacije. Ali kot je obširneje opisano v Erjavec (2010): šele odprtost jezikovnih virov za prenos omogoča njihovo polno izkoriščanje, zagotavljanje takšnega dostopa pa bi pravzaprav morala biti moralna zaveza izdelovalcev vseh jezikovnih virov, ki so nastali s pomočjo javnih sredstev.

Pogodba z besedilodajalci Gigafide onemogoča nadaljnje razširjanje celotnih besedil, vključenih v korpus, dovoljuje pa, da se omogoči poln dostop do 10 % posameznega besedila – 4. člen se namreč glasi:

Imetnik pravic dovoli, da se do 10 % dela uporabi na način, kot to določa licenca Creative Commons. V tem delu na naročnika neizključno, neodplačno in brez časovnih omejitev prenaša pravico reprodukcije, distribucije, dajanja v najem, priobčitve javnosti in predelave avtorskega dela, ki je predmet te pogodbe in njegovih predelav v skladu ter na način, kot to določa licenca Creative Commons: "priznanje avtorstva" + "nekomercialno" + "deljenje pod istimi pogoji". Ta licenca dovoli uporabnikom avtorsko delo in njegove predelave reproducirati, distribuirati, dajati v najem, priobčiti javnosti in predelovati samo pod pogojem, da navedejo avtorja, da ne gre za komercialno uporabo in da tudi oni naprej širijo izvirna dela/predelave pod istimi pogoji.

Zaradi navedenega smo iz Gigafide in KRES-a vzorčili podkorpusa, ki zadostujeta kriteriju "do 10 %", natančneje: program za vzorčenje smo nastavili – kot je bilo že pojasnjeno – na 9 %.

Korpusa ccGigafida in ccKRES sta odprta za prenos pod licenco Creative Commons Priznanje avtorstva-Nekomercialno. Licenca, na kratko označena s CC BY-NC, določa, da je dovoljeno reproduciranje, distribuiranje, dajanje v najem in priobčevanje korpusa javnosti, kot tudi predelava korpusa pod pogojem, da se prizna avtorstvo korpusa oz. besedil in da se ga ne uporablja v komercialne namene. Priznanje avtorstva pomeni, da je pri uporabi korpusa treba navesti ime korpusa, za posamezne iztržke navesti tudi izvirnega avtorja oz. besedilodajalca, v strokovnih in znanstvenih publikacijah pa citirati ustrezno bibliografijo, ki ta korpus opisuje, enako, kot je to že sicer navada pri citiranju raziskav.

S korpusoma ccGigafida in ccKRES tako omogočamo tretjim osebam, da pod čim bolj liberalnimi pogoji poglobljeno raziskujejo slovenski jezik tako z jezikoslovnega kot računalniškega oz. jezikovno-tehnološkega vidika. Pri slednjem je najbolj pomembna izdelava sekundarnih jezikovnih virov, kot so frekvenčni seznami besed in lem, besednih zvez in terminov ter modelov za jezikoslovno označevanje.

	Č	R	L	S	D	I
1.	*v*	*in*	*biti*	*in*	*ta*	*ta*
2.	*leto*	*se*	*se*	*z*	*in*	*da*
3.	*ob*	*z*	*on*	*ali*	*člen*	*pa*
4.	*slovenski*	*on*	*jaz*	*on*	*se*	*biti*
5.	*občina*	*lahko*	*da*	*v*	*za*	*ne*
6.	*odstotek*	*ali*	*in*	*se*	*ali*	*se*
7.	*tekma*	*svoj*	*ne*	*ki*	*da*	*zakon*
8.	*tolar*	*ti*	*reči*	*lahko*	*biti*	*o*
9.	*za*	*pri*	*ko*	*ta*	*zakon*	*jaz*
10.	*mesto*	*ki*	*kaj*	*pri*	*ne*	*če*
11.	*nov*	*slika*	*ti*	*če*	*ki*	*člen*
12.	*ura*	*ne*	*ta*	*str.*	*o*	*za*
13.	*milijon*	*jaz*	*vedeti*	*drug*	*v*	*ti*
14.	*predsednik*	*zelo*	*moj*	*kot*	*če*	*kaj*
15.	*včeraj*	*če*	*videti*	*kateri*	*z*	*imeti*

Tabela 2: Leme, značilne za taksonomske kategorije ccGigafide.

5. Primerjava taksonomskih kategorij ccGigafide

Kot je razvidno iz Tabele 1, so besedila v Gigafidi razdeljena v šest taksonomskih kategorij: *leposlovje* (L), *stvarna besedila* (S),[10] *časopisi* (Č), *revije* (R), *drugo* (D) in *internet* (I). V tem razdelku obravnavamo posamezne kategorije v Gigafidi z dveh vidikov:

a) Za vsako kategorijo v nadaljevanju v ločenih točkah najprej podamo eno ali dve tabeli, ki vsebujeta število besed za posamezno kategorijo, in praviloma deset založb, avtorjev ali naslovov, ki v ta del korpusa prispevajo največ besed.

b) Drugi vidik predstavitve zajema prikaz najbolj specifične leksike za posamezno kategorijo. Tu smo uporabili metodo frekvenčnega profila (angl. *frequency profiling*), ki sta jo vpeljala Rayson in Garside (2000), in z njo poiskali leme, ki vsako posamezno kategorijo značilno loči od preostalega dela Gigafide. Za izračun smo uporabili kar ccGigafido, saj to na rezultate ne vpliva, je pa zato obdelava bistveno hitrejša.

Za izdelavo frekvenčnega profila smo najprej izdelali frekvenčni seznam lem vsakega od podkorpusov ter preostalega dela ccGigafide, nato pa za vsako lemo izračunali njeno logaritemsko verjetnost (angl. *log-likelihood*, LL). LL upošteva frekvenci elementa, kot tudi velikosti obeh korpusov, ki jih primerjamo, in večji, kot je, bolj je element značilen za enega od njiju. V Tabeli 2 prikazujemo prvih 15 lem, ki so značilne za vsako od taksonomskih kategorij.

[10] Da ne bi bilo poimenovanje *stvarna besedila* zavajajoče, naj poudarimo, da smo to kategorijo pripisali le knjigam, v celoti gledano pa »stvarno« oz. nefikcijsko vsebino seveda vsebujejo tudi vse druge neleposlovne kategorije (časopisi itd.; več o razlogih za tovrstno poimenovanje gl. v Logar Berginc in Šuster, 2009: 63).

Ugotovimo lahko, da po številu polnopomenskih besed v vrhnjem delu seznama izstopajo časopisi, saj je pri njih med prvimi 15 lemami polnopomenskih besed kar 12, medtem ko pri ostalih kategorijah prevladujemo zaimki, vezniki, predlogi in členek *ne*. Pri slednjih tudi sicer razlike niso velike, saj se vsaj v štirih taksonomskih kategorijah že v tem delu lestvice pojavijo *in*, *se*, *ne* in *če*, po drugi strani pa se *svoj* pojavlja samo pri revijah, *ko* in *moj* samo pri leposlovju, *drug*, *kot* in *kateri* samo pri stvarnih besedilih ter *pa* samo pri internetu. Od 16. mesta naprej pa tudi v ostalih kategorijah, ne le pri časopisih, prevladujejo polnopomenske besede. Pri analizi leksike v naslednjih razdelkih smo se omejili na samostalnike, glagole in pridevnike do 100. mesta.

5.1. Leposlovje

Kot kaže Tabela 3, je od založb v leposlovju po številu besed najbolj zastopana Mladinska knjiga, saj v tem delu korpusa prispeva več kot tretjino besed. Tabela 4 kaže po obsegu najbolj zastopane leposlovne avtorje. Skoraj četrtina besedil po količini besed tu nima pripisanega avtorja, med prvimi devetimi imeni pa je le en avtor domači, vsi drugi so v korpusu zastopani s prevodi (gre za visoko brane avtorje trivialnih del – z izjemo romana *Paradiso* J. Lezame Lime).

Založba	**Število besed**
VSE	23.969.196
Mladinska knjiga	8.310.664
DZS	3.689.486
Karantanija	2.287.703
Študentska založba	1.945.974
Didakta	1.556.171
Delo Revije	818.131
Litera	672.474
Tuma	594.222
Mohorjeva družba	361.093
Genija	258.474

Tabela 3: Založbe z največ besedami v kategoriji leposlovje.

Avtor	**Število besed**
VSE	23.969.196
neznani avtor	5.008.529
Barbara Cartland	818.131
Joachim Friedrich	391.131
José Lezama Lima	382.056
Danielle Steel	368.872
Dan Brown	358.990
Mary Higgins Clark	317.543
Maeve Binchy	297.992
John Grisham	297.591
Edo Rodošek	278.754

Tabela 4: Avtorji z največ besedami v kategoriji leposlovje.

Leksikalna analiza po metodi frekvenčnega profila nam je pokazala, da je kategorija leposlovje izrazito glagolska, saj je do 100. mesta na seznamu kar četrtina glagolov, in sicer predvsem glagolov sporočanja, mišljenja in zaznavanja (*reči*, *govoriti*, *vprašati*, *povedati*, *vedeti*, *misliti*, *pomisliti*, *zdeti se*, *slišati*, *videti*, *gledati*, *pogledati*) ter premikanja (*iti*, *priti*, *oditi*, *vrniti*, *stopiti*, *obrniti*). Samostalnikov je manj in poimenujejo (del) človeka ali prostor: *roka*, *oči*, *glava*, *obraz*, *pogled*, *glas*, *oče*, *mama*, *gospod*; *soba*, *hiša*; izstopata še *miza* in *trenutek*. Pridevnika ni nobenega.

5.2. Stvarna besedila

Knjige, ki smo jih označili s stvarna besedila, je v največjem obsegu prispevala založba DZS (Tabela 5), v celoti imamo tu 1.082 različnih (znanih) naslovov. Tabela 6 kaže, da desetini stvarnih besedil naslova nismo uspeli določiti.

Založba	**Število besed**
VSE	50.387.335
DZS	14.078.488
Mladinska knjiga	6.152.240
Krtina	2.321.758
Desk	2.066.592
GV Založba	1.744.891
Zavod RS za šolstvo	1.691.456
Založba /*cf.	1.599.493
Tehniška založba Slovenije	1.275.446
Fakulteta za socialno delo UL	1.188.566
Cistercijanska opatija Stična	1.119.858

Tabela 5: Založbe z največ besedami v kategoriji knjižnih stvarnih besedil.

Naslov	**Število besed**
VSE	50.387.335
neznani naslov	5.506.017
Sociologija	355.008
Učenje in poučevanje tujih jezikov na Slovenskem	345.003
Vojna zgodovina	283.566
Vrtnarski priročnik	251.809
Evropsko kmetijsko pravo	250.598
Kog: krajepis in zgodovinopis	223.474
Družinska enciklopedija zdravil	222.442
Annales	219.417
Slovensko domobranstvo	217.884
Sodobna politična filozofija: uvod	215.055

Tabela 6: Naslovi z največ besedami v kategoriji knjižnih stvarnih besedil.

Z leksikalna analizo smo ugotovili, da je v stvarnih besedilih več samostalnikov, ki jih po (prvotnem) pomenu lahko povežemo z izobraževanjem (*učitelj*, *znanje*, *učenje*, *učenec*), strokovnimi besedili (*stran*, *slika*, *primer*, *naloga*, *poglavje*) in računalništvom (*okno*, *vnos*, *sistem*, *datoteka*), ter abstraktnejših, na značilnost predmeta ali proces vezanih izrazov (*oblika*, *vrsta*, *način*, *proces*, *uporaba*, *stopnja*), pa tudi zelo raznorodna poimenovanja tipa *življenje*, *človek*, *oseba*, *otrok*, *skupina*, *bog*, *telo*, *rastlina*, *odnos*, *razvoj*, *moč* in *potreba*. Med šestimi pridevniki izstopata *družben* in *socialen*, glagolov je samo pet (*uporabljati*, *uporabiti*, *morati*, *postati*, *gledati*).

5.3. Časopisi

V Tabeli 7 je prvih deset časopisov, ki v Gigafido prispevajo največ besed. Z izjemo *Celjana* so vsi v vrhu branosti (na lestvici NRB 2010 se razvrščajo od drugega do 24. mesta).

Naslov	Število besed
VSE	663.664.965
Dnevnik	181.336.239
Delo	149.252.977
Ekipa	46.154.899
Gorenjski glas	39.008.344
Večer	33.414.300
Dolenjski list	31.224.786
Nedeljski dnevnik	27.007.794
Finance	21.580.873
Kmečki glas	20.968.294
Celjan	10.835.696

Tabela 7: Časopisi z največ besedami.

Pri leksikalni analizi do 100. mesta v kategoriji časopisi močno prevladujejo samostalniki, glagola sta le dva (*prodati*, *dejati*). Okvirno lahko razpoznamo naslednje teme: gospodarstvo, finance, javna uprava (*direktor*, *uprava*, *vodstvo*, *podjetje*, *trg*, *tolar*, *evro*, *milijon*, *milijarda*, *delnica*, *banka*, *občina*, *mesto*, *Slovenija*), domača in tuja politika (*predsednik*, *srečanje*, *ministrstvo*, *minister*, *župan*) ter šport (*tekma*, *zmaga*, *trener*, *prvenstvo*, *liga*, *klub*, *sezona*, *pokal*, *turnir*). Značilni pridevniki so v tem delu: *slovenski*, *domač*, *evropski*, *svetoven*, *državen*, *občinski*, *mesten*; *kulturen*, *kmetijski*, *turističen*; *nov*, *mlad*; *zadnji*, *nekdanji*, *letošnji*, *letni* in *prihodnji*, ter samostalniki in prislovi s časovnim pomenom: *leto*, *ura*, *sobota*, *nedelja*, *teden*; *lani*, *včeraj*, *letos*.

5.4. Revije

Med revijami je največ besed v korpus prispevala *Mladina* (Tabela 8). Kot pri časopisih gre tudi tu za visoko brane naslove (na lestvici NRB 2010 se z izjemo *Maga* uvrščajo na mesta od osem do 69).

Naslov	Število besed
VSE	255.271.089
Mladina	33.870.249
Jana	13.458.466
Hopla	11.965.594
Monitor	10.246.819
Avto magazin	9.256.420
Nova	7.614.379
Življenje in tehnika	6.722.699
Viva	6.422.352
Moj mikro	5.823.997
Mag	5.501.255

Tabela 8: Revije z največ besedami.

Leksikalni profil je pokazal, da je v kategoriji revije na lestvici šest glagolov (*imeti*, *najti*, *uporabljati*, *omogočati*, *postati*, *potrebovati*), samostalniki pa na eni strani najbolj izrazito kažejo žensko/moško tematiko oz. splošnejše življenjske teme (*koža*, *telo*, *ženska*, *moški*, *prijatelj*, *model*, *motor*, *km*, *hitrost*, *avtomobil*, *moč*, *olje*, *oprema*; *življenje*, *volja*, *bolezen*) ter računalništvo (*računalnik*, *zaslon*, *arhiv*, *plošča*), po drugi strani pa značilno besedišče revijalnega tipa medija, kot je *slika*, *stran*, *revija*, *fotografija*, *foto*, *članek*. Med pridevniki, ki jih je sedem, izstopajo *spleten*, *moden* in *pravi*.

5.5. Drugo

V kategorijo drugo, ki obsega manj kot 8 milijonov besed, sta od znanih besedilodajalcev največ besed prispevala Državni zbor RS (zapisi sej) in RTV Slovenija (podnapisi, postprodukcijska besedila). Ker smo sem umestili tudi besedilni drobiž in podobna že za FidoPLUS zbrana neobjavljena besedila, je tu obsežen tudi vir "neznani založnik".

Založba	Število besed
VSE	7.951.450
Državni zbor RS	3.637.520
neznani založnik	1.639.653
RTV Slovenija	1.577.539

Tabela 9: "Založbe" z največ besedami v kategoriji drugo.

Leksikalna analiza je pokazala, da je kategorija drugo povsem pravno-upravna, značilni samostalniki v njej so: *člen*, *zakon*, *odstavek*, *republika*, *Slovenija*, *amandma*, *oseba*, *predlog*, *postopek*, *država*, *pravica*, *organ*, *poslanec*, *pogodbenica*, *vlada*, *sklep*, značilni pridevniki pa *državen*, *določen*, *praven*, in *pristojen*. Izstopajo še *hvala*, *gospod*, *beseda* in *lep*, ki prihajajo iz državnozborskega *"Gospod predsednik, hvala lepa za besedo"*. Glagolov je 13, med njimi: *dejati*, *imeti*, *uporabljati*, *reči*, *iti*, *misliti* in *morati*.

5.6. Internet

V tem delu Gigafide smo največ besed pri novičarskih portali dobili s *siol.net*, *24ur.com* in *rtvslo.si*, pri ustanovah in podjetjih pa s spletnih strani Državnega zbora RS, Vrhovnega sodišča RS in Informacijskega pooblaščenca RS.

Vir	Število besed
VSE	185.758.467
Novičarski portali:	
siol.net	36.103.293
24ur.com	34.963.385
rtvslo.si	27.294.954
Ustanove, podjetja:	
dz-rs.si	27.737.001
sodisce.si	5.776.609
ip-rs.si	3.735.755

Tabela 10: Vir z največ besedami v kategoriji internet.

Tudi za ta del korpusa so po leksikalnem profilu predvsem značilni samostalniki, povezani s pravom ali upravo, npr. *zakon*, *člen*, *sodišče*, *postopek*, *odstavek*, *vlada*, *pravica*, *organ*, *Slovenija*, *država*, *zadeva*, *stranka*, *republika*, *predsednik*, *pogodba*, ter sorodni pridevniki: *javen, praven*, *državen*, *ustaven*, *uraden*, *deloven*; *določen*, *naveden*. Po pregledu konkordanc lahko istemu tematskemu sklopu pridružimo še samostalnike, kot so

predlog, *zadeva*, *podatek*, *podlaga*, *oseba* ipd. Izstopajo *Pahor* ter zopet *gospod*, *hvala* in *lep*. Glagolov je tu 12, npr. *imeti*, *reči*, *vedeti*, *videti*, *iti*, *misliti* in *moči*.

Rezultati leksikalne analize po metodi frekvenčnega profila so bili deloma pričakovani, deloma pa presenetljivi. Ker smo predhodno dokaj dobro poznali besedila, zajeta v Gigafido, vključno z njihovimi taksonomskimi kategorijami, smo približno táko tematsko pokritost, o kakršni lahko okvirno sklepamo glede na dobljene značilne leme, pričakovali pri stvarnih besedilih, časopisih, revijah in v kategoriji drugo, presenetili pa sta nas pretežna ter pestra glagolskost leposlovja ter tolikšna pravno-upravnost internetnega dela. Ta se je v analizi izkazal kot zelo soroden kategoriji drugo, pri kateri slabo polovico besedil predstavljajo zapisi sej državnega zbora.

Po naknadnem pregledu spletnega dela Gigafide s tega zornega kota je v zvezi s tem mogoče dati naslednje pojasnilo: kot je bilo razvidno že zgoraj, smo z interneta poskusno in dokaj naključno zbrali predstavitvene strani podjetij ter državnih, izobraževalnih ipd. ustanov ter novičarske spletne strani. Obseg besed s strani podjetij je v okviru celotnega internetnega dela le 4-odstoten, dve tretjini preostalega dela pa prihajata z novičarskih strani, od tega največ besed prinašajo *siol.net* (19 %), *24ur.com* (19 %) in *rtvslo.si* (15 %). Očitno je torej to glavni vir besedišča tipa *zakon*, *člen*, *sodišče*, ki se je izkazalo kot značilno za to taksonomsko kategorijo (medtem ko smo sami – sicer povsem na pamet – pričakovali večjo podobnost s časopisi in revijami), pridružimo pa mu lahko vsaj še 25-odstotni delež spletnih besedil, ki prihaja s strani državnih ustanov (*dz-rs.si*, *sodisce.si*, *ip-rs.si* itd.). Primerjava frekvenc je tako izzvala nekaj premislekov, ki jih podajamo v zaključku.

6. Zaključki

Dostop do celotnih korpusov omogoča dvoje: bolj poglobljene jezikoslovne študije, ki jih ni mogoče izvesti s konkordančnikom, ter uporabo korpusov kot učnih in testnih podatkovnih množic za razvoj jezikovnih tehnologij. Obojega smo se zavedali, zato smo poleg prek konkordančnika prosto dostopnih korpusov sodobne pisne slovenščine Gigafide in KRES-a po postopku vzorčenja, ki smo ga opisali v prispevku, izdelali še dva korpusa – njuni prostodostopni različici v obsegu 100 milijonov besed (ccGigafida) oz. 10 milijonov besed (ccKRES).

Oba oz. vse štiri korpuse bo mogoče celovito ovrednotiti šele po različnih analizah. Eno od njih – izdelavo frekvenčnega profila vsake od taksonomskih kategorij ccGigafide – smo izvedli sami in ugotovili, da je večinoma dala, lahko bi rekli, neizstopajoče rezultate – z izjemo internetne kategorije. Pri tej smo bili v času zbiranja besedil predvsem osredotočeni na razvoj nove metodologije pajkanja, (avtomatskega zbiranja) vključno z odstranjevanjem spremnih in vnaprej pripravljenih besedil ter dvojnikov in približnih dvojnikov, manj pa na vsebinsko razpršenost v kombinaciji z dinamiko pajkanj ter obsegom pridobljenih besedil z določenih, čeprav zelo dinamičnih spletnih mest. Pri nadaljnjem pridobivanju besedil s spleta bomo temu vsekakor posvetili več pozornosti. Sicer pa v nadaljevanju raziskave med drugim načrtujemo tudi izdelavo frekvenčnega profila taksonomskih kategorij KRES-a, ki bo dala prvi uvid v njegovo »realno« uravnoteženost, ter enako primerjavo s katerim od referenčnih korpusov drugih jezikov, ki bi še dodatno osvetlila to, kar smo zgoraj pri frekvenčnem profilu taksonomskih kategorij ccGigafide glede na predhodno poznavanje v korpus vključenih besedil označili kot nepresenetljivo.

Zahvala

Avtorja se zahvaljujeta anonimnima recenzentoma za koristne pripombe in nasvete. Operacijo, v okviru katere je nastala raziskava, delno financira Evropska unija iz Evropskega socialnega sklada ter Ministrstvo za izobraževanje, znanost, kulturo in šport Republike Slovenije. Operacija se izvaja v okviru Operativnega programa razvoja človeških virov za obdobje 2007–2013, razvojne prioritete: razvoj človeških virov in vseživljenjskega učenja; prednostne usmeritve: izboljšanje kakovosti in učinkovitosti sistemov izobraževanja in usposabljanja 2007–2013.

Literatura

Arhar Holdt, Š., Gorjanc, V., 2007. Korpus FidaPLUS: nova generacija slovenskega referenčnega korpusa. *Jezik in slovstvo*, 52(2): 95–110.

Arhar Holdt, Š., Kosem, I., Logar Berginc, N., 2012. Izdelava korpusa Gigafida in njegovega spletnega vmesnika. V T. Erjavec, J. Žganec Gros (ur.), *Zbornik Osme konference Jezikovne tehnologije*. Ljubljana: Institut Jožef Stefan.

Erjavec, T., 2010. MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. *LREC 2010, 7th International Conference on Language Resources and Evaluations: Proceedings*: 2544–2547. Malta.

Erjavec, T., Krek, S., 2008. Oblikoskladenjske specifikacije in označeni korpusi JOS. V T. Erjavec, J. Žganec Gros (ur.), *Zbornik šeste konference Jezikovne tehnologije*: 49–53. Ljubljana: Institut Jožef Stefan.

Kazalnik 1: Standard za redno zbiranje pisnega gradiva za referenčni korpus, 2009. Dostopno prek: http://www.slovenscina.eu.

Krek, S., Grčar, M., Dobrovoljc, K., 2012. Označevalnik za slovenski jezik Obeliks. V T. Erjavec, J. Žganec Gros (ur.), *Zbornik Osme konference Jezikovne tehnologije*. Ljubljana: Institut Jožef Stefan.

Logar Berginc, N., Šuster, S., 2009. Gradnja novega korpusa slovenščine. *Jezik in slovstvo*, 54(3–4): 57–68.

Logar Berginc, N., Grčar, M., Brakus, M., Erjavec, T., Arhar Holdt, Š., Krek, S., 2012. *Korpusi slovenskega jezika* Gigafida, KRES, ccGigafida *in* ccKRES*: gradnja, vsebina, uporaba.* Ljubljana: Trojina, zavod za uporabno slovenistiko; Fakulteta za družbene vede.

Rayson, P., Garside, R., 2000. Comparing corpora using frequency profiling. *Proceedings of the ACL Workshop on Comparing Corpora*: 1–6. Hong Kong.

Steinberger, R., et al., 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*: 24–26. Genoa.

TEI Consortium, 2007. TEI P5: *Guidelines for Electronic Text Encoding and Interchange*. Dostopno prek: http://www.tei-c.org/Guidelines/P5.

Weaving sloWNet using window-based co-occurence features

Darja Fišer,* Maciej Piasecki, † Bartosz Broda†

* Department of Translation, Facult of Arts, University of Ljubljana
Aškerčeva 2, SI-1000 Ljubljana, Slovenia
darja.fiser@ff.uni-lj.si
† Department of Artificial Intelligence, Institute of Informatics, Wroclaw University of Technology
ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{bartosz.broda, maciej.piasecki}@pwr.wroc.pl

Abstract

This paper presents the first results of using statistical methods and linguistically annotated corpus data to extract lists of semantically similar words that are then incorporated into an existing wordnet for Slovene. The approach was originally developed for Polish but is attractive for other languages as well because, apart from a large corpus, it requires minimal NLP tools and resources, and can therefore be easily applied to any language that is still lacking an extensive wordnet or a similar semantic lexicon. Another important advantage of the adopted approach is that it relies on real linguistic evidence harvested from a corpus, yielding a linguistically sound organization of the vocabulary. As all the previous approaches used for the construction of Slovene wordnet were transfer-based and relied on the English Princeton WordNet, the encouraging results obtained in the presented experiment will be a welcome complement to the existing semantic network.

Spletanje sloWNeta na podlagi informacij o sopojavljanju besed v korpusu

V prispevku predstavljamo prve rezultate raziskave, v kateri smo z uporabo statističnih metod in jezikoslovno označenih korpusnih podatkov izluščili sezname semantično podobnih besed, ki smo jih nato vključili v wordnet za slovenščino. Pristop je bil prvotno razvit za poljščino, vendar je privlačen tudi za druge jezike, saj zanj razen obsežnega korpusa potrebujemo minimalna jezikovnotehnološka orodja in vire, zato ga je enostavno uporabiti tudi za jezike, za katere obsežen wordnet ali podoben semantični leksikon še ne obstaja. Druga pomembna prednost uporabljenega pristopa pa je, da temelji na izpričani jezikovni rabi, pridobljeni iz korpusa, ki se nato kaže v jezikovno utemeljeni organizaciji besedišča v izdelani semantični mreži. Glede na to, da so vsi naši dosedanji pristopi za izdelovo slovenskega wordneta celotno strukturo prevzeli iz Princeton WordNeta, ki je bil izdelan za angleščino, bodo spodbudni rezultati, dobljeni s pričujočo metodo, koristno dopolnjevali obstoječo semantično mrežo.

1. Introduction

sloWNet, a wordnet for Slovene, has been developed in a number of steps, taking advantage of several types of available bi- and multilingual language resources, such as bilingual dictionaries, parallel corpora and Wikipedia (Fišer and Sagot, 2008). All these approaches have in common that they take over the structure of Princeton WordNet (Fellbaum, 1998), the oldest and most extensive existing wordnet that was developed for English, and find Slovene equivalents for the same set of concepts.

The work presented in this paper tackles the problem from a completely different angle as it extracts all the relevant lexico-semantic information from a single resource, the largest Slovene reference corpus Gigafida (Logar Berginc and Šuster, 2009), yielding language-motivated lists of semantically related words and a linguistically sound organization of the vocabulary. The aim of this paper is to adapt the wordnet expansion algorithms, originally developed for Polish, to Slovene in order to test whether they work for another language as well. With the analysis of the first results we also wish to outline further refinements and enhancements of the approach for future work on fully automated methods of wordnet expansion for Slovene.

This paper is structured as follows: in the next section we present related work. Then, we focus on the resources and tools that were used in the experiment. In Section 4 we give an overview of the experimental setup, evaluate and discuss the results. We then conclude the paper with some final remarks and ideas for future work.

2. Related work

The task of extending a wordnet with additional literals or synsets consists of two parts: first, word pairs of sufficient semantic relatedness need to be extracted from a large corpus, and then they need to be attached to the most appropriate place in the existing semantic network.

Automatic methods for the extraction of semantically related words from corpora fall into two main frameworks: pattern-based (Hearst, 1992) and those that follow the Distributional Hypothesis (Harris, 1968). The pattern-based approaches rely on a list of lexico-syntactic patterns in which two lexical units frequently occur in an identifiable lexical semantic relation, e.g., hypernymy (Pantel and Pennacchiotti, 2006). On the other hand, the distributional- based approaches assume that the similarity of distributions of some lexical units across different lexico-syntactic contexts is evidence of their close semantic relation. The stronger the similarity, the closer the meanings of the lexical units are. Unlike pattern-based approaches, which are limited only to the words that co-occur in a particular pattern, distributional-based techniques can be used for any word pair. Because high recall is an important desideratum in the work presented in this paper, we have opted for the latter.

Many measures of semantic relatedness have been proposed (cf. Ruge, 1992; Lin and Pantel, 2002; Weeds and Weir, 2005). They all share the starting point, which is the construction of a coincidence matrix of co-occurrences of lexical units (rows) and their lexico-syntactic contexts (columns) from a large corpus.

Their main differences between them are the following:

(1) how contexts are defined,
(2) how raw frequencies are normalized, and
(3) how the final value of the measure is calculated.

We have experimented with several different settings reported in literature in our previous work (cf. Piasecki and Broda, 2007; Broda and Piasecki, 2008), and are using the best-performing settings in this work (see Section 3.3).

Once lists of highly semantically related words have been generated, they need to be attached to the most appropriate positions in the existing semantic network. Most known taxonomy induction methods utilize only the existing hypernymy structure in incremental wordnet expansion. Several machine-learning methods have been used to induce a taxonomy from hypernym-hyponym pairs, such as decision trees (Witschel, 2005) or k-nearest neighbors (Widdows, 2003) for a limited set of domains of concrete and frequent nouns. In their seminal paper, Snow et al. (2006) propose a probabilistic wordnet-expansion method based on a probabilistic model of the taxonomy which reports promising results that however were not reproduced successfully in a reimplementation of their algorithm (see Piasecki et al., 2012a).

The approach used in this paper goes beyond the related work in three respects. First, in our previous work (Piasecki et al., 2012a), the wordnet hypernymy structure is perceived as intrinsically interlinked to other wordnet relations. Thus, we aim at utilizing all different types of links in the expansion of Slovene wordnet as well. Second, the algorithm is based on the assumption that the relation extraction method produces some noise in the results, so we cannot identify the exact place (synset) for a new lemma as such but an area (a wordnet subgraph). And last, contrary to a rich body of the related work, we do not assume any shape of the lexical semantic network, but we try to build it in a way that faithfully reflects the language data.

3. Resources and tools used

3.1. Gigafida

The Gigafida corpus is a 1.15 billion word reference corpus of Slovene and is as such currently the largest and most extensive text collection of Slovene (Arhar Holdt et al., 2012). It has been developed within the national project Communication in Slovene (2007-2013) and contains texts of various types and genres such as literary texts, newspaper articles and Internet contents that were published between 1995 and 2011. The corpus has been split into paragraphs and sentences, tokenized, part-of-speech tagged and lemmatized, so that is readily available for use via a concordancer as well as for NLP applications.

3.2. sloWNet

sloWNet is a concept-based semantic lexicon in which nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (synsets) which are then organized into a hierarchical network with lexical and semantic relations, such as hyper- and hyponymy, antonymy, meronymy etc. The concepts that synsets represent are defined with a short gloss and usage examples while most synsets also have a domain label and a mapping to the SUMO/MILO ontology.

sloWNet is based on a Princeton WordNet that was originally developed for the English language (Fellbaum, 1998). Slovene equivalents for synsets were obtained automatically by leveraging existing bi- and multilingual resources, such as a bilingual dictionary, a multilingual parallel corpus and Wikipedia (see Fišer and Sagot, 2008). Recently, a large-scale extension of sloWNet has been achieved by training a maximum entropy classifier in order to determine appropriate senses of translation candidates extracted from heterogeneous bilingual resources (see Sagot and Fišer, 2012a). In addition, automatic detection of candidate outliers has been performed within the framework of distributional semantics by comparing the immediate neighborhood of literals in sloWNet and their contexts in a reference corpus (see Sagot and Fišer, 2012b) with the goal of eliminating noise from the automatically generated resource.

The most recent version of sloWNet has 82,721 literals, which are organized into 42,919 synsets. Apart from single words sloWNet contains many multi-word expressions and proper names as well. Nouns are still by far the most frequent, representing more than 70% of all synsets. While 66% of all the literals in sloWNet are monosemous, their average polysemy level is 2.07.

The methodology of sloWNet construction has three important implications that we try to address in this work:

(1) The resource is based on a semantic network originally produced for a foreign language, so it might be biased towards the organization and distinction of senses typical of English and therefore inadequately reflects the semantic inventory of Slovene.

(2) Slovene equivalents for synsets were harvested from several already available language resources of limited coverage, which is why we were able to obtain equivalents only for some synsets while the rest are still empty, leaving gaps in the network.

(3) Due to automatic generation of synsets, word-sense disambiguation was not perfect, resulting in noisy synsets that have a negative impact on applications using sloWNet, and should therefore be eliminated as far as possible in the shortest possible time.

3.3. The SuperMatrix system for Distributional Semantics

SuperMatrix is a system for semantics analysis of text, especially aimed at supporting automatic acquisition of lexical semantic relations from large corpora (Broda and Piasecki, 2008). The main functionality of *SuperMatrix* is related to the automated construction of Measures of Semantic Relatedness (MSRs) on the basis of a corpus, and testing them on the basis of a wordnet. An MSR is a function that takes a pair of words and returns a value, which describes how closely semantically related the two words are. MSR construction follows a typical blueprint: corpus preprocessing, co-occurrence matrix construction, matrix filtering and transformation, and row similarity computation.

Corpus preprocessing depends on the available language tools. However, for morphologically rich languages lemmatization is a minimal requirement for obtaining a useful MSR. In addition, a corpus parsed by a shallow parser or a dependency parser is a good basis for the construction of a highly accurate MSR, i.e., an MSR which assigns higher values for pairs of lemmas linked by one of the lexico-semantic relations, e.g. synonymy, hyper-/hyponymy, holo-/meronymy and other relations described in wordnets.

Data collected from a corpus contain a lot of statistical noise, e.g. very low frequencies, accidental co-occurrences due to errors produced by language tools, thus the stored data must be filtered and transformed before they can be used for similarity calculations. In addition, raw frequencies produce skewed results, which is why several weighting algorithms have been implemented in *SuperMatrix*. Our previous experiments show that the Point-wise Mutual Information (PMI) measure (Lin and Pantel, 2002) produces the best results. *SuperMatrix* can also reduce dimensions of a matrix using, for example, singular value decomposition. Finally, a vector similarity measure is applied to the matrix in order to obtain a ranked list of similar lemmas. *SuperMatrix* offers most well-known similarity measures but it has been shown that the simple cosine measure produces best results in most cases.

The system also supports an automated evaluation of the selected MSR using synonymy tests that are automatically generated from wordnet, called *Wordnet-Based Synonymy Test* (WBST). The test is described in details in (Piasecki et al., 2009) but the procedure is quite straightforward. Each test item consists of a question word selected from the wordnet data, its synonym (the correct answer) taken from the same synset (or its direct hypernym in the case of singleton synsets including only the question word) and *k* distractors (words taken form other synsets). The task is to select the most related word to the question word among the presented candidates using only the MSR value. For example, for word *svet (council)* one has to choose between *gomolj* (tuber), *izvirnost* (originality), *odbor* (committee) – the correct answer – and *odobravanje* (approval).

3.4. WordnetWeaver

WordnetWeaver is a tool that extends the wordnet editing system called *WordnetLoom* (Piasecki et al., 2012b) with an automated wordnet expansion facility. It utilizes the results of the Activation Area Attachment Algorithm (AAAA) that generates suggested attachment positions for new lemmas, not yet present in wordnet. A suggested attachment is a synset to which a new lexical unit for the given new lemma can be added as a synonym or linked via a lexical or semantic relation, such as hypo- or hypernymy. Moreover, as all automated methods for the extraction of the lexico-semantic relations produce some errors, attachment points in *WordnetWeaver* are presented in the context of attachment areas – subgraphs of the wordnet hypernymy graph such that each synset of the selected subgraph express a strong enough semantic relation to the new lemma according to AAAA.

WordnetWeaver then presents top-scored suggestion in a visual, graph-based editor and enables their verification, correction as well as free manual editing of the wordnet structure. Contrary to other automated wordnet construction methods mentioned in Section 2, the aim of AAAA is to generate suggestions for lexicographers, who make the final wordnet expansion decisions, not to expand the wordnet fully automatically. Thus, AAAA is intentionally set up for slight sense over-generation in order to increase the coverage. The refinement of AAAA that would allow fully automated wordnet expansion is still an open research question.

The input to AAAA are sets of triples: *<l1, l2, w>,* where *l1* s a new lemma and *l2* a lemma already in wordnet. They are linked with a lexico-semantic relation according to a corpus-based relation extraction method, and *w* is the weight assigned to the pair by the given method. We refer to such a set of triples as a knowledge source (KS). AAAA does not assume a probabilistic interpretation of the weights and can work with any set of knowledge sources of any types, produced by any method. Each KS can also have an assigned weight, e.g. expressing the KS accuracy obtained from manual inspection of a sample.

Taking triples from the desired KSs, the AAAA algorithm is composed of two steps. First, the *semantic fit* between the input lemma *l1* and each synset *X* in a wordnet is calculated on the basis of the KSs and the neighborhood of *X*. And then, connected subgraphs *(activation areas)* of the lexical-semantic network are identified, (for details see Piasecki et al., 2012a, Broda et al., 2011).

AAAA has so far been successfully applied to the development of the Polish wordnet (plWordNet) on a practical scale (Piasecki et al., 2009). Also, an automated evaluation of the AAAA performance on Princeton WordNet (Fellbaum, 1998) has been performed (Broda et al., 2011).

4. Experimental setup

The application of the AAAA algorithm to a new language is limited only by the available language resources and corpus processing tools. The minimum requirements are: a large enough corpus and a means for constructing an MSR from it. For morphologically rich languages, Part-of-Speech tagging and lemmatization is also very useful.

In this initial experiment on Slovene wordnet extension with *WordnetWeaver,* we have limited our work to the most frequent single-word nouns, i.e. nouns that occurred at least 1,000 times in the Gigafida corpus. There were 36,026 such nouns, 8,981 of which are already in sloWNet. This was a pragmatic decision in order to examine the first results as quickly as possible and make any necessary changes for future large-scale experiments. But the selected setting is not a limiting factor of the algorithm as such as most of the methods developed for Polish were aimed at low- frequency data (see Piasecki et al., 2009). On the other hand, the results for very frequent words should be better due to the statistical nature of applied methods.

The corpus was PoS-tagged and lemmatized already. It was then converted to a simple plain-text format. In addition, sloWNet had to be converted to the plWordNet XML format for use in *WordnetWeaver*. Apart from that, no other changes were required, which is a great advantage of the tools that were initially developed for Polish because this means that they can be used with other resources and for other languages with relatively little effort.

4.1. Extracting semantically related words

The measure of semantic relatedness is the most fundamental knowledge source for AAAA as it has good coverage (i.e. it provides similarity values for every pair of lemmas that are frequent enough in the corpus), and facilitates the discovery of lexical-semantic relations between words. In comparison to a KS that contains pairs of semantically related lemmas extracted with manually constructed patterns, which has a much higher precision than MSR, the coverage of the pattern-based KS is much lower as only a limited number of pairs can be found in the corpus.

As work on dependency parsers for Slovene is still on-going and we wanted to avoid additional manual work required for pattern-based approaches in this preliminary work, the MSR was constructed with a simple window-based approach. That is, target lemmas are described by all the other content lemmas (nouns, adjectives, verbs, adverbs) co-occurring in a small text window (3 lemmas before and after the target lemma), stopping at paragraph boundaries.

Since there is no *a piori* best method for MSR development and several are implemented in SuperMatrix, we selected the best-performing one with WBSTs based on the existing part of sloWNet. We generated questions with three detractors and a correct answer. On the 20,308 generated questions we achieved the best results for PMI weighting extended with the discounting factor and cosine similarity function (Lin and Pantel, 2002). MSR chose the correct answer in 72.37% of all the questions in WBST.

4.2. Attaching the words to sloWNet

The most straightforward adaptation of AAAA to sloWNet requires importing sloWNet to the *WordnetWeaver* scheme and a preparation of knowledge sources.

We have prepared two KSs based on MSR. The first one is based on the similarity lists for lemmas. That is, for each lemma l_x we compute 20 most similar lemmas l_y using the above described MSR. This KS then takes the form of pairs $<l_x, l_y, msr(x,y)>$, where $msr(x,y)$ is a value of MSR between the two lemmas.

The other KS uses *bi-directional* similarity lists. It is a subset of the above knowledge source with additional filtering. For l_x the pair $<l_x, l_y, msr(x,y)>$ is included only if there is also a pair $<l_y, l_x, msr(y,x)>$ among the 20 most similar items for l_y.

4.3. Evaluation of the results

WordnetWeaver and AAAA were designed to help a linguist in expanding an existing wordnet structure with new lemmas. Thus, the evaluation of the algorithm's performance should focus on this practical aspect. In order to gain a comprehensive insight into the performance of the adopted approach, we perform the results both automatically and manually.

4.3.1. Automatic evaluation

For automatic evaluation of the results, we follow the evaluation methodology proposed by (Broda et al., 2011). The idea of the evaluation is simple: first, we remove some literals from the existing sloWNet structure; then we run AAAA for those literals and see how close to the original place in sloWNet (along hyper-/hyponymy paths) the removed literals were re- attached by the AAAA. Ideally, we would like to remove all occurrences of one lemma in sloWNet at a time and then reattach it, in order to alter sloWNet structure as little as possible, but this is computationally very expensive. Thus, we remove a package of 50 lemmas at a time. For evaluation purposes, we randomly selected a sample of the 1,000 nouns meeting the frequency threshold that was also set to 1,000 (see Section 3).

Several evaluation strategies are possible, each giving a different perspective on the algorithm performance (Broda et al., 2011). From the lexicographers' point of view, the algorithm performs well if there is at least one correct suggestion that is relatively close to the proper place in a wordnet structure, i.e., the *closest path* strategy. For a given lemma, this method only checks the attachment of the closest path to one of the lemma's original position in the wordnet. On the other hand, the *strongest* supported strategy evaluates only the highest-ranked suggestion provided by the wordnet expansion algorithm. The last strategy we use evaluates *all* the propositions returned by the algorithm.

Table 1 presents the results of the described evaluation methodology for all three strategies. The *acceptable distance* to the original place was set to 6 by the lexicographers during the construction of plWordNet (Piasecki et al., 2009). The distance is measured on the hypo-/hypernymy and mero-/holonymy graphs with the exception that we can only traverse one edge of mero-/holonymy (as this relation can take us to completely unrelated parts of the wordnet very quickly).

Dist.	Closest[%]	Best[%]	All[%]
0	15.0	5.9	3.7
1	19.7	13.9	4.6
2	19.0	13.9	6.0
3	11.7	8.2	4.9
4	8.1	9.0	5.3
5	5.5	6.4	6.8
6	0.2	0.7	0.8
Σ	79.2	57.9	32.2

Table 1: Results of the automatic evaluation procedure for sloWNet expansion.

The achieved results are significantly lower than for Polish (Broda et al., 2011), which was expected as we have employed much simpler and less precise, window-based MSR, and we did not used additional, pattern-based KSs. On the other hand, the results are encouraging as for almost 80% of the words the algorithm suggested at least one correct place for attachment. Also, the correct attachment places are mostly close to the original place in

the wordnet structure (i.e., the results are shifted towards closer distances than 6). AAAA provided a suggestion for 94% of words from the random sample and found 29.6% of word senses for each word on average.

4.3.2. Manual evaluation

For a more qualitative insight into the results, we also performed a manual evaluation on 100 random lemmas included in the automatic evaluation. In manual evaluation, 5 highest-ranking attachment suggestions were checked for each lemma, amounting to 500 candidate-attachment pairs.

The evaluated lemmas were first categorized into monosemous or polysemous. Based on the attachment suggestions for polysemous lemmas, we checked whether our algorithm was able to detect only one of its senses or more. Next, we tried to label each attachment suggestion with one of the 10 lexico-semantic relations: *synonymy, hypernymy, hyponymy, holonymy, meronymy, co-hyponymy, co-meronymy, antonymy, close, vague*, or *no relation*. The *no relation* label is intended for clear errors of the algorithm. The *close* label is used for cases where the candidate-attachment pair is clearly semantically related but the relation type is not found in the current version of sloWNet (e.g. *Occupation-Place* such as *pošta-poštar* [post-postman], *Activity-Occupation* such as *učenje-učitelj* [teaching-teacher]). The *vague* label, on the other hand, is used for cases where the candidate-attachment pair is in a more loose associative relation that will probably not be encoded in wordnet (e.g. same semantic field such as *politika-debata* [politics-debate]).

Overall, the results of manual evaluation are very encouraging as no cases were found where all the attachment suggestions for a lemma would be completely unrelated. What is more, only 1 out of 100 lemma received no better attachment suggestion than a vague association, and an additional 1 got at best a closely related one. On the other hand, as many as 38 lemmas had no erroneous attachment suggestions, which means that the lexicographers who are responsible for selecting the best attachment candidates will be presented with very little noise that would slow down their work.

Category	Freq.	%
synonym	22	4.40%
hypernym	74	14.80%
hyponym	9	1.80%
holonym	9	1.80%
meronym	12	2.40%
antonym	1	0.20%
co-hyponym	40	8.00%
co-meronym	2	0.40%
closely related	171	34.20%
vaguely related	50	10.00%
unrelated	110	22.00%
total	500	100.00%

Table 2: Frequency counts of association candidates per relation type.

As Table 2 shows, almost 34% of the suggested association candidates can easily be labeled with one of the standard lexico-semantic relation types from wordnet. By far the most frequent one is the hypernymy relation that was selected in almost 15% of the cases. There were quite a lot of co-hyponymy (8%) and synonymy (4%) attachments as well while the rest of the relations were much more rare. A further 34% of the suggestions were very closely related to the lemmas, 10% were loosely associated to them while 22% of the association candidates were not related at all to the lemmas they were assigned to.

When analyzing the semantic nature of the randomly selected lemmas in the evaluation sample, we observe that 62% of them are monosemous and 38% polysemous. This is very similar to the polysemy level of nouns in the latest version of sloWNet, where 66% of the literals are monosemous. A single sense prevailed for 58% of the otherwise polysemous lemmas in the evaluation sample, while association candidates refer to different senses in 42% of the cases. This is a well-known phenomenon of distributional semantics where a Zipfian distribution of senses in the corpus causes skewed context vectors of polysemous words, which are thus heavily biased towards the most frequent sense in the corpus.

Table 3 shows frequency counts of semantic categories that appeared at least once among the association suggestions per lemma. Because we counted all the relation types that were suggested for each lemma, and a single lemma could have suggestions belonging to a single category or up to five different categories, the total count is more than 100. Hypernymy and co-hyponymy are still the most frequent in this setting, suggested for 60% and 28% of the lemmas, respectively. Both are more frequently suggested for monosemous nouns, while polysemous ones have more suggestions for synonyms, hyponyms, holonyms, meronyms and co-meronyms. Polysemous nouns contain a slightly higher number of erroneous attachment candidates and a much higher number of vaguely and closely related suggestions than polysemous ones. Interestingly, the polysemous nouns for which only one sense was detected by the algorithm, contain the least noise and vague association candidates.

Cat.	Mono.	Poly.			Σ
		1 sense detected	>1 sense detected	Σ poly	
	62	22	16	38	100
syn	11	5	3	8	19
hyper	40	13	7	20	60
hypo	3	1	2	3	6
holo	4	2	3	5	9
mero	4	5	2	7	11
anto	1	0	0	0	1
co-hypo	19	6	3	9	28
co-mero	1	1	0	1	2
close	51	16	2	18	69
vague	22	4	5	9	31
error	37	11	10	21	58

Table 3: Frequency counts of lemmas with at least 1 association suggestion per category.

5. Conclusions

In this paper we presented the first results of applying *WordnetWeaver* to Slovene data in order to extend Slovene wordnet. The approach uses statistical methods to extract lists of semantically similar words from a large reference corpus of Slovene, and then identifies the part of the wordnet hierarchy these words should be attached to. Automatic and manual evaluations of the results show that the algorithm was successfully ported to a new language and is already useful in its most basic setting. However, the state-of-the-art results for Polish suggest that further improvements of measures of semantic relatedness are still possible, for example by using a constraint-based approach, a dependency parser, and testing more measures with more parameters. Similarly, the attachment algorithm could further be improved by optimizing parameters of the algorithms, for example by using meta-heuristics like in (Kłyk et al., 2012), and providing additional knowledge sources, such as pattern-based lists of semantically related word pairs.

In the future, we wish to investigate methods that would enable us to extend the current functionality of the attachment algorithm to expand sloWNet fully automatically, requiring no human intervention for reaching the final decision where to add a new word in wordnet. A somewhat different but very interesting area of research would be to adapt the attachment algorithm to be able to use corpus data in order to analyze the semantic network in sloWNet that is based on Princeton WordNet and find suspicious areas in the network that does not correspond to the linguistic evidence harvested from the corpus and should therefore be improved.

Acknowledgments

We would like to thank Tomaž Erjavec for performing conversions of sloWNet into the plWordNet XML format and of the Gigafida corpus into plain text.

6. References

Š. Arhar Holdt, I. Kosem and N. Logar Berginc. 2012. Izdelava korpusa Gigafida in njegovega spletnega vmesnika. In *Proceedings of 8th Eighth Language Technologies Conference IS-LTC-12*. Ljubljana, Slovenia.

Z. S. Harris. 1968. *Mathematical Structures of Language*. Interscience Publishers, New York.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceeedings of COLING-92*. Nantes, France, pp. 539–545.

Ł. Kłyk, P. B. Myszkowski, B. Broda, M. Piasecki and D. Urbansky. 2012. Metaheuristics for Tuning Model Parameters in Two Natural Language Processing Applications. In *Proceedings of the 15th AIMSA conference*, Varna, Bulgaria.

C. Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automat- ically harvesting semantic relations. In *Proceedings of COLING-ACL-06*, Sydney, Australia, pp. 113–120.

G. Ruge. 1992. Experiments on linguistically-based term associations. *Information Processing and Management* 28(3), pp. 317–332.

Lin, D. and Pantel, P. 2002. Concept discovery from text. In Proceedings of *COLING-02*, Taipei, Taiwan, pp. 577–583.

M. Piasecki and B. Broda. 2007. Semantic similarity measure of Polish nouns based on linguistic features. *Business Information Systems 10th International Conference, Volume 4439 of Lecture Notes in Computer Science*, Springer.

B. Broda and M. Piasecki. 2008. SuperMatrix: a General Tool for Lexical Semantic Knowledge Acquisition. *Speech and Language Technology conference, Volume 11 of Lecture Notes in Computer Science*, Springer, pp. 239–254.

M. Piasecki, S. Szpakowicz and B. Broda. 2009. *A Wordnet from the Ground Up*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław.

R. Snow, D. Jurafsky and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of ACL-06*, pp. 801–808.

M. Piasecki, R. Kurc, R. Ramocki and B. Broda. Lexical Activation Area Attachment Algorithm for Wordnet Expansion. 2012a. In *Proceedings of the 15th AIMSA conference*, Varna, Bulgaria.

M. Piasecki, M. Marcińczuk, R. Ramocki, M. Maziarz. 2012b. WordnetLoom: a Wordnet Development System Integrating Form-based and Graph-based Perspectives. *International Journal on Data Mining, Modelling and Management*.

B. Broda, R. Kurc, M. Piasecki, R. Ramocki. 2011. Evaluation Method for Automated Wordnet Expansion. In *Security and Intelligent Information Systems*. Springer, 2011.

D. Widdows. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of NACL-03*.

H. F. Witschel. 2005. Using decision trees and text mining techniques for extending taxonomies. In *Proceedings of Learning and Extending Lexical Ontologies by using Machine Learning Methods, Workshop at ICML-05*.

J. Weeds, D. Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics* 31(4), pp. 439–475.

Speech Act Based Classification of Email Messages in Croatian Language

Tin Franović, Jan Šnajder

University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
{tin.franovic, jan.snajder}@fer.hr

Abstract

Speech acts provide an effective way of summarizing the intended purpose of an email message. In this paper we address the task of speech act classification of email messages in Croatian language. We frame the task as a multilabel text classification problem. We perform thorough evaluation using six machine learning algorithms on message-level, paragraph-level, and sentence-level features. Using message-level features, we achieved an overall best F1 score of over 94%.

Razvrščanje sporočil elektronske pošte v hrvaškem jeziku na podlagi govornih dejanj

Govorna dejanja predstavljajo učinkovit način za povzemanje namena sporočila elektronske pošte. V članku obravnavamo nalogo razvrščanja sporočil elektronske pošte v hrvaškem jeziku na podlagi govornih dejanj. Nalogo opredelimo kot problem razvrščanja besedila na podlagi oznak. Izvedemo poglobljeno evalvacijo z uporabo šestih algoritmov strojnega učenja in več nabori značilk na različnih ravneh – na ravni sporočila, na ravni odstavka ter na ravni stavka. Z uporabo značilk na ravni sporočila smo dosegli najboljši F1 izid preko 94%.

1. Introduction

The increase in popularity of email as means of business and personal communication is reflected in the amount of messages users are required to deal with on a daily basis. Recent surveys indicate that most people who use email for business purposes spend up to two hours a day reading, writing, and sorting email messages. This clearly indicates that there is a need for automated classification of email messages, which would drastically reduce the amount of time users spend on reading and sorting them. Classification of incoming email messages provides the user with information about the predicted importance or content of the messages before the user even reads them. This allows the user to focus on the messages considered important or interesting. Email classification has first become popular through *spam* filtering, which removes from the inbox the messages classified as unsolicited and places them in a special folder. Another solution is the filtering of messages classified as potentially important into a special folder called the *priority inbox*. Both techniques have successfully been implemented in widespread email clients.

The two aforementioned methods filter the messages based on their predicted importance. While importance-based filtering is convenient for most users, it is often difficult to predict what users will find important in a particular situation or context. The alternative to importance-based filtering is content-based classification, which labels each message based on its content, leaving it to the user to decide on the importance of the message.

This paper focuses on using speech acts expressed in email messages in the Croatian language for the purpose of content-based email classification. Speech acts are illocutionary acts that attempt to convey meaning from the speaker (or writer) to the listener (or reader) (Searle, 1965). In the context of email classification, speech acts provide an effective way of summarizing the intended purpose of the message. By labeling the email messages with speech acts which they contain, we enable the user to decide on which messages to focus first while reading. In this paper, we frame the speech act classification problem as a multilabel text classification problem and address it using supervised machine learning. We perform thorough evaluation experiments using six machine learning algorithms and three types of features extracted at three discourse levels (message, paragraph, and sentence level). We evaluate our speech act classifiers on a manually annotated collection of email messages in the Croatian language.

The rest of the paper is structured as follows. In the next section we give a brief overview of previous work on speech act classification. In Section 3 we describe our approach to speech act classification of email messages in Croatian. In Section 4 we evaluate the classifiers and discuss the results. Section 5 concludes the paper.

2. Related Work

The study of speech act classification (or *dialogue act classification*, as it is sometimes referred to) is one of the interesting challenges in natural language processing (NLP). From the NLP perspective, speech act classification is interesting especially for dialogue-based human-computer interaction. Successful dialogue systems are capable of understanding the speaker's intention and the message the speaker wishes to convey. Interpreting the speaker's intention is usually accomplished by analyzing and classifying speech acts. The *Clarity* project (Finke et al., 1998) is one of the first works in which speech acts are used in an attempt to understand dialogues. The focus of the project was to infer three levels of discourse structure in Spanish telephone conversations: speech acts, dialogue games, and discourse segments. The AutoTutor system (Marineau et al.,

2000) is an English computer tutor sensitive to speech acts from the previous dialogue turn, allowing the tutor to select the next action according to the speaker's intent. Keizer (2001) designed a conversational agent for the Dutch language that probabilistically interprets dialogue acts. Serafin et al. (2003) employ Latent Semantic Analysis (LSA) to classify speech acts from a corpus of tutoring dialogues in Spanish. Louwerse and Crossley (2006) use n-gram algorithms to classify speech acts from English dialogues on a location map reconstruction topic.

Relevant to the work presented in this paper is the use of speech acts for content-based email classification. Cohen et al. (2004) presented a system for classication of email messages in English based on supervised machine learning and a custom taxonomy of speech acts. In their subsequent work, Carvalho and Cohen (2006) exploit the linguistic aspects of the content-based classification problem by combining message preprocessing and n-gram feature extraction in order to improve the classification.

3. Speech Act Based Message Classification

3.1. Dataset annotation

There are several email datasets publicly available on the Internet, such as the *Enron* dataset (Klimt and Yang, 2004). However, none of these sets is in Croatian language. We therefore decided to first build a suitable dataset. An email dataset can essentially be obtained in two ways: by simulating a communication process (i.e., a business project communication), where different people take on different roles, as done by Cohen et al. (2004), or by finding a group of volunteers willing to provide their email messages sent over a period of time. We used the latter approach, mainly because volunteers were readily available and because the former method would take more time and resources. The total number of messages, collected from five sources, is 1337. Four sources contain personal emails provided by volunteers, while the fifth consists of messages exchanged during the course of a small student project.

For annotation, we used a set of 13 different speech acts, which can be divided into five groups according to Searle's classification (Searle, 1965):

- Assertives (AMEND, PREDICT, CONCLUDE);
- Directives (REQUEST, REMIND, SUGGEST);
- Expressives (APOLOGIZE, GREET, THANK);
- Commisives (COMMIT, REFUSE, WARN);
- Declarations (DELIVER).

The message annotation was split between two annotators, each annotating approximately one half of the dataset. The annotators were asked to annotate in each email portions of text that contain a speech act. The size of these portions may vary from a few words to larger portions spanning over several sentences. As a general rule, one speech act annotation could not span over multiple paragraphs. The total number of messages is 1337, the number of paragraphs is 4468 paragraphs and number of words 76,760. The number of annotated speech acts in the dataset is 4498, and for different speech acts the number of annotations is between 14 for the REFUSE speech act and 1069 for the GREET speech act. On average, a speech act annotation contains 17.06 words, with CONCLUDE being the longest on average (32.1 words), while GREET being the shortest (5.99 words per annotation).

Table 1: κ-statistic for all speech acts

Speech act	κ	Speech act	κ
AMEND	0.714	REFUSE	0.000
APOLOGIZE	0.856	REMIND	0.747
COMMIT	0.851	REQUEST	0.589
CONCLUDE	0.005	SUGGEST	0.544
DELIVER	0.792	THANK	0.949
GREET	0.779	WARN	0.174
PREDICT	0.267		

Table 2: Classifier performance on speech acts (% F1)

	NB	k-NN	SVM	DS	AB	RDR
DELIVER	69.70	83.72	88.16	85.71	87.50	**88.51**
AMEND	**79.31**	71.43	77.97	72.29	74.63	77.27
COMMIT	62.45	67.44	78.61	79.37	81.97	**83.75**
REMIND	60.87	63.64	75.00	76.92	**94.74**	76.92
SUGGEST	67.06	70.27	**76.84**	76.27	75.12	71.50
REQUEST	69.69	75.44	**78.76**	70.57	75.23	74.46

The two annotators double-annotated 15% of the dataset, on which we evaluated the inter-annotator agreement. The κ statistic (Carletta, 1996), computed separately for each speech act, is shown in Table 1. On some speech acts (REFUSE, CONCLUDE, WARN) the agreement was considerably low, thus we decided to exclude these speech act from further consideration. After removing the infrequent speech acts and speech acts with low inter-annotator agreement, we ended up with six speech acts: DELIVER, AMEND, COMMIT, REMIND, SUGGEST, and REQUEST. The removed speech acts are: APOLOGIZE, CONCLUDE, GREET, PREDICT, REFUSE, THANK, and WARN.

3.2. Message preprocessing

Message preprocessing consisted of stop-word removal, stemming, and the extraction of training examples. We created a separate training set for every speech act. Using the information provided by each annotation (original message, start and end point of annotation), we extracted text segments corresponding to the sentence, paragraph, and message levels. At the message level, we use the whole original message text. At the paragraph and sentence levels, we extract the text segments that enclose the start and end points of the annotation. If an annotation spans over multiple sentences, all of the sentences are included. Negative examples for every speech act are sampled from the set of text segments not annotated with the corresponding speech act. The number of negative examples was chosen to be approximately the same as the number of positive examples.

In order to reduce the dimensionality of the input space and eliminate the morphological variation, we applied a

Table 3: Classifier performance on discourse levels (% F1)

	Message	Paragraph	Sentence
DELIVER	86.59	83.64	**88.51**
AMEND	**79.31**	77.27	72.38
COMMIT	**83.75**	81.97	78.93
REMIND	**94.74**	76.92	69.57
SUGGEST	71.88	**76.84**	69.74
REQUEST	70.09	**78.76**	72.19
Overall	94.74	83.64	78.93

Table 4: Overall classifier performance (% F1)

	Message	Paragraph	Sentence
NB	**79.31**	69.70	72.38
k-NN	72.73	75.44	**83.72**
SVM	83.87	81.55	**88.16**
DS	78.65	79.37	**85.71**
AB	**94.74**	83.54	87.50
RDR	86.59	83.64	**88.51**

simple stemming procedure: we removed the the suffix of each word after the last vowel (including the vowel itself) if the length of the suffix is less than half the length of the word. Stemming reduced the number of terms from 15,100 to 11,856. Apart from stemming, we optionally employ stop-word filtering. Stop-words are common function words that, in the context of content-based text classification, are usually filtered out because they carry little semantic information. We used a list of 2024 Croatian stop words.

3.3. Training classifiers

For the classification experiment we use Rapid Miner, an open-source data mining environment that simplifies the training process and provides a variety of classifiers to choose from. We experiment with six different models: SVMs (Support Vector Machines), naive Bayes (NB), k-NN (k-Nearest Neighbors), Decision Stump (DS), AdaBoost (with Decision Stump as the weaker learner), and RDR (Ripple Down Rule). For all models, apart from RDR, we experiment with two term weighting schemes: TF (Term Frequency) and TF-IDF (Term Frequency – Inverted Document Frequency). For RDR, we use binary weights in order to obtain interpretable rules, which are based on the presence or absence of a term in a message. We train a separate classifier for every speech act, term weighting scheme, and discourse level. Because we are considering six speech acts, three term weighting schemes (one for RDR and the other two for the other models), three discourse levels, and a total of six different classifier types, the total number of models trained is 198. Additionally, we have trained all models using feature sets with reduced dimensionality obtained by removing the stop-words.

For training and validation we used 70% of the dataset, while the remaining 30% we used as a held-out test set. The training process includes the optimization of model parameters (except for NB and DS, which have no model parameters), which we accomplished using grid-search and cross-validation. For every parameter combination, a model is trained and evaluated using 10-fold cross-validation and the optimal parameters are chosen based on the F1 score averaged over ten folds. The optimal model is then re-trained on the whole training set and evaluated on the held-out set.

4. Evaluation

4.1. Classifier performance

Table 2 shows the performance of the six classifiers on the six different speech acts in terms of the F1 score. Here we show the performance of the best-performing models regardless on the discourse level or features used. The SVM and RDR classifiers consistently outperform other considered classifiers, with F1 scores reaching over 88%. SVMs not only performed well, but also had the lowest difference between the best and the worst performance, ranging from 75% (for the REMIND speech act) to 88.16% (for the DELIVER speech act). AdaBoost also showed a consistently good performance, and was the best performing classifier for the REMIND speech act. DS showed surprisingly good results, considering the simplicity of the model.

It can also be seen that most of the classifiers perform best on the DELIVER speech act. On the other hand, the REMIND speech act proved to be the most difficult to classify, which may be attributed to the fact that this speech act had by far the lowest number of training examples.

4.2. Discourse level

Table 3 shows the classifier performance on the three different discourse levels. We again show the performance of best-performing classifiers, regardless of features used.

The results exhibit no particular global regularities, such as that better performance may be obtained on sentences rather than on the complete messages, as might have been expected. However, the results may help us understand what are the levels on which particular speech acts are usually expressed. For instance, a reminder to someone is rarely expressed with a single sentence, thus it would be expected to see that for this particular speech act a classifier performs better on the message or paragraph level. On the other hand, deliveries are usually expressed in a small number of words, which is why classification at the sentence level showed to perform the best.

Overall, classification at the message level has shown to perform best for most speech acts, followed by the paragraph level. This could be attributed to the fact that all classifiers have a very high recall, and more surrounding text is needed to filter out the false positives.

4.3. Features

Table 5 shows the results obtained by choosing the best-performing classifier for each pair of speech act and feature type. In general, stop-word removal seems not to influence the classification performance. In the case when there is no stop-word removal, the performance of all three feature types was comparable in that there is no consistent pattern where one feature type outperforms the others, with only the binary feature under-performing for the REQUEST

Table 5: Classifier performance with respect to feature types (% F1)

	With stop-words			Without stop-words		
	Binary	TF	TF-IDF	Binary	TF	TF-IDF
DELIVER	**88.51**	87.50	88.00	**88.51**	88.16	87.96
AMEND	70.07	77.19	**79.31**	77.27	75.86	**77.19**
COMMIT	**83.75**	79.37	81.63	78.82	79.76	**81.97**
REMIND	76.92	76.92	**77.78**	75.00	**94.74**	77.78
SUGGEST	71.50	**76.84**	76.27	68.40	73.08	**73.68**
REQUEST	61.90	**78.76**	78.10	74.46	**78.08**	77.53

speech act. The differences between the F1 scores using different feature types were usually confined within 3%, which shows that the problem at hand is generally robust with respect to the term weighting schemes used.

4.4. Overall performance

The best performance of each classifier for a particular discourse level is presented in Table 4. Most classifiers show their best performance on the sentence level, which is in contradiction with the observation that for most speech acts the best classification is achieved on the message level. This, however, can be explained by taking into account that these results are highly influenced by the very high performance of all classifiers on the DELIVER speech act. The overall performance of the classifiers is relatively high compared to reported results for the English language: our F1 scores range from 79.31% to 94.74%, whereas Cohen et al. (2004) report F1 scores from 44% to 85%.

5. Conclusion

Speech acts provide an effective way of summarizing the intended purpose of email messages. We addressed the task of speech act classification of email classification in Croatian language. We framed this task as a multilabel text classification problem and performed thorough evaluation using six machine learning algorithms and three types of features (message-level, paragraph-level, and sentence-level features). We have shown that the discourse level and feature type do not significantly influence the performance. However, we were able to demonstrate that certain speech acts are more accurately classified at a particular discourse level. Using message-level features, we achieved an overall best F1 score of over 94%. The obtained F1 scores are notably higher than those reported in previous work.

An issue that we have not addressed in this paper is the practicaly usability of speech act classification for importance-based email classification; we leave this investigation for future work. Also for future work, we intend to further explore the relationship between the discourse levels and the speech act. Another possible direction of research would be to employ information extraction methods to augment each speech act with additional information such as named entities, temporal expressions, etc.

6. Acknowledgments

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under Grant 036-1300646-1986. We thank the anonymous reviewers for their comments.

7. References

J. Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, pages 249–254.

V. R. Carvalho and W. W. Cohen. 2006. Improving "email speech acts" analysis via n-gram selection. In *Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech*, pages 35–41.

W. W. Cohen, V. R. Carvalho, and T. M. Mitchell. 2004. Learning to classify email into "speech acts". In *Proceedings of EMNLP 2004*, pages 309–316.

M. Finke, M. Lapata, A. Lavie, L. Levin, L. Mayfield Tomokiyo, T. Polzin, K. Ries, A. Waibel, and K. Zechner. 1998. Clarity: Inferring discourse structure from speech. In *In Proc. of Workshop on Applying Machine Learning to Discourse Processing*.

S. Keizer. 2001. A Bayesian approach to dialogue act classification. In *BI-DIALOG 2001: Proc. of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue*, pages 210–218.

B. Klimt and Y. Yang. 2004. The Enron corpus: A new dataset for email classification research. In *Machine Learning: ECML 2004*, pages 217–226.

M. M. Louwerse and S. A. Crossley. 2006. Dialog act classification using n-gram algorithms. In *FLAIRS Conference*, pages 758–763.

J. Marineau, P. Wiemer-Hastings, D. Harter, B. Olde, P. Chipman, A. Karnavat, V. Pomeroy, A. Graesser, and the Tutoring Research Group. 2000. Classification of speech acts in tutorial dialog. In *Proc. of the workshop on modeling human teaching tactics and strategies, Intelligent Tutoring Systems 2000*, pages 65–71.

J. R. Searle. 1965. What is a speech act? *The Philosophy of Language, Oxford University Press*, pages 44–46.

R. Serafin, B. Di Eugenio, and M. Glass. 2003. Latent semantic analysis for dialogue act classification. In *Proceedings of HLT-NAACL 2003–short papers*, volume 2 of *NAACL-Short '03*, pages 94–96, Stroudsburg, PA, USA. Association for Computational Linguistics.

CroNER: A State-of-the-Art Named Entity Recognition and Classification for Croatian Language

Goran Glavaš, Mladen Karan, Frane Šarić, Jan Šnajder, Jure Mijić, Artur Šilić, Bojana Dalbelo Bašić

University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
takelab@fer.hr

Abstract

In this paper we present CroNER, a named entity recognition and classification system for Croatian language based on supervised sequence labeling with conditional random fields (CRF). We use a rich set of lexical and gazetteer-based features and different methods for enforcing document-level label consistency. Extensive evaluation shows that our method achieves state-of-the-art results (MUC F1 90.73%, Exact F1 87.42%) when compared to existing NERC systems for Croatian and other Slavic languages.

CroNER: orodje za prepoznavanje in klasifikacijo imenskih entitet v hrvaščini

V pričujočem prispevku predstavljamo CroNER, sistem za prepoznavanje in klasifikacijo imenskih entitet za hrvaščino, ki temelji na nadzorovanemu označevanju s pomočjo pogojnih naključnih polj (*conditional random fields* – CRF). Za označevanje uporabimo bogat nabor leksikalnih lastnosti ter imenik, doslednost oznak na ravni dokumenta pa dosežemo z različnimi metodami. Obsežno vrednotenje rezultatov in primerjava z drugimi tovrstnimi sistemi za hrvaščino in ostale slovanske jezike kažeta, da naša metoda sodi med najuspešnejše (MUC F1 90,73%, Exact F1 87,42%).

1. Introduction

Named Entity Recognition and Classification (NERC) is a well-known natural language processing (NLP) and Information Extraction (IE) task. NERC aims to extract and classify all names (*enamexes*), temporal expressions (*timexes*), and numerical expressions (*numexes*) appearing in natural language texts. The classes of named entities typically extracted by NERC systems are names of people, organizations, and locations as well as dates, temporal expressions, monetary expressions, and percentages.

In this paper we present CroNER, a supervised NERC for the Croatian language. We use sequence labeling with conditional random fields (CRF) (Lafferty et al., 2001) to extract and classify named entities from newspaper text. We use a rich set of features, including lexical and gazetteer-based features, with many of them incorporating morphological and lexical peculiarities of the Croatian language. We implemented two different methods for document-level consistency of NE labels: postprocessing rules (hard consistency constraint) and a two-stage CRF (soft consistency constraint). Postprocessing rules are hand-crafted patterns designed to extract or re-label named entities omitted or misclassified by the CRF model. Two-stage CRF (Krishnan and Manning, 2006) aims to consolidate NE label predictions on document and corpus level by employing a second CRF model that uses features computed from the output of the first CRF model. We evaluate the performance of the system using standard MUC and Exact NERC evaluation schemes (Nadeau and Sekine, 2007).

The rest of the paper is structured as follows. In Section 2 we present related work on named entity extraction for Croatian and other Slavic languages. Section 3 discusses the details of corpus annotation. In Section 4 we thoroughly describe the feature set and the extensions used (rule-based postprocessing and two-stage CRF). Section 5 presents experimental setup and evaluation results. In Section 6 we conclude and outline future work.

2. Related Work

Identifying references to named entities in text was recognized as one of the important subtasks of IE, and it has been a target of intense research for the last twenty years. The task was formalized at the Sixth Message Understanding Conference in 1995 (Grishman and Sundheim, 1996). There is a large body of NERC work for English (Mikheev et al., 1998; McCallum and Li, 2003; Etzioni et al., 2005; Krishnan and Manning, 2006) and other major languages (Faruqui and Padó, 2010; Yu et al., 1998; Cucchiarelli and Velardi, 2001; Poibeau, 2003). Substantially less research has targeted Slavic (especially South Slavic) languages; NERC systems have been reported for Russian (Popov et al., 2004), Polish (Piskorski, 2004; Marcińczuk and Janicki, 2012), Czech (Kravalová and Žabokrtskỳ, 2009), and Bulgarian (Da Silva et al., 2004; Georgiev et al., 2009). Georgiev et al. (2009) showed that CRF-based NERC with a rich set of features outperforms all other methods for Bulgarian, as well as other Slavic languages.

The rule-based system by Bekavac and Tadić (2007), which uses a cascade of finite-state transducers, is the only reported work on NERC for Croatian language. Ljubešić et al. (2008) propose a method for generating a morphological lexicon of organizational names, a valuable resource for morphologically rich languages. We used a similar approach to expand morphological lexica with inflectional

forms of Croatian proper names, but we include first names, surnames, and toponyms in addition to organization names.

To the best of our knowledge, we are the first to use supervised machine learning for named entity recognition and classification in Croatian language. Using a machine learning method, we avoid the need for specialized linguistic knowledge required to design a rule-based system. This way we also avoid the explicit modelling of complex dependencies between rules and their application order. We instead focus on designing a rich set of features and let the CRF algorithm uncover the dependencies between them.

3. Corpus Annotation

The training and testing corpus consists of 591 news articles (about 310,000 tokens) from the Croatian newspaper *Vjesnik*, spanning years 1999 to 2009. The preprocessing of the corpus involved sentence splitting and tokenization. For annotation we used seven standard MUC-7 types: *Organization*, *Person*, *Location*, *Date*, *Time*, *Money*, and *Percent*. We also introduced five additional types: *Ethnic* (names of ethnic groups), *PersonPossessive* (possesive adjectives derived from person names), *Product* (names of branded products), *OrganizationAsLocation* (organization names used as metonyms for locations, as in *"The entrance of the PBZ bank building"*), and *LocationAsOrganization* (location names used as metonyms for organizations, as in *"Zagreb has sent a demarche to Rome"*). The additional types were introduced for experimental reasons; in this work only the *Ethnic* tag was retained, while other additional tags were not used (i.e., the *Product* tag was discarded, while the remaining three subtype tags were mapped to the corresponding basic tags). Thus, in the end we trained our models using eight types of named entities.

The annotation guidelines we used are similar to MUC-7 guidelines, with some adjustments specifically for the Croatian language. The corpus was independently annotated by six annotators. To ensure high annotation quality, the annotators were first asked to independently annotate a calibration set of about 10,000 tokens. On this set, all the disagreements have been resolved by consensus, the borderlines were discussed, and the guidelines revised accordingly. Afterwards, each of the remaining documents was annotated by two independent annotators, while a third annotator resolved the disagreements. For annotating we used an in-house developed annotation tool.

The inter-annotator agreement (calculated in terms of MUC F1 and Exact F1 score and averaged over all pairs of annotators) is shown in Table 1. The inter-annotated was measured on a subset of about 10,000 tokens that was annotated by all six annotators. Notice that the overall quality of the annotations improved after resolving the disagreements, but – because each subset was resolved by a single annotator – we cannot objectively measure the resulting improvement in annotation quality.

Table 1: Inter-annotator agreement

Tag	F1 Exact	F1 MUC
Person	98.05	98.55
Ethnic	97.19	97.19
Percent	92.00	96.77
Location	93.95	94.93
Money	91.95	94.15
Organization	89.35	93.58
Date	71.47	85.79
Time	67.55	71.04

4. CroNER

CroNER is based on sequence labeling with CRF. We use the CRFsuite (Okazaki, 2007) implementation of CRF. At the token level, named entities are annotated according to the Begins-Inside-Outside (B-I-O) scheme, often used for sequence labeling tasks. Following is a description of the features used for sentence-level label prediction and the techniques for imposing document-level label consistency.

4.1. Sentence-level features

Most of the features can be characterized as lexical, gazetteer-based, or numerical. Some of the features were *templated* on a window of size two, both to the left and to the right of the current word. This means that that the feature vector for the current word consists of features for this word, two previous words, and two following words.

Lexical features. The following is the list of the lexical features used (templated features are indicated as such).

1. Word, lemma, stem, and POS tag (*templated*) – For lemmatization we use the morphological lexicon developed by Šnajder et al. (2008). For stemming, we simply remove the word's suffix after the last vowel (or the penultimate vowel, if the last letter is a vowel). Words shorter than 5 letters are not stemmed. For POS tagging, we use a statistical tagger with five basic tags.
2. Full and short shape of the word – describe the ordering of uppercased and lowercased letters in the word. For example, *"Zagreb"* has the shape *"ULLLLL"* and short shape *"UL"*, while *"iPhone"* has the shape *"LULLLL"* and short shape *"LUL"*.
3. Sentence start – indicates whether the token is the first token of the sentence.
4. Word ending – the suffix of the word taken from the last vowel till the end of the word (or the penultimate vowel, if the last letter of the word is a vowel).
5. Capitalization and uppercase (*templated*) – indicates whether the word is capitalized or entirely in uppercase (e.g., an acronym).
6. Acronym declension – indicates whether the word is a declension of an acronym (e.g., *"HOO-om"*, *"HDZ-a"*). Declension of acronyms in Croatian language follows predictable patterns (Babić et al., 1996).
7. Initials – indicates whether a token is an initial, i.e., a single uppercase letter followed by a period.
8. Cases – concatenation of all possible cases for the word, based on morpho-syntactic descriptors (MSDs)

from the morphological lexicon. If the word has two or more MSDs with differing cases, we concatenate them in alphabetical order. We also add one Boolean feature for each individual case (*isNominative*, *isGenitive*, *isDative*, *isAcusative*, and *isInstrumental*).

9. Bigram features – concatenations of the previously described features computed for two consecutive tokens: *word bigram*, *lemma bigram*, *POS bigram*, *shape bigram*, and *cases bigram*.
10. Lemmas in window – all lemmas within a symmetric window of size 5 from the current token.
11. MSDs in window – all MSDs of the words within a symmetric window of size 5 from the current token.

Gazetteer-based features. Information about the presence of named entities from predefined gazetteers has been shown to be an important information for NERC (Nadeau and Sekine, 2007). We use several gazetteers: first names, surnames, ethnics, organizations, cities, streets, and countries gazetteers. The last four gazetteers have multi-word entries. The following is a list of gazetteer-based features.

1. Gazetteer match – indicates whether the lemma matches a gazetteer entry (used for gazetteers with single-word entries: names, surnames, and ethnics).
2. Starts gazetteer match – indicates whether there is any sequence of words starting with the current word that fully matches a gazetteer entry. E.g., in *"usluge Zavoda za javno zdravstvo"* (*services of the Public Health Department*), the word *"Zavoda"* would have this feature set to *true* because the organizations gazetteer contains *"Zavod za javno zdravstvo"*.
3. Stemmed gazetteer match – similar to the previous feature, but considers stems instead of lemmas. This feature is used only for the organizations gazetteer.
4. Gazetteer match length – the length (number of words) of the gazetteer entry whose first token matches the current token (e.g., for token *"Zavod"* in text *"usluge Zavoda za javno zdravstvo"*, the length would be 4).
5. Inside gazetteer match – indicates whether a word is inside the phrase that matches a gazetteer entry (e.g., true for tokens *"za"*, *"javno"*, and *"zdravstvo"* in organization entry *"Zavod za javno zdravstvo"*).

Both the text and the gazetteer entries were lemmatized before looking for matches. As gazetteers predominantly contain proper nouns, we needed to extend the morphological lexicon with the inflectional forms of proper names. We did this automatically with a set of rules following the paradigms for proper names declension (Babić et al., 1996). We expanded both Croatian and foreign proper names.

Some simple preprocessing steps were applied for all gazetteers. All entries containing non-alphabetic characters were removed. We considered all words with more than 10% non-capitalized occurrences in the corpus to be common words and removed such entries. The rationale was to eliminate common word entries from the gazetteers in order to reduce the noise in the training set. For example, *"Luka"* is a very common personal name, but also a frequent common noun (*port*). Capitalization frequencies required for the above analysis were gathered from the *Vjesnik* corpus, a collection of 270,000 newspaper articles.

The major source of the Croatian names and surnames was the Croatian telephone directory. For English names, we used Stanford NER[1] to extract names from the NYT corpus[2] and Wikipedia. The compiled gazetteers for personal names and surnames contain 13,618 Croatian first names, 64,240 Croatian surnames, 70,488 foreign first names, and 228,134 foreign surnames. For locations we use three gazetteers – for streets, countries and cities. The street names (52,593 entries) were extracted from the Croatian telephone directory. Country names in Croatian (276 entries) were obtained from Wikipedia. The cities gazetteer (289,707 entries) was constructed using the telephone directory and internet sources. The organizations gazetteer (3035 entries) was created from several different sources, and includes names of institutions (e.g., *Ministry of Science*, *Louvre*), political parties (e.g., SDP, HDZ), international organizations (e.g., *UNESCO*, *NATO*), local and foreign companies, newspaper names, and sports teams. Finally, we compiled the ethnics gazetteer (940 entries) automatically from country names using the appropriate rules of Croatian grammar (Babić et al., 1996).

Numerical features. We used the following features to deal specifically with numbers (occurring in numexes and timexes):

1. Integer or decimal number – indicates whether the word is an integer or a decimal number;
2. Two/four digit integer – indicates whether the token is a two digit (useful for recognizing numexes) or a four digit integer (useful for recognizing years in dates);
3. Number followed by a period – indicates whether the token is an integer followed by a period (a good clue for dates and currencies);
4. Currency – indicates whether a token is a currency marker (e.g., *"$"* or *"EUR"*). We compiled a currency gazetteer that includes all major world currencies.

4.2. Document-level consistency

The CRF model predicts the sequence of B-I-O labels on the sentence level. It is therefore possible to have at the document level differing labels for the same named entity. The goal of the document-level label consistency postprocessing is to unify the labels of named entities on the document level. We experimented with incorporating document-level consistency into our model as both soft constraints (two-stage CRF) and hard constraints (hand-crafted postprocessing rules).

Two-stage CRF. The two-stage CRF (Krishnan and Manning, 2006) is a model that accounts for non-local dependencies between named entities. The main idea is to employ a second CRF that uses both local features (same features the first CRF uses) and non-local features computed

[1] http://nlp.stanford.edu/software/CRF-NER.shtml
[2] The New York Times Annotated Corpus, (2008), LDC.

on the output of the first CRF. We use three document-level features computed from the output of the first CRF:

1. The most frequent lemma label – the most frequent label assigned to a given lemma in the document (e.g., *B_Person* or *I_Organization*);
2. The most frequent NE label – the most frequent label assigned to a given NE mention in the document;
3. The most frequent superentity label – a superentity is a mention of the same entity that contains two or more tokens (e.g., *"Ivan Horvat"* vs. *"Horvat"*, or *"Zavod za javno zdravstvo"* vs. *"Zavod"*). This feature represents the most frequent label assigned to all the superentities of a given entity within the document.

Postprocessing rules (PPR). We created two sets of postprocessing rules: one to enforce document-level consistency (hard constraint) and another one to improve the recall on numexes and timexes. The rules for enforcing document-level label consistency work as follows. First, we collect all the different named entities recognized by the CRF model and identify the most frequent label assigned to each of them. Then we correct (i.e., re-label) NE instances that were assigned a different label from the most frequently assigned one. In the second step, we search for the potential false negatives (i.e., mentions of named entities from the collection that were omitted by the CRF model). If found, omitted mentions are also assigned the most frequent label for the corresponding named entity.

The rules for improving the recall for numexes are in fact token-level regular expressions. For currencies and percentages the rules are defined as follows:

1. $[num][num|prep|conj]^*[currencyMarker]$ – the currency expression starts with a number, followed by either numbers, prepositions, or conjunctions, and ends with a currency clue. When written in words, numbers often contain conjunctions. E.g., in *"trideset i pet"* (*thirty five*), word *"i"* is a conjunction. Ranges are often expressed using prepositions; e.g., *"30 do 50 milijuna kuna"* (*30 to 50 million kuna*);
2. $[num][num|prep|conj]^*[percentClue]$ – the rule for percentages is similar to the rule for currencies, except for requiring that the phrase ends with a percent clue (*"posto"* or *"%"*) instead of a currency marker.

For timex (time) class we use the following three rules:

1. [u]$[number][timeword]$ – captures phrases like *"u 12.30 sati"* (*at 12.30 o'clock*), where *number* is an appropriately formatted number and *timeword* is a word from a predefined list of time-related words, e.g., *"sati"* (*o'clock*);
2. $[mod]?[preposition]?[daytimeword][mod]?$ – captures phrases like *"rano u jutro"* (*early in the morning*). Here *mod* represents a modifying word, e.g.. *"rano"* (*early*);
3. $[modGen][daytimeword]$ – captures phrases like *"tijekom podneva"* (*during the afternoon*), where *modGen* is a modifier that governs a noun in genitive case; e.g., *"prije"* (*before*).

5. Evaluation

We measured the performance of four different models: single CRF (1-CRF), two-stage CRF (2-CRF), single CRF with postprocessing rules (1-CRF + PPR), and two-stage CRF with postprocessing rules (2-CRF + PPR). In Tables 2 and 3 we report the performance in terms of precision, recall, and F1 for MUC (allows for extent overlap instead of an exact extent match) and Exact (requires that both extent and class match) evaluation schemes (Nadeau and Sekine, 2007), respectively. Results are reported separately for each NE class. We also report both micro- and macro-averaged overall performance for each of the four models. The results were obtained with 10-fold cross validation on the entire annotated corpus.

5.1. Result analysis

Regarding the enamex classes, the performance for organizations is significantly (5–7%) worse than for persons and locations. This is expected, because in Croatian many organization instances are multi-word expressions, whereas person and location mentions more often consist of only one or two words. The lower inter-annotator agreement (cf. Table 1) for organizations supports this assumption.

The results show that 2-CRF outperforms 1-CRF consistently on main enamex classes (*Person*, *Organization*, and *Location*); the improvement is between half a point (*Location*) and a full point (*Organization*). The 1-CRF + PPR model similarly outperforms 1-CRF (e.g., 0.8 point increase for *Person*). However, the 2-CRF + PPR model brings negligible gain when compared to either 2-CRF or 1-CRF + PPR (on average 0.1 point for enamex classes). This indicates that both the second stage CRF and postprocessing rules ensure document-level consistency in a similar fashion, hence combining them does not lead to significant performance improvements.

For numexes, the second CRF model seems not to improve the performance, whereas the postprocessing rules significantly improve the performance. This improvement is to be attributed to the use of extraction rules for numexes, implying that document-level consistency is not an issue for numexes. Postprocessing rules for currencies and percents increase the recall and keep the precision on the same level. For temporal expressions, however, increase in recall is accompanied by a proportional decrease in precision. Deeper inspection reveals that this is mostly due to inconsistent annotations of timexes, as confirmed by the very low inter-annotator agreement for these classes (cf. Table 1).

As expected, Exact evaluation results are generally lower than MUC results. However, for most classes the decrease in performance is not significant. Exceptions to this are *Organization*, *Date*, and *Time* classes, for which the decrease in performance is 7%, 7%, and 11%, respectively. Many organization instances consist of four or more words, and in such cases our models – though able to recognize the mention – often fail to exactly match its extent. The most common errors include omitting the last word or adding an extra word at the end. The performance on the three mentioned classes is also limited by the annotation quality; these classes are in fact the ones on which human annotators agreed the least (cf. Table 1).

Table 2: CroNER MUC evaluation results

	1-CRF			2-CRF			1-CRF + PPR			2-CRF + PPR		
NE Class	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Person	91.31	92.12	91.71	91.76	93.26	92.50	91.13	93.58	92.34	91.62	93.68	**92.64**
Location	89.27	89.77	89.52	89.83	90.30	**90.06**	88.30	91.00	89.63	89.00	90.46	89.72
Organization	88.15	81.65	84.78	88.66	82.94	**85.71**	85.51	84.74	85.13	86.43	84.11	85.25
Ethnic	96.82	90.56	93.59	97.73	90.55	94.01	97.74	90.56	94.01	98.29	90.56	**94.27**
Date	93.72	82.35	87.67	93.48	82.02	87.38	93.55	83.05	**87.99**	93.56	82.47	87.67
Time	91.86	50.22	64.94	91.74	49.33	64.16	76.96	78.67	77.80	77.06	79.11	**78.07**
Currency	99.54	87.30	93.02	99.32	88.10	93.37	99.20	99.20	**99.20**	99.20	99.20	**99.20**
Percent	100.00	96.43	98.18	100.00	96.21	98.07	99.54	97.77	**98.65**	99.54	97.77	**98.65**
Overall Micro	90.67	87.21	88.91	91.07	87.99	89.51	89.48	89.43	89.45	90.09	89.09	**89.59**
Overall Macro	93.84	83.80	88.78	94.06	84.08	88.79	91.49	89.82	90.65	91.83	89.67	**90.73**

Table 3: CroNER Exact evaluation results

	1-CRF			2-CRF			1-CRF + PPR			2-CRF + PPR		
NE Class	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Person	89.42	90.22	89.81	89.92	91.38	**90.64**	89.06	91.46	90.24	89.62	91.64	90.62
Location	87.60	88.09	87.84	88.11	88.57	**88.34**	86.58	89.21	87.87	87.34	88.74	88.03
Organization	80.79	74.83	77.70	81.05	75.82	**78.35**	77.26	76.94	77.10	78.58	76.57	77.56
Ethnic	96.82	90.56	93.59	97.74	90.56	94.01	97.73	90.56	94.01	98.29	90.56	**94.27**
Date	86.19	75.73	80.62	85.98	75.44	80.37	85.73	76.10	**80.63**	85.95	75.77	80.54
Time	87.80	48.00	62.07	88.43	47.55	61.85	66.08	67.55	66.81	66.23	68.00	**67.10**
Currency	95.93	84.13	89.64	95.75	84.92	90.01	96.45	97.22	**96.84**	96.27	97.22	96.74
Percent	95.60	92.19	93.86	95.82	92.19	93.97	98.86	97.09	**97.97**	98.86	97.10	**97.97**
Overall Micro	86.84	83.53	85.15	87.19	84.24	**85.69**	85.30	85.36	85.33	86.08	85.17	85.62
Overall Macro	90.00	80.47	84.97	90.35	80.80	85.31	87.21	85.76	86.49	87.64	87.20	**87.42**

Table 4 shows the performance of the best-performing model (2-CRF + PPR) depending on the size of the training set. (25%, 50%, 75%, and 100% of the training data). Expectedly, the performance generally improves as the size of the training set increases. However, the improvement from using 75% data to using 100% data is relatively small, suggesting that no significant increase in performance could be gained from annotating a larger corpus.

5.2. Discussion

Unfortunately, our results are not directly comparable to other reported results because of the differences in (1) language (though very similar, all Slavic languages have their own peculiarities), (2) NE types (e.g., some use only four classes: *Person*, *Location*, *Organization*, and *Miscellaneous*), or (3) evaluation methodology (non-adherence to standard evaluation methodology, such as in the work by Bekavac and Tadić (2007)). Nonetheless, the comparison might still be informative to some extent. Bekavac and Tadić (2007) report a 79% F1-score on persons, 89% on organizations, and 95% on locations, although it must be noted that for the latter two classes their evaluation was limited to selected subsets of NE instances. Our results seem to be better than those reported for other Slavic languages: Polish – 82.4% F1, (Piskorski, 2004), Czech – 76% F1, Russian – 70.9% F1 (Popov et al., 2004). Only the best reported results for Bulgarian are comparable to our results: 89.6% overall F1, persons 92.79%, locations 90.06%, organizations 89.73% (Georgiev et al., 2009). These comparisons suggest that CroNER is a state-of-the-art NERC system when considering the Slavic languages.

6. Conclusion and Future Work

We have presented CroNER, a NERC system for Croatian based on sequence labeling with CRF. CroNER uses a rich set of lexical and gazetteer-based features achieving good recognition and classification results. We have shown how enforcing document-level label consistency (either through postprocessing rules or a second CRF model capturing non-local dependencies) can further improve NERC performance. The experimental results indicate that, as regards the Slavic languages, CroNER is a state-of-the-art named entity recognition and classification system.

The work presented here could be extended in several ways. First, the annotated set should be revised, considering that the inter-annotator agreement is rather low on some classes. Secondly, a systematic feature selection (e.g., wrapper feature selection) may be performed in order to select an optimal subset of features. Thirdly, we plan to employ classification using more fine-grained NE labels.

Table 4: CroNER performance depending on the size of the training set (CRF-2 + PPR)

Evaluation	Set size (tokens)	Person	Loc.	Org.	Ethnic	Date	Time	Curr.	Perc.	Ov. Micro	Ov. Macro
MUC	25% (75k)	92.51	82.69	79.95	92.30	79.46	78.74	100.00	98.99	86.01	88.08
	50% (155k)	92.56	87.56	82.60	93.70	85.01	76.40	99.62	98.64	88.05	89.51
	75% (230k)	92.19	88.81	85.00	94.87	87.30	76.84	99.59	98.77	89.07	90.42
	100% (310k)	92.64	89.72	85.25	94.27	87.67	78.07	99.20	98.65	**89.59**	**90.73**
Exact	25% (75)	90.17	79.50	69.53	92.30	71.57	59.84	96.97	98.32	80.65	82.28
	50% (155k)	90.59	85.04	73.66	93.70	76.47	62.17	97.51	97.74	83.35	84.61
	75% (230k)	90.06	86.71	77.25	94.87	79.45	65.40	97.24	97.84	84.80	86.10
	100% (310k)	90.62	88.03	77.56	94.27	80.54	67.10	96.74	97.97	**85.62**	**87.42**

7. Acknowledgments

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia, under Grant 036-1300646-1986.

8. References

S. Babić, B. Finka, and M. Moguš. 1996. *Hrvatski pravopis*. Školska knjiga.

B. Bekavac and M. Tadić. 2007. Implementation of Croatian NERC system. In *Proc. of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, pages 11–18.

A. Cucchiarelli and P. Velardi. 2001. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131.

J.F. Da Silva, Z. Kozareva, and GP Lopes. 2004. Cluster analysis and classification of named entities. In *Proc. Conference on Language Resources and Evaluation*, pages 321–324.

O. Etzioni, M. Cafarella, D. Downey, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

M. Faruqui and S. Padó. 2010. Training and evaluating a German named entity recognizer with semantic generalization. *Semantic Approaches in Natural Language Processing*, page 129.

G. Georgiev, P. Nakov, K. Ganchev, P. Osenova, and K. Simov. 2009. Feature-rich named entity recognition for Bulgarian using conditional random fields. In *Proc. of the Conference on Recent Advances in Natural Language Processing (RANLP'2009)*, pages 113–117.

R. Grishman and B. Sundheim. 1996. Message Understanding Conference-6: A brief history. In *Proc. of COLING*, volume 96, pages 466–471.

J. Kravalová and Z. Žabokrtskỳ. 2009. Czech named entity corpus and SVM-based recognizer. In *Proc. of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 194–201.

V. Krishnan and C.D. Manning. 2006. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 1121–1128.

J. Lafferty, A. McCallum, and F.C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

N. Ljubešić, T. Lauc, and D. Boras. 2008. Generating a morphological lexicon of organization entity names. In *Proc. of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*.

M. Marcińczuk and M. Janicki. 2012. Optimizing CRF-based model for proper name recognition in Polish texts. *Computational Linguistics and Intelligent Text Processing*, pages 258–269.

A. McCallum and W. Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191.

A. Mikheev, C. Grover, and M. Moens. 1998. Description of the LTG system used for MUC-7. In *Proc. of 7th Message Understanding Conference (MUC-7)*.

D. Nadeau and S. Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

N. Okazaki. 2007. CRFsuite: a fast implementation of conditional random fields (CRFs).

J. Piskorski. 2004. Extraction of Polish named entities. In *Proc. of the Fourth International Conference on Language Resources and Evaluation, LREC*, pages 313–316.

T. Poibeau. 2003. The multilingual named entity recognition framework. In *Proc. of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 155–158.

B. Popov, A. Kirilov, D. Maynard, and D. Manov. 2004. Creation of reusable components and language resources for named entity recognition in Russian. In *Proc. of the Fourth International Conference on Language Resources and Evaluation, LREC*, pages 309–312.

J. Šnajder, B.D. Bašić, and M. Tadić. 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing & Management*, 44(5):1720–1731.

S. Yu, S. Bai, and P. Wu. 1998. Description of the Kent Ridge Digital Labs system used for MUC-7. In *Proc. of the Seventh Message Understanding Conference*.

Toward computational modeling of the comprehension deficit in Broca's aphasia

Milan Gnjatović, Vlado Delić

Faculty of Technical Sciences
University of Novi Sad
Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia
milangnjatovic@yahoo.com, vdelic@uns.ac.rs

Abstract

Broca's aphasics suffer a highly restricted receptive disorder of syntax (Grodzinsky, 2000). They have severe comprehension difficulties with syntactic structures containing transformational operations (i.e., syntactic movement), and exhibit near-normal performance in most other domains of syntax. However, despite the deficit in receptive mechanisms of grammatical analysis, Broca's aphasics use, in certain cases, semantic cues (i.e., the general knowledge of the world) to get around their deficit. Understanding of cognitive mechanisms that underlay this behavior may be valuable for researchers that aim at addressing the question of enabling dialogue systems to process spontaneously produced user's utterances of different syntactic forms with no explicit syntactic expectations. This paper presents a cognitively-inspired approach to computational modeling of the comprehension deficit in Broca's aphasia. We consider a neurolinguistics insight into the comprehension deficit, and introduce three basic requirements for an approach aimed at modeling this deficit. Finally, we discuss that the focus tree is appropriate for modeling the most salient aspects of the comprehension deficit in Broca's aphasia.

V smeri računalniškega modeliranja motenj v razumevanju pri Brockovi afaziji

Pacienti z Brockovo afazijo imajo zelo specifično motnjo pri razumevanju sintakse (Grodzinsky, 2000). Imajo resne težave pri razumevanju sintaksnih struktur, ki vključujejo t.i. sintaksni premik, medtem ko je njihovo razumevanje v preostalih domenah sintakse blizu normalnemu. Vendar lahko ti pacienti navkljub motnji v receptivnem mehanizmu gramatične analize v določenih primerih uporabljajo semantične namige (t.j. splošno znanje o svetu), da bi zaobšli motnjo pri razumevanju sintakse. Razumevanje kognitivnih mehanizmov, ki botrujejo takemu obnašanju, je lahko ključnega pomena za razvoj sistemov dialoga za obdelavo spontano izgovorjenih sporočil z različnimi sintaksnimi strukturami brez predefinirane gramatike. Članek predstavlja kognitivno naravnan pristop k računalniškemu modeliranju motenj v razumevanju pri Brockovi afaziji. Obravnava nevrolingvistični vpogled v deficit pri razumevanju ter predlaga tri osnovne zahteve za modeliranje te motnje. Iz zaključne diskusije je razvidno, da je model fokusnega drevesa primeren za modeliranje najbolj izrazitih vidikov motnje v razumevanju pri Brockovi afaziji.

1. Introduction

One of the fundamental understandings of language is that it is a modularly organized neurological entity (Grodzinsky, 2000, pp. 1–3). The insight in the cognitive neuroscience shows that syntax is anatomically distinguished from semantics and the lexicon. Discussing the neurolinguistic model of language perception, Grodzinsky notes that it is widely accepted that syntax is represented in the part of the left anterior cortex (i.e., Broca's area and its vicinity), while semantics and the lexicon are located temporoparietally around the Sylvian fissure. This anatomical distinction may be illustrated by observing a specific language impairment—Broca's aphasia.

The term "aphasia" refers to an impairment of language ability caused by a brain injury due to stroke, brain tumor, head trauma, etc. There are many types of aphasia that, with respect to the location of the brain injury, affect different communicative skills, e.g., coming up with specific lexical items, generating syntactic strings of words, comprehension, repetition, etc. (for a detailed overview cf. Obler and Gjerlow (1999)). Here, we consider one particular type of aphasia, i.e., Broca's aphasia, caused by an injury to a part of the brain in the left frontal lobe, called Broca's area, and its vicinity (represented in Brodmann's cytoarchitectonic map as areas 44 and 45, cf. Dronkers et al. (2007, p. 2)). It is commonly characterized by a nonfluent and effortful speech that contains only content words, while function words, morphemes and syntactic constructions are missing (Van der Meulen, 2004, p. 6). However, we focus on the less salient, although not less fundamental, comprehension deficit in Broca's aphasia that has a syntactic character. Broca's aphasics suffer a highly restricted receptive disorder of syntax. They have severe comprehension difficulties with syntactic structures containing transformational operations (i.e., syntactic movement), and exhibit near-normal performance in most other domains of syntax (Grodzinsky, 2000, p. 4).

It is important to note that Broca's aphasics rely on use of the lexicon and the general knowledge of the world in order to get around their comprehension deficit. In other words, despite the deficit in receptive mechanisms of grammatical analysis, they use, in certain cases, semantic cues in order to correctly interpret the given input. Understanding of cognitive mechanisms that underlay this behavior may be valuable for researchers that aim at addressing the question of enabling dialogue systems to process spontaneously produced user's utterances of different syntactic forms with no explicit syntactic expectations.

This paper presents a cognitively-inspired approach to computational modeling of the comprehension deficit in Broca's aphasia. It expands upon previous work. Our approach is based on the focus tree—a computational model

Table 1: Description of the pictures used in the study of Caramazza and Zurif (1976). The list is adopted and adjusted from the work of Van der Meulen (2004, p. 8).

Picture description	***Change***
(1) A dog chasing a brown cat.	Correct response
(2) A dog chasing a *black* cat.	Lexical change (adjective)
(3) A dog *biting* a brown cat.	Lexical change (verb)
(4) A dog *biting* a *black* cat.	Lexical change (adjective and verb)
(5) A cat chasing a brown dog.	Syntactic change (subject-object reversal)

of attentional state in task-oriented human-machine interaction (Gnjatović et al., 2011). From the methodological point of view, the focus tree is inspired by human information processing system, i.e., it is a computationally appropriate representation of attentional information that imitates the function of a focus of attention in human perception. It integrates neurocognitive understanding of the focus of attention (Bledowski et al., 2010; Oberauer and Lange, 2009) and notions of attention in computational (Grosz and Sidner, 1986) and corpora linguistics (Gnjatović and Rösner, 2010).

The paper is organized as follows. Section 2. considers the comprehension deficit in Broca's aphasia in more detail. Bases on this neurolinguistics insight, Section 3. introduces the basic requirements for an approach aimed at modeling this deficit. In Section 4., we discuss that the focus tree is appropriate for modeling the most salient aspects of the comprehension deficit in Broca's aphasia.

2. A neurolinguistics insight into the comprehension deficit in Broca's aphasia

In terms of Grodzinsky (2000, pp. 2–3), syntax constitutes a central combinatorial aspect of language. From the functional point of view, syntax is related to capacity to produce and analyze meaningful expressions through rule-based combinations. The role of Broca's area in syntax is highly specific and related to computation of transformational relations between moved phrasal constituents and their extraction sites. This is in line with brain imaging studies indicating that in language comprehension Broca's area is activated when higher levels of linguistic processing are required (D'Ausilio et al., 2010). To illustrate this, we refer to the milestone study conducted by Caramazza and Zurif (1976). In a part of their study, Broca's patients were asked to select a picture that represents the object relative clause "The cat that the dog is chasing is brown". Each subject was given two pictures—a picture that represents the correct answer, and one of the four pictures that represent incorrect situations, as described in Table 1.

(i) Broca's aphasics do not have impairment in their lexicon, but in syntax (Grodzinsky, 2000, p. 4). As summarized by Van der Meulen (2004, pp. 7–8), the study shows that Broca's patients never mistakenly selected the pictures (2), (3) or (4), i.e., they did not make lexical errors. On the other hand, when they had to choose between the pictures (1) and (5), the study reports chance-level performance of patients (i.e., guessing). This experimental condition is particularly illustrative for their comprehension deficit. The clause "The cat that the dog is chasing is brown" contains two noun phrases ("the cat" and "the dog"), the first carrying the Theme role, and the second carrying the Agent role. If these phrases changed their positions in the clause, they would also change their semantic roles, but the new clause (i.e., "The *dog* that the *cat* is chasing is brown") would still be semantically possible. That is why these clauses are referred to as *semantically reversible*. In order to interpret these clauses, the listener must correctly assign the semantic roles to the noun phrases. However, it should be noted that both these noun phrases are animate and, when observed outside of the syntactic structure of the given clause, could be assigned the Agent role. Therefore, assignment of the semantic roles in this case is determined only by the syntactic structure of the clause. Due to their deficit in receptive mechanisms of grammatical analysis, Broca's patients cannot distinguish between these two semantic cases, and, thus, they can only try to guess the Agent of the action, which, in turn, results in chance-level performance.

(ii) Broca's aphasics are able to use semantic cues (i.e., the general knowledge of the world) to get around their comprehension deficit (Grodzinsky, 2000, p. 4). The study of Caramazza and Zurif also considers the following *semantically irreversible* sentence: "The apple that the boy is eating is red". The patients were confronted with the same syntactic structure as in the previous sentence, but, in this case, it is not possible to reverse the semantic roles, i.e., the interpretation that the apple eats the boy is semantically incorrect. It is important to note that Broca's aphasics are able to use their knowledge of the world to correctly interpret this sentence (i.e., to assign the Agent role to the boy), although they cannot comprehend the underlying syntactic structure (Van der Meulen, 2004, pp. 7–8).

Grodzinsky (2000, pp. 4–7) and Van der Meulen (2004, pp. 21–26) provide useful overviews of this deficit in receptive mechanisms of grammatical analysis that is characteristics for Broca's aphasia. Here, we highlight some aspects that are most relevant for this contribution. A widely accepted patterns of comprehension data taken from dozens of experiments that investigated aphasics' interpretive abilities are summarized in Tables 2 and 3 (Grodzinsky, 2000, pp. 4–5). The clauses given in Table 2 reflect construction types that Broca's aphasics correctly interpret at above chance-level of performance, while the clauses given in Table 3 reflect construction types with chance-level of performance.

(iii) Broca's aphasics can comprehend basic phrase syntactic structures. The experimental record shows that Broca's aphasics are able to comprehend basic syntactic trees (i.e., phrase structures) for simple sentences that do not contain intrasentential dependency relations, such as active sentences, e.g., "the girl pushed the boy" (6) or "a dog chasing a brown cat" (1), etc. (Grodzinsky, 2000, p. 4). This observation is in line with Chomsky's notion of *kernel sentences* (cf. also Kay (2000)). According to him, every sentence of the language either belongs to the kernel

Table 2: Clause patterns with above chance-level performance, adopted and adjusted from the work of Grodzinsky (2000, p. 5).

Clause pattern
(6) The girl pushed the boy.
(7) The girl who pushed the boy was tall.
(8) Show me the girl who pushed the boy.
(9) It is the girl who pushed the boy.
(10) The boy was interested in the girl.

Table 3: Clause patterns with chance-level performance, adopted and adjusted from the work of Grodzinsky (2000, p. 5).

Clause pattern
(11) The boy was pushed by the girl.
(12) The boy who the girl pushed was tall.
(13) Show me the boy who the girl pushed.
(14) It is the boy who the girl pushed.

or can be derived from the strings underlying one or more kernel sentences by a sequence of one or more transformation (Chomsky, 1957, p. 45). The kernel consists of simple, declarative, active sentences that reflect basic grammatical relations such as subject-predicate or verb-object, i.e., the terminal strings underlying the kernel sentences are derived by a simple system of phrase structure (Chomsky, 1957, pp. 61, 80).

(iv) Broca's aphasics have difficulties in comprehending sentences with syntactic movement. Syntactic movement is a grammatical transformation in which a sentence constituent is pronounced in a different position than the one in which it is generated (Van der Meulen, 2004, p. 10). For example, passive sentences "the boy was pushed by the girl" (11) is derived from its active counterpart "the girl pushed the boy." (6) through NP-movement of the object, as illustrated in Fig. 1 (Van der Meulen, 2004, pp. 22-23). This grammatical transformation restrains Broca's aphasics' ability to comprehend the given passive sentence. It should be noted that comprehension difficulties are related to syntactic movement, and not to the passive morphology. Sentence (11) is a verbal passive sentence and, thus, includes NP-movement. In contrast to this, sentence "the boy was interested in the girl" (10) is adjectival passive and does not include syntactic movement. Therefore, Broca's aphasics are able to comprehend the latter sentence.

Still, not every syntactic movement equally affects comprehension ability of Broca's aphasics. Subject relative sentence "show me the girl who pushed the boy" (8) contains syntactic movement of the subject (cf. Fig. 2), while object

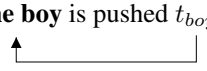


Figure 1: Verbal passive sentence derived through NP-movement of the object (cf. Van der Meulen (2004, p. 23)).

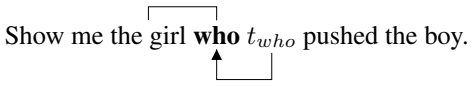


Figure 2: Subject relative sentence derived through *wh*-movement of the subject (cf. Van der Meulen (2004, p. 24)).

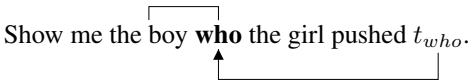


Figure 3: Object relative sentence derived through *wh*-movement of the object (cf. Van der Meulen (2004, p. 24)).

relative sentence "show me the boy who the girl pushed" (13) contains syntactic movement of the object (cf. Fig. 3). The experimental record shows that Broca's aphasics can comprehend the first sentence, but not the latter. A similar observation holds for subject cleft (9) and object cleft sentences (14). This subject/object asymmetry may be summarized as follows—Broca's aphasics have intact comprehension of sentences involving movement out of the subject position, and impaired comprehension of sentences involving movement out of the object position (Van der Meulen, 2004, pp. 24-25).

3. Basic requirements

Based on the discussion from the previous section, we introduce three basic requirements for developing a model of the comprehension deficit in Broca's aphasia:

- *Performance requirement:* The model should interpret the following sentences at above-chance level of performance: basic phrase syntactic structures, sentences containing syntactic movement out of the subject position (cf. Table 2), and semantically irreversible sentences that contain syntactic movement out of the object position (e.g., "the apple was eaten by the boy"). The model should interpret semantically reversible sentences containing syntactic movement out of the object position (cf. Table 3) at chance-level of performance.

- *Methodological requirement:* The model should rely on use of the lexicon and semantic cues (i.e., the general knowledge of the world), rather than on syntactic analysis.

- *Parsimony requirement:* The model should be as simple as possible (cf. Mirman et al. (2011, p. 61)). In this particular case, it means that the model should give an economical account of the comprehension deficit in Broca's aphasia.

Here, we use the verb "to interpret" in a restricted scope. It refers to identifying two fundamental semantic relationships of sentence constituents, i.e., the semantic roles of Agent and Theme. In the previous discussion, we stated that Broca's aphasics have difficulties in comprehending semantically reversible sentences that contain

syntactic movement out of the object position. In contrast to this, Kay (2000) notes that Grodzinsky's comprehension deficit data (cf. Tables 2 and 3) can be more economically accounted for without reference to movement, with traditional grammatical concepts that are less theory-internal and more empirically based. His point of departure is that in the canonical clause of English language (i.e., simple, active and declarative clause like clause (6) in Table 2), the subject comes first, followed by the verb and object. Therefore, the interpretive strategy employed by English-speaking Broca's aphasics may be formulated as follows—a logical subject precedes its coarguments. Following Kay, the chance-level comprehension by the Broca's aphasics occurs if and only if a clause constituent that carries the Theme role precedes a clause constituent that carries the Agent role. This rather simple rule appears to be appropriate for predicting whether a given clause should be interpreted at chance or above-chance level. However, identifying semantic roles would require some sort of syntactic analysis (cf. Gildea and Jurafsky (2002)), which violates the methodological requirement. In the following section, we discuss that the focus tree is appropriate for modeling the considered aspects of the comprehension deficit in Broca's aphasia, while still satisfying the introduced requirements.

4. Applying the focus tree

The focus tree model was primarily introduced to address the research question of robust automatic processing of different syntactic forms of spontaneously uttered users' commands with no explicit syntactic expectations (Gnjatović et al., 2011). This model was implemented within several prototypical dialogue systems with diverse interaction domains, including: solving problems in a graphics system (Gnjatović and Rösner, 2008; Gnjatović and Rösner, 2007), retrieving textual contents from web sites over the telephone line (Gnjatović et al., 2011), identifying the semantic entities of Figure and Ground in a spatial context (Gnjatović and Delić, in press), etc. These implementations demonstrated that the focus tree model enables the system to handle flexible mapping relations between the spontaneously produced user's commands and the system's actions, including processing of ellipses, context-dependent commands, constituent negations, anaphora, nonverbal commands, pauses in the conversation, etc. Here, we discuss only those aspects of the model that relate to the research question of modeling the considered aspects of the comprehension deficit in Broca's aphasia.

4.1. Knowledge representation

For the purpose of this discussion, let us assume that the knowledge of the world includes three animate entities (i.e., a boy, a girl, and a dog) and an inanimate entity (i.e., an apple). It also includes the following actions: the boy and girl can push each other, the boy can wash himself (in a reflexive sense) and he can wash the dog, the girl can eat the apple, and the dog can eat the apple. Thus, the animate entities may carry both the Agent and the Theme role, while the inanimate entity can carry only the Theme role. The focus tree that represents this restricted knowledge of the world is given in Fig. 4.

Table 4: Possible interpretations in the scope of the restricted knowledge of the world.

Interpretation (semantic role labeling)
I_1 — Agent: *the boy*; Theme: *the girl*; Action: *push*;
I_2 — Agent: *the boy*; Theme: *the boy*; Action: *wash*;
I_3 — Agent: *the boy*; Theme: *the dog*; Action: *wash*;
I_4 — Agent: *the girl*; Theme: *the boy*; Action: *push*;
I_5 — Agent: *the girl*; Theme: *the apple*; Action: *eat*;
I_6 — Agent: *the dog*; Theme: *the apple*; Action: *eat*;

The entities that may carry the Agent role are represented by nodes at the second level of the focus tree, the actions are represented by nodes at the third level, and the entities that could be assigned the Theme role are represented by nodes at the fourth level. Each direct path from the root node to a terminal node represents a possible interpretation. Thus, the given focus tree contains six possible interpretations in the scope of the restricted knowledge of the world. All encapsulated interpretations are described in Table 4. Although these descriptions are fairly self-explanatory, we note that the verb "wash" in interpretation I_2 is reflexive, e.g., as in "the boy washed in the river". The noun phrase "the boy" carries both the Agent and the Theme role. Therefore, this interpretation does not include a node at the fourth level.

For a given input sentence, the model should choose an interpretation. For example, I_4 interprets sentences (6)–(9) (cf. Table 2) and sentences (11)–(14) (cf. Table 2). However, according to the performance requirement, the model of the comprehension deficit should correctly interpret only sentences (6)–(9). For sentences (11)–(14), the model should randomly choose between two possible interpretations, I_1 and I_4, only one of which represents the correct interpretation. This is discussed in the next subsection.

4.2. Sentence processing

Since Broca's aphasics do not have impairment in their lexicon, the system based on the focus tree model is allowed to recognize and extract noun phrases (NP), verbs (V), and certain verb phrases (VP) that relate to entities and actions from the restricted knowledge of the world. For example, noun phrases "the boy" and "John" may be recognized as relating to the animate entity *boy*, while verb forms "pushes" and "pushed" may be recognized as relating to the action *push*. We refer to these extracted sentence chunks as to *focus stimuli*. In our approach, the lexicon comprises of preset focus stimuli (i.e., NP, V, VP). The general idea underlying the interpretation of a given sentence is that the system detects paths that include nodes relating to all extracted focus stimuli.

Still, extraction of verb phrases deserves additional explanation. We recall that Broca's aphasics have intact comprehension of sentences involving movement out of the subject position, and impaired comprehension of sentences involving movement out of the object position. In

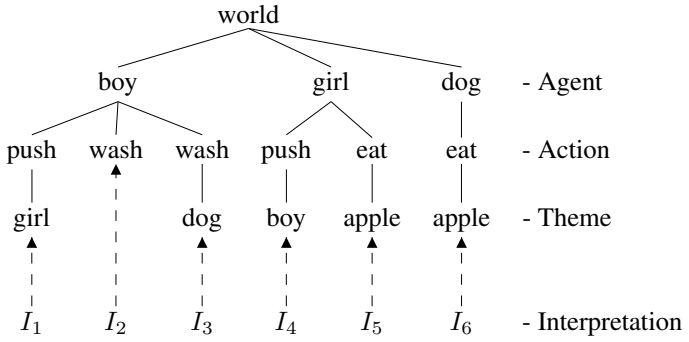


Figure 4: The focus tree representing the restricted knowledge of the world.

other words, movement out of the object position appears to be a critical syntactic transformation. On the other hand, Broca's aphasics can correctly interpret simple, active and declarative sentences, e.g., the canonical Subject-Verb-Object clauses of English language. Therefore, we introduce a more economical account of the comprehension deficit data that does not involve syntax analysis. At the level of surface (linguistic) expression, if the given sentence contains a *canonical* verb phrase VP $\rightarrow$ V NP, then NP involved in the verb phrase caries the Theme role, and the Agent role is assigned to other (if any) NP involved in the sentence. Otherwise, no assumption on semantic role labeling can be made based only on the surface elements. It should be noted, however, that extraction of such verb phrases is not a matter of syntactic analysis. These phrases are part of a preset lexicon, and their recognition is reduced to string matching.

We illustrate sentence processing with the following examples.

(i) Show me the girl that pushed the boy. The system extracts the following focus stimuli from the sentence: "the girl", "the boy", "pushed", and "pushed the boy". Since a canonical verb phase is recognized, the system concludes that the Theme role should be assigned to the entity *boy*. Consequently, the Agent role is assigned to the entity *girl*. Finally, the system unambiguously determines that I_4 is the interpretation of the given sentence, because it includes nodes that relate to all extracted focus stimuli, and the entity *boy* carries the Theme role. All sentences (6)–(9) are processed in the same manner.

(ii) The boy was pushed by the girl. The system extracts the following focus stimuli: "the girl", "the boy", and "pushed". Since no canonical verb phrase was identified, the system cannot assign semantic roles at this point. Instead, it tries to find possible interpretations that relate to the extracted focus stimuli. In the given focus tree, there are two possible interpretations that satisfy this condition: I_1 and I_4. The system randomly choose one of them. Therefore, the interpretation of this sentence results in a chance-level performance, as required for semantically reversible, verbal passive sentences. All sentences (11)–(14) are processed in the same manner.

(iii) The apple was eaten by the dog. In contrast to the previous case, this verbal passive sentence is semantically irreversible, and should be correctly interpreted by the system. The system extracts the following focus stimuli: "the apple", "the dog", and "eaten". Although no canonical verb phase was detected, and the system cannot make any assumptions on semantic role labeling based only on the surface elements, it can use semantic cues. Namely, in the given focus tree, there is only one interpretation, i.e., I_6, that contains nodes that relate to all extracted focus stimuli.

(iv) John washed in the river. This sentence contains only two focus stimuli: "John" and "washed", the first relating to the entity *boy*, and the second to the action *wash*. In the given focus tree, there are two interpretations, i.e., I_2 and I_3, that include nodes related to these stimuli. However, these interpretations differ in one point: all nodes contained in I_2 are related to some focus stimulus, while there is one node in I_3 (i.e., the terminal node *dog*) that does not relate to any focus stimuli. Therefore, the system select I_2 as more appropriate interpretation. It should be noted that if the given sentence contained any chunk that could relate to entity *dog* (e.g., the noun "dog", dog's name or an anaphoric reference to this entity), the system would select interpretation I_3.

(v) She ate the apple. The interpretation of this sentence depends on the knowledge of the world. If this knowledge includes a female dog, then pronoun "she" (which is also included in the lexicon) may refer to two entities; *girl* and *dog*. In this case, both interpretations I_5 and I_6 would be applicable. Otherwise, the system unambiguously determines that I_5 is the interpretation of the given sentence.

5. Discussion and conclusion

This paper presented a cognitively-inspired approach to computational modeling of the comprehension deficit in Broca's aphasia. We considered a neurolinguistics insight, and, based on this, introduced three basic requirements for an approach aimed at modeling this comprehension deficit. Then, we discussed that the focus tree is appropriate for modeling the most salient aspects of the comprehension deficit in Broca's aphasia.

The discussion in this paper was primarily focused on English language. The proposed model exploits, to some extent, the fact that fixed word order in English is used to indicate the Theme semantic relations. Still, it does not mean that the model is not applicable to free word-order languages, like Serbian. In Serbian, case is conveyed by noun-inflections (Lukatela et al., 1995, pp. 96–7). Case markers

and other agreement markers are used in comprehending relative clauses. For example, the English sentence "The girl pushed the boy" may be translated into two Serbian sentences having the same meaning but different word orders: "Девојчица$_{(nom)}$ је гурнула дечка$_{(accus)}$" and "Дечка$_{(accus)}$ је гурнула девојчица$_{(nom)}$". In order to enable the system to use these markers to label the semantic roles, two minor changes are required at the implementation level: the lexicon should be expanded to include inflected forms, and the inflected forms should be appropriately related to nodes in the focus tree. However, case and agreement markers are not always sufficient. For example, the Serbian sentence "Зец$_{(nom)}$ кога јури пас$_{(nom)}$ је браон" cannot be correctly interpreted only on the basis of markers. This is analogous to the case of the English sentence "The rabbit that the dog is chasing is brown" (i.e., English translation of the Serbian sentence) which cannot be correctly interpreted only on the basis of word order.

Finally, it should be noted that this approach is not intended to address all diverse aspects of the comprehension deficit in Broca'aphasia, but just the most salient aspects of this phenomenon. Understanding of mechanisms that underlay the interpretive strategy of Broca's aphasics to use semantic cues in order to get around their deficit and to correctly interpret the given input may provide a better insight into potentialities and limitations of semantic analysis in human-machine interaction.

6. Acknowledgments

The presented study is performed as part of the projects "Design of Robots as Assistive Technology for the Treatment of Children with Developmental Disorders" (III44008) and "Development of Dialogue Systems for Serbian and Other South Slavic Languages" (TR32035), funded by the Ministry of Education and Science of the Republic of Serbia. The responsibility for the content of this paper lies with the authors.

7. References

C. Bledowski, J. Kaiser, and B. Rahm. 2010. Basic operations in working memory: contributions from functional imaging studies. *Behavioural Brain Research*, 214(2):172–9.

A. Caramazza and E.B. Zurif. 1976. Dissociation of algorithmic and heuristic processes in language comprehension: Evidence from aphasia. *Brain and Language*, 3:572–582.

N. Chomsky. 1957. *Syntactic Structures*. Mouton, London.

A. D'Ausilio, L. Craighero, and L. Fadiga. 2010. The contribution of the frontal lobe to the perception of speech. *Journal of Neurolinguistics*.

N. F. Dronkers, O. Plaisant, M. T. Iba-Zizen, and E. A. Cabanis. 2007. Paul Broca's historic cases: high resolution MR imaging of the brains of Leborgne and Lelong. *Brain*, 130(5):1432–1441.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

M. Gnjatović and V. Delić. in press. Attention and linguistic encoding of motion events in human-machine interaction. In S. Halupka-Rešetar, M. Marković, T. Milićev, and N. Milićević, editors, *Selected papers from the 3rd International Conference of Syntax, Phonology and Language Analysis*. Cambridge Scholar Publishing.

M. Gnjatović and D. Rösner. 2007. An approach to processing of user's commands in human-machine interaction. In *Proceedings of the 3rd Language and Technology Conference (LTC'07)*, pages 152–156, Adam Mickiewicz University, Poznan, Poland.

M. Gnjatović and D. Rösner. 2008. Adaptive Dialogue Management in the NIMITEK Prototype System. In *Proceedings of the 4th IEEE Tutorial and Research Workshop Perception and Interactive Technologies for Speech-Based Systems (PIT'08)*, volume 5078 of *Lecture Notes in Computer Science*, pages 14–25, Kloster Irsee, Germany. Springer.

M. Gnjatović and D. Rösner. 2010. Inducing genuine emotions in simulated speech-based human-machine interaction: The nimitek corpus. *IEEE Transactions on Affective Computing*, 1:132–144.

M. Gnjatović, M. Janev, and V. Delić. 2011. Focus tree: Modeling attentional information in task-oriented human-machine interaction. *Applied Intelligence*. 10.1007/s10489-011-0329-5.

Y. Grodzinsky. 2000. The neurology of syntax: Language use without Broca's area. *Behavioral and Brain Sciences*, 23(01):1–21.

B.J. Grosz and C.L. Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204.

P. Kay. 2000. Comprehension deficits of Broca's aphasics provide no evidence for traces. *Behavioral and Brain Sciences*, 23(01):37–38.

K. Lukatela, D. Shankweiler, and S. Crain. 1995. Syntactic processing in agrammatic aphasia by speakers of a Slavic language. *Brain & Language*, 49(1):95–115.

D. Mirman, E. Yee, S. Blumstein, and J.S. Magnuson. 2011. Theories of spoken word recognition deficits in aphasia: Evidence from eye-tracking and computational modeling. *Brain & Language*, 117:53–68.

K. Oberauer and E.B. Lange. 2009. Activation and binding in verbal working memory: A dual-process model for the recognition of nonwords. *Cognitive Psychology*, 58(1):102136.

L. Obler and K. Gjerlow. 1999. *Language and the Brain*. Cambridge University Press.

A.C. Van der Meulen. 2004. *Syntactic movement and comprehension deficits in Broca's aphasia*. Ph.D. thesis, Netherlands Graduate School of Linguistics, Utrecht, Netherlands.

Redundant Information Reduction in FST-Based Pronunciation Lexicon Compression

Žiga Golob*, Uliana Dorofeeva*, Jerneja Žganec Gros*, Milena Gros†, Simon Dobrišek†

*Alpineon d.o.o.
Ljubljana, Slovenija
{ziga.golob}@alpineon.si
†Faculty of Electrical Engineering
University of Ljubljana, Slovenia
simon.dobrisek@fe.uni-lj.si

Abstract

Finite-state transducers are frequently used for pronunciation lexicon representations in speech engines, in which memory and processing resources are scarce. This paper proposes a method for further reducing the memory footprint of finite-state transducers representing pronunciation lexicons. A combination of grapheme-to-allophone rules with a finite-state transducer is proposed, which yields a 65% smaller finite-state transducer than conventional approaches.

Zmanjševanje odvečnosti informacije pri kompaktnem zapisu slovarjev izgovorjav s pomočjo končnih pretvornikov

Končni pretvorniki se pogosto uporabljajo za predstavitev slovarjev izgovorjav v sintetizatorjih govora, v katerih je na voljo omejena količina pomnilnika ter procesorske moči. V tem članku je predstavljena metoda za nadaljnjo zmanjševanje potrebne količine pomnilnika za predstavitev slovarja izgovorjav s pomočjo končnega pretvornika. Predlagana je uporaba kombinacije grafemsko-alofonskih pravil ter končnih pretvornikov, kar omogoča izgradnjo 65% manjših končnih pretvornikov kot s pomočjo klasičnih postopkov.

1. Introduction

Consistent and accurate determination of word pronunciation is critical to the success of many speech technology applications. Most state-of-the-art speech engines performing automatic speech recognition (ASR) and text-to-speech synthesis (TTS) rely on lexicons, which contain pronunciation information for many words. To provide maximum coverage of the words, multiword expressions, or even phrases that commonly occur in a given application domain, application-specific words, or phrase pronunciations may be required, especially for application-specific proper nouns such as personal names or names of locations.

Pronunciation lexicons for speech engines contain grapheme and allophone transcription of lexical words (Šef et al., 2004). The "x-sampa-SI-reduced" phonetic alphabet, a subset of the X-SAMPA set as defined for Slovenian (Zemljak et al., 2002), is used in allophone transcriptions. An example of a pronunciation lexicon for a few Slovenian words is shown in Figure 1.

```
...
opera       o:pEra
operah      o:pErah
operam      o:pEram
operama     o:pErama
operami     o:pErami
opere       o:pErE
operi       o:pEri
...
```

Figure 1. An excerpt from a Slovenian pronunciation lexicon.

The storage and run-time processing of pronunciation lexicons is memory consuming, especially for highly inflected languages, where pronunciation lexicons typically contain over one million lexical items. In some systems with limited memory resources—for example, in speech engines for embedded systems or multilingual speech engines—using large pronunciation lexicons is not feasible. In addition, the search time in such lexicons may be long if the lexicons are too large to be stored in the main memory or cache.

Therefore, *memory-efficient representations* of pronunciation lexicons enabling fast lookup are mandatory in order to address these limitations. Another disadvantage of inefficient lexicon representation is the unnecessary use of system resources.

State-of-the-art pronunciation lexicon representation techniques used in speech engines are based on structures called finite-state automata (FSA) as in (Daciuk, 2011) and finite-state transducers (FSTs) (Dobrišek et al., 2010; Rojc et al., 2007); very similar structures are also called tries (Ristov, 2005).

This paper discusses new possibilities for reducing the size of pronunciation lexicon representation using FSTs. We report encouraging results that were obtained by removing redundant information from the allophone transcription prior to building the FST.

2. FST representations of pronunciation lexicons

A FST differs from FSA in that when it accepts a symbol it also outputs another symbol. In this way it can translate an input string into an output string. An example of a FST representing a simple pronunciation lexicon from Table 1 is shown in Figure 2.

GRAPHEMES	ALLOPHONES
hiša	hi:Sa
hišo	hi:SO
hiter	hi:t@r

Table 1. Pronunciation lexicon with three lexical items.

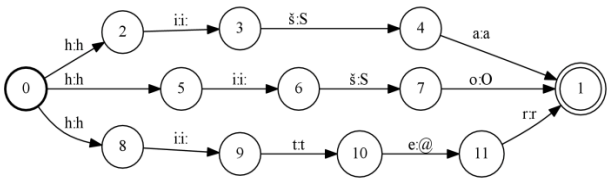


Figure 2. A FST representing a simple pronunciation lexicon from the example in Table 1. By convention, the states are represented as circles and marked with their unique number. The initial state is represented by a bold circle and final states by double circles. An input label i and an output label o are marked on the corresponding directed arc as i : o.

At the beginning, the FST is in its initial state. For every symbol in the input string, the FST changes its state according to the transition function and emits an output symbol. If it moves to a final state when it accepts all the symbols of the input string, we say that it has *accepted* the input string. At that moment the output string, which is composed of all the emitted symbols, becomes valid.

Many efficient algorithms have been developed for FSTs (Cyril et al., 2007), such as union, concatenation, intersection, determinization, minimization, and so on. When using minimization and determinization, the FST becomes a very convenient representation of pronunciation lexicons (Mohri, 1994a; Dobrišek et al., 2009; Rojc et al., 2011). If one excludes all heteronyms (words with the same spelling but different pronunciations), all acyclic FSTs representing pronunciation lexicons can be determinized (Mohri, 1996), and therefore acyclic FSTs are frequently used for pronunciation lexicon representations in speech engines. For representing heteronyms, p-subsequential FSTs can be used (Cyril et al., 2002). For deterministic FSTs, the existence of a minimal FST has been proven(Mohri, 1994b). Hereinafter we denote a minimized and determinized FST as MDFST. Minimization of a FST transforms the FST into an equivalent FST with a minimal number of states. A MDFST also exhibits a minimal number of transitions (Mohri, 1997). The two deterministic FSTs are said to be equivalent if for every sequence of input symbols they generate the same sequence of output symbols.

Table 2 shows the reduction of the number of states by minimization and determinization for the SI-Pron Slovenian pronunciation dictionary containing 1,239,401 lexical items.

TYPE	STATES	TRANSITIONS
FST	11,404,858	12,644,257
MDFST	217,300	517,225

Table 2. Comparison of the number of states and transitions of FST and MDFST representing the SI-Pron pronunciation lexicon.

The FST in Figure 2 shows that there are three possible transitions from the initial state with the same input symbols. In a deterministic FST there is no state with more than one transition with the same input symbol. The advantage of a deterministic FST is lookup speed, which is linearly dependent only on the length of the input string and not dependent on the size of the FST.

In Figure 3 an equivalent FST to the one from Figure 2 is shown, which has been both determinized and minimized (MDFST).

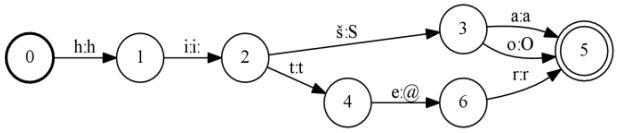


Figure 3. Minimized and determinized FST from Figure 2: MDFST.

3. Language resources

This paper reports results for Slovenian pronunciation lexicon SI-Pron containing 1,239,410 lexical entries (Žganec Gros et al., 2006).

The lexicon contains grapheme and allophone transcriptions for all lexical entries. It also contains additional information such as syllabification, morphological information, and stress positions. Table 3 shows some statistical properties of the lexicon.

LEXICON	WORDS	GRAPHEMES	ALLOPHONES
SI-Pron	1,239k	12,644k	12,713k

Table 3. Statistical properties of the three chosen pronunciation lexicons.

All homographs (words with the same spelling but different meaning) were removed from the lexicon.

4. Experiments and results

It has been reported that for Slovenian it is possible to achieve a grapheme-to-allophone transcription accuracy of 99.1% by using a set of context-dependent rules if the stress position and the transcription of the graphemes *e* and *o* in stressed syllables are known in advance (Gros, 1999).

The idea of the following experiment is to remove all unnecessary information from the pronunciation lexicon and to model the information left with a simpler and possibly lighter FST.

The potential of this approach becomes clearer if one recalls the basic principle of FST minimization. When minimizing a FST, equivalent states are merged. Equivalent states are those that have transitions with the same input and output symbols targeting the same or equivalent states.

According to (Gros, 1999), the necessary information for reconstructing the allophone transcription from the grapheme transcription in Slovenian is the lexical stress position and the transcriptions of the graphemes *e* and *o* in stressed syllables, which we were able to model with only three different symbols in the FST output alphabet. This highly reduced number of different output symbols offers more possibilities for the state transitions to have equal input and output symbols and a greater chance for the possible states to merge.

In this experiment we built a FST representing the pronunciation lexicon emitting information containing only the stress position and the transcription variation of graphemes *e* and *o* in stressed syllables.

In Slovenian, when words of foreign origin are included, the six graphemic vowels *a*, *e*, *i*, *o*, *u*, and *y* along with the reduced vowel schwa @ can be stressed. A lexical item can have multiple lexical stress positions.

The lexical stress position can be modeled implicitly with a FST if, for every accepted grapheme, it emits information on whether it is stressed or unstressed. The reduced vowel @ is not part of the grapheme symbol set. It appears in allophone transcriptions before the consonant *r* when preceded and followed by a consonant (e.g., *potrt* → *pOt@"rt*, *prvi* → *p@"rvi*). Therefore, if the reduced vowel @ is stressed, the FST outputs the information about stress when it accepts the consonant *r*.

The allophone transcriptions of the stressed vowels *e* and *o* can be either close *e:* and *o:* or open *E:* and *O:*. To model this information, for every accepted vowel *e* or *o*, the FST has to emit information on whether the vowel is unstressed, stressed open, or stressed close. Therefore, there are three possible output symbols.

Table 4 shows the possible output symbols of the FST if we group both the stress and the transcription of the graphemes *e* and *o* into one model. Table 5 shows an alternative grouping.

FST INPUT SYMBOL	FST OUTPUT SYMBOL
unstressed grapheme	0
stressed vowel *a, i, u, y*; consonant *r* following a stressed reduced vowel @; stressed vowels *e* and *o* with open transcription	1
stressed vowels *e* and *o* with close transcription	2

Table 4. Modeling the FST information output for stress position and the grapheme e and o transcription. Stressed vowels e and o with open transcriptions are grouped with other stressed vowels.

FST INPUT SYMBOL	FST OUTPUT SYMBOL
unstressed grapheme	0
stressed vowel *a, i, u, y*; consonant *r* following a stressed reduced vowel @; stressed vowels *e* and *o* with close transcription	1
stressed vowels *e* and *o* with open transcription	2

Table 5. Modeling the FST information output for stress position and the grapheme e and o transcription. Stressed vowels e and o with close transcriptions are grouped with other stressed vowels.

Tables 4 and 5 represent two possible mappings between FST input and output symbols. Table 6 shows a few examples of FST input and output strings derived from the mapping presented in Table 4 for a few Slovenian words.

The first two temporary lexicons were constructed for the SI-Pron pronunciation lexicon based on the mappings from Tables 4 and 5. These two lexicons represented the alignments of the symbols (graphemes) in input strings and symbols (numbers) in output strings. Then we built the MDFST from these temporary lexicons using the OpenFST tool.

FST INPUT STRING	FST OUTPUT STRING	ALLOPHONE TRANSCRIPTION
medved	010000	mE:dvEd
prvi	0100	p@"rvi
roža	0200	ro:Za

Table 6. FST output strings for three Slovenian words (*medved, prvi, roža*) and their complete allophone transcription in the last column.

Table 7 shows the comparison between different approaches. The results in Table 7 show a considerable 65% reduction in the number of states (using the mapping from Table 5) compared to MDFST storing the complete allophone transcription.

We also compared the sizes (in MB) of different data structures representing the information stored in pronunciation lexicons. We disregarded the sizes of program code that are necessary to manipulate the data structures because, if implemented properly, they are negligible in size in comparison to the data structures.

ALGORITHM	STATES	TRANSITIONS
FST	11,404,858	12,644,257
MDFST	217,300	517,225
MDFST + information reduction (table 7 mapping)	78,329	246,674
MDFST + information reduction (table 8 mapping)	76,846	242,851
FSA	49,741	155,988

Table 7. Comparison between different approaches to representing the information in the SI-Pron pronunciation lexicon with FST. The finite-state acceptor (FSA) stores the information when the specific input string is valid. It represents the lower bound of the number of states of the FST with the same input alphabet and accepting the same language.

The SI-Pron lexicon as UTF8 encoded text represents the baseline and 100% size. We used the OpenFST tool to build a MDFST representing the complete lexicon information and a MDFST representing only the necessary information for rule-based grapheme-to-allophone conversion.

We also implemented our own more compact representation of the FST similar to the implementation for finite state automata found in (Daciuk, 2011).

It is interesting to compare the lexicon reduction techniques used to standard methods used in text compression, such as zip, even though standard text compression methods are not useful for solving our problem because their data have to be decompressed completely to their full size before they can be used. The results are shown in Table 8.

In its compact form, the MDFST structure representing the reduced information of the SI-Pron pronunciation lexicon is over 40 times smaller than the original UTF8-

encoded text representation as seen in Table 8. It is also three times smaller than the MDFST representing the complete allophone transcription.

	Size [kB]	Size [%]
SI-Pron (UTF8 encoded text)	30,657	100
Compressed SI-Pron (zip)	6,071	19.8
MDFST (OpenFST)	9,948	32.4
MDFST (compact representation)	2,287	7.7
MDFST + information reduction: Table 4 mapping (OpenFST)	4,084	13.3
MDFST + information reduction: Table 4 mapping (compact representation)	708	2.3

Table 8. Size of data structures representing information in the SI-Pron pronunciation lexicon.

5. Conclusion

Finite-state transducers are frequently used for pronunciation lexicon representations in speech engines, in which memory and processing resources are scarce.

A method for further memory footprint reduction of finite-state transducers representing pronunciation lexicons was proposed in the paper. A combination of grapheme-to-allophone rules with a FST yielded a 65% smaller finite-state transducer than conventional approaches.

All the information that can be reconstructed with a set of context-dependent rules was removed from the allophone transcription in the Slovenian pronunciation lexicon. By building the MDFST for the new lexicon, we succeeded in significantly reducing the number of states by 65% and in achieving an implementation over three times smaller.

The proposed method can be used for efficiently representing Slovenian pronunciation lexicon resources; the use of a similar principle could also be considered for other languages with similar pronunciation properties.

6. Acknowledgements

The research work by the first author was partially financed by the European Union, European Social Fund, the framework of the Operational Programme for Human Resources Development for the Period 2007–2013 under contract no. P-MR-10/94.

7. References

Cyril A., Mohri M., 2002. p-Subsequentiable Transducers. Proceedings of the Seventh International Conference on Implementation and Application of Automata (CIAA 2002), Tours, France. pp. 24–34.

Cyril A., Michael R., Johan S., Wojciech S., Mohri M., 2007. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. Proceedings of the 12th International Conference on Implementation and Application of Automata (CIAA 2007). Lecture Notes in Computer Science, Prague, Springer-Verlag, Heidelberg, Germany, 4783: 11-23.

Daciuk J., 2011. Smaller Representation of Finite State Automata. Proceedings of the 16th International Conference on Implementation and Application of Automata, pp. 118–129.

Dobrišek S., Vesnicer B., Mihelič F., 2009. A Sequential Minimization Algorithm for Finite-State Pronunciation Lexicon Models. Proceedings of Interspeech 2009, International Speech Communication Association, Brighton, UK, pp. 720–723.

Dobrišek S., Žibert J., Mihelič F., 2010. Towards the Optimal Minimization of a Pronunciation Dictionary Model. Petr Sojka, Ales Horak, Ivan Kopecek and Karel Pala (Eds.). TSD-2010. Lecture Notes in Computer Science, Brno, Springer, pp. 267–274.

Gros J., Mihelič F., 1999. Acquisition of an Extensive Rule Set for Slovene Grapheme-to-Allophone Transcription. Proceedings 6th European Conference on Speech Communication and Technology. September 5–9. 1999. Eurospeech 1999. Budapest, 5: 2075–2078.

Mohri M., 1994a. Compact Representations by Finite-State Transducers. 32nd Meeting of the Association for Computational Linguistics (ACL '94). Proceedings of the Conference. Las Cruces. NM, pp. 204–209.

Mohri M., 1994b. Minimization of Sequential Transducers. Proceedings of the 5th Annual Symposium on Combinatorial Pattern Matching (CPM '94), Maxime Crochemore and Dan Gusfield (Eds.). Vol. 807 of *Lecture Notes in Computer Science*, Asilomar, CA, Springer-Verlag, Berlin, pp. 151–163.

Mohri M., 1996. On Some Applications of Finite-State Automata Theory to Natural Language Processing. *Journal of Natural Language Engineering*, 2: 61–80.

Mohri M., 1997. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 33: 269–311.

Ristov S., 2005. LZ Trie and Dictionary Compression. Jurnual Software-Practice & Experience, pp. 445–465.

Rojc M., Kačič Z., 2007. Time and Space-Efficient Architecture for a Corpus-Based Text-to-Speech Synthesis System. Speech Communication, 49: 230–249.

Rojc M., Mlakar I., 2011. Multilingual and Multimodal Corpus-Based Text-to-Speech System – PLATTOS. Ipšić Ivo (Ed.). Speech and Language Technologies, InTech, Available from: http://www.intechopen.com/books/speech-and-language-technologies/multilingual-and-multimodal-corpus-based-text-to-speech-system-plattos. Accessed 2012 June 6.

Šef T., Gams M., 2004. Data mining for creating accentuation rules, Applied. Artificial Intelligence, vol. 17, pp. 395-410.

Zemljak M., Kačič Z., Dobrišek S, Gros J, Weiss P., 2002, Računalniški simbolni fonetični zapis slovenskega govora. Slavistična revija, 50: 159–169.

Žganec-Gros J., Cvetko-Oresnik V., Jakopin P., 2006. SI-Pron Pronunciation Lexicon: A New Language Resource for Slovenian. Informatica, 30: 447–452.

Obeliks:
statistični oblikoskladenjski označevalnik in lematizator za slovenski jezik

Miha Grčar,[1] Simon Krek,[2] Kaja Dobrovoljc[3]

[1, 2] Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
miha.grcar@ijs.si, simon.krek@ijs.si
[3] Trojina, zavod za uporabno slovenistiko
Partizanska cesta 5, 4220 Škofja Loka
kaja.dobrovoljc@trojina.si

Povzetek

V prispevku opisujemo označevalnik za slovenščino Obeliks, ki je bil izdelan v okviru projekta Sporazumevanje v slovenskem jeziku. Označevalnik je sestavljen iz treh komponent: tokenizacijskega modula, ki za stavčno segmentacijo in tokenizacijo uporablja pravila, oblikoskladenjskega označevalnika ter različice lematizatorja LemmaGen, ki je prilagojena za delovanje v kombinaciji z označevalnikom. Učno množico pri opisani različici označevalnika predstavlja učni korpus ssj500k, ki je označen po tabeli oznak JOS. Pri sistemu JOS s 1.903 možnimi oznakami je natančnost označevalnika 91,34 % za celotno oznako in 98,30 % za vrhnjo kategorijo (POS). Natančnost lematizacije je 97,88 % ob upoštevanju velike začetnice ter 98,55 % na ravni črkovnega niza. V prispevku predstavljamo zasnovo označevalnika ter analizo natančnosti označevanja. Označevalnik je prosto dostopen na spletu.

Obeliks: Statistical Morphosyntactic Tagger and Lemmatizer for Slovene

The paper describes Obeliks, a new statistical tagger for Slovene developed within the "Communication in Slovene" project. The new tool consists of three modules: a rule-based sentence splitter and tokenizer, a morphosyntactic tagger, and a version of the LemmaGen lemmatizer which works in combination with the tagger. Obeliks is trained on the ssj500k corpus tagged according to the JOS tagset. In the JOS system which includes 1,903 possible tags, the tagger achieved 91.34% accuracy for all tags and 98.30% for POS only. Lemmatization accuracy is 97.88% with capitalization included and 98.55% for all-lowercase letters. The paper presents the design of the tagger and the analysis of the tagging accuracy. Obeliks is freely available for download on the Web.

1. Uvod

V prispevku opisujemo statistični označevalnik, ki je nastal v okviru projekta Sporazumevanje v slovenskem jeziku.[1] Označevalnik implementira programska oprema Obeliks in zajema (1) segmentacijo in tokenizacijo, (2) oblikoskladenjsko označevanje in (3) lematizacijo.

Segmentacija je proces označevanja začetkov in koncev delov besedil po izbranem kriteriju. V skladu s tradicijo, ki se opira na tipično organizacijo delov besedil pri jezikih z alfabetsko pisavo, ki za označevanje smiselno zaključenih delov besedil uporabljajo ločila (npr. latinična in cirilična), se kot najpomembnejši segment pojavlja stavek, tipično ločen s piko, klicajem ali vprašajem, v nekaterih primerih tudi s tropičjem, pomišljajem ali na drug, manj običajen način.

Tokenizacija je postopek prepoznavanja in določanja posamezih korpusnih pojavnic, ki se na splošno delijo v dve večji skupini. Prva skupina so ločila in simboli, v drugo skupino pa spadajo elementi, ki so zanimivi za kasnejše jezikoslovno označevanje in jih pogosto opredeljujemo z oznako "beseda", čeprav mednje spadajo tudi števila, spletni naslovi ali kombinacije ločil, števk in črk (npr. Micro$oft, U2, AC/DC itd.), ki jih na prvi ravni morda niti ne bi prepoznali pod tem imenom.

Pripisovanje oblikoskladenjskih oznak oz. oblikoskladenjsko označevanje (*POS-tagging*, *part-of-speech tagging*, *word-class tagging*) je ena od najstarejših in najpogostejših oblik dodajanja interpretativnih informacij jezikoslovne narave besedilom, pri čemer posamezni pojavnici v korpusu pripišemo, v kateri osnovni besednovrstni razred spada v specifičnem jezku ter lastnosti, ki jih izkazuje znotraj razreda.

Pri oblikoskladenjskem označevanju je pomembno, kateri model označevanja izberemo; ta je navadno opredeljen s tabelo oznak (*tag set*), ki vsebujejo različno število možnih oznak. Za slovenščino obstaja več tabel oznak; za učenje označevalnika Obeliks je bila izbrana tabela oznak JOS (Erjavec et al., 2010), ki vsebuje 1.903 možne oznake z dvanajstimi vrhnjimi kategorijami.[2] To število je zelo veliko in zato predstavlja zahteven problem za statistične označevalnike.

Zadnja faza v procesu označevanja je lematizacija, ki je jezikovnotehnološki proces pripisovanja osnovne oblike korpusnim pojavnicam pri tistih besednih vrstah, ki so pregibne in tvorijo oblikoslovno paradigmo.

Označevalnik Obeliks procese segmentacije, tokenizacije, oblikoskladenjskega označevanja in lematizacije združuje v enoten proces, ki ga podrobneje opisujemo v naslednjem poglavju.

2. Postopek označevanja

V naslednjih podpoglavjih opisujemo osnovne postopke, tj. segmentacijo in tokenizacijo, oblikoskladenjsko označevanje ter lematizacijo, s poudarkom na oblikoskladenjskem označevanju, ki tvori jedro označevalnika in je osrednja tema tega prispevka.

2.1. Segmentacija in tokenizacija

Segmentacija in tokenizacija v označevalniku Obeliks temeljita na pravilih, ki so bila opredeljena v samostojnem tokenizacijskem programu, opisanem v (Krek, 2010), in so nekoliko prilagojena za konsistentno uporabo v sklopu tokenizacija-oblikoskladenjsko označevanje-lematizacija.

[1] http://www.slovenscina.eu

[2] http://nl.ijs.si/jos/msd/html-sl/index.html.

Izhodišče segmentacijsko-tokenizacijskih načel je slediti čim bolj intuitivnim in razumljivim pravilom, da se izognemo presenečenjem in nerazumevanju pri kasnejših obdelavah tokeniziranih besedil, kar velja predvsem za človeške uporabnike označenih besedil. Kot daleč najbolj problematični ločili se pri tem procesu izkazujeta vezaj in pika.

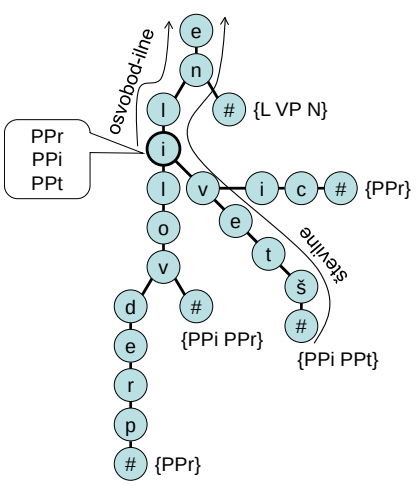
2.2. Oblikoskladenjsko označevanje

Delovanje označevalnika – kot sestavne komponente Obeliksa – temelji na principu nadzorovanega strojnega učenja (angl. *supervised machine learning*) (Mitchell, 1997), kar pomeni, da se označevanja nauči na ročno označenih primerih iz učnega korpusa. Označevalnik v fazi učenja zgradi model znanja, ki mu nato omogoča označevanje novih (še neoznačenih) besed. Osnovne komponente označevalnika so: (i) drevo končnic (*suffix trie*), (ii) algoritem za tvorjenje vektorjev značilk (*feature vectors*), (iii) algoritem za učenje in (iv) algoritem za označevanje. Te koncepte in postopke podrobneje opisujemo v naslednjih podpoglavjih.

2.2.1. Drevo končnic

Drevo končnic je ena poglavitnih komponent označevalnika. Zgrajeno je na podlagi učnega korpusa in leksikona ter "zna" za vsako besedo določiti nabor potencialnih oznak. Glavna prednost drevesa končnic pred običajnim leksikonom je, da lahko deluje tudi pri besedah, ki jih v drevesu ni – v takem primeru algoritem tvori nabor potencialnih oznak na podlagi končnice.

Kot primer navajamo drevo končnic (slika 1), ki vsebuje besede *predvolilne*, *volilne*, *ne*, *številne* in *civilne* (zapisane so od zadnje do prve črke in zaključene z znakom #, od korena drevesa proti listom). Vozlišče z zaključnim znakom vsebuje nabor oznak, ki jih je mogoče pripisati besedi (npr. besedi *ne* lahko pripišemo eno od oznak iz množice {L, VP, N}). Četudi besede ni v drevesu (npr. *osvobodilne*), lahko na podlagi končnice (v tem primeru *-ilne*) oz. pripadajočega vozlišča (označeno na sliki) sklepamo na oblikoslovne oznake, tako da izračunamo unijo vseh množic oznak v poddrevesu s korenom v vozlišču.



Slika 1: Preprost primer drevesa končnic.

kontekst (besede)	tri predhodne besede (w_{-3}, w_{-2}, w_{-1}), trenutna beseda (w_0) in tri sledeče besede (w_{+1}, w_{+2}, w_{+3})
kontekst (oblikoslovne oznake in množice oznak iz drevesa končnic)	tri predhodne oblikoslovne oznake (p_{-3}, p_{-2}, p_{-1}), množice oznak iz drevesa končnic za trenutno besedo in sledeče besede (a_0, a_{+1}, a_{+2}, a_{+3}), posamezne oznake iz množic a (m_0, m_{+1}, m_{+2}, m_{+3})
predpone in končnice	predpone in končnice dolžine od 1 do 4 za tri predhodne besede ($p_{pos,len}$, $s_{pos,len}$, $-3 \le pos \le -1$, $1 \le len \le 4$), trenutno besedo ($p_{0,len}$, $s_{0,len}$, $1 \le len \le 4$) in tri sledeče besede ($p_{pos,len}$, $s_{pos,len}$, $1 \le pos \le 3$, $1 \le len \le 4$)
črkovne in ostale značilke	- Ali beseda vsebuje znak, ki ni črka ali številka? - Ali beseda vsebuje številko? - Ali beseda vsebuje veliko črko? - Ali se beseda začenja z veliko začetnico [in ni prva beseda v stavku]?

Tabela 1: Značilke za oblikoslovno označevanje z uporabo strojnega učenja.

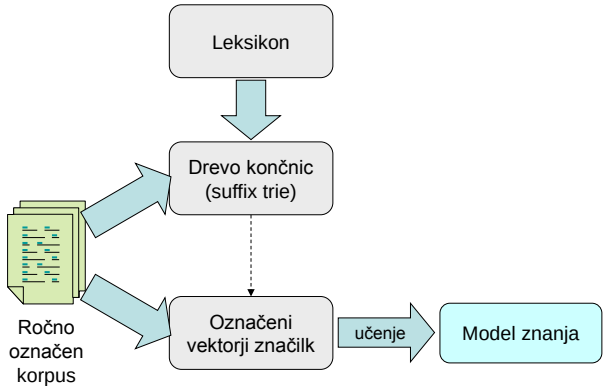
2.2.2. Vektorji značilk

Vsak primer (tj. beseda) je, kot je to navada v strojnem učenju, opisan z vektorjem značilk. Značilke in njihove vrednosti lahko dojemamo kot lastnosti besede, ki pa morajo biti takega tipa, da samostojno ali v kombinaciji z drugimi lastnostmi določajo neko specifiko oblikoslovne oznake, ki se pritiče besedi. Obeliks za značilke uporablja (i) kontekst besede (tj. predhodne in sledeče besede), (ii) predhodne oblikoslovne oznake, (iii) oznake iz drevesa končnic za trenutno in sledeče besede, (iv) predpone in končnice ter (vi) črkovne in ostale značilke, kot je to razvidno iz tabele 1. Značilke so povzete po (Giménez & Márquez, 2004), kjer so tudi bolj podrobno obrazložene. Primer tvorjenja vektorja značilk podaja tabela 2.

L D P ? Še v najboljših časih je redko delovalo.	
w_{-3}=še	$p_{-1,3}$=naj
w_{-2}=v	...
w_{-1}=najboljših	$p_{0,1}$=č
…	$p_{0,2}$=ča
w_{+3}=delovalo	…
p_{-3}=L	$p_{+1,1}$=j
p_{-2}=D	$p_{+1,2}$=je
p_{-1}=P	…
a_0={S}	$p_{+3,1}$=d
a_{+1}={G, Z}	$p_{+3,2}$=de
a_{+2}={R}	…
a_{+3}={P, S, G, …}	$s_{-3,1}$=e
m_0=S	$s_{-3,2}$=še
m_{+1}=G	$s_{-2,1}$=v
m_{+1}=Z	$s_{-1,1}$=h
…	$s_{-1,2}$=ih
m_{+3}=P	…

m_{+3}=S	$s_{0,1}$=h
m_{+3}=G	$s_{0,2}$=ih
…	…
$p_{-3,1}$=š	$s_{+1,1}$=e
$p_{-3,2}$=še	$s_{+1,2}$=je
$p_{-2,1}$=v	…
$p_{-1,1}$=n	$s_{+3,1}$=o
$p_{-1,2}$=na	$s_{+3,2}$=lo
vsebuje znak, ki ni črka ali številka = ne	
vsebuje številko = ne	
vsebuje veliko črko = ne	
se začenja z veliko začetnico = ne	

Tabela 2: Vrednosti značilk za besedo *časih* v stavku *Še v najboljših časih je redko delovalo*.



Slika 2: Učenje označevalnika: (1) iz ročno označenega korpusa in leksikona zgradimo drevo končnic, (2) za vsako besedo tvorimo vektor značilk (pri tem postopku uporabljamo tudi drevo končnic), (3) na podlagi označenih vektorjev značilk tvorimo model znanja.

2.2.3. Algoritem za učenje

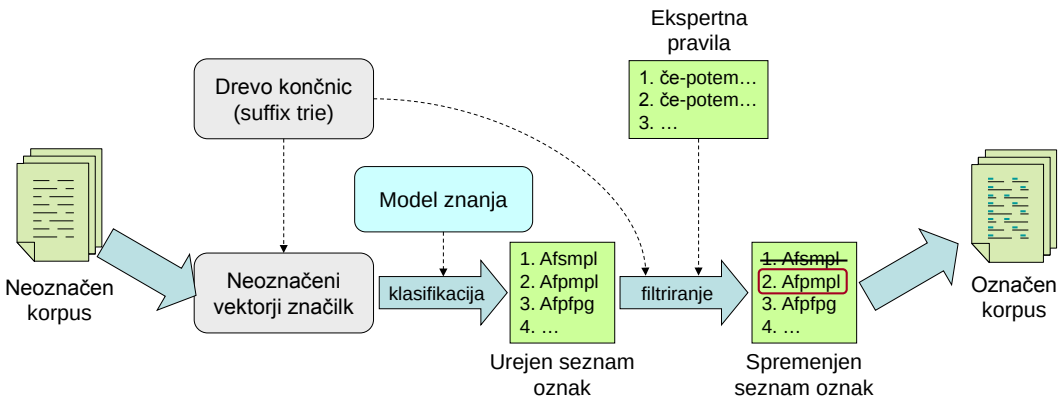
Za učenje potrebujemo ročno označeni korpus in leksikon, iz katerih najprej zgradimo drevo končnic. Nato za vsako besedo iz učnega korpusa tvorimo vektor značilk (pri tem za določanje vrednosti značilk *a* in *m* (glej tabeli 1 in 2) uporabimo drevo končnic). Ti vektorji značilk so označeni z oblikoslovnimi oznakami pripadajočih besed in zato lahko uporabimo algoritem za nadzorovano učenje, ki na podlagi označenih vektorjev tvori model znanja. Model vsebuje informacijo o tem, katere lastnosti (tj. pari značilka=vrednost) združujejo določene oblikoslovne kategorije in jih hkrati ločujejo od ostalih. Algoritem, ki ga Obeliks uporablja za učenje, temelji na principu maksimalne entropije (*maximum entropy*) in se pogosto uporablja za označevanje sekvenc (Ratnaparkhi, 1996; Nigam et al. 1999). Postopek učenja prikazuje slika 2.

2.2.4. Algoritem za označevanje

V fazi označevanja algoritem besedam iz neoznačenega korpusa pripiše oblikoslovne oznake (tvori označen korpus). Za vsako besedo najprej tvori vektor značilk na popolnoma enak način kot v fazi učenja. Besede oz. pripadajoči vektorji značilk zaporedno vstopajo v klasifikacijski algoritem, ki na podlagi zgrajenega modela znanja pripiše besedi neko oblikoslovno oznako. Pripisovanje oznake poteka tako, da klasifikacijski algoritem najprej uredi vse oznake padajoče po verjetnosti, da pripadajo dani besedi. Nato s tega seznama nekatere oznake odstranimo. V primeru, da je beseda (v celoti) v drevesu končnic, kar pomeni, da je vsebovana tudi v učnem korpusu in/ali leksikonu, potem so s seznama najprej odstranjene vse oznake, ki v drevesu končnic ne pripadajo besedi. Naknadno seznam spremenimo (ne glede na to, ali drevo končnic vsebuje besedo ali ne) z uporabo ekspertnih pravil, ki jih povzemamo v prilogi. Prvo oznako z vrha tako dobljenega seznama na koncu pripišemo besedi. Postopek označevanja je prikazan na sliki 3.

2.3. Lematizacija

Obeliks za lematizacijo besed uporablja LemmaGen[3] (Juršič, Mozetič, Erjavec, & Lavrač, 2010), implementacijo algoritma za lematizacijo na osnovi pravil tipa RDR (*Ripple Down Rules*). Za vsako kategorijo oblikoslovne oznake (samostalnik, pridevnik, glagol itd.) Obeliks iz učnega korpusa in leksikona zgradi ločeno lematizacijsko drevo. Lematizacija se izvede po označevanju in uporabi oblikoslovno oznako besede za izbiro lematizacijskega drevesa. Če npr., oblikoslovna oznaka predstavlja samostalnik, potem se za lematizacijo pripadajoče besede uporabi "samostalniško" lematizacijsko drevo. Na ta način Obeliks tvori bolj smiselne pare oznaka-lema, saj npr., glagolu *pozná* ne pripiše pridevniške leme *pozen*, temveč glagolsko lemo *poznati*, in obratno.



Slika 3: Označevanje besedila: (1) za vsako besedo tvorimo vektor značilk, (2) klasifikacijski algoritem na podlagi modela znanja, drevesa končnic in ekspertnih pravil pripiše besedi oblikoslovno oznako.

[3] http://lemmatise.ijs.si/

3. Učni korpus ssj500k[4]

Učni korpus ssj500k je bil tako kot označevalnik Obeliks izdelan v okviru projekta Sporazumevanje v slovenskem jeziku in temelji na obeh učnih korpusih, izdelanih v okviru projekta JOS. Sestavljata ga celotni korpus jos100k ter dodatnih 400.000 besed iz enomilijonskega korpusa jos1M. Vsi jezikoslovni metapodatki (oznake, leme, tokenizacija) so bili v korpusu ssj500k še enkrat ročno pregledani, povečana je bila množica skladenjsko označenih in ročno pregledanih povedi. V delu, ki ga zajema korpus jos100k, so bile dodane informacije o lastnih imenih za potrebe strojnih prepoznavalnikov imenskih entitet. Za razliko od korpusov jos100k in jos1M je bila v korpusu ssj500k v celoti ročno pregledana in popravljena tudi stavčna segmentacija in tokenizacija, kar omogoča tudi preverjanje uspešnosti algoritmov pri teh dveh postopkih. Številčni podatki o elementih v korpusu ssj500k so v tabeli 3.

Oznaka	Opis	ssj500k
	besedilo	1.677
<p>	odstavek	8.137
<s>	stavek oz. poved	27.829
<w>	beseda	500.295
<c>	ločilo/simbol	85.953
<w> + <c>	pojavnica	586.248
<links>	element s skladenjskimi povezavami	11.411
<link>	skladenjska povezava	235.865
<chunks>	element s povezavami na imenske entitete	2.178
<chunk>	imenska entiteta	4.398

Tabela 3: Število elementov v učnem korpusu ssj500k

Učni korpus ssj500k je prosto dostopen na spletnih straneh projekta SSJ[5] pod licenco Creative Commons Priznanje avtorstva-Nekomercialno 3.0.[6]

4. Analiza označevanja

Na korpusu ssj500k je bilo izvedeno desetkratno prečno preverjanje natančnosti označevalnika, ki daje naslednje rezultate:

Kategorija	Dod. pogoj	%
natančnost na znanih besedah		93,09
natančnost na neznanih besedah		54,03
skupna natančnost		92,49
natančnost na znanih besedah	(kat.)	98,72
natančnost na neznanih besedah	(kat.)	87,24
skupna natančnost	(kat.)	98,55
natančnost na znanih besedah	(brez ločil)	92,04
natančnost na neznanih besedah	(brez ločil)	53,99
skupna natančnost	(brez ločil)	91,34
natančnost na znanih besedah	(kat., brez ločil)	98,50
natančnost na neznanih besedah	(kat, brez ločil)	87,22
skupna natančnost	(kat., brez ločil)	98,30
natančnost lematizacije	(brez ločil)	97,88
natančnost lematizacije	(male črke, brez ločil)	98,55

Tabela 4: Natančnost označevalnika.

10-kratno prečno preverjanje je metoda, s katero lahko ocenimo natančnost označevalnika na neznanem besedilu (ob predpogoju zadostne heterogenosti označenega korpusa, ki pa je izpolnjen v primeru korpusa ssj500k).[7] Naj na tem mestu še razložimo, da so znane besede tiste besede, ki so vsebovane bodisi v leksikonu bodisi v učnem korpusu, neznane pa tiste, ki se pojavijo izključno v korpusu, ki ga označujemo.

Rezultati kažejo, da ima Obeliks doslej najboljši rezultat pri statističnih označevalnikih, ki so bili uporabljeni za slovenščino (prim. Džeroski et al., 2000). Natančnost označevanja ob upoštevanju zgolj pojavnic, ki so označene kot beseda tj. <w>, brez pojavnic, ki jih je tokenizator opredelil kot ločilo, tj. <c>, je 91,34 %. Ob upoštevanju vrhnje kategorije, ki jih je v tabeli oznak JOS dvanajst, je označevalnik uspešen v 98,30 % primerov. Natančnost lematizatorja je 97,88 % ob upoštevanju velike začetnice pri lemi ter 98,55 % v primeru, da upoštevamo zgolj črkovni niz brez razlikovanja med velikimi in malimi črkami.

Podrobnejše analize napak pri prečnem preverjanju kažejo natančnejšo sliko. Če pri lematizaciji najprej prikažemo napake lematizatorja, pri čemer ne upoštevamo razlike med velikimi in malimi črkami, se napake pojavljajo pri naslednjih oblikoslovnih oznakah (tabela 5):

Oznaka	Opis	Št. Napak
S (Sl+So)	samostalnik (lastno ime + občni)	2.914 (1.790+1.124)
P	pridevnik	1.991
G	glagol	749
R	prislov	577
Z	zaimek	530
K	števnik	127
N	neuvrščeno	116
L	členek	55
V	veznik	30
D	predlog	10
M	medmet	2
O	okrajšava	1
Skupaj		7.102

Tabela 5: Napake lematizacije (ne razlikujemo med velikimi in malimi črkami).

[4] http://www.slovenscina.eu/Vsebine/Sl/Kazalniki/K10.aspx
[5] http://www.slovenscina.eu/tehnologije/ucni-korpus
[6] http://creativecommons.org/licenses/by-nc/3.0/deed.sl
[7] http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29

To tabelo lahko primerjamo s tabelo, v kateri so upoštevane razlike med velikimi in malimi črkami. Rezultate navajamo po besednih vrstah v tabeli 6.

Oznaka	Opis	Št. napak
S (Sl+So)	samostalnik (lastno ime + občni)	5.696 (4.308+1.388)
P	pridevnik	2.082
G	glagol	749
R	prislov	578
Z	zaimek	530
N	številnik	356
K	neuvrščeno	136
L	členek	56
V	veznik	30
D	predlog	12
M	medmet	4
O	okrajšava	2
Skupaj		10.231

Tabela 6: Napake lematizacije (upoštevane male in velike črke).

Pričakovano (prim. Krek, 2010) se na prvem mestu pojavljajo lastna imena, ki so nepredvidljiva in v večini primerov rešljiva zgolj s širjenjem leksikona. Pri pridevnikih, ki so na drugem mestu za samostalniki, gre v treh četrtinah primerov (1.621) za zamenjavo s prislovi. Kategorija lastnih imen in parov pridevnik-prislov torej prispevata približno polovico vseh napak pri lematizaciji.

V tabeli 7 nadalje navajamo napake označevalnika v primerih, kjer je bila oblikoskladenjska kategorija pripisane oznake pravilna, celotna oblikoskladenjska oznaka pa napačna.

Oznaka	Opis	Št. napak
S	samostalnik	17.402
P	pridevnik	8.519
Z	zaimek	5.338
K	številnik	1.428
G	glagol	1.117
D	predlog	818
R	prislov	6
Skupaj		34.628

Tabela 7: Napake označevanja (kategorija pravilna).

Zanimiva je tudi podrobna analiza po posameznih oznakah. V tabeli 8 navajamo prvih petnajst oznak, pri katerih se označevalnik najpogosteje moti.

Zap. št.	Oznaka	Št. napak
1	Sometn	1.657
2	Sozmt	1.405
3	Somei	1.340
4	Slmei	1.243
5	Sozmi	986
6	Slzei	939
7	Soset	702
8	Sosei	660
9	Ppnzmt	611
10	Ppnmetd	528
11	Slmer	474
12	Sometd	459
13	Ppnzmi	443
14	Sozer	431
15	Slmetd	430

Tabela 8: 15 najbolj težavnih oznak (kategorija pravilna)

Analiza kaže, da je denimo pri samostalniku težavna kategorija par imenovalnik-tožilnik moškega spola (oznaki *Sometn*-*Somei*) ali podoben par pri ženskem spolu množine samostalnika (oznake *Sozmt*-*Sozmi*). Predpostavimo lahko, da bo prišlo do težav tam, kjer so oblike v oblikoslovni paradigmi identične, razdalje do okoliških besed, ki bi lahko razdvoumile pravilno oznako, pa tipično daljše od tistih, ki jih upošteva označevalnik.

V tabeli 9 prikazujemo podatke o primerih, kjer se je označevalnik zmotil že pri oblikoskladenjski kategoriji.

Oznaka	Opis	Št. napak
P	pridevnik	2.063
S	samostalnik	1.474
R	prislov	1.189
Z	zaimek	1.028
N	neuvrščeno	762
V	veznik	686
G	glagol	681
L	členek	242
K	številnik	133
D	predlog	128
M	medmet	27
O	okrajšava	4
Skupaj		8.417

Tabela 9: Napake označevanja (kategorija napačna).

Pri tej vrsti napak je najpogostejša zamenjava prislovov s pridevniki in obratno, ki tudi gledano v celoti predstavlja velik izziv za označevalnik (in posledično torej tudi za lematizator). Kot zanimivost v tabeli 10 navajamo še tri najtežje primere glede na posamezno obliko oz. pojavnico.

Oblika	Št.	Oznaka1 / Lema1	Oznaka2 / Lema2
jih	618	Zotzmt--k / *on*	Zotmmt--k / *on*
vse	379	Zc-set / *ves*	Rsn / *vse*
kaj	298	Zv-set / *kaj*	Rsn / *kaj*

Tabela 10: Najtežje oblike oz. pojavnice. Drugi stolpec predstavlja št. napak pri označevanju oblike, tretji (pravilna oznaka) in četrti stolpec (pripisana napačna oznaka) pa podajata najpogostejšo napako pri označevanju oblike.

Prva je oblika *jih* zaimkovne leme *on* v ženskem ali moškem spolu (tožilnika). V tem primeru gre za

razreševanje stavčne ali celo medstavčne koreference, kar je zahtevna naloga za označevalnik. Drugi je zaimek *ves* v srednjem spolu ednine imenovalnika (npr. *vse kaže, da...*) ali prislov z lemo *vse* (npr. *vse večji sum...*). Tretja je oblika *kaj* kot vprašalni zaimek (npr. *kaj je to*) ali prislov (npr. *nič kaj prida*).

Pomembno je torej, da se pri interpretaciji besedil, strojno obdelanih z lematizatorji in označevalniki, zavedamo tistih točk, kjer do napak prihaja pogosteje, in se ne zanašamo zgolj na skupno oceno natančnosti, kot jo kaže desetkratno prečno preverjanje.

5. Zaključek

V prispevku smo opisali novo orodje za označevanje besedil v slovenskem jeziku Obeliks, ki vsebuje module za segmentacijo in tokenizacijo, lematizacijo in oblikoskladenjsko označevanje ter kombinira statistične modele z ekspertnimi pravili za slovenski jezik. Obeliks je dostopen v dveh oblikah: (1) kot programska oprema na portalu SourceForge[8] in (2) kot spletni servis na straneh projekta "Sporazumevanje v slovenskem jeziku".[9] Projekt Obeliks smo zastavili kot dolgoročni projekt, ki bo v prihodnje doživel nadgradnjo, ki bo izhajala iz predstavljene analize napak pri prečnem preverjanju na učnem korpusu. Nadgradnja bo usmerjena predvsem v izboljšanje ekspertnih pravil in povečanje leksikona.

Literatura

Džeroski S., Erjavec T., Zavrel J. (2000). Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. *Second International Conference on Language Resources and Evaluation, LREC'00*, pp. 1099-1104.

Erjavec, T., Fišer, D., Krek, S., Ledinek, N. (2010). The JOS Linguistically Tagged Corpus of Slovene. V *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)* (str. 1806-1809). Pariz: ELRA.

Giménez, J., & Márquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04).* Lisbon, Portugal.

Juršič, M., Mozetič, I., Erjavec, T., & Lavrač, N. (2010). LemmaGen : multilingual lemmatisation with induced Ripple-Down rules. *J. univers. comput. sci.*

Krek, S. (2010). *Pridobivanje jezikovnih podatkov iz besedilnih korpusov za namen izdelave enojezičnih slovarjev in slovnic*: doktorska disertacija. Univ. v Ljubljani, Filozofska fakulteta, Oddelek za slovenistiko.

Mitchell, T. M. (1997). *Machine Learning.* McGraw-Hill.

Nigam, K., Lafferty, J., & McCallum, A. (1999). Using Maximum Entropy for Text Classification. *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, (pp. 61-67).

Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-Of-Speech Tagging. *Proceedings of Empirical Methods in Natural Language Processing*, (pp. 133-142).

Priloga (ekspertna pravila)[10]

w – beseda (kot je zapisana v besedilu)
w_L – beseda w zapisana z malimi črkami
L – seznam dovoljenih oznak
(vhodna spremenljivka in hkrati rezultat aplikacije pravil)
T={ } – množica za filtriranje seznama L (sprva prazna)

(1) če je w ena sama črka
    v T dodaj oznako Some
    če S_{Dm1} vsebuje w_L, v T dodaj oznako Dm
    če S_{Dt1} vsebuje w_L, v T dodaj oznako Dt
    če S_{DoDr1} vsebuje w_L, v T dodaj oznaki Do, Dr
    če S_{Dd1} vsebuje w_L, v T dodaj oznako Dd
    če S_{Krg1} vsebuje w_L, v T dodaj oznako Krg
    če S_{N1} vsebuje w_L, v T dodaj oznako N
    če w_L="a", v T dodaj oznaki Vp, Rsn
  sicer: če je w oblike *številka-končnica*, pri čemer S_{KaKo} vsebuje končnico
    v T dodaj oznake Ka* (“*” pomeni poljubne vrednosti preostalih atributov oznake)
  sicer: če w vsebuje vsaj eno črko in vsaj eno številko ter vsebuje izključno črke in številke
    v T dodaj oznake S*, N, Kag
  sicer: če w vsebuje vsaj eno številko in nobene črke
    če se w konča s piko, v T dodaj oznako Kav
    sicer: v T dodaj oznako Kag
  sicer: v T dodaj vse oznake iz L

(2) če S_{Di} ne vsebuje w_L, potem iz T odstrani oznako Di
(3) če S_{Dr} ne vsebuje w_L, potem iz T odstrani oznako Dr
(4) če S_{Dd} ne vsebuje w_L, potem iz T odstrani oznako Dd
(5) če S_{Dt} ne vsebuje w_L, potem iz T odstrani oznako Dt
(6) če S_{Dm} ne vsebuje w_L, potem iz T odstrani oznako Dm
(7) če S_{Do} ne vsebuje w_L, potem iz T odstrani oznako Do
(8) če S_{Vp} ne vsebuje w_L, potem iz T odstrani oznako Vp
(9) če S_{Vd} ne vsebuje w_L, potem iz T odstrani oznako Vd
(10) če S_L ne vsebuje w_L, potem iz T odstrani oznako L
(11) če S_Z ne vsebuje w_L, potem iz T odstrani oznake Z*
(12) če w ne vsebuje samih črk, potem iz T odstrani oznake M, G*
(13) če w ni kombinacija črk in pomišljajev, potem iz T odstrani oznake P*, R*
(14) če w ni kombinacija črk, številk in pomišljajev, potem iz T odstrani oznake oznake S*
(15) če S_O ne vsebuje w_L in hkrati w ni ena sama črka, ki ji sledi pika, potem iz T odstrani oznako O
(16) če w ni zaporedje rimskih številk, ki mu ali mu ne sledi pika, potem iz T odstrani oznake Kr*
(17) če se w ne začenja z neko predpono iz S_{KbPr}, potem iz T odstrani oznake Kb*
(18) če se w ne začenja s številko, potem iz T odstrani oznake Ka*
(19) če S_{Gp} ne vsebuje w_L, potem iz T odstrani oznake Gp*
(20) nazadnje iz L odstrani vse oznake, ki jih ni v T (pri tem L ohrani vrstni red oznak, ki jih vsebuje)

[8] http://sourceforge.net/projects/obeliks/ (navodila za uporabo se nahajajo na pripadajoči wiki-strani)
[9] http://oznacevalnik.slovenscina.eu/
[10] Seznami besed, ki so del definicije ekspertnih pravil (S_{Dm1}, S_{Dt1}, S_{DoDr1}…), so dostopni na spletni strani https://sourceforge.net/p/obeliks/wiki/SeznamiBesed/

Merjenje berljivosti japonščine kot tujega jezika na korpusu učbeniških besedil

Kristina Hmeljak Sangawa

Filozofska fakulteta, Univerza v Ljubljani
Aškerčeva 2, 1000 Ljubljana, Slovenija
kristina.hmeljak@ff.uni-lj.si

Povzetek

Samodejno merjenje jezikovne težavnosti je koristen postopek, ko pišemo, prirejamo ali izbiramo besedila za manj sposobne bralce tujega jezika. Po vzoru postopkov, ki so bili razviti za merjenje berljivosti angleških besedil, sta v zadnjih letih nastali dve orodji za merjenje berljivosti japonskih besedil za materne govorce. V preizkusu teh orodij na korpusu besedil iz učbenikov japonščine kot tujega jezika je bilo ugotovljeno, da sta ti orodji le delno uporabni za tuje bralce japonščine. Na istem korpusu sta bila preverjena dva kazatelja berljivosti: ugotovljeno je bilo, da je povprečna dolžina povedi preprost, a učinkovit pokazatelj besedilne težavnosti, razmerje med različnicami in pojavnicami znakov pa nekoliko manj zanesljiv pokazatelj.

Assessment of the readability of Japanese as a foreign language on a textbook corpus

Automatic measurement of readability is a useful procedure when writing, editing or selecting texts for weak readers. Drawing on research on English readability, some formulas and tools have been developed recently for the assessment of Japanese text readability for native speakers of Japanese. Three such tools were applied to a corpus of textbooks of Japanese as a foreign language, and found to be only partially useful in predicting readability for foreign learners. Two indicators of readability were tested on the same corpus; average sentence length was found to strongly correlate with nominal text difficulty in the textbook corpus, while character type-to-token ratio was found to have a weaker correlation.

1. Uvod

Branje je ena od osnovnih dejavnosti, na katerih sloni učenje tujega jezika, in lahko najbolj učinkovito pripomore k razvoju bralčevih jezikovnih sposobnosti takrat, ko je berilo ne samo zanimivo, ampak tudi razumljivo, torej primerne težavnosti za bralca, ki se jezika šele uči.

Bralno gradivo primerne težavnosti za začetne bralce tujega jezika učitelji običajno pripravijo tako, da sestavijo ali priredijo besedila in pri tem uporabijo samo jezikovne strukture in besedišče, ki ga ciljni bralci obvladajo, gradivo za bolj izkušene bralce pa pogosto tako, da izberejo primerno težka besedila na osnovi objektivnih ali subjektivnih kriterijev. Oba postopka se že od nekdaj izvajata ročno, k obema pa lahko veliko pripomorejo jezikovne tehnologije.

Pri pisanju in prirejanju besedil so koristna orodja za besedilno analizo, ki izpostavijo jezikovne strukture in besedišče nad določeno stopnjo težavnosti (npr. vse besede ali slovnične vzorce, ki niso vključeni v dani seznam že predelane učne snovi). Pri izbiri besedil za določeno stopnjo jezikovne sposobnosti pa je koristna splošna ocena težavnosti besedila, na osnovi katere množico razpoložljivih besedil razvrstimo po težavnosti in iz nje izberemo besedilo želene stopnje.

Glede na ogromno količino besedil, ki so na spletu dostopna v digitalni obliki, lahko namreč zlahka pridobimo veliko število besedil na izbrano temo, izbor primerno težkega besedila pa je zamudno opravilo, pri katerem je samodejna ocena berljivosti (težavnosti) besedila lahko v veliko pomoč tako učitelju kot tudi učencu tujega jezika.

Zato da lahko besedila razvrstimo po stopnji težavnosti, potrebujemo lestvico težavnosti in metodo, s katero besedila razvrstimo po tej lestvici. Za merjenje berljivosti angleških besedil obstaja že veliko postopkov, podprtih z obsežnimi raziskavami. Tudi za ocenjevanje stopnje berljivosti japonščine kot maternega jezika obstaja nekaj formul in orodij, toda o berljivosti japonščine kot tujega jezika je bilo doslej le malo raziskav.

Kot prvi korak k ocenjevanju berljivosti japonščine kot tujega jezika smo zato zgradili korpus besedil iz učbenikov japonščine kot tujega jezika in na njem preizkusili obstoječa orodja in dva od najbolj pogosto uporabljanih kazateljev berljivosti.

V prispevku so v drugem razdelku predstavljeni dosedanji pristopi k strojnemu ocenjevanju berljivosti, vključno z merjenjem berljivosti japonščine. V tretjem razdelku je predstavljen poskus uporabe orodij, ki so bila razvita za merjenje berljivosti japonščine kot maternega jezika, na korpusu učbenikov japonščine kot tujega jezika. V četrtem razdelku so prikazani rezultati meritev dveh preprostih, a robustnih kazateljev berljivosti na istem korpusu, v zadnjem razdelku pa so prikazane možnosti za nadaljnje delo.

2. Postopki za merjenje berljivosti

Branje je kompleksen proces, na razumevanje branega besedila pa vpliva veliko dejavnikov, ki jih moramo upoštevati, ko ocenjujemo berljivost določenega besedila: samo besedilo in njegove sestavine (znaki, besede, struktura, pomen), bralec in njegovo znanje (jezikovno, kulturno in širše) ter cilj oziroma namen branja.

2.1. Pristopi k merjenju berljivosti

Pri merjenju berljivosti oziroma težavnosti besedil v dosedanjih raziskavah lahko ločimo dva glavna pristopa.

Prvi pristop je statistična jezikovna analiza, pri kateri se različni merljivi dejavniki v besedilu (npr. povprečna dolžina povedi ali besed) izmerijo na korpusu besedil z vnaprej znanimi oz. določenimi stopnjami težavnosti, nato pa se tisti dejavniki, ki se izkažejo za učinkovite kazatelje berljivosti, na osnovi regresijske analize sestavijo v formulo, v kateri ima vsak dejavnik primerno utežitev.

Drugi pristop je jezikovno modeliranje, pri katerem se - ravno tako na osnovi korpusa, ki je sestavljen iz podkorpusov različnih vnaprej določenih težavnostnih stopenj - izoblikuje serija jezikovnih modelov z rastočo stopnjo

težavnosti; pri oceni berljivosti poljubnega besedila se potem to besedilo primerja z vsemi modeli s pomočjo klasifikatorjev in oceni s stopnjo težavnosti modela, ki je besedilu najbližji.

2.2. Referenčne lestvice za merjenje berljivosti

Pri obeh pristopih je prvi korak določitev težavnostne lestvice in izoblikovanje referenčnega korpusa, na osnovi katerega se nato ali razvije formula, ki vsebuje različne dejavnike berljivosti, ali zgradijo jezikovni modeli, ki nato služijo za kategoriziranje besedil.

Referenčna lestvica, ki se najbolj pogosto zasledi v literaturi o merjenju berljivosti za bralce maternega jezika, je zelo intuitivna lestvica: to so šolski razredi oziroma število let šolanja v določenem jeziku (npr. od 1. razreda osnovne šole do zadnjega razreda gimnazije ipd.), saj naj bi ti odgovarjali jezikovni in splošneje kognitivni razvojni stopnji bralcev. Referenčni korpus pri raziskavah, ki uporabljajo to lestvico, je pogosto zbirka besedil iz učbenikov za vsak razred šolanja. Ta izbor sloni na predpostavki, da so strokovnjaki, ki so izdelali učbenike, na podlagi svojih izkušenj uporabljali jezik, ki je primeren za določeno starostno stopnjo in šolski razred. Pri merjenju težavnosti za tuje bralce smo zasledili samo en podoben pristop k določanju težavnostne lestvice (François 2009), ki uporablja Skupni evropski referenčni okvir za jezike (CEFR).

Druga možna referenčna lestvica je odstotek populacije, ki je sposobna razumeti določeno besedilo. Pri tem pristopu se najprej pripravi zbirka različnih besedil, s testiranjem na reprezentativnem vzorcu populacije govorcev določenega jezika se nato ugotovi, kolikšen odstotek bralcev besedilo razume, na osnovi tega pa se besedila razporedijo v poljubno število stopenj.

Tretji, zelo redek pristop (Sato 2011), je uporaba reprezentativnega korpusa, pri čemer se vsa besedila v korpusu, ki naj bi bil reprezentativna slika vseh besedil v določenem jeziku, s pomočjo obstoječega orodja razporedijo po težavnosti in nato razdelijo v izbrano število stopenj (podkorpusov), ki služijo kot referenčna lestvica.

2.3. Merjenje berljivosti v angleščini

Merjenje berljivosti angleških besedil ima verjetno najdaljšo in najobsežnejšo tradicijo raziskovanja. Od leta 1923, ko sta Lively in Pressy predstavili prvo formulo za merjenje berljivosti, je bilo razvitih na stotine formul in postopkov za ocenjevanje berljivosti (DuBay 2004, 2006). Lastnosti besedil, ki se najbolj pogosto omenjajo v teh raziskavah, so povprečno število besed v povedi, povprečno število zlogov v besedah, povprečno število črk v besedah, odstotek besed iz seznamov osnovnega besedišča, idr.

Pri prvih formulah so se lastnosti besedila (število besed v povedi ipd.) štele ročno, zato so vsebovale samo lahko merljive dejavnike. S pojavom jezikovnih tehnologij pa so se pri merjenju berljivosti začele uporabljati tudi druge, bolj zahtevne analize, kot npr. bolj podrobna analiza besedišča na osnovi obsežnih seznamov pogostosti posameznih besed ali večbesednih enot v velikih korpusih (Anagnostou in Weir 2007), analiza skladenjske strukture s pomočjo samodejnih skladenjskih analizatorjev, merjenje razmerja med različnicami in pojavnicami, merjenje leksikalne kohezije, diskurzne analize ipd. (Graesser idr. 2004, Barzilay in Lapata 2008, Pitler in Nenkova 2008). Novejše raziskave, ki uporabljajo jezikovne modele n-gramov in klasifikatorje za ocenjevanje berljivosti besedil, so se izkazale za robustne in posebej primerne za spletna besedila z nepopolnimi povedmi (Collins-Thompson in Callan 2004, Schwarm in Ostendorf 2005, Heilmann idr. 2007, Feng idr. 2010)

2.4. Merjenje berljivosti v japonščini

Na Japonskem je Morioka leta 1952 objavil verjetno prvo raziskavo o berljivosti japonščine po vzoru formul za ugotavljanje berljivosti angleščine. Temu je sledilo nekaj drugih predlogov formul (Sakamoto 1962, Tateishi idr. 1988, Pichl in Narita 2007) oziroma dejavnikov berljivosti, kot npr. gostote besedišča (Sano in Maruyama 2008), toda nobena od predlaganih formul se ni širše uporabila za merjenje berljivosti japonščine. Šele v zadnjih letih sta dve skupini raziskovalcev razvili dve metodi za merjenje berljivosti besedil za bralce japonščine kot maternega jezika. Prvo metodo (RRL) je razvila Shibasaki s sodelavci (Shibasaki idr. 2008, 2010) z uporabo regresijske analize na korpusu učbenikov japonskega jezika za japonske otroke v dvanajstih razredih od prvega razreda osnovne šole do tretjega (zadnjega) razreda gimnazije. Drugo metodo je razvil Sato s sodelavci (Sato idr. 2008, Sato 2011), ki pa uporablja jezikovno modeliranje: v prvi metodi (Obi T13) ustvari jezikovne modele unigramov znakov v besedilih učbenikov za vse predmete v 12 razredih japonskih šol; v drugi metodi (Obi B9) pa so jezikovni modeli bigramov znakov oz. pismenk v besedilih osnovani na uravnoteženem korpusu japonskega pisanega jezika (BCCWJ, Maekawa idr. 2010). Pri določanju stopnje težavnosti oz. berljivosti določenega besedila, orodji Obi T13 in Obi B9 ocenita težavnostno stopnjo besedila tako, da ga primerjata s tako izdelanimi jezikovnimi modeli. Tako RRL kot Obi sta dostopni na strežnikih obeh raziskovalnih skupin.

Objavljena so tudi poročila o nekaterih raziskavah o berljivosti japonščine kot tujega jezika (Kawamura 1999a, Kitamura idr. 2009, Yamura-Takei idr. 2005, Mizushima idr. 2011, Nakamura idr. 2012, Yoshihashi idr. 2007, Mizuno idr. 2008, Hazelbeck in Saito 2009), toda razen metode, ki jo predstavljajo Kawamura, Kitamura in sodelavci in ki meri samo težavnost besedišča v določenem besedilu, ostale raziskave ne nudijo še dostopne in zaključene metode za celovito oceno berljivosti japonščine kot tujega jezika.

3. Korpus besedil iz učbenikov japonščine kot tujega jezika

Kot prvi korak k preverjanju uporabnosti obstoječih orodij, ki so bili razviti za materne govorce japonščine, tudi za merjenje berljivosti japonščine kot tujega jezika, smo sestavili korpus besedil iz petih različnih serij učbenikov japonščine kot tujega jezika. Medtem ko je pri določanju težavnostnih stopenj za materne govorce možna razporeditev besedil po starosti ali šolskem razredu bralcev, ki besedila razumejo, pa za učence japonščine kot tujega jezika taka lestvica ne obstaja. Za druge namene se sicer pogosto uporabljajo smernice standardnega izpita iz znanja japonščine kot tujega jezika (Japanese Language Proficiency Test), ki pa je od leta 2010 prešel s štirih na pet stopenj znanja in - po vzoru Skupnega evropskega referenčnega okvira za učenje, poučevanje in ocenjevanje tu-

jih jezikov (CEFR) - prešel z opisa stopenj s seznami besedišča in slovničnih struktur, ki so predvidene za vsako stopnjo, na bolj opisne ocene sposobnosti (npr. razumeti predavanja o strokovnih temah pri naravni hitrosti ipd.) , ki naj bi jih bili tuji učenci japonščine sposobni na vsaki stopnji, brez seznamov struktur in besedišča, tako da je težko uporaben kot referenčna lestvica za strojno merjenje jezikovnih dejavnikov.

Glede na to, da se v različnih jezikovnih programih, izobraževalnih ustanovah in učbeniških založbah uporabljajo različne lestvice težavnosti za japonščino, nismo sestavili korpusa s homogenimi podkorpusi npr. začetniške, nadaljevalne in izpopolnjevalne japonščine, pač pa smo za prvo preverjanje uporabnosti obstoječih orodij zbrali pet zbirk učbenikov za tuje učence japonščine, ki vsebujejo vsaka po nekaj knjig postopoma rastoče težavnosti, nato pa primerjali ocene berljivosti obstoječih orodij znotraj vsake posamezne zbirke, saj je zelo verjetno, da so se pri vsaki zbirki učbenikov dosledno uporabljala ista načela za razporejanje besedil po težavnosti. Zbrali smo besedila iz spodaj navedenih učbenikov.

1) *Japanese Graded Readers - Yomu yomu bunko* (2005-2009) založbe Ask, zbirka postopnih lahkih beril za začetnike in nadaljevalce na 4 nivojih (v nadaljevanju JGR1 do JGR4); v besedilu vsakega nivoja se uporabljajo samo besede in slovnični vzorci iz strogo določenega nabora besed in vzorcev, ki je osnovan na petih priljubljenih učbeniških serijah in delno sovpada s seznami JLPT.

2) Učbeniki zavoda Bunka Institute of Language *Shin bunka shokyû nihongo II* (2000), *Bunka chûkyû nihongo I, II* (1994, 1997) založbe Bonjinsha, v nadaljevanju BunA2, BunB1 in BunB2. To je serija učbenikov, ki vključuje vaje za branje, pisanje, poslušanje in govorjenje, zato smo za analizo uporabili le berila in dialoge za vajo iz branja.

3) Serija učbenikov Centra za tuje študente Univerze Sanno *Enjoyable Task Reading in Japanese - Nihongo o tanoshiku yomu hon: Pre-intermediate - Shochûkyû* (1996)*, Intermediate - Chûkyû* (1991) in *Pre-advanced - Chûjôkyû* (1993) založbe Univerze Sanno, v nadaljevanju Joy1, Joy2 in Joy3.

4) Serija učbenikov avtorice Nobuko Mizutani *Sôgô nihongo - Introduction to Japanese - An Integrated Course,* založbe Bonjinsha: *Introduction to Intermediate Japanese - Shokyû kara chûkyû e* (1990), *Intermediate Japanese - First Semester - Chûkyû zenki* (1989) ter *Intermediate Japanese - Chûkyû* (1987); druga in tretja knjiga se vsaka delita na dve seriji osnovnih in nadgrajevalnih beril, v nadaljevanju Mizu1, Mizu2a, Mizu2b, Mizu3a in Mizu3b.

5) Učbeniki Tokijske univerze za tuje jezike (Tokyo University of Foreign Studies) *Shokyû nihongo* (1994), *Chûkyû nihongo* (1994) in *Jôkyû nihongo* (1998), v nadaljevanju TUFS-Ad (dialogi), TUFS-Ar (berila), TUFS-B in TUFS-C.

Vse učbenike oziroma berila smo preslikali, obdelali s programom za razpoznavanje znakov, ročno preverili in popravili, normalizirali in vsako besedilo vsakega učbenika shranili v ločenih datotekah.

Obseg korpusa, število vsebovanih besedil in povprečna dolžina besedil v vsakem učbeniku (po številu znakov, kot je običajno pri merjenju dolžine japonskih besedil) so predstavljeni v Tabeli 1.

	Učbenik	št. besedil	št. znakov	povprečno št. znakov/besedilo
1)	JGR1	17	23569	1386
	JGR2	15	44039	2936
	JGR3	12	52413	4368
	JGR4	11	83967	7633
2)	BunA2	19	40540	2134
	BunB1	21	17327	825
	BunB2	15	18697	1246
3)	Joy1	17	18794	1106
	Joy2	9	10541	1171
	Joy3	9	16764	1863
4)	Mizu1	15	8925	595
	Mizu2a	12	12795	1066
	Mizu2b	12	5332	444
	Mizu3a	12	11927	994
	Mizu3b	12	5000	417
5)	TufsAd	28	25471	910
	TufsAr	10	7393	739
	TufsB	21	26662	1270
	TufsC	10	28140	2814

Tabela 1: Obseg korpusa učbenikov japonščine kot tujega jezika

4. Aplikacija obstoječih orodij za merjenje berljivosti na korpusu

Na zgoraj opisanem korpusu učbenikov smo preverjali učinkovitost obstoječih metod za ocenjevanje težavnosti japonščine kot maternega jezika, ki smo jih omenili v prejšnjem razdelku:

- Obi T13, ki oceni besedilo na lestvici 13 razredov od osnovne šole do prvega leta univerze (Sato idr. 2008);
- Obi B9, ki oceni besedilo na lestvici devetih stopenj, določenih na osnovi porazdelitve besedil v reprezentativnem korpusu japonskega jezika BCCWJ (Sato 2011),
- metodo RRL, ki podobno kot T13 ocenjuje besedila na lestvici šolskih razredov, toda samo na lestvici prvih devet razredov obveznega šolanja (Shibasaki idr. 2010).

Z vsemi tremi orodji smo izmerili predvideno stopnjo berljivosti besedil v zgoraj opisanem korpusu učbenikov japonščine kot tujega jezika.

4.1. Rezultati in analiza meritev

Primerjava ocen berljivosti besedil iz različnih serij učbenikov, kot smo omenili v razdelku 3, ni smiselna, zato smo primerjali nominalni rang učbenikov znotraj iste serije, kot ga predvidevajo avtorji učbenika sami, z rangi ocen treh uporabljenih orodij. S pomočjo spletnega orodja (Wessa 2010) smo izračunali Spearmanovo korelacijo rangov ρ (Kendall 1970), ki nakazuje ujemanje med razporeditvijo besedil po težavnostni stopnji v seriji sami in razporeditvijo besedil, kot jo ocenjujejo tri orodja. Podatek o korelciji rangov je podan samo kot vodilo pri oceni primernosti, saj gre za majhne razpone rangov.

Kot je razvidno iz tabele 2, metodi Obi T13 in Obi B9 pravilno razporedita večino serij, se pa ne ujemata z lestvico težavnosti zbirke Mizutani. Metoda RRL se ravno tako ujema pri nekaterih serijah, ponuja pa posebej presenetljiv rezultat pri zbirki JGR, ki vsebuje najlažja besedila, kjer je korelacija nakazana celo kot negativna.

Učbenik	Obi T13	ρ	Obi B9	ρ	RRL	ρ
JGR1	5	0.8	1	0.8	6.5	-0.65
JGR2	6		1		6.5	
JGR3	6		1		6	
JGR4	6		3		6.4	
BunA2	5	1	2	1	6.5	1
BunB1	6		4		6.8	
BunB2	9		5		6.9	
Joy1	6	1	1	1	5.9	1
Joy2	8		4		6.6	
Joy3	9		5		7	
Mizu1	9	0.5	5	0.5	6.5	0.9
Mizu2a	9		5		7	
Mizu2b	9		5		7.5	
Mizu3a	9		5		7.3	
Mizu3b	9		5		8.7	
TufsAd	3	1	1	0.95	4.2	0.9
TufsAr	5		1		4.2	
TufsB	9		5		7.2	
TufsC	10		7		7.2	

Tabela 2: Obijeva in RRL-jeva ocena berljivosti besedil v korpusu učbenikov japonščine kot tujega jezika

Kot možen razlog za to neujemanje se ponuja dejstvo, da lahko tako na obe metodi Obi, ki sta osnovani na modelu unigramov oz. bigramov znakov v besedilih, kot tudi na metodo RRL, ki ravno tako preverja uporabljane pismenke, zelo vpliva izbor znakov pri zapisu japonščine.

Za zapis japonščine se namreč uporabljajo trije nabori znakov: dve zlogovnici (hiragana in katakana), pri katerih vsak od 46 znakov zapisuje en zlog, ter približno 2000 kitajskih pismenk, pri katerih vsak znak zapisuje eno besedo ali morfem. V standardnem zapisu japonskih besedil se za skoraj vse polnopomenske besede uporabljajo kitajske pismenke, zlogovnici pa samo za obrazila, funkcijske besede in tujke, toda v učbenikih in besedilih za otroke, ki se šele učijo branja, se v vsakem razredu uporabljajo samo točno določene pismenke, ki se jih po veljavnih smernicah japonskega ministrstva za šolstvo otroci naučijo v vsakem razredu. V besedilih za 1. razred osnovne šole se tako uporabljata zlogovnici ter prvih 80 pismenk, v besedilih za 2. razred znaki iz 1. razreda in dodatno še 160 pismenk iz standardnega seznama itd. Vse besede, ki se v standardnem pisanju zapišejo s kitajskimi pismenkami, ki jih otroci v določenem razredu še ne spoznajo, se v tako prirejenih besedilih zapišejo fonetično, z zlogovnico hiragana, brez uporabe pismenk, ki jih otroci ne poznajo.

Pri besedilih, ki so pisana za japonske otroke, je torej že iz nabora znakov, ki jih besedilo uporablja, dokaj jasno razvidno, za kateri razred so napisana. Zato je verjetno, da vse tri metode ocenjevanja berljivosti, ki slonijo na merjenju vsebovanih znakov, uspešno ugotovijo, za kateri razred je besedilo napisano, če je napisano po standardnih navodilih za japonska šolska besedila.

V korpusu učbenikov japonščine kot tujega jezika se v seriji TUFS uporablja podoben sistem zapisovanja, po katerem so besedila v učbeniku za začetnike zapisana z omejenim naborom kitajskih pismenk, učbenik za srednjo stopnjo s postopno več pismenkami, učbenik za višjo stopnjo pa v standardnem zapisu. To je možen razlog za uspešno razvrstitev besedil po metodah Obi in RRL. Po drugi strani pa so v zbirki JGR besedila zapisana v standardnem zapisu, ob vseh pismenkah pa je z zlogovnico še dodan glasovni zapis (furigana). Pri strojni obdelavi besedil smo dodatne glasovne zapise odstranili, ker drugače obdelava ni mogoča, zato so se v besedilih pojavile pismenke, ki bi bile verjetno težko berljive za začetnike, ki ne poznajo kitajskih pismenk, ko ne bi bilo zraven glasovnega zapisa, dejansko pa so besedila izredno berljiva in razumljiva tudi začetnikom, kot smo lahko ugotovili med lastnim poukom japonščine za začetnike.

To hipotezo smo preverili tako, da smo ista besedila iz zbirke JGR prepisali po kriterijih za zapis besedil za prvih šest razredov osnovne šole in ugotovili, da obe orodji ocenjujeta isto besedilo kot bistveno bolj berljivo, če je zapisano v zlogovnici, kot pa če je zapisano s pismenkami.

Glede na to, da obstaja več spletnih orodij, s katerimi lahko poljubnemu besedilu v japonščini samodejno dodamo zapis glasovnih vrednosti v zlogovnici, ali samodejno obdelamo besedilo tako, da se izpiše v zlogovnici, je potem pri branju besedil v elektronski obliki uporaba pismenk namesto zlogovnice ali obratno pravzaprav nepomembna. Iz tega lahko sklepamo, da so obstoječa orodja za ocenjevanje berljivosti japonščine kot maternega jezika samo delno uporabna pri ocenjevanju japonščine kot tujega jezika.

5. Merjenje dejavnikov berljivosti v učbenikih japonščine kot tujega jezika

Kot prvi korak k razvoju formule za merjenje berljivosti japonskih besedil za bralce japonščine kot tujega jezika smo na istem korpusu učbenikov preverili dve izmed lastnosti besedil, ki so se doslej uporabile v merjenju berljivosti japonščine.

5.1. Jezikovni dejavniki berljivosti japonščine

V dosedanjih raziskavah o berljivosti japonščine so se kot kazatelji berljivosti merile naslednje merljive lastnosti japonskih besedil:

- pri zapisu: razmerje med številom znakov različnih naborov, t.j. hiragane, katakane, kitajskih pismenk in latinice (Morioka 1952, Sakamoto 1964, Tateisi idr. 1988, Pichl in Narita 2007); delež kitajskih pismenk različnih stopenj po izbranih referenčnih seznamih (Pichl in Narita 2007 uporabljata seznam pismenk japonskega Ministrstva za šolstvo za pouk v osnovni in srednji šoli, Kawamura 1998 uporablja seznam JLPT); delež pismenk z več kot 15 potezami (Pichl in Narita 2007); dolžina sosledij znakov iz istega nabora (Tateisi idr. 1988); razmerje med številom ločil in drugih znakov (Morioka 1952); razmerje med številom pik in vejic kot kazatelj skladenjske zapletenosti

(Tateisi idr. 1988); delež besed z dodanim glasovnim zapisom kot kazatelj težavnosti besedišča (Morioka 1952);

- pri besedišču: delež besed iz različnih stopenj referenčnih seznamov, kot npr. seznam pogostosti (Sakamoto 1962, Kawamura idr. 2008), tf*idf glede na časopisni korpus (Kitamura idr. 2009), seznam besedišča za JLPT (Kawamura 1998), domačnost (Kawamura 2008); delež funkcijskih besed iz seznama JLPT (Mizuno idr. 2008); delež besed japonskega, kitajskega in drugega izvora, delež okrajšav, delež lastnih imen, delež abstraktnih besed v vlogi osebka (Morioka 1952); besediščna gostota po Hallidayevi formuli, t.j. razmerje med številom polnopomenskih besed in številom povedkov (Sano in Maruyama 2008);

- pri skladnji: dolžina povedi (Morioka 1952, Sakamoto 1962, Tateisi idr. 1988, Oono in Inazumi 2007, Shibasaki 2008); dolžina stavkov (Morioka 1952); število povedkov na poved (Shibasaki 2008); delež prisamostalniških odvisnikov (Mizushima idr. 2011); delež neizraženih udeleženskih vlog (Yamura-Takei 2008) ali neizraženih osebkov (Nakamura idr. 2012).

- na ravni besedila: dolžina besedila (Morikawa idr. (2010) omenjajo dolžino 10000 znakov kot najdaljšo sprejemljivo za nadaljevalne učence); delež premega govora (Morioka 1952);

- na ravni sloga: delež povedkov v neformalnem in formalnem slogu (Morioka 1952); delež pogovornih členkov in polnil (Morioka 1952).

Izmed teh smo izbrali dve najbolj osnovni in zanesljivo merljivi: povprečno dolžino povedi in razmerje med različnicami in pojavnicami (TTR) znakov. Povprečna dolžina povedi se je izkazala kot najbolj zanesljiv med merljivimi kazatelji težavnosti v angleških besedilih (Feng idr. 2010), kar se tudi ujema z raziskavami o pomnjenju povedi (Goetz idr. 1981), TTR znakov pa je lahko dober kazatelj pestrosti in posledično težavnosti besedišča.

Povprečno dolžino povedi smo izmerili tako, da smo v besedilih prelomili vrstice po vseh ločilih, ki označujejo konec povedi (。 ？ ！」 』) in ročno preverili prelom vrstic pri oklepajskih znakih 」 in 『, ki se lahko uporabljata tudi sredi povedi. Nato smo prešteli število vrstic v vsakem besedilu (vključno z vrsticami naslovov, ki niso vsebovale ločil, a so bile že v prvotnih besedilih ločene od ostalega besedila) ter delili število znakov v besedilu s številom vrstic. Dolžino smo merili po številu znakov in ne besed, ker je tokenizacija (deljenje na besede) pri japonščini, ki se zapisuje brez presledkov med besedami, lahko vir napak pri štetju, še posebej pri sestavljenih besedah, ki jih različna orodja za tokenizacijo različno obravnavajo.

Izmerili smo tudi razmerje med različnicami in pojavnicami znakov (TTR). Tudi tu smo zaradi možnih napak pri tokenizaciji merili TTR znakov raje kot TTR besed. Ker na to razmerje močno vpliva dolžina besedila, smo dolžino besedil normalizirali na dolžino najkrajšega besedila v korpusu, ki je štelo 270 znakov, tako da smo črpali začetnih 270 znakov vsakega besedila. Nato smo na vsakem vzorcu izmerili število različnic in pojavnic, izračunali razmerje med njima ter izračunali povprečje teh razmerij v vsakem učbeniku.

Rezultati analize so povzeti v tabeli 3. Tudi tu je Spearmanova korelacija rangov ρ izračunana samo informativno. Kot je razvidno iz tabele, povprečna dolžina povedi v vseh serijah učbenikov raste vzporedno s stopnjo težavnosti učbenika. Tudi pri besedilih v učbenikih japonščine kot tujega jezika se je torej dolžina povedi izkazala za preprosto, a obenem učinkovito mero besedilne težavnosti.

Po drugi strani TTR manj učinkovito ocenjuje stopnje težavnosti v učbenikih. Razlog za to je lahko neprimerni način meritve (prekratki vzorci besedil), lahko izbor besedil samih - torej neprimernost učbeniške serije kot referenčne lestvice za merjenje težavnosti besedil, ali pa neprimernost te mere za razvrščanje besedil v japonščini kot tujem jeziku.

	Učbenik	povprečno št. znakov/poved	ρ	TTR znakov	ρ
1)	JGR1	14.8	1.0	0.28	1.0
	JGR2	19.9		0.31	
	JGR3	21.9		0.33	
	JGR4	25.9		0.38	
2)	BunA2	20.6	1.0	0.38	0.87
	BunB1	30.1		0.41	
	BunB2	37.3		0.41	
3)	Joy1	25.8	1.0	0.35	1.0
	Joy2	28.5		0.37	
	Joy3	36.6		0.43	
4)	Mizu1	32.9	1.0	0.37	0.82
	Mizu2a	35		0.42	
	Mizu2b	39.4		0.48	
	Mizu3a	39.7		0.42	
	Mizu3b	33		0.51	
5)	TufsAd	22.4	1.0	0.28	1.0
	TufsAr	30.8		0.31	
	TufsB	37.1		0.39	
	TufsC	44.8		0.42	

Tabela 3: Kazatelji berljivosti v korpusu učbenikov japonščine kot tujega jezika

6. Zaključek

V prispevku smo predstavili korpus besedil iz učbenikov japonščine kot tujega jezika, ki lahko služi kot osnova za izdelavo formule za merjenje berljivosti japonskih besedil za tuje bralce. Ugotovljeno je bilo, da so obstoječe mere berljivosti za rojene govorce japonščine le delno uporabne za ugotavljanje berljivosti japonščine kot tujega jezika. Na korpusu je bila preverjena koristnost dveh preprostih kazateljev merljivosti, od katerih se je povprečna dolžina povedi izkazala za koristno mero. Analiza drugih besedilnih lastnosti, ki lahko služijo pri samodejnem ocenjevanju besedilne težavnosti, bo predmet naših prihodnjih raziskav.

7. Literatura

Anagnostou, N. in Weir, G., 2007. Average Collocation Frequency as an Indicator of Semantic Complexity. V *Proceedings of ICTATLL 2007*.

Barzilay, R. in Lapata, M., 2008. Modeling local coherence: An entity-based approach, *Computational Linguistics* 34:1, 1-34.

Collins-Thompson, K. in Callan, J., 2004. A Language Modeling Approach to Predicting Reading Difficulty. V *HLT-NAACL 2004 Main Proceedings*, Boston: ACL. 193--200.

DuBay, W., 2004. *The principles of readability*. Costa Mesa: Impact Information.

DuBay, W., 2006. *The Classic Readability Studies*. Costa Mesa: Impact Information.

Feng, L., Jansche, M., Huenerfauth, M. in Elhadad, N., 2010. A Comparison of Features for Automatic Readability Assessment. V *23rd International Conference on Computational Linguistics, Poster Volume*. Beijing: COLING. 276--284.

François, T., 2009. Combining a Statistical Language Model with Logistic Regression to Predict the Lexical and Syntactic Difficulty of Texts for FFL. V *Proceedings of the Student Research Workshop at EACL 2009*, Athens: ACL.

Goetz, E., Anderson, R. in Schallert, D.. 1981. The representation of sentences in memory, *Journal of Verbal Learning & Verbal Behavior.* 20:4. 369-385.

Graesser, A., McNamara, D., Louwerse, M. in Cai, Z., 2004. Coh-Metrix: Analysis of text on cohesion and language, *Behavior Research Methods, Instruments, & Computers* 36:193-202.

Hazelbeck, G. in Saito, H., 2009. A Corpus-based E-learning System for Japanese Vocabulary, *Information and Media Technologies* 4:4. 1104-1128.

Heilman, M., Collins-Thompson, K., Callan, J. in Eskenazi, M., 2007. Combining lexical and grammatical features to improve readability measures for first and second language texts. V *Proceedings of the Human Language Technology conference*, Rochester, NY: ACL. 460--467.

Kawamura, Y., 1999. Kanji no nan'ido hantei shisutemu 'Kanji checker' o mochiita tekisuto no bunseki - Analysis of Japanese Textbook Using the``Kanji Level Checker". *Tokyo kokusai daigaku ronsô.* 59:73-87.

Kendall M.G., 1970. *Rank correlation methods*. London: Griffin

Kitamura, T., Tomioka, Y. in Kawamura, Y., 2009. Development and evaluation of a word level rating system based on inverse document frequency. *JLEM* 16:1. 52-53.

Maekawa, K., Yamazaki, M., Maruyama, T., Yamaguchi, M., Ogura, H., Kashino, W., Ogiso, T., Koiso, H. in Den, Y., 2010. Design, compilation, and preliminary analyses of Balanced Corpus of Contemporary Written Japanese. *Proceedings of the 10th LREC*, Valletta: LREC. 1483-1486.

Mizuno, S., Ooyama, H., Kobayashi, T. idr. 2008. Nihongo dokkai shien no tame no gogigoto no yôrei chûshutsu shisutemu no kôchiku. *Kyôiku-gakushû o shien suru gengoshori.* Tokio: NLP. 63-66.

Mizushima, H., Uchida, S., Kitamura, T. in Kawamura, Y., 2011. Gakushûsha ni totte nankai na kôbun no jidôkenshutsu. *JLEM*. 18:1. 64-65.

Morikawa, Y., Nagasu, M., Haruna, H. in Kitamura, T., 2010. Nihongo dokkai gakushû shien saito "tutor.bunko" no kôsô to kaihatsu. *Kônan daigaku jôhô kyôiku kenkyû sentaa kiyô.* 9(3).

Morioka, K., 1952. Yomiyasusa no kisoteki kenkyû. V *Shôwa 26 nendo Kokuritsu kokugo kenkyûjô nenpô*. Tokio: Kokken. 91-108.

Nakamura, K., Kitamaura, T. in Kawamura, Y., 2012. Kensaku enjin o mochiita shukaku shôryakubun no jidô hantei. *JLEM*. 19:1.

Pichl, L. in Narita, J., 2007. Readability Factors of Japanese Text Classification. *Databases in Networked Information Systems*, Berlin: Springer. 132-138.

Pitler, E. in Nenkova, A., 2008. Revisiting readability: a unified framework for predicting text quality. V *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg: ACL. 186--195.

Sakamoto, I., 1962. Jidô yomimono no goihijû no hyôteihô. *Kyôiku shinrigaku nenpô.* 1:107.

Sano, M. and Maruyama, T., 2008. Lexical Density in Japanese Texts: classifying text samples in the Balanced Corpus of Contemporary Written Japanese (BCCWJ). *Proceedings of ISFC 35: Voices Around the World*, Sydney: ISFC. 359-364.

Sato, S., Matsuyoshi, S. in Kondoh, Y., 2008. Automatic Assessment of Japanese Text Readability Based on a Textbook Corpus. V *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech: LREC.

Sato, S., 2011. Measuring Text Readability Based on Balanced Corpus, *IPSJ Journal,* 52:4, 1777-1789.

Schwarm, S. in Ostendorf, M., 2005. Reading level assessment using support vector machines and statistical language models. V *Proceedings of the 43rd Annual Meeting*, Stroudsburg: ACL. 523--530.

Shibasaki, H., Tamaoka, K., Yamamoto K., Kanô, M., Hara, S. in Lee, J., 2008. Nihongo koopasu o ôyôshita bunshô no nan'ido sokutei no kenkyû. V *Tokutei ryôiki kenkyû 'Nihongo koopasu' Heisei 19 nendo kôkai waakushoppu.* 125-130.

Shibaraki, H. in Tamaoka, K., 2010. Constructing a formula to predict school grades 1-9 based on Japanese language school textbooks. *Nihongo kyôiku kôgakukai ronbunshi.* 33:4. 449-458.

Tateisi, Y., Ono, Y. in Yamada, H., 1988. A computer readability formula of Japanese texts for machine scoring. V *Proceedings of the 12th conference on Computational linguistics*, Morristown: ACL. 649--654.

Wessa, P., 2012. *Free Statistics Software*, Office for Research Development and Education, v. 1.1.23-r7. [http://www.wessa.net/]

Yamura-Takei, M., Aizawa, T. in Fujiwara, Mo., 2005. Diversity of zeros in Japanese discourse: A corpus analysis and a tool for language teachers. *Proceedings of PACLING 2005*, Tokyo: PACLING. 358-367.

Yoshihashi, K. in Nishina, K., 2007. Japanese composition support system displaying co-occurrences and example sentences. *Symposium on large-scale knowledge resources (LKR2007).* 119-122.

Kako dobro programi popravljajo vejice v slovenščini

Peter Holozan

Amebis, d. o. o.
Bakovnik 3, 1241 Kamnik
peter.holozan@amebis.si

Povzetek

Za slovenščino obstaja dva slovnična pregledovalnika, ki med drugim popravljata tudi napake pri vejicah v besedilu, to sta Besana in LanguageTool. S pomočjo zbrane baze primerov napačne rabe vejic (kjer je večina primerov pridobljena iz korpusa Šolar) je bil izračunan priklic za iskanje manjkajočih in odvečnih vejic za oba programa. Pokazalo se je, da oba programa kar dobro odkrivata manjkajoče vejice, pri čemer je Besana uspešnejša, ker zna opozoriti, da vejica manjka, tudi v primerih, ko sicer ne zna točno postaviti vejice. Pri odvečnih vejicah je Besana manj uspešna, LanguageTool pa jih sploh ne odkriva.

How well programs correct commas in Slovenian

There are two grammar checkers for Slovenian, Besana and Language Tool, and both are capable of correcting incorrect positions of commas in text. Thanks to the collection of examples of wrong usage of commas (the majority of the examples came from Šolar) recall for missing and superfluous commas was calculated for both programs. The results showed that both programs successfully detect missing commas, but Besana got better results because it is able to warn us when a comma is missing also in cases where it is not able to place a comma correctly. In superfluous commas Besana is less successful, but LanguageTool does not even detect them.

1 Uvod

Amebis že več kot 20 let razvija slovnični pregledovalnik Besana, ki med drugim popravlja tudi napake pri postavljanju vejic v slovenskih besedilih.

Takoj se zastavi vprašanje, kolikšen delež napak pri vejicah Besana odkrije. Vendar je na to vprašanje težko odgovoriti brez dovolj velikega korpusa besedil, v katerem bi bile označene napake pri vejicah.

S korpusom Šolar (Rozman et al., 2010), v katerem so napake označene, se je pokazala možnost za preizkus Besane in hkrati še za primerjavo s slovenskimi pravili za LanguageTool, odprtokodni slovnični pregledovalnik, ki je vključen tudi v LibreOffice.

Najprej bosta na kratko opisana Besana in LanguageTool.

Sledil bo opis korpusov Šolar (v katerem so zbrani šolski pisni izdelki in učiteljski popravki) in Kust (kjer besedila tujcev, ki se učijo slovenščino) (Rozman et al., 2010) in metode dela.

Na koncu bodo predstavljeni rezultati preizkušanja in možnosti za izboljšanje preizkušanja.

2 Besana

Besana[1] (kar je okrajšava za BESedna ANAliza) je slovnični pregledovalnik, ki ga razvija podjetje Amebis. Namenjen je iskanju napak v besedilih, in sicer predvsem slovničnih, to je takih, ki jih črkovalnik ne more odkriti.

Odkriva npr. neujemanje med pridevniki in samostalniki v sklonu spolu in številu, napačne sklone za predlogi, napačne variante predlogov s/z oz. k/h, napačne predloge pri krajevnih imenih, zanikanje s tožilnikom, napačno tvorbo trpnika, nekatere tipične neknjižne uporabe, napačno dvojino, napačne velike/male začetnice, presledke pri ločilih ipd. Pomemben del pa je tudi opozarjanje na napake pri vejicah, in sicer tako na manjkajoče kot odvečne vejice.

Besana lahko deluje kot samostojen program (Besana Mini) ali pa je vključena kot preverjanje slovnice v Microsoft Word ali LibreOffice. Dodatek pri Besani je še pregibnik, tj. program za pregibanje (spreganje, sklanjanje) besed.

Besana odkriva morebitne napake na dva načina: osnovni način je stavčni analizator, ki ima vgrajene tudi tipične napake (opis nekaterih tipičnih napak, ki jih sprejema analizator, in težav, ki lahko zaradi tega nastanejo, je v Holozan (2006)) oz. so tipične napake skupaj s podatki o vrstnih oznakah vgrajene že v leksikalno podatkovno zbirko ASES (Arhar, Holozan, 2009). Kadar pa analizatorju analiza ne uspe, uporabi Besana pomožna pravila, ki pa so vgrajena neposredno v kodo programa in jih uporabniki ne morejo prilagajati (lahko pa pri vseh vrstah napak nastavijo, ali želijo, da jih Besana opozarja nanje).

Prva verzija Besane (za okolje DOS) je bila napisana v jeziku C, zdaj pa je napisana v jeziku C++.

3 LanguageTool

LanguageTool[2] je odprtokodni program za preverjanje sloga in slovnice. Podpira angleščino, francoščino, nemščino, poljščino, nizozemščino romunščino in še množico drugih jezikov, med katerimi je tudi slovenščina. Odkriva napake, ki jih črkovalniki ne morejo. Program deluje s pravili, ki so narejena za vsak jezik posebej. Osnovna pravila so zapisana v formatu XML, podpira pa tudi kompleksna pravila, napisana v jeziku Java.

Prvotni avtor programa je Daniel Naber, ki je program razvil v okviru svojega diplomskega dela, zdaj pa pri razvoju sodelujejo tudi drugi razvijalci, še posebno pri pisanju jezikovno odvisnih pravil. Prva verzija je bila napisana v programskem jeziku Python, zdaj pa je napisana v jeziku Java.

Je prosto dostopen pod licenco LGPL. Glavni vzdrževalec za slovenska pravila je Martin Srebotnjak, 30. 6. 2012 je bilo za slovenščino 85 pravil (od teh 41 za manjkajoče vejice).

[1] http://besana.amebis.si

[2] http://www.languagetool.org/

```
<rulegroup name="Manjkajoča vejica pred 'zato'" id="ZATO_BREZ_VEJICE">
<rule>
<pattern mark_from="1" mark_to="-1" case_sensitive="yes">
<token regexp="yes" negate="yes">[,\(;:\–]|[Ii]n|ter|[Aa]li|[Ss]amo|[Ll]e|[Zz]golj|[Pp]redvsem|[Ll]ahko|[Aa]mpak|glavnem|[Mm]orda</token>
<token>zato</token>
<token negate="yes">,</token>
</pattern>
<message>Ponavadi je pred 'zato' vejica: <suggestion>, zato</suggestion>!</message>
<short>Najbrž manjka vejica pred 'zato'</short>
<example type="correct">Bilo je vroče<marker>, zato</marker> sva se slekli.</example>
<example type="incorrect">Bilo je vroče<marker> zato</marker> sva se slekli.</example>
</rule>
</rulegroup>
```

Slika 1. Primer pravila iz LanguageTool za odkrivanje manjkajočih vejic pred *zato*.

Na sliki 1 je primer pravila za manjkajoče vejice pred veznikom *zato*. Pravilo pravi, da če najde v besedilu besedo *zato*, pred katero ni ločil ,(;:- in ne besed *in, ter, ali, samo, le, zgolj, predvsem, lahko, ampak, glavnem, morda*, potem opozori, da najbrž manjka pred tem *zato* vejica.

LanguageTool lahko deluje kot samostojni program ali pa je vključen v LibreOffice oz. OpenOffice.

4 Priprava testnih podatkov

Večina primerov za napake pri postavljanju vejic je bila zbrana iz korpusa Šolar, del pa tudi iz korpusa KUST in drugih virov.

Osnovni namen zbirke napak pri vejicah je ugotavljanje priklica, ker vsebuje le povedi z napačno postavljenimi vejicami, ne pa tudi pravilnih povedi, kar bi bilo potrebno za izračun prave natančnosti. Natančnost se zato izračuna le nad primeri z napakami, kot nadomestek pa se lahko izračuna še nad temi primeri s popravljenimi napakami.

Za uporabnike slovničnih pregledovalnikov, ki sami poznajo slovnična pravila, natančnost ni tako zelo pomembna (seveda do neke mere, preveč lažnih napak vseeno postane moteče), saj sami hitro ocenijo, ali gre res za napačno postavljeno vejico in jim program predvsem pomaga pri primerih, ko spregledajo napačno rabo vejico; po drugi strani pa uporabniki, ki sami slovničnih pravil ne poznajo, lahko preveč zaupajo slovničnim pregledovalnikom in zanje bo pomembna tudi visoka natančnost.

4.1 Korpus Šolar

Korpus šolskih pisnih izdelkov Šolar je korpus besedil, ki so jih učenci slovenskih osnovnih in srednjih šol samostojno tvorili pri pouku. Zajeta so besedila, kjer je slovenščina materni jezik avtorjev, niso bila napisana posebej za projekt, ampak so del šolske produkcije, in jezikovni popravki so realni, kakršne so naredili učitelji (Rozman et al., 2010).

V korpusu so besedila učencev od 6. do 9. razreda osnovnih šol in od 1. do 5. letnika srednjih šol.

šola	število/delež besedil	število/delež besed
osnovna šola	505 / 18,7 %	133423 / 13,8 %
poklicno izobraževanje	183 / 6,8 %	52422 / 5,4 %
strokovno izobraževanje	843 / 31,2 %	261496 / 27 %
gimnazija	1172 / 43,3 %	520136 / 53,8 %

Tabela 1: Besedila v korpusu Šolar po šolah.

Tabela 1 prikazuje deleže po vrstah šol, ki so vključene v korpus.

82,3 % besedil je bilo zajeto pri predmetu slovenščina, druga pa pri drugih predmetih (npr. psihologija, sociologija, zgodovina), večina besedil je iz let 2009 in 2010. Pomemben kriterij pri gradnji korpusa je bila tudi regijska uravnoteženost (Rozman et al., 2010).

V korpus so vključeni tudi popravki napak, in sicer tako popravki, ki so jih naredili učitelji (razmerje med besedili, ki so imela učiteljske poprave, in nepopravljenimi besedili je 1459:1292), kot popravki, ki so jih vnesli sestavljavci korpusa (vendar je bilo to narejeno le nad delom korpusa).

4.1.1 Priprava primerov

Za pripravo podatkov so bili uporabljeni popravki iz korpusa, in sicer je bil narejen postopek, ki je vpisal manjkajoče vejice (z znakom ¤) in označil odvečne (z znakom ÷). Korpus je bil nato razdeljen na povedi in izločene so bile vse povedi, ki niso vsebovale odvečnih ali manjkajočih vejic.

Vendar se je kasneje izkazalo, da postopek ni bil popolnoma natančen. Tako je npr. kot odvečne označil vejice, ki sicer v resnici niso odvečne, vendar so bile v popravkih nadomeščene s pomišljajem ali s piko. Tudi pri odvečnih vejicah so bile težave, ker so bile v popravkih kdaj izbrisane posamezne besede ali pa je bilo kaj dodanega. Te primere je bilo treba ročno popraviti, delno si je dalo pomagati s tem, da je bil v takih primerih velikokrat stranski učinek odvečen presledek pred znakom za manjkajočo ali odvečno vejico. Del napak je najbrž ostal, vendar ročno preverjanje rezultatov ni pokazalo, da bi bilo tega zelo veliko, tako da to ne bi smelo bistveno vplivati na rezultate.

Na nekatere od teh napak (in verjetno tudi na napake v samem Šolarju, kjer so popravki za tisto, kar so opazili učitelji, kar pa niso nujno vse napake) pa je pokazal tudi pregled primerov, kjer Besana ni bila uspešna. Te napake so bile popravljene, tako da je zdaj baza primerov napak pri vejicah natančnejša, kot je sam Šolar, bi pa bilo smiselno nekoč v prihodnosti nazaj združiti to bazo primerov, pridobljenih iz Šolarja, s samim korpusom Šolar in tako po eni strani dobiti še seznam pravilnih primerov, da bo mogoče povedati pravo natančnost, po drugi strani pa se bodo pokazale še napake v korpusu Šolar.

4.2 Korpus KUST

Korpus KUST (korpus usvajanja slovenščine kot tujega jezika) je zbirka besedil, ki so jih napisali govorci drugih jezikov, ki se učijo slovenščine. Tak korpus je bil predlagan v (Stritar, 2006), besedila so bila zbrana v

okviru projekta ESS Uspešno vključevanje otrok, učencev in dijakov migrantov v vzgojo in izobraževanje, ki ga je izvajal Center za slovenščino kot drugi/tuji jezik Filozofske fakultete Univerze v Ljubljani (Rozman et al., 2010). Besedila so bila napisana na roko in pretipkana v okviru projekta, dodani so bili podatki o tipu dokumenta, vrsti in stopnji tečaja, na katerem je besedilo nastalo, starosti avtorja besedila, državi izvora, kraju bivanja in njegovem prvem jeziku. V besedilih so bili prekriti podatki, ki razkrivajo identiteto avtorja (če je besedilu npr. omenjeno ime Janez, je to nadomeščeno z XImeX, in to ne glede na sklon). V korpusu je skupaj 32.117 besed v 306 besedilih (Rozman et al., 2010).

jezik	delež
španščina	31,7 %
italijanščina	29,9 %
italijanščina dvojezično s slovenščino	1,8 %
angleščina	11,2 %
srbščina	8,4 %
nemščina	4,3 %
nemščina dvojezično s slovenščino	1,5 %
bosanščina	2,7 %
hrvaščina	0,8 %
makedonščina	3,7 %

Tabela 2: Deleži (glede na število besed) glede na prvi jezik avtorja v korpusu KUST.

Tabela 2 prikazuje deleže glede na prvi jezik avtorja, drugi jeziki imajo največ 1,8 % (Rozman et al., 2010). Starost avtorjev je bila med 13 in 21 let, večina jih je imelo med 16 in 19 let (Rozman et al., 2010).

Primeri iz korpusa KUST so bili vključeni zaradi domneve, da je tukaj več napak kot v besedilih rojenih govorcev in da bi lahko slovnični pregledovalnik po potrebi prilagodili, da bi bil še bolj uporaben za tujce, ki se učijo slovensko.

4.2.1 Priprava primerov

Za pripravo primerov je bilo treba izhajati iz nepopravljenih besedil, saj napake v korpusu KUST še niso označene. Besedilo je bilo razrezano na povedi in razporejeno glede na prvi jezik učenca, pri čemer sem združil hrvaščino, srbščino in bosanščino v isto skupino. Potem je bilo treba ročno poiskati manjkajoče in odvečne vejice, kar je vzelo precej časa, občasno pa je bilo tudi zapleteno razvozlati, kaj je pisec pravzaprav mislil. Na mesta, kjer manjkajo vejice, je bil dopisan znak ¤, odvečne vejice pa so bile nadomeščene z znakom ÷. Jeziki, za katere je bilo manj besedil v korpusu (nemščina, angleščina in srbščina/hrvaščina/bosanščina), so bili obdelani v celoti, kjer je bilo besedil veliko (italijanščina in španščina), je bila obdelana prva tretjina korpusa (ker bi bilo ročno popravljanje vejic preveč dolgotrajno in še več primerov najbrž ne bi bistveno vplivalo na rezultate), vsi drugi jeziki, kjer je bilo le malo besedil oz. besed v njih, pa so bili izpuščeni.

4.3 Drugi viri

Primeri iz korpusov Kust in Šolar so bili dodani že obstoječi bazi primerov za napake pri vejicah. Primeri v njej so bili zbrani iz različnih jezikovnih priročnikov, ki obravnavajo vejice, vaj, člankov na temo vejic (npr. (Šek Mertük, 2011)), diplomske naloge (Žibert, 2006) in drugih primerov za vejice, ki so bili zbrani med razvojem programa Besana v podjetju Amebis.

Vendar pa ti primeri ne odražajo pravega razmerja napak, ker je to tipično le izbor, v priročnikih je običajno več primerov za vejice, ki jih je teže postaviti, v besedilih se pa taki primeri ne pojavljajo tako pogosto. Zato niso tako uporabni za določanje uporabnosti slovničnih pregledovalnikov, seveda so pa uporabni za razvoj pregledovalnikov, saj je ravno pri težjih vejicah pomembno, da bi jih prav postavili.

Iz dela teh virov in iz primerov tretjega letnika gimnazije iz Šolarja je narejen krajši testni nabor, ki se ne uporablja pri izboljševanju Besane. Rezultati tega nabora služijo za preverjanje, da izboljšave niso preveč prilagojene na točno določene primere.

4.4 Priprava baze primerov

Baza je izvedena kot seznam vrstic s tremi stolpci, pri čemer je v prvem stolpcu oznaka kategorije (ki npr. pove, ali gre za primere, ki se bodo uporabljali za kontrolno skupino pri dodajanju pravil za vejico ali za osnovne podatke, ločuje pa tudi podatke iz korpusov Kust in Šolar od drugih podatkov).

V drugem stolpcu je številčni[3] podatek o viru. Primeri iz korpusa Kust so razdeljeni glede na prvi jezik učenca (nemščina, angleščina, španščina, italijanščina, srbščina/hrvaščina/bosanščina), primeri iz korpusa Šolar pa glede na razred oz. letnik in vrsto šole (6. do 9. razred osnovne šole, 1. do 3. in 5. letnik poklicne šole, 1. do 4. letni srednje strokovne šole in 1. do 4. letnik gimnazije). Možno bi bilo tudi združiti podatke iz korpusa Šolar v le štiri kategorije (osnovna šola, poklicna šola, strokovna šola in gimnazija), vendar me je zanimalo tudi, koliko se rezultati znotraj teh skupin ujemajo po letnikih, in ne le povprečki. Slabost te odločitve pa so bolj nepregledne tabele (seveda pa je možno podatke potem po potrebi združevati).

V tretjem stolpcu so primeri stavkov. Manjkajoče vejice so označene z znakom ¤, odvečne pa nadomeščene z znakom ÷.

Baza je izvedena kot preglednica v programu, da pa jo programi potem laže uporabljajo, se naredi izvoz v besedilno obliko, kjer so vrstice ločene s kodo za novo vrstico, stolpci pa s kodo za tabulator.

```
5	25	Ta hčerka še ni pri kruhu¤ saj se šola kot frizerka v Mariboru.
5	25	Veliko časa tudi preživim pred televizijo÷ ter me zanimajo znanstvene odajo.
5	25	Nato ju je policija izsledila v kampu¤ vendar predrzni najstnici sta ušle¤ zatekli sta se zapuščenim otrokom.
5	25	Če ju opazite¤ se javite na policiski postaji.
```

Slika 2. Izsek iz baze primerov (8. razred OŠ iz korpusa Šolar).

[3] Številčne oznake so bile izbrane, da lahko program neposredno zapisuje rezultate v tabelo. Dodana je manjša baza z legendo oznak virov.

Na sliki 2 je delček primerov, zbranih iz korpusa Šolar. Opazno je, da je v besedilu tudi veliko drugih napak, torej mora biti postopek za odkrivanje napak pri vejicah čim bolj odporen na druge napake.

vir	število manjkajočih vejic	število odvečnih vejic
testni nabor	795	203
KUST	401	104
Šolar - osnovna šola	2332	415
Šolar - poklicno izobraževanje	1285	189
Šolar - strokovno izobraževanje	4310	985
Šolar - gimnazija	3362	1155

Tabela 3: Število zbranih primerov.

V tabeli 3 je zbrano število primerov napak pri vejicah po kategorijah. Primeri iz korpusa Šolar močno prevladujejo.

4.4.1 Opcijske vejice

V nekaterih primerih je slovnično možno oboje: da vejica bodisi je bodisi je ni. Take vejice največkrat vplivajo na pomen besedila oz. vsaj na poudarek.

Pravilno postavljanje takih vejic za zdaj presega zmožnosti računalniškega preverjanja (možno bi bilo kvečjemu opozarjanje na morebitno dvoumnost v takih primerih), zato taki primeri niso bili uvrščeni v bazo primerov.

Posebni primer so vejice pri datumih v primerih: *v nedeljo, 9. maja (bo)*, kjer Slovenski pravopis 2001 dopušča tudi pisanje brez vejice. Besana v tem primeru opozarja, če vejice ni, je pa to ločena kategorija napake, tako da je mogoče opozarjanje na vejice v takih primerih preprosto izključiti, če je uporabnik ne želi uporabljati.

5 Postopek preizkušanja

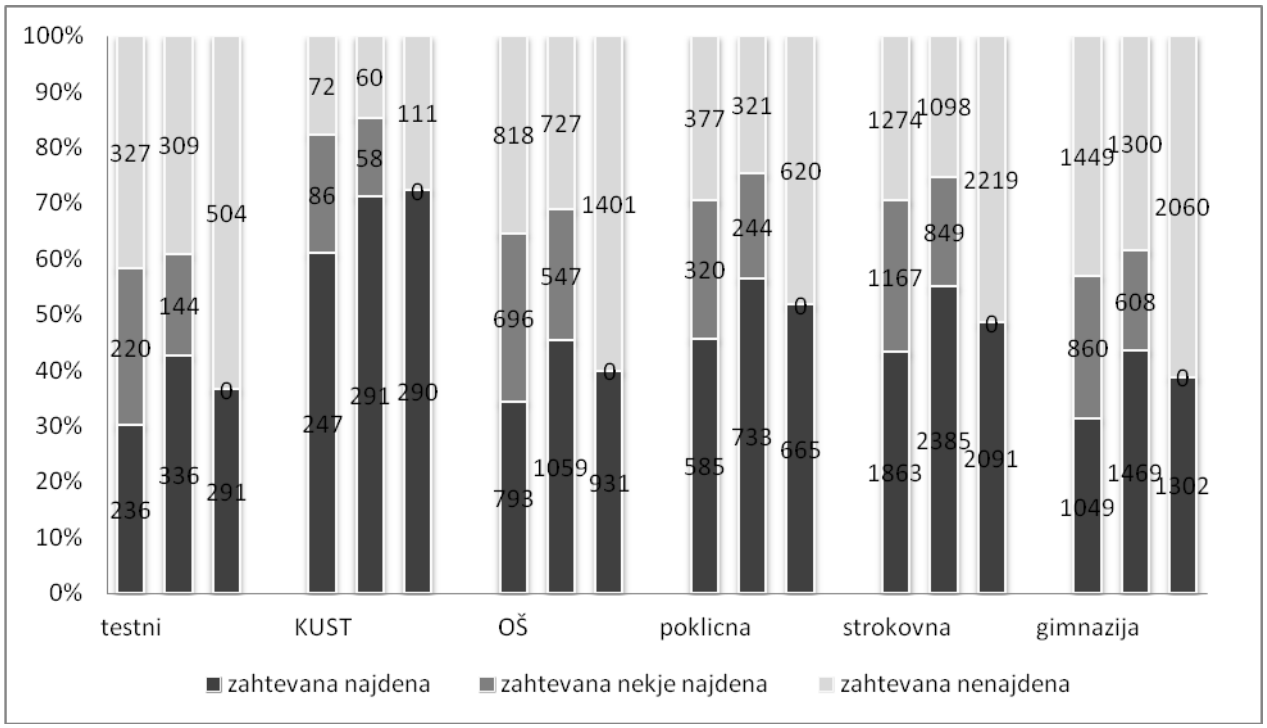
Za preizkušanje je bil uporabljen program Vejicar, ki je sicer namenjen preizkušanju zanesljivosti Besane. Program uporabi bazo primerov, kjer so označene manjkajoče in odvečne vejice (z znakoma ¤ in ÷). Za vsak primer najprej nadomesti znak za odvečno vejico (÷) z vejico in izbriše znak za manjkajočo vejico (¤). Stavek spusti skozi Besano in ugotovi, ali je Besana pravilno opozorila na manjkajoče in odvečne vejice. Rezultate ob skupnem rezultatu izračunava še za posamezne vire in kategorije.

Program Vejicar je bil dopolnjen še za uporabo rezultatov LanguageTool, in sicer tako, da so bila pravila v LanguageTool popravljena tako, da dodajajo pri manjkajoči vejici znak ¤ namesto vejice, datoteka, ki je vsebovala primere (vsakega v svoji vrstici), je bila pognana skozi program (pri čemer so bila vključena le pravila za vejice) in rezultati so bili dodani kot dodatni stolpec v vhodu programa Vejicar.

6 Rezultati

Program Vejicar izpiše rezultate v datoteko, in sicer v obliki tabele, v kateri so elementi ločeni s tabulatorjem, tako da je rezultate preprosto prek odložišča prenesti v preglednico v Excelu, kjer se potem naredijo dodatne obdelave.

Ker je bilo v zadnjem letu iskanje napak pri vejicah v Besani precej izpopolnjeno (tudi po zaslugi zbrane baze primerov), sta bila zabeležena dva rezultata Besane, ki sta narazen 9 mesecev, v tretjem stolpcu pa je rezultat LanguageTool. Korpus Šolar je razdeljen glede na vrsto šole.



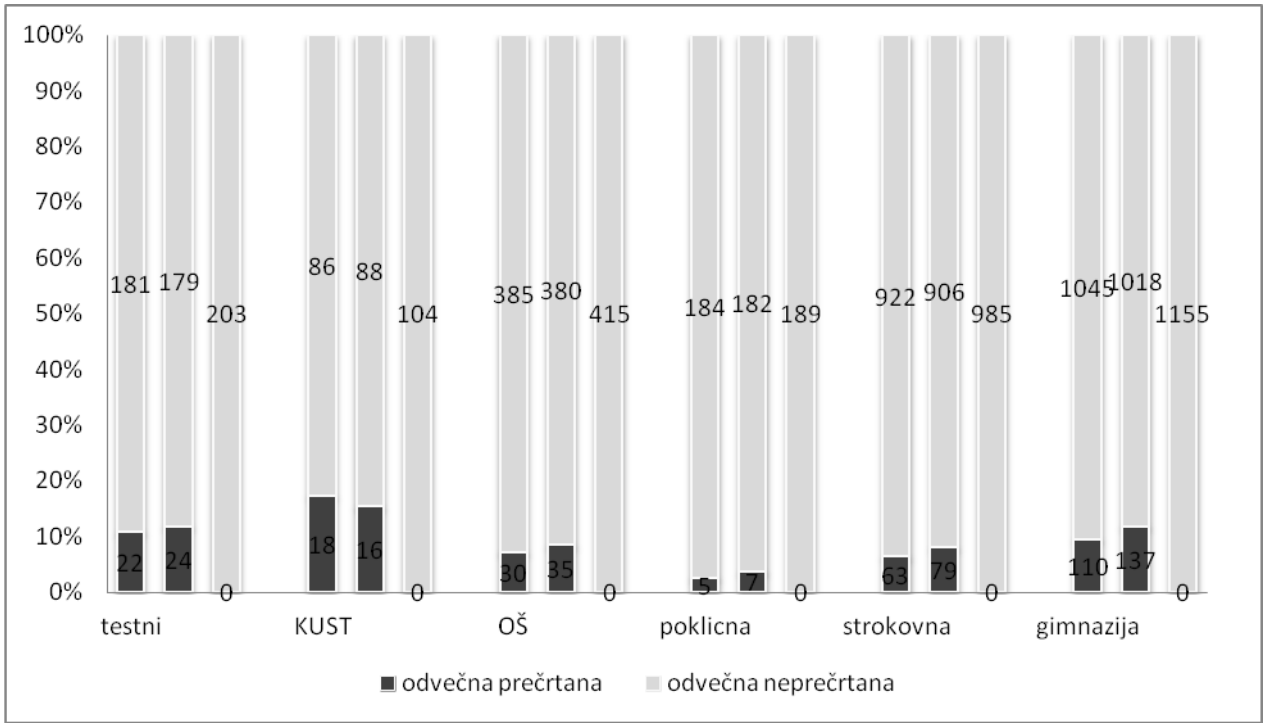
Slika 3: Priklic manjkajočih vejic (po stolpcih: Besana septembra 2011, Besana julija 2012, LanguageTool maja 2012)

Besana opozarja na manjkajoče vejice na dva načina. Pri prvem točno določi mesto, kje manjka vejice, največkrat je to na veznikih, ki zahtevajo vejico. Pri drugem načinu pa le opozori, da vejica manjka nekje v stavku, ne zna pa ugotoviti, kje to je. Do takih opozoril pride npr. v primerih, ko najde v stavku dve osebni glagolski obliki brez vmesnega veznika ali vejice, kar je jasen znak, da bi morala nekje vmes biti vejica.

Rezultat kaže, da ravno ta opozorila, kjer vejica ni natančno najdena, naredijo večino razlike med Besano in LanguageTool. Rezultati obeh programov pa so precej odvisni od piscev, oba bolje delujeta v primeru, ko znajo pisci slabše slovenščino in pozabljajo na vejice pred tipičnimi vezniki.

Natančnost opozarjanja na manjkajoče vejice (le nad primeri z napakami pri vejicah, tako odvečnimi kot manjkajočimi) je pri Besani 87,8 %, pri LanguageTool pa 81,5 %. Za pravo natančnost bi bila potrebna testna baza, ki bi vsebovala tudi povedi s pravilnimi vejicami.

Uporaba korpusa Šolar pa je pokazala, da niso težava le manjkajoče vejice, ampak tudi odvečne. V primerih iz korpusa Šolar skupaj manjka 11340 vejic, odvečnih pa je 2744, kar pomeni, da je v skoraj 20 % primerov problematičnih vejic težava to, da je vejica odveč. Če dodamo še to, da Besana popravi 60 do 70 % manjkajočih vejic, je potem odvečnih vejic skoraj toliko kot manjkajočih, kajti rezultati programov pri odvečnih vejicah so precej slabši kot pri manjkajočih.



Slika 4: Priklic odvečnih vejic (po stolpcih: Besana septembra 2011, Besana julija 2012, LanguageTool maja 2012)

LanguageTool sploh ne išče odvečnih vejic in tudi Besana jih najde le okoli 10 %. To kaže, da je pri odvečnih vejicah še veliko prostora za izboljšave slovničnih pregledovalnikov.

Zanimivo pa je, da ima Besana pri odvečnih vejicah boljši rezultat pri gimnaziji kot pri poklicni oz. strokovni šoli, kar je ravno obratno kot pri manjkajočih vejicah.

7 Kako naprej

Treba bi bilo zbrati še bazo primerov, kjer bi bile vključene tudi pravilne povedi, da bi bilo mogoče izračunati še natančnost popravljanja vejic. Pri izdelavi take baze je treba biti previden, saj kakovost lekture besedil zelo vpliva na rezultat natančnosti, saj v primeru, da za preizkušanje uporabimo besedila, kjer skoraj ni napak, dobimo zelo slab rezultat za natančnost (Helfrich, Music, 2000). Zato je treba zbrati realna besedila, ki še niso bila lektorirana, po možnosti taka, kjer obstaja tudi lektorirana verzija, da ni treba ročno iskati napak. Idealno morajo biti pisci čim bolj raznoliki, z različnim jezikovnim znanjem.

V bazo primerov bi bilo smiselno dodati tudi posebej označene primere za opcijske vejice, torej vejice, ki lahko so ali ne, pri čemer se bolj ali manj spremeni pomen. Zbirka teh primerov bi omogočila preizkušanje tudi takih primerov in morda razvoj postopkov za opozarjanje na morebitne dvoumne primere.

Naslednji korak je podrobnejša analiza, na katerih mestih tipično manjkajo oz. so odveč vejice, s pomočjo česar bo mogoče dopolniti pravila za iskanje manjkajočih oz. odvečnih vejic.

V prihodnosti bo treba v slovničnih pregledovalnikih posvetiti večjo pozornost odkrivanju odvečnih vejic, saj kaže, da je tudi to pogosta težava pri pisanju. Zanimivo pri odvečnih vejicah je, da so v 30 % do 50 % pred vezniki, ki sicer tipično zahtevajo vejico (npr. *ko*, *ki*, *da*), a je v teh primerih ne sme biti, npr. zaradi tega, ker je spredaj kakšen drug veznik ali pa členek (ta podatek je iz pripravljalnega poskusa podrobnejše analize).

Po drugi strani pa bilo smiselno uporabiti korpus Šolar tudi za celovit preizkus slovničnih pregledovalnikov, ki se ne bi omejil le na napake pri vejicah, ampak bi zajel vse napake. Bo pa v tem primeru, posebno pri slogovnih popravkih, treba paziti na to, da popravki niso nujno absolutni, ampak bi tudi dva lektorja različno popravila isto besedilo, kar pomeni, da bo težko narediti popolnoma samodejno preizkušanje, ampak bo treba rezultate ocenjevati ročno, kar bo žal omejilo velikost vzorca, ki ga bo mogoče preizkusiti.

8 Sklep

Rezultat nad testnim naborom je naslednji:

kategorija	Besana	LanguageTool
priklic manjkajočih vejic	60,8 %	36,6 %
natančnost pri manjkajočih vejicah	87,8 %	79,0 %
priklic odvečnih vejic	11,8 %	/
natančnost pri odvečnih vejicah	100 %	/

Tabela 4: Rezultati nad testnim naborom.

Rezultati nad testnim naborom so nekaj slabši, kot so sicer rezultati nad korpusom Šolar, še posebej pa nad korpusom KUST, kar kaže na to, da je v testnem naboru večji delež napak, ki jih je težko odkriti.

Tako Besana kot LanguageTool kar uspešno odkrivata manjkajoče vejice v slovenskih besedilih, Besana jih odkrije dobrih 20 % več, in to predvsem na račun opozoril, kjer ne zna točno postaviti vejice, ve pa, da nekje manjka. Težava pri teh opozorilih pa je, da mora uporabnik dovolj dobro poznati pravila za postavljanje vejic, da lahko sam postavi vejico na pravo mesto. Prav tako pa niso uporabna za popolnoma samodejno postavljanje vejic (npr. pri razpoznavi govora), zato bi bilo dobro v čim več primerih določiti pravo mesto, kje manjka vejica. Spremembe Besane v zadnjem letu so bile izrazito v tej smeri, kar kažejo tudi rezultati na sliki 3 (pred temi popravki je imel LanguageTool boljši rezultat kot Besana, če bi upoštevali le natančno določene manjkajoče vejice).

Besana je tudi bolj natančna pri opozarjanju na manjkajoče vejice, brez nove baze primerov pa ni mogoče reči, kakšna je v resnici natančnost obeh programov.

Odvečne vejice delajo težave obema programoma, LanguageTool jih sploh ne odkriva, Besana pa jih odkrije le okoli 10 %.

Zanimivo bi bilo rezultate za slovenščino primerjati z rezultati za druge jezike, vendar se tu pojavi težava, da večinoma preizkušajo programe za preverjanje slovnice v celoti in pri rezultatih niso posebej navedeni rezultati za vejice. Primer, kjer so rezultati za vejice navedeni ločeno, je za latvijščino v Deksne, Skadiņš (2011), kjer so med tipi najdenih napak navedeni štirje tipi, ki zadevajo vejice. Priklic je med 14 % in 56,3 %, natančnost pa med 70,4 % in 91,3 %. Žal pa ni navedeno, koliko je primerov za posamezen tip, zato ni mogoče izračunati skupnega rezultata za vejice. Ni pa tudi jasno, ali gre le za manjkajoče ali tudi za odvečne vejice, vprašanje je, ali navedeni štirje tipi pokrivajo vse problematične vejice, zaradi različnih pravil o postavljanju vejic pa so rezultati tudi težko primerljivi.

9 Literatura

Arhar, Š., Holozan, P., 2009. ASES – leksikalna podatkovna zbirka za razvoj slovenskih jezikovnih tehnologij. V V. Mikolič (ur.), *Jezikovni korpusi v medkulturni komunikaciji*. Koper: Založba Annales.

Deksne, D., Skadiņš, R. 2011. CFG Based Grammar Checker for Latvian. V *Proceedings of the 18th Nordic Conference of Computational Linguistics NODALIDA 2011*. Riga, Latvia.

Helfrich, A., Music, B. 2000. Design and evaluation of grammar checkers in multiple languages. V *Proceedings COLING '00 Proceedings of the 18th conference on Computational linguistics - Volume 2*. Stroudsburg, PA, ZDA: Association for Computational Linguistics.

Holozan, P., 2006. Dodatne dvoumnosti zaradi popustljivosti analizatorja pri analizi slovenskih stavkov. V T. Erjavec in J. Žganec Gros (ur.), *Zbornik 5. slovenske in 1. mednarodne konference JEZIKOVNE TEHNOLOGIJE 2006*. Ljubljana: IJS.

Rozman, T., Stritar, M., Krapš Vodopivec, I., Kosem, I., Krek, S., 2010. *Nova didaktika poučevanja slovenskega jezika : sporazumevanje v slovenskem jeziku*. Ljubljana: Ministrstvo za šolstvo in šport: Amebis. http://www.slovenscina.eu/Media/Kazalniki/Kazalnik15/Nova_didaktika_Sporazumevanje.pdf.

Stritar, M., 2006: Oblikovanje korpusa usvajanja slovenščine kot tujega jezika, V T. Erjavec, J. Žganec Gros (ur.), *Zbornik 5. slovenske in 1. mednarodne konference Jezikovne tehnologije 2006*. Ljubljana: IJS.

Šek Mertük, P., 2011: Vejica premalo ali preveč pri študentih razrednega pouka. *Revija za elementarno izobraževanje*. Letnik 4, št. 1–2..

Žibert, Ž., 2006. *Slovenska vejica: balast ali skladenjska nujnost slovenskega knjižnega jezika?*. Diplomska naloga. Ljubljana: Fakulteta za družbene vede, Univerza v Ljubljani.

Umetno tvorjenje slovenskega govora s pomočjo odprto kodnih orodij ter prikritih Markovovih modelov

Tadej Justin[1], France Mihelič[1], Janez Žibert[2]

[1] Univerza v Ljubljani,Fakulteta za elektrotehniko, LUKS, Tržaška 25, 1000 Ljubljana
{tadej.justin, france.mihelic}@fe.uni-lj.si
[2] Univerza na Primorskem, FAMNIT, Glagoljaška 8, 6000 Koper
janez.zibert@upr.si

Povzetek

Na področju tvorjenja umetnega govora (sinteze) se je do danes uveljavilo veliko prosto dostopnih orodji, ki omogočajo gradnjo sistemov za umetno tvorjenje govora (sintetizatorjev). Programska orodja omogočajo s pomočjo vgrajenih uveljavljenih postopkov na uporabniku prijazen način udejanjiti lasten sintetizator govora. Ker so takšni sistemi močno odvisni od jezika, pri tovrstnih orodjih ne najdemo potrebne podpore za gradnjo slovenskega sintetizatorja, kakor tudi ni mogoče zaslediti udejanjenega glasu za slovenski govor. S pomočjo odprto-kodnih programskih orodji za obdelavo govora, obdelavo besedila in njuni analizi, je z nekaj truda veščemu uporabniku omogočeno izdelati tudi jezikovno podporo. S pomočjo analize besedila je mogoče posneti majhno zbirko govora. Z ustrezno urejeno zbirko govora pa je omogočeno, da lahko izgradimo sintetizator govora. V tem prispevku želimo predstaviti prosto dostopna programska okolja za gradnjo sintetizatorjev s pomočjo prikritih Markovovih modelov (PMM), kakor tudi ključne korake za izdelavo slovenske jezikovne podpore ter zajem lastne zbirke slovenskega govora za namen gradnje lastnega sintetizatorja.

Development of slovenian HMM speech synthesis system with the use of open source software

Nowadays in the field of speech synthesis there are many open source frameworks, which allow us to build the text to speech systems. Some of them obtain also the user friendly interface, which allow the inexperienced user to build its own text to speech systems. Since such systems are well dependent on the language, the each language support have to be individually investigated. Currently there is no available language support for Slovenian language and also we did not find any open source Slovenian voice. With some basic knowledge of speech and natural language processing skilled user can build the new language support and based on written language data record small speech database. With obtained database the HMM based text to speech system can easily be build. With this article we want to present the open source frameworks and other helpful software for building the HMM based speech synthesis systems. We present the key steps for building the Slovenian language support, recording small Slovenian speech database and the basic steps, how to obtain the HMM text to speech system with the recorded data.

1. Uvod

Za končne uporabnike je na trgu mogoče najti veliko komercialno dostopnih orodij, ki omogočajo kvalitetno tvorjenje umetnega govora. Poleg takih so prisotna tudi orodja, ki razvijalcem omogočajo enostaven razvoj sintetizatorjev govora. Ker so taki sistemi močno odvisni od jezika, v večini primerov slovenskega jezika ni med podprtimi jeziki. Le malo je razvijalcev, ki se odločijo za gradnjo potrebne jezikovne podpore, ter njeno implementacijo v prosto dostopna programska okolja. Poleg jezikovne podpore je tudi predpogoj za izgradnjo slovenskega sintetizatorja fonemsko in količinsko bogata zbirka slovenskega govora, kar razvijalcem velikokrat predstavlja dolgotrajno ter drago pridobivanje tovrstnih podatkov. To je le nekaj navedb, zakaj med prosto dostopnimi sintetizatorji, še ni takega, ki bi omogočal slovensko sintezo. Avtorji menimo, da ravno prosto dostopna orodja omogočajo poglobljeno poznavanje takih sistemov, saj nudijo uporabniku natančen vpogled tudi v izvorno programsko kodo. V zadnjem času razvojna orodja v veliki meri temeljijo na strokovni javnosti uveljavljenemu postopku za umetno tvorjenje govora s pomočjo prikritih Markovovih modelov (ang. Hidden Markov Models), v nadaljevanju PMM. Njegova uporaba ne zahteva velike procesorske in pomnilniške zmogljivosti računalniških sistemov, kljub temu pa dosega solidne rezultate pri tvorjenju naravnega in razumljivega govora. Med orodji za razvoj tovrstnih sistemov, ki so v zadnjem času med bolj zastopanimi, velika večina temelji na orodju HTS[1] (Yoshimura et al., 1999), ki je izdan kot sistemski popravek za orodje HTK (Young et al., 2006). Orodje ne omogoča za sintezo nujno predpripravo od jezika odvisnih parametrov, vseeno je često uporabljen kot glavna komponenta za izgradnjo sintetizatorjev govora s pomočjo PMM. Med prosto dostopnimi orodji, ki imajo vgrajeno orodje HTS, sta nedvomno najbolj zastopana Festival[2] in Mary TTS[3] (Schröder et al., 2011). Prvega so razvili raziskovalci centra za raziskovanje govornih tehnologij na univerzi v Edinburgu, drugi pa je delo skupine raziskovalcev v laboratoriju za jezikovne tehnologije na nemškem raziskovalnem inštitutu za umetno inteligenco DFKI. S poznavanjem komponent tako enega kot drugega je mogoče udejanjiti tudi jezikovno podporo, ki je predpogoj za gradnjo sintetizatorja s pomočjo PMM.

V tem prispevku želimo spodbuditi razvoj odprto kodnih slovenskih sintetizatorjev govora, kakor tudi predstaviti potrebne korake za gradnjo jezikovne podpore za slovenski jezik. S pomočjo analize besedila nad slovenskim leposlovjem, predstavljamo tudi enostavno snemanje fonetično bogatih povedi, s katerimi lahko slovenski go-

[1] http://hts.sp.nitech.ac.jp/
[2] http://www.festvox.org/festival
[3] http://mary.dfki.de

vorci enostavno posnamejo govorno podatkovno zbirko ter s pomočjo prej naštetih orodji zgradijo lastni sintetizator govora.

2. Gradnja slovenske jezikovne podpore za sintezo govora

Ključna za izgradnjo sintetizatorja je ustrezno pridobljena in označena govorna zbirka. V tem prispevku se osredotočamo na pridobivanje ter analizo besedila lektorirane slovenske pisane besede.

2.1. Pridobivanje slovenskega besedila

V splošnem nam je pisana beseda sama po sebi na voljo na svetovnem spletu. Z razvojem spletnih programskih jezikov so se uveljavili tudi spletni "čitalci" (ang. parsers), ki predstavljajo nepogrešljiv pripomoček pri zbiranju besedila iz svetovnega spleta. Vseeno se je potrebno vprašati o kakovosti pridobljenega slovenskega besedila ter nenazadnje tudi o intelektualni lastnini, avtorstvu in z njim povezanim pravom.

Pri gradnji obsežnejših zbirk besedila iz svetovnega spleta velikokrat pridobimo nekakovostne podatke, ali pa take, ki jih je potrebno ročno obdelati. Da bi si prihranili dragocen čas namenjen ročnemu pregledovanju zbirke besedil, smo se odločili, da besedilo za naše potrebe pridobimo s pomočjo slovenskega leposlovja. Tako besedilo predstavlja lektorirano besedilo, kjer ob enostavnih pravilih izločanja tujih besed, znakov ter številk, lahko pridobimo dovolj dober čistopis, ki nam nudi osnovo za analizo slovenske pisane besede,hkrati pa predstavlja tudi zbirko za pridobivanje fonetično bogatih sklopov povedi, s katerimi lahko posnamemo dovolj bogato fonetično zbirko slovenskega govora za namen gradnje sintetizatorja.

2.2. Analiza pridobljenih povedi

Pridobljen čistopis smo razdelili v tri skupine. Prva zajema povedne povedi (zaključijo se s piko) ter nedokončane povedi (zaključijo se s tremi pikami). Druga vzklične povedi (zaključijo se s klicajem) ter tretja vprašalne povedi (zaključijo se z vprašajem). Taka razdelitev je bila vnaprej predvidena, saj si pri gradnji sintetizatorja želimo zajeti tudi raznoliko stavčno intonacijo, ki omogoča, da je umetni govor bolj naraven. Zato moramo pri zajemu zbirke govora vključiti tudi primerno število povedi vseh treh skupin. Da bi iz čistopisa avtomatsko pridobili čimbolj kakovostne povedi za snemanje zbirke govora, smo izločiti tudi povedi, ki so prekratke ali predolge. Za prve velja, da niso fonetično dovolj bogate. Pri drugih pa je mogoče našteti dva vzroka za njihovo izločitev. Prvi je pogojen z natančnostjo avtomatske poravnave (ang. forced-aligment) govornih enot v akustičnem signalu s fonemskim prepisom besedila, saj je ob predolgih povedih podvržen večji napaki. Drugi vzrok pa izhaja iz uporabniku prijaznega zajema zbirke govora, saj neprofesionalno branje enostavnejših povedi pripomore k boljšemu razumevanju ter posledično boljšemu izražanju. Celoten skupek besedila smo obdelali, tako da smo izločili povedi, ki obsegajo manj kot tri in več kot petnajst besed. Količina povedi v posameznih skupinah je prikazana v tabeli 1.

2.3. Fonemska analiza povedi

Vez med pisanim besedilom (grafemi) ter akustičnim signalom lahko predstavimo z zapisom izgovorjene besede s pomočjo fonetike. Plod dolgoletnega dela sodelavcev v Laboratoriju za umetno zaznavanje, sisteme in kibernetiko na Fakulteti za elektrotehniko, Univerze v Ljubljani, je tudi program za avtomatsko fonemsko grafemsko pretvorbo, ki jo je mogoče uporabljati za namen sinteze slovenskega govora, kakor tudi za potrebe sorodnega področja razpoznavanja slovenskega govora (Gros, 1997). Sestoji iz fonetičnih pravil in skrbno ročno pridobljenega slovarja z več kot 35.000 besedami. S pomočjo programa tako pridobimo fonemski zapis posameznih besed. Za potrebe tega prispevka smo realizirali pretvorbo s 43-timi fonemi ter jim dodali oznako za premor.

V kolikor ne bi razpolagali z avtomatsko grafemsko fonemsko pretvorbo, bi jo lahko z uporabo naprednih algoritmov za obdelavo jezika, ki jih omogočata tudi programska okolja Mary TTS in Festival pridobili avtomatično. Predpogoj za učenje odločitvenih dreves za avtomatsko odločanje nad pravili fonetike je skrbno pripravljen grafemsko fonemski slovar z nekaj več tisoč vnosi ter skrbno določene značilke posameznih fonemov jezika.

Pri gradnji sintetizatorjev s pomočjo PMM, se za modeliranje običajno uporablja trifonske govorne enote. Ti predstavljajo skupek treh povezanih monofonov. S tem pa se število vseh možnih govornih enot bistveno poveča in v našem primeru znaša 85184 vseh možnih trifonov. V kolikor si želimo, da udejanjimo kvaliteten sintetizator, je pomembno, da razpolagamo z zbirko govora, ki ima čim večjo raznolikost možnih trifonov. V tabeli 1 lahko preverimo tudi trifonsko zastopanost podskupin čistopisa slovenskega leposlovja. S pomočjo trifonske zastopanosti posameznih sklopov povedi lahko pridobimo lestvico najbolj pogostih trifonov v obravnavani zbirki slovenskega besedila, kar lahko s pridom uporabimo pri izbiri manjšega števila fonemsko bogatih povedi za snemanje zbirke slovenskega govora.

2.4. Izbira fonemsko bogatih povedi za snemanje govorne zbirke

Izbira manjšega nabora povedi v namen snemanja govornih signalov, ki so najbolj značilne za obravnavan jezik in so hkrati tudi fonemsko bogate ni enostavna naloga. V literaturi se pojavljata dva pristopa. Prvi temelji na naključni izbiri povedi, toliko časa, da skupek povedi zajame vse osnovne govorne enote. Drugi se obračajo na difonsko analizo ter difonsko prozodično porazdeljenost. Pri zadnjem mora uporabnik sam definirati končno število povedi, ki jih želi pridobiti, kakor tudi mora vsakemu monofonu pripisati fonemske značilnosti.

V literaturi (Yamagishi et al., 2010) je možno zaslediti, da je mogoče udejanjiti sintetizator s pomočjo PMM z govorno zbirko s skupno dolžino manj kot 15 minut govora istega govorca. Pri čemer mora govorna zbirka vsebovati vse monofone, obenem pa je potrebno težiti tudi k čim večji trifonski zastopanosti. Da bi navedeno preverili tudi za slovenski jezik, smo povedi, s katerimi smo posneli govorno zbirko izbrali na naslednji način. Najprej smo vsak sklop (povedne, vzklične in vprašalne) povedi dodatno ure-

Tabela 1: Analiza čistopisa zbirke povedi slovenskega leposlovja

Analizirana postavka	Povedne povedi	Vprašalne povedi	Vzklične povedi	Skupaj
Št. povedi	166223	14552	9891	190666
Št. edinstvenih besed	106507	12462	17811	112126
Št. vseh besed	1434211	66694	98548	1599453
Št. edinstvenih trifonov	28218	15336	17591	28673
Procent zastopanosti trifonov	33,12	18,00	20,65	33,66
Št. vseh trifonov	6580055	280368	432537	7292960

Tabela 2: Analiza sklopa izbranih 200 povedi za snemanje zbirke slovenskega govora

Analizirana postavka	Povedne povedi	Vprašalne povedi	Vzklične povedi	Skupaj
Št. povedi	160	20	20	200
Št. edinstvenih besed	115	115	111	102
Št. vseh besed	1476	152	139	1767
Št. edinstvenih trifonov	3568	610	577	4004
Procent zastopanosti trifonov	4,19	0,72	0,68	4,7
Št. vseh trifonov	7014	707	605	8376

dili v pod-sklope glede na količino besed. Vsaki povedi smo pripisali monofonsko zastopanost in jih uredili po velikosti od največje do najmanjše zastopanosti. Iz vsakega pod sklopa smo proporcionalno glede na celotno zastopanost posamezne skupine povedi v zbirki naključno izbrali 160 povednih povedi, 20 vzkličnih povedi ter 20 vprašalnih povedi. Pri tem smo upoštevali, da smo iz vsakega pod sklopa naključno izbirali le v prvi polovici (bolj zastopane) monofonske zastopanosti. Zastopanost sklopa povedi prikazuje tabela 2.

3. Zajem zbirke govora

Za snemanje govora na podlagi besedilne zbirke smo uporabili program Red Start, ki je del programskega orodja Mary TTS. Program omogoča snemanje predvidenih povedi tako, da uporabniku nudi velik izpis povedi, omogoča enostaven izris posnetega akustičnega signala v časovnem in frekvenčnem prostoru, detekcijo šuma, ter večkratno ponavljanje snemanja izbrane povedi. S snemanjem zbirke smo pridobili približno 10 min govora.

4. Splošen sistem PMM sistema za umetno tvorjenje govora

Za udejanjanje lastnega sintezizatorja s pomočjo tehnike PMM je potrebno poznavanje orodja HTS ter njegovih komponent. V splošnem sestoji iz treh ločenih komponent, ki si sledijo po vrstnem redu; analiza govornega signala, učni proces sintetizatorja ter generiranje akustičnega signala (sinteza).

4.1. Priprava govornih signalov za gradnjo PMM sintetizatorja

Vsak akustični signal, ki je namenjen gradnji PMM sintetizatorja mora biti opremljen z grafemskim prepisom. S pomočjo grafemsko fonemske pretvorbe lahko pridobimo tudi fonemski zapis akustičnega signal. Za uspešno gradnjo PMM sintetizatorja moramo priskrbeti tudi časovno poravnavo med fonemi in akustičnim signalom. V kolikor imamo na razpolago ročno označeno fonemsko poravnano, lahko pričakujemo boljšo kvaliteto sintetiziranega govora, kar nakazuje, da natančna poravnava fonemov in akustičnega signala igra pomembno vlogo pri gradnji PMM sintetizatorja. Velikokrat ne razpolagamo z ročno označeno zbirko govornih signalov. V takem primeru moramo uporabiti avtomatsko označevanje trajanja posameznih fonemov vsakega govornega signala v zbirki. Postopki za avtomatično označevanje niso tako natančni kot ročna poravnava, vendar je natančnost pri primerno obsežnem govornem signalu zadovoljiva. Avtomatično poravnavo govorne zbirke lahko izvedemo z orodjem EHMM (Prahallad et al., 2006) ali kar neposredno z uporabo orodja HTK.

Ko pridobimo fonemsko poravnavo z govornim signalom, lahko pričnemo s potrebnim luščenjem značilk govornega signala. Moderni sistemi omogočajo luščenje spektralnih značilk, kot tudi značilk govornega trakta. Najbolj pogosto uporabljene spektralne značilke so koeficienti melodičnega kepstra (ang. mel frequency cepstral coefficinets, MFCC). Pridobimo jih lahko s pomočjo orodja SPTK (Takuda et al., 2009) ali ESP (Black et al., 2003). Za pravilno modeliranje osnovne frekvence potrebujemo tudi ustrezna orodja. Za to je na voljo več prosto dostopnih programov, v našem delu smo uporabili orodje Praat (Boersma and Weenink, 2001).

4.2. Gradnja PMM sintetizatorja

Učni proces je podoben učnemu procesu pri razpoznavanju govora. Spektralne značilke ter značilke vzbujanja (kot na primer logaritem osnovne frekvence ter njene dinamične zančilke ali trajanje posamezne osnovne enote) se pridobivajo iz zbirke govornih posnetkov in so modelirane z množico kontekstno odvisnih PMM. Parametre PMM modelov nato na podlagi teh značilk določimo po kriteriju maksimalnega verjetja (ang. maximum likelihood), v nadalje-

vanju ML,

$$\hat{\boldsymbol{\lambda}} = arg\,max_{\lambda}\,\{p(\boldsymbol{O}|\boldsymbol{W},\lambda)\}, \quad (1)$$

kjer je $\boldsymbol{\lambda}$ vektor parametrov modela,$\boldsymbol{O}$ vektor učnih podatkov in $\boldsymbol{W}$ vektor zaporednih manjših besednih enot, ki pripadajo vektorju O. Parametri govora, $o = o_1, ..., o_T$, se tvorijo s pomočjo vektorja ocenjenih modelov $\hat{\boldsymbol{\lambda}}$ za dan vektor manjših besednih enot, ki bodo sintetizirane, ω, na način, da bodo izhodne verjetnosti največje

$$\hat{\boldsymbol{o}} = arg\,max_{o}\,\{p(\boldsymbol{o}|\omega,\hat{\boldsymbol{\lambda}})\}. \quad (2)$$

Modeliranje trajanja govornega segmenta je enostavnejše, saj v splošnem lahko na podlagi informacije koliko časa ostaja PMM v nekem stanju ocenimo trajanje. Trajanje lahko enostavno ocenimo iz matrike porazdelitve verjetnosti prehodov med stanji. Proces učenja smo v tem prispevku izvedli s pomočjo orodja HTS (Yoshimura et al., 1999).

4.3. Izvedba PMM sintetizatorja s tehnikami prilagajanja

Z razvojem tehnik prilagajanja (ang. adaptation) govorca na različna akustična okolja pri razpoznavanju govora, so se te tehnike začele uporabljati tudi pri sistemih za umetno tvorjenje govora. Tako moderni sistemi za umetno tvorjenje govora s pomočjo PMM omogočajo najprej gradnjo splošnega modela posameznega jezika, ki zajema posplošen akustični model, posplošen model trajanja ter posplošen model vzbujanja. S tehnikami kot sta postopek maksimizacije posteriorne porazdelitve (ang. maximum posteriori, MAP) ter postopek linearne regresije z maksimalnim verjetjem (ang. constrained, maksimum likelihood regression, CMLLR) je mogoče vplivati na parametre PMM splošnega modela govora z namenom prilagajanja parametrov splošnih modelov na posameznega govorca. S takim načinom pridobimo umetno tvorjen govor, ki je podoben lastnemu govoru govorca, oziroma posnetkov, ki jih imamo na razpolago za prilagajanje, pri čemer izrabimo tudi vse akustične značilnosti splošnega modela govora. Že z relativno majhno količino posnetkov namenjenih za prilagajanje je mogoče prilagoditi splošni modela jezika. Na tem mestu moramo opozoriti, da v kolikor fonemi v govoru niso zastopani, se tudi slabše prilagodijo na splošni model, kar se odraža tudi pri sintezi. S pomočjo demonstracijskih programov na spletni strani skupine HTS, smo tako izvedli tudi sintetizator na način, da smo adaptirali posneto govorno zbirko na splošen model, ki smo ga izvedli s pomočjo petih govorcev (1 ženska govorka, 4 moški govorci) zbirke VNTV (Žibert and Mihelič, 2000).

4.4. Sinteza govora

Proces sinteze je obrnjen proces razpoznave govora. Vhod v postopek predstavlja niz osnovnih govornih enot. Na podlagi tega niza se ustrezni modeli povežejo v verigo. Nato se izbere najbolj verjetna pot (niz stanj) skozi to verigo glede na porazdelitve verjetnosti trajanj posameznih stanj. Ta najbolj verjetna veriga odda niz vektorjev v katerih so združene značilke vzbujanja ter govornega trakta. Iz poteka značilk vzbujanja se nazadnje tvori še vzbujanje, ki ga vodimo na vhod filtra, ki smo ga določili s pomočjo spektralnih značilk. Na izhodu filtra dobimo sintetiziran govor (Yoshimura et al., 1999).

5. Zaključek

Čeprav v prispevku nismo ovrednotili udejanjenih sistemov za umetno tvorjenje govora, smo pokazali, da je mogoče s pomočjo PMM sinteze z relativno majhno količino podatkov in pravilno izbiro slovenske pisane besede za gradnjo govorne zbirke izdelati sintetizator slovenskega govora. V nadaljnjem delu bomo nedvomno ovrednotiti udejanjene sisteme ter preizkusili tudi naprednejše algoritme za pridobivanje fonemsko bogatih sklopov povedi v namen snemanja ustreznih govornih zbirk, primernih za sintezo govora. Obenem pa bomo poizkusili določiti še priporočljive najmanjše količine govornega materiala, s katerim lahko še udejanjimo slovenski sintetizator govora s pomočjo PMM.

6. Literatura

A. W. Black, Taylor P., Caley R., Clark R., in S. King. 2003. The edingburgh speech tools library. Tehnično poročilo.

Paul Boersma in David Weenink. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345.

Jerneja Gros. 1997. *Samodejno tvorjenje govora iz besedil*. Doktorsko delo, Fakulteta za elektrotehniko, Univerza v Ljubljani.

Kishore Prahallad, Alan W Black, in Ravishankhar Mosur. 2006. Sub-phonetic modeling for capturing pronunciation variation in conversational speech synthesis. V: *in Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*.

Marc Schröder, Marcela Charfuelan, Sathish Pammi, in Ingmar Steiner. 2011. Open source voice creation toolkit for the mary tts platform. V: *Proceedings of Interspeech 2011*. ISCA.

K. Takuda, Masuko T., Koishida K., Sako S., Zen H. Imai S., in Kobayashi T. 2009. Speech signal processing toolkit (sptk). Tehnično poročilo.

Junichi Yamagishi, Bela Usabaev, Simon King, Oliver Watts, John Dines, Jilei Tian, Yong Guan, Rile Hu, Keiichiro Oura, Yi-Jian Wu, Keiichi Tokuda, Reima Karhila, in Mikko Kurimo. 2010. Thousands of voices for hmm-based speech synthesis: analysis and application of tts systems built on various asr corpora. *Trans. Audio, Speech and Lang. Proc.*, 18(5):984–1004, July.

Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, in Tadashi Kitamura. 1999. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. V: *EUROSPEECH'99*, str. –1–1.

S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, in P. C. Woodland. 2006. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK.

Janez Žibert in France Mihelič. 2000. Slovenian weather forecast speech database. V: *Proc, SoftCOM*, str. 199–206.

Distributional Semantics Approach to Detecting Synonyms in Croatian Language

Mladen Karan, Jan Šnajder, Bojana Dalbelo Bašić

University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
{mladen.karan, jan.snajder, bojana.dalbelo}@fer.hr

Abstract

Identifying synonyms is important for many natural language processing and information retrieval applications. In this paper we address the task of automatically identifying synonyms in Croatian language using distributional semantic models (DSM). We build several DSMs using latent semantic analysis (LSA) and random indexing (RI) on the large hrWaC corpus. We evaluate the models on a dictionary-based similarity test – a set of synonymy questions generated automatically from a machine readable dictionary. Results indicate that LSA models outperform RI models on this task, with accuracy of 68.7%, 68.2%, and 61.6% on nouns, adjectives, and verbs, respectively. We analyze how word frequency and polysemy level affect the performance and discuss common causes of synonym misidentification.

Prepoznavanje hrvaških sopomenk s pomočjo distribucijske semantike

Prepoznavanje sopomenk je pomembno za številne aplikacije na področju jezikovnih tehnologij in poizvedovanja po informacijah. V pričujočem prispevku se ukvarjamo z avtomatskim prepoznavanjem sopomenk v hrvaščini, pri čemer uporabljamo modele distribucijske semantike (DSM). S pomočjo latentne semantične analize (LSA) in naključnega indeksiranja (RI) iz korpusa hrWaC zgradimo več različnih modelov. Modele nato ovrednotimo s pomočjo testov sinonimije, ki so avtomatsko izluščeni iz strojno berljivega slovarja. Rezultati kažejo, da so modeli, zgrajeni s pomočjo LSA, za to nalogo uspešnejši, njihova natančnost pa je 68,7% za samostalnike, 68,2% za pridevnike in 61,6% za glagole. V prispevku analiziramo tudi, kako pogostost pojavljanja besed v korpusu in stopnja njihove večpomenskosti vplivajo na rezultate in razpravljamo o najpogostejših razlogih za napake, do katerih pri prepoznavanju prihaja.

1. Introduction

The task of automatically determining the semantic similarity between words (e.g., *car* and *truck*) and semantic relatedness between words (e.g., *car* and *driver*) has received a lot of attention, and many semantic similarity measures (SSMs) have been proposed in the literature. Synonymy is a paradigmatic type of semantic relation between words that are substitutable in any possible context without change in meaning (*absolute synonyms*), substitutable in different contexts depending on their sense (*sense synonyms*), or substitutable in some contexts (*near-synonyms*). Numerous natural language processing and information retrieval applications can benefit from knowledge about synonyms, including word sense disambiguation (Agirre and Soroa, 2009), query expansion (Pantel et al., 2009), text similarity between short texts (Šarić et al., 2012), paraphrasing (Zhao et al., 2009), natural language generation (Inkpen and Hirst, 2004), WordNet acquisition (Broda et al., 2008), and text simplification (Inui et al., 2003).

Methods of semantic modelling can be divided into two broad categories: knowledge-based and distributional methods. The former rely on manually constructed resources, such as WordNet (Miller, 1995) or Wikipedia, to provide information required to measure relatedness. While such approaches work quite well, a resource like WordNet is often of limited coverage and, more importantly, it is not available for every language. The alternative to knowledge-based methods are distributional semantic models (DSMs). DSMs model the meaning of a word by the distribution of its contexts; two words are considered similar if they have similar context distributions. Often used DSMs are latent semantic analysis (LSA) (Deerwester et al., 1990) and random indexing (RI) (Sahlgren, 2005). More recently, approaches have been proposed that combine information from Wikipedia with distributional analysis (Gabrilovich and Markovitch, 2007; Hassan and Mihalcea, 2011). Unlike the knowledge-based approaches, DSMs require no linguistic resources other than a corpus (more sophisticated models, e.g., (Padó and Lapata, 2007), may also require tagging or parsing). This makes DSMs ideal for languages lacking basic lexical resources such as WordNet. On the other hand, the disadvantage of DSMs over their knowledge-based counterparts is that DSMs cannot easily distinguish between the various types of semantic relations (similarity vs. relatedness, paradigmatic vs. syntagmatic relations). Moreover, DSMs are typically token-based, and therefore unable to distinguish between the different senses of polysemous words. These two issues make synonymy detection using DSMs a very challenging task.

In this paper we address the task of identifying synonyms in Croatian language using DSMs. Our primary motivation is the automatic acquisition of WordNet synsets, as proposed by Broda et al. (2008). We perform large-scale experiments with two basic models (LSA and RI) built using the large Croatian Web as Corpus – hrWaC (Ljubešić and Erjavec, 2011). Inspired by the approach proposed by Landauer and Dumais (1997) and refined by Freitag et al. (2005), we evaluate our models on a dictionary-based sim-

ilarity test (DBST) – a set of synonym questions generated automatically from a machine readable dictionary. To the best of our knowledge, this is the first work that addresses the task of synonymy detection for Croatian language.

The rest of the paper is structured as follows. Section 2 gives a summary of the related work. In Section 3 we describe the construction of DSMs. Evaluation results are presented in Section 4. We conclude in Section 5.

2. Related Work

The first to tackle the synonymy detection task using LSA were Landauer and Dumais (1997). To evaluate their approach, they used a set of synonymy matching questions, as they appear in the *Test of English as a Foreign Language* (TOEFL), a standardized test administered by the Educational Testing Service (ETS). Each synonym question consists of a target word and four answer words, of which one is a synonym of the question word, while the other three are distractors (incorrect answers). The task consists of identifying which of the four words is a synonym of the target word. Landauer and Dumais (1997) reported an accuracy of 64.4% on a set of 80 TOEFL questions. The best reported result on the set of TOEFL questions is 97.5% (Turney et al., 2003), achieved by a combination of methods.

Because work of Turney et al. (2003) essentially solved the synonymy detection task on TOEFL questions, Freitag et al. (2005) proposed a somewhat harder test – the *WordNet-based similarity test* (WBST). The test uses WordNet synsets to generate a large number of questions resembling TOEFL questions. Additional measures are taken to ensure that the distractors are not in a synonymy relation with the answer word. Because WBST has a lot more questions (23,570) than TOEFL has, WBST gives much more reliable performance estimates than TOEFL. The best reported result on WBST is 72.2% (Freitag et al., 2005), achieved using feature-based DSM. Features are unnormalized tokens, sometimes augmented with direction (left vs. right occurrence) and distance information, appearing within a context window of a target term. Freitag et al. (2005) also introduced a vector similarity measure specially tailored to the task of synonymy detection. A similar approach was used for Polish language (Broda et al., 2008; Piasecki et al., 2007). In contrast to Freitag et al. (2005), Broda et al. (2008) use a richer feature set incorporating lexical, morphological, and syntactic information. They also use feature selection methods and an additional feature weighting scheme to accentuate the most informative features of a particular target term. Since WBST may not be stringent enough to demonstrate the advantages of more sophisticated SSMs, Piasecki et al. (2007) proposed the *Extended WordNet based similarity test* (EWBST). This test extends the regular WBST by deliberately using similar and related (but not synonymous) words as distractors. Consequently, EWBST is much harder than WBST, but gives a better estimate of how well a SSM identifies synonyms.

3. Model Construction

3.1. Corpus and preprocessing

To build the DSMs, we use the large Croatian Web as a Corpus (hrWaC) (Ljubešić and Erjavec, 2011). To our knowledge, this is the largest available corpus of Croatian texts. In order to reduce the noise in the corpus, introduced by the use of informal language, we removed from the corpus all documents acquired from discussion forums and blogs. For reasons of computational efficiency, we also filtered out all words with a frequency below 50. This left us with a corpus containing 5,647,652 documents, 1.37 G tokens, 3.89 M word-form types, and 215,499 lemmas. Each document is further split into paragraphs (because the corpus was acquired from the web, the division into paragraphs is not consistent across all documents). To account for the morphological variation, which would disperse distribution vectors over inflectional forms and result in less reliable probability estimates, we employed lemmatization. To this end, we use the semi-automatically acquired morphological lexicon for Croatian language (Šnajder et al., 2008). We did not POS-tag the corpus; in cases of lemma ambiguity, we consider all possible lemmas when building DSMs. Moreover, we did not remove stop-words because all models have weighting schemes that give less emphasis to less discriminant words. Notice that we could have applied more sophisticated preprocessing techniqes, including POS tagging and parsing, but we leave this for future research.

3.2. Latent semantic analysis

Latent semantic analysis (LSA) (Deerwester et al., 1990) is a DSM based on the singular value decomposition (SVD) of a term-context co-occurrence matrix. A context vector for each of n contexts (documents or paragraphs) is extracted from the corpus. Elements of the context vector, corresponding to context c, are occurrence counts in c for each of the m target terms. The context vectors constitute the columns of a term-context co-occurrence matrix A. The row i of matrix A can be interpreted as a distribution of contexts conditioned on word i. Once the matrix A is constructed, it is decomposed by SVD, resulting in three matrices U, D, and V, such that $A = UDV^T$. The rows of U model for each term the distributions of a new set of contexts defined by DV^T. The final step is performing a dimensionality reduction by discarding all but the k largest singular values and the corresponding singular vectors. The semantic relatedness of words can be measured by comparing the corresponding rows of the reduced matrix U.

In our experiments, after constructing the term-context matrix, we apply the classical *tf-idf* weighting scheme. The inverse document frequency is defined as $idf(w) = \log \frac{D}{Q}$, where D is the total of number of documents in our corpus and Q is the number of documents containing word w.

The large size of the hrWaC corpus is reflected in the dimensions of our term-context co-occurrence matrices. The matrix has 215,499 rows (target terms); for document contexts the matrix has 5,647,652 columns and 827.7 M non-zero elements, while for paragraph contexts it had 29,763,686 columns and 1.16 G non-zero elements. For SVD computation we use the freely available ARPACK library.[1] For comparing the vectors, we use the cosine similarity measure.

[1] http://www.caam.rice.edu/software/ARPACK/

3.2.1. Random indexing

Random indexing (RI) (Sahlgren, 2005) is another kind of DSM that is, much like LSA, based on dimensionality reduction. For each context a random *index vector* is generated: a sparse d-dimensional vector containing a small number of randomly generated non-zero values. The so-obtained index vectors are shown to be nearly orthogonal (Sahlgren, 2005). Next, the distributional vectors for each target term are generated. Initially, all distributional vectors are d-dimensional null-vectors. The corpus text is scanned, and each time a term t is associated with context c, the index vector of c is added to the distributional vector of t. Eventually, target terms associated with similar contexts will tend to have similar distributional vectors. This is equivalent to constructing the entire term-context co-occurrence matrix and performing dimensionality reduction using random projection; the rows of the projection matrix are in fact the index vectors. The semantic relatedness of target terms can now be measured by comparing their distributional vectors.

In our models the index vectors are generated with dimension 100 (2 random non-zero elements) and 500 (4 random non-zero elements). The non-zero elements are chosen so that they contain an equal number of $+1$ and -1 values. We apply RI using documents, paragraphs, and neighboring words as contexts. In case of the first two, identically as for LSA, a term is associated with a context (a paragraph or a document) if it appears in the context. In the latter case, a term is associated with the context (the neighboring words) appearing within a ± 5 word window around the term. To take into account that some words are more informative than others, before adding the index vector of context word w, we weigh the whole index vector by the inverse document frequency score of w. Similarly as with LSA, we use the cosine similarity for vector comparison.

4. Evaluation

4.1. Dictionary-based similarity test

Because the Croatian WordNet (Raffaelli et al., 2008) is not yet available, we could not directly follow the approach by Freitag et al. (2005) to generate the similarity test. Instead, we relied on a machine readable dictionary derived from the monolingual Croatian dictionary (Anić, 2003). The dictionary lists over 68,500 lexemes divided into almost 100,000 sense entries. For each lexeme, the dictionary provides, inter alia, a basic morphological description and a gloss containing a short description of the word (or a description of every sense of a polysemous word). In many cases the gloss also contains implicit references to synonyms. In most cases these references follow a regular pattern and the referent can be extracted automatically. Using a few heuristic rules, we extracted automatically the synonym references from glosses and established synonymy links between entries. We extracted 43,311 synonym references (an average of 0.44 links per sense). Notice that the synonymy references are often ambiguous as they may refer to a polysemous word. Many ambiguous references could be resolved automatically, but this was not required in our case because we need not distinguish between senses of polysemious words when generating the questions. Interestingly, about 5000 words to which the synonymy references referred to were missing in the dictionary, thus we automatically added these entries to the dictionary.

Table 1: Example questions from the nouns part of DBST (the correct answers are A, D, C, and A)

```
težak (farmer):
  A.  poljoprivrednik (farmer)
  B.  umjetnost (art)
  C.  radijacija (radiation)
  D.  bod (point)
krov (roof, home):
  A.  zgrada (building)
  B.  izvršilac (executant)
  C.  sanjkalište (sled run)
  D.  dom (home)
karakter (character):
  A.  detalj (detail)
  B.  kruška (pear)
  C.  lice (face, character)
  D.  maharadža (maharadja)
jaran (friend):
  A.  drug (friend)
  B.  krivovjerje (heresy)
  C.  sulfit (sulfite)
  D.  ekscentričnost (eccentricity)
```

To generate the questions of our dictionary-based similarity test (DBST), we proceeded as follows. Using the synonymy links, we generate from the sense dictionary all pairs of synonymous words (the target word and the correct answer word), such that both words appear in our corpus. To make the test more realistic and more difficult, we use string-distance measures to filter out from this list pairs of words that seem to be morphologically or orthographically related. To generate the distractors for a question, we choose at random three words from the same part-of-speech, subject to the following constraints: (1) a chosen word appears in the corpus, (2) it is not in a (transitive) synonymy relation with any of the other four words used for that question, and (3) it is morphologically and orthographically unrelated to other four words. To check whether two words are in a synonymy relation, for each word we first collect 100 words to which it has transitive synonymy links, by performing a breadth-first search on the symmetric closure of the synonymy graph originating from the corresponding word (thereby disregarding the differences between senses). We then consider two words to be in a (transitive) synonymy relation if one word is contained in the set of 100 synonymy-linked words of the other word. Collecting 100 synonymy-linked words ensures that we have collected all potential synonyms of a given word and that therefore no distractor will be a synonym of any other distractor nor the target word. The requirement that the distractors are not in a synonymy relation makes the test more realistic, as synonymous distractors might be discarded from being the correct answers without consideration. The described procedure yields a set of 11,276 questions, of which 6446 for nouns (N), 2704 for adjectives (A),

and 2126 for verbs (V). Example questions are given in Table 1 (for the sake of brevity some senses are omitted). Notice that, by using a dictionary-based evaluation, we subscribe to the definition of synonymy used in compiling the dictionary. The choice of a dictionary (characterized also by its coverage, sense granularity, etc.), together with the strategy used for generating the distractors, determines the appropriateness of a DBST as a means to evaluate DSMs for synonym identification.

DSMs in general tend to perform better for high-frequency words than for low-frequency words, as demonstrated by Piasecki et al. (2007). To test how word frequency affects the model performance, we generated two additional questions sets: one for low/medium-frequency band ($100 \leq f < 1000$) and one for high-frequency band ($f \geq 1000$). Both the target word and the answer words come from the corresponding frequency band, thus the questions contain more or less frequency-balanced words. We did not generate a separate low-frequency set because it would not have a sufficient number of questions.

Apart by frequency, we expect our models to be influenced by the level of polysemy. To test this assumption, we divide the questions based on their *polysemy levels*. Following Freitag et al. (2005), we define the polysemy level of a question to be the sum of the number of senses in the dictionary of its target and answer words.

4.2. Result analysis

Table 2 shows the accuracy of the models on the generated test sets. We evaluated 10 models: six RI models and four LSA models. The models were built with either 100 or 500 dimensions on contexts consisting of documents (D), paragraphs (P), or words within a window (W), as described in Section 3. Best results are given in bold. Notice that all models outperform the accuracy baseline of 25%.

A general observation is that the LSA models consistently outperform the RI models. The LSA500P model performed best in almost all experiments and outperformed the second-best model (LSA100P) by a significant margin. These results suggest that LSA may be better suited for the task of synonym detection in Croatian language. Results also reveal that a higher-dimensional model almost always significantly outperforms the corresponding lower-dimensional model. This indicates that the number of dimensions plays an important role in our task. The optimal number of dimensions for the task of identifying synonyms may differ when compared to other semantic similarity tasks (e.g., relatedness); to confirm this, additional experiments are required. The results seem to suggest that an additional increase in dimensionality may further improve the performance. We carried out additional experiments and concluded that this is not the case: improvement can only be observed until a plateau is reached at around 200 dimensions.

With respect to the context definition, results suggest that a smaller context – a window and especially a paragraph – gives better performance for LSA, while RI benefits more from a larger context – the entire document. While using a larger context is better for modelling long distance co-occurrences, using a smaller context prevents less relevant words occurring far from the target term from introducing noise into the distributions.

With respect to the word's part-of-speech, we can make two general observations: the performance on nouns and adjectives is comparable (slightly better on adjectives in most cases), while on verbs it is consistently lower. Identical behaviour can be observed for English in the results reported by Freitag et al. (2005).

As regards the frequency bands, results suggest that lower frequency has a detrimental effect on the performance of most models. This is expected, because distributional vectors of high-frequency words are built using more data, allowing for better modelling of word meaning. Notice, however, that high frequency words also tend to be more polysemous, which may again decrease the performance. In our case, however, it seems that higher word frequency still results in better performance. The same was confirmed in (Piasecki et al., 2007; Broda et al., 2008). Notice that in a realistic scenario the target and the answer words will not come from the same frequency band. In this respect, mixed frequencies results give a more realistic performance estimate.

In Table 3 we give the results for the best-performing LSA500P model with respect to polysemy levels of questions. As expected, models perform worse on questions with higher polysemy levels. Distributional representations of each sense of a polysemous word get merged into a single distributional representation – a mixture of distributions. For questions with high polysemy level, the corresponding distributional vectors are blurred and the similarity comparisons between such vectors are less meaningful.

Because the synonym questions contain randomly chosen distractors, the accuracy can vary on different test instances. To measure the variance in accuracy, we generated 30 test instances and calculated the performance of the best-performing model (LSA500P) across all test instances. Results proved to be quite stable: the maximum standard deviation of accuracy was 0.8% (obtained on verbs). This suggests that, owing to the relatively large number of questions, our DBST provides reliable accuracy estimates.

4.3. Error analysis

Most cases of synonym misidentification can be attributed to polysemy or low frequency of the target term in the corpus. We identified four typical causes of errors.

1. Homonyms and homographs – As Croatian is a highly inflected language, it is often the case that two different (often completely unrelated) words share some word-forms. For each ambiguous word-form we considered all its possible lemmas, thereby introducing an interference between the corresponding distributional vectors. The distributional vector of term t will model not only contexts of t, but also a number of additional contexts introduced by words that share word-forms with t. Even if the number of shared word-forms is small, the interference can still be very detrimental if the frequency difference between the words is large (i.e., one very frequent word-form may distort the distributional vector of a less frequent target term). A

Table 2: Accuracy for all considered models

	Mixed freq.			Low/medium freq.			High freq.		
Model	A	N	V	A	N	V	A	N	V
RI100D	42.0	41.8	36.2	39.7	29.5	29.3	43.9	46.5	37.7
RI500D	58.1	53.2	47.2	22.4	41.8	25.2	57.4	57.0	48.7
RI100P	40.7	39.5	36.3	24.1	31.9	29.3	43.6	42.5	37.8
RI500P	54.9	51.7	43.0	44.8	40.9	29.3	56.4	56.0	45.8
RI100W	56.8	49.8	43.5	44.8	42.4	35.0	52.5	48.8	40.3
RI500W	54.4	48.9	43.2	36.2	43.0	43.1	51.5	48.1	39.2
LSA100D	54.1	55.2	43.6	50.0	51.3	43.1	57.7	58.9	44.7
LSA500D	61.2	59.1	50.4	51.7	53.0	45.5	63.5	64.2	51.3
LSA100P	63.2	66.0	55.6	60.3	**67.7**	60.2	64.8	67.0	57.7
LSA500P	**68.2**	**68.7**	**61.6**	**67.2**	67.4	**65.0**	**69.3**	**70.1**	**60.3**

Table 3: Accuracy for model LSA500P with respect to different polysemy levels

	Mixed freq.			Low/medium freq.			High freq.		
Polysemy level	A	N	V	A	N	V	A	N	V
5–7	72.6	74.3	67.7	66.7	69.7	76.5	75.9	81.2	66.0
8–10	66.8	70.9	62.9	62.5	66.9	62.1	69.2	74.3	68.8
11–13	60.2	64.2	64.6	100.0	54.7	55.2	60.9	67.8	57.3
14–16	63.6	61.1	56.4	–	33.3	100.0	57.8	63.9	61.8
17–	47.6	60.0	58.4	–	–	–	46.5	64.0	52.9

case in point is the first question from Table 1: the word *težak* is a homonym, which in fact is more often used as an adjective (*hard, heavy*) than a noun. Obviously, this problem could for the most part be solved by a morphological disambiguation of the corpus.

2. Semantically related distractors – In some questions one of the distractors is a word that, albeit not synonymous, is semantically related to the target word. A case in point is the second question from Table 1: while the correct answer *dom (home)* does receive the second-best similarity score, the highest score goes to *zgrada (building)*. This is because the word *krov* in its dominant sense of *roof* happens to be more related to *zgrada* than to *dom*. While the polysemy of *krov* certainly contributes to misidentification, the dominant cause of misidentification is the fact that the semantically related word *zgrada* was among the distractors. Notice that, depending on the application, false synonyms that are paradigmatically related to the target word may still be usable in practice. However, manual inspection revealed that most false synonyms are syntagmatically related to the target word. To avoid this kind of error, we would need a method to distinguish between synonymy and general semantic relatedness.

3. Rare senses – In some cases the target word and the correct answer are sense synonyms via a very specific and seldom used sense. Contexts of such senses make a very small fraction of the total contexts on which the distributional vectors are built. Consequently, such senses are poorly modelled and the performance for them is worse. A case in point is the third question from Table 1: the correct answer *lice (face, character)* is almost never used in the sense of *character*, except in a few phrases.

4. Rare variants – In some cases the target word and the correct answer are not only sense synonyms, but also *variants*, i.e., they differ in register, dialect, or affect. This is often accompanied by a big difference in frequency; e.g., a dialectal form occurs very rarely in the corpus. Because the meanings of rare words are poorely modelled, this may lead to misidentification. An example is the fourth question from Table 1: the word *jaran (friend)* is an informal dialectal word used much less frequently than its sense synonym *drug (friend)*.

5. Conclusion and Future Work

In this paper we have addressed the task of automatically identifying synonyms in Croatian language using distributional semantic models (DSMs). We build several DSMs using latent semantic analysis (LSA) and random indexing (RI) on the large hrWaC corpus and evaluated the models on a dictionary-based similarity test. Results indicate that LSA models outperform RI models on this task. The best accuracy was obtained using LSA (500 dimensions, paragraph context): 68.7%, 68.2%, and 61.6% on

nouns, adjectives, and verbs, respectively. Our results are along the lines of those obtained for English by Freitag et al. (2005). Compared to the result for Polish (Piasecki et al., 2007; Broda et al., 2008), our result is slightly worse, however, we make at present no use of rich morphological and syntactic features.

For future work we intend to address the most common causes of synonym misidentification discussed above. Following the work of Piasecki et al. (2007), we plan to develop a more stringent version of the similarity test. Another possibility for future research is to experiment with ways to mitigate the negative effect of polysemy by employing WSD techniques prior to building the distributional vectors, as done by Fišer et al. (2012). Finally, it would be interesting to experiment with many other types of distributional semantic models, such as the grammatical feature models (Freitag et al., 2005; Piasecki et al., 2007), the syntax-based model (Padó and Lapata, 2007), or Wikipedia-based models (Hassan and Mihalcea, 2011; Gabrilovich and Markovitch, 2007).

6. Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions. This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under Grant 036-1300646-1986.

7. References

E. Agirre and A. Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41.

V. Anić. 2003. *Veliki rječnik hrvatskoga jezika*. Novi Liber.

B. Broda, M. Derwojedowa, M. Piasecki, and S. Szpakowicz. 2008. Corpus-based semantic relatedness for the construction of polish WordNet. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC'08)*.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6).

D. Fišer, N. Ljubešić, and O. Kubelka. 2012. Addressing polysemy in bilingual lexicon extraction from comparable corpora.

D. Freitag, M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, and Z. Wang. 2005. New experiments in distributional representations of synonymy. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 25–32.

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on artificial intelligence*, volume 6, page 12.

S. Hassan and R. Mihalcea. 2011. Semantic relatedness using salient semantic analysis. In *Proceedings of AAAI Conference on Artificial Intelligence*.

D.Z. Inkpen and G. Hirst. 2004. Near-synonym choice in natural language generation. In *Recent Advances in Natural Language Processing*, volume 3, pages 141–152.

K. Inui, A. Fujita, T. Takahashi, R. Iida, and T. Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 9–16.

T.K. Landauer and S.T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review; Psychological Review*, 104(2):211.

N. Ljubešić and T. Erjavec. 2011. hrWaC and slWaC: compiling web corpora for Croatian and Slovene. In *Text, Speech and Dialogue*, pages 395–402. Springer.

G.A. Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

J. Šnajder, B. Dalbelo Bašić, and M. Tadić. 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing & Management*, 44(5).

S. Padó and M. Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

P. Pantel, E. Crestan, A. Borkovsky, A.M. Popescu, and V. Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 938–947.

M. Piasecki, S. Szpakowicz, and B. Broda. 2007. Extended similarity test for the evaluation of semantic similarity functions. *Vetulani (Vetulani, 2007)*, pages 104–108.

I. Raffaelli, M. Tadić, B. Bekavac, and Ž. Agić. 2008. Building croatian wordnet. In *Proceedings of the 4th Global WordNet Conference, Szeged, Global WordNet Association*, pages 349–359.

M. Sahlgren. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5.

F. Šarić, G. Glavaš, M. Karan, J. Šnajder, and B. Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity.

P. Turney, M.L. Littman, J. Bigham, V. Shnayder, et al. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems.

S. Zhao, X. Lan, T. Liu, and S. Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 834–842.

Avtomatsko luščenje leksikalnih podatkov iz korpusa

Iztok Kosem*, Polona Gantar, Simon Krek*****

* Trojina, zavod za uporabno slovenistiko
Partizanska cesta 5, SI-4220 Škofja Loka
iztok.kosem@trojina.si
** Inštitut za slovenski jezik Frana Ramovša
Novi trg 4, SI-1000 Ljubljana
apolonija.gantar@guest.arnes.si
*** Amebis d.o.o.
Bakovnik 3, SI-1241 Kamnik
simon.krek@guest.arnes.si

Povzetek

Pri oblikovanju leksikalne baze za slovenščino v okviru projekta *Sporazumevanje v slovenskem jeziku* smo del gesel izdelali s postopkom avtomatskega luščenja leksikalnih podatkov iz korpusa Gigafida prek orodja Sketch Engine neposredno v program za izdelavo leksikalne baze iLex. V prispevku opisujemo posamezne korake pri pripravi avtomatizacijskega postopka, zlasti prilagoditev slovnice besednih skic, izdelavo konfiguracije za aplikacijo GDEX za izbor dobrih korpusnih zgledov in pripravo API skripte. Na kratko predstavimo prve rezultate izvedenega postopka in predlagamo izboljšave tako na ravni metodologije kot tudi vključitve dodatnih jezikovnotehnoloških funkcionalnosti pri avtomatičnem luščenju jezikovnih podatkov.

Automatic extraction of lexical information from a corpus

A selection of entries in the lexical database for Slovene, an activity within the Communication in Slovene project, was compilled using the automatic extraction of lexical information from the Gigafida corpus (via the Sketch Engine corpus tool) and importing the obtained information directly into the dictionary writing system iLex. The paper describes individual steps in the preparation of the automatic extraction procedure, especially the adjustment of sketch grammar, development of the GDEX configuration for the selection of good corpus examples, and the programming of the API script. We briefly present the initial results and suggest improvements in methodology of automatic extraction of lexical data, and inclusion of additional language technologies.

1. Uvod

Sodobne tehnologije so prinesle velike spremembe na področju leksikografije, tako za slovarske uporabnike kot za leksikografe. Nova generacija slovarskih uporabnikov, ki je vešča uporabe elektronskih medijev, zlasti spleta, zahteva zelo hiter in uporabniku prijazen dostop do podatkov o sodobnem jeziku, s tem pa posredno odloča tudi o vrsti, obliki in načinu njihove predstavitve. Če uporabniki česa ne najdejo v (spletnem) slovarju – do katerih največkrat dostopajo prav tako prek spletnih iskalnikov (Lorentzen in Theilgaard, 2012) –, obstaja velika verjetnost, da bodo podatek skušali poiskati na spletu.

Težava, s katero se soočajo leksikografi, je, da adekvaten opis jezika še vedno zahteva precej časa, tudi zaradi vse večjih besedilnih korpusov, ki jih je potrebno analizirati. Zato razvijalci leksikografskih orodij iščejo nove rešitve, ki bi leksikografom olajšale delo in jim pomagale leksikografske podatke čim hitreje spraviti do uporabnika. V zadnjih letih so bila tako razvita orodja, kot je Sketch Engine s funkcijami Besedna skica (ang. Word sketches; Kilgarriff in Rundell, 2002) in Kliksikografija (ang. Tickbox lexicography), od katerih prva nudi hiter pregled slovničnih in kolokacijskih vzorcev, v katerih se besede sopojavljajo, druga pa hiter in preprost prenos želenih podatkov v slovarsko orodje. Tovrstna orodja smo s pridom uporabili tudi pri oblikovanju Leksikalne baze za slovenščino (LBS), ki je potekalo v obdobju 2008–2012 pri projektu *Sporazumevanje v slovenskem jeziku* (http://www.slovenscina.eu/; SSJ[1]).

Vendar pa mora leksikograf kljub uporabi omenjenih polavtomatskih funkcionalnosti še vedno analizirati velike količine podatkov, na podlagi katerih se odloča o njegovi relevantnosti in primernosti za vključitev v podatkovno bazo ali slovar. Zamuden pa je tudi postopek vnašanja podatkov v slovarsko orodje, saj ne obstaja neposreden prehod med Besednimi skicami in različnimi strukturami slovarskih baz ali priročnikov.

Pri gradnji LBS smo predvideli, da bi se proces izdelave gesla precej skrajšal, če bi imel leksikograf možnost avtomatsko izluščiti relevantne podatke o besedi neposredno iz korpusa v program za izdelavo slovarja, jih pregledati, selekcionirati in po potrebi dopolniti. Dodatna analiza korpusa ne bi bila več potrebna oz. bi se zreducirala na preverjanje aktualne rabe.

Zamisel smo začeli uresničevati ob zaključku aktivnosti, saj je bilo pomembno, da sta bila vrsta leksikalno-gramatičnih podatkov in način njihove organizacije v LBS dokončno opredeljena z DTD strukturo in da je bila zadostna količina gesel ročno izdelana Na ta način je bilo mogoče predvideti optimalne rešitve, ki bi jih prinesel postopek avtomatizacije.

V tem prispevku opisujemo posamezne korake pri vzpostavljanju postopka avtomatskega luščenja podatkov za izdelavo slovarskih gesel ter prva opažanja.

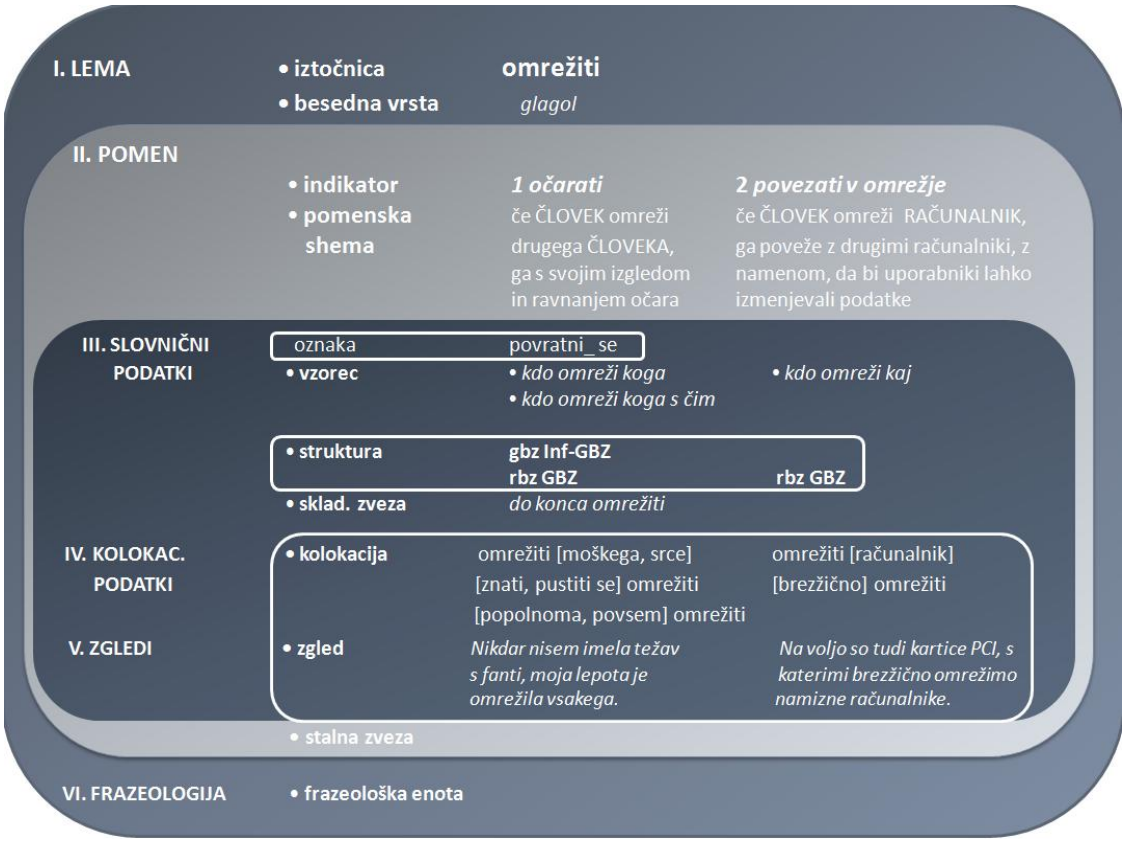
[1] Operacijo, v okviru katere je nastala raziskava, delno financira Evropska unija iz Evropskega socialnega sklada ter Ministrstvo za izobraževanje, znanost, kulturo in šport Republike Slovenije.

2. Leksikalna baza za slovenščino glede na avtomatsko luščenje leksikalnih podatkov iz korpusa

Leksikalna baza za slovenščino je zasnovana z dvema temeljnima ciljema: (a) zagotoviti relevantne leksikografske podatke, zlasti pomenske, skladenjsko-pomenske, kolokabilne, besednozvezne, frazeološke ipd. za izdelavo sodobnih jezikovnih priročnikov za slovenščino in (b) izdelati jezikovni vir, ki bo zagotavljal računalniško procesljive jezikovne podatke, ki so interpretirani in označeni v skladu z jezikoslovnimi kategorijami (Gantar in Krek, 2011). Celotna leksikalna baza je na voljo v formatu XML, posamezni njeni deli pa so določeni z DTD strukturo, tako da jih je mogoče avtomatsko identificirati. S tem naj bi LBS služila izdelavi in izboljšanju orodij za avtomatsko analizo jezika, kot je skladenjsko razčlenjevanje, večnivojsko označevanje, pomensko razdvoumljanje besed itd. Tako strukturirane podatke pa smo uspešno uporabili tudi pri postopku avtomatizacije.

Jezikovni podatki so v LBS strukturirani na šestih medsebojno povezanih nivojih: najvišji nivo vključuje lemo, podatke o besedni vrsti in korpusni frekvenci. Sledi semantični nivo, ki predvideva pomenske podatke v treh različnih oblikah: kot *indikatorje* za oblikovanje pomenskega menija, kot t. i. *pomenske sheme*, izdelane po vzoru baze FrameNet s formaliziranimi semantičnimi tipi na posameznih mestih prototipičnega vezljivostnega vzorca in kot *stavčne definicije*. Vsaka leksikalna enota, kamor štejemo (pod)pomene, stalne zveze in frazeološke enote, je na skladenjskem nivoju opredeljena glede na to, ali nastopa v določenih zanjo tipičnih *skladenjskih strukturah*, pri glagolih pa je ta podatek razviden iz *stavčnih vzorcev*. Na kolokacijskem nivoju sledijo skladenjskim strukturam *kolokacije*, ki so tipične leksikalne zapolnitve skladenjskih struktur, in t. i. *skladenjske zveze*, ki so pomensko in strukturno ustaljeni delčki jezika, kot na primer: *[deževati] od jutra do večera; [angažirati se] v smeri česa; [vožnja, voziti] pod vplivom alkohola*. *Stalne zveze* in *frazeološke enote* so od pomenov neodvisne in imajo lastno notranjo zgradbo, ki predvideva vse elemente na enak način kot posamezni pomeni besede. Vsi našteti podatki so potrjeni s *korpusnimi zgledi*.

Kot je razvidno iz spodnje slike, smo s postopkom avtomatizacije pridobili podatke na ravni skladenjskih struktur, pripadajočih kolokacij in relevantnih korpusnih zgledov (v okvirčkih). Poleg tega še podatke o tipičnem skladenjskem ali besedilnem obnašanju leme v korpusu, kot je denimo sopojavljanje z lastnimi imeni ali količinskimi izrazi, možnost tretjeosebne rabe glagola ali nastopanje v *se*-glagolskih ali citatnih konstrukcijah.



Slika 1: Podatki, pridobljeni s pomočjo avtomatskih postopkov, v strukturi LBS

3. Avtomatsko luščenje leksikalnih podatkov

Odločitev za izvedbo postopka avtomatičnega luščenja leksikalnih podatkov iz korpusa (ALLP) izhaja iz potrebe po skrajšanju časa in zmanjšanju stroškov pri izdelavi slovarskih priročnikov ter po drugi strani iz novih možnosti, ki jih ponujajo sodobna orodja za analizo velikih besedilnih korpusov.

3.1. Metodologija in zaporedje postopkov

Ideja ALLP predvideva prenos relevantnih leksikalnih podatkov iz korpusa Gigafida prek orodja Sketch Engine (Kilgarriff et al., 2004; SkE) oz. aplikacije Besedne skice in v leksikalni bazi registriranih skladenjskih struktur z uporabo API skripte (ang. Application Programming Interface) v program iLex (Erlandsen, 2004), v katerem se izdeluje LBS. Kot relevantne leksikalne podatke upoštevamo kolokabilno okolje besede ter dobre, tj. berljive ter pomensko in skladenjsko relevantne korpusne zglede, ki kolokacije potrjujejo v realnem besedilnem okolju. Te podatke zagotavlja aplikacija Besedne skice v orodju SkE, vendar pa je bilo za njihovo avtomatsko luščenje potrebno izdelati oz. slovenščini prilagoditi slovnico besednih skic (Krek, 2012) in pripraviti konfiguracijo orodja GDEX za pridobivanje dobrih korpusnih zgledov. Pred začetkom ALLP smo s sodelavci ekipe SkE v Brnu pripravili še API skripto, prilagodili DTD leksikalne baze novim elementom in atributom, specifičnim za ALLP, namestili korpus Gigafida v orodje SkE ter pripravili izbor relativno frekventnih in po možnosti enopomenskih lem za prvo fazo avtomatskega luščenja.

3.2. Izbor lem

Zaradi obvladljivosti količine podatkov pri evalvaciji izluščenih gesel in posledično zaradi možnosti postopnega izboljševanja nastavitev orodja GDEX in API skripte smo pri naboru lem upoštevali tri parametre: (a) relevantno frekvenco za posamezno besedno vrsto, kar pomeni predvsem dovolj obsežno besedno skico; (b) potencialno enopomenskost glede na slovenski Wordnet (Fišer, 2009; sloWNet) in SSKJ; (c) vključenost v sloWNet zaradi možnosti nadaljnjih povezav in nevključenost v SSKJ zaradi potencialnih novih izrazov in pomenov. Pri prvih poskusih ALLP smo se omejili na manj pogoste leme (pribl. 600 pojavitev v korpusu Gigafida), vendar pa so izdelane besedne skice pokazale premalo relevantnih podatkov, zato smo v nadaljevanju pri posamezni besedni vrsti določili do pet frekvenčnih skupin, znotraj njih pa smo se osredotočili na tiste frekvenčne razpone, ki so zagotavljali obvladljivo število lem ter hkrati ponudili optimalno besedno skico za pomensko relativno nerazvejano lemo.

Pri določanju enopomenskih ali pomensko manj zahtevnih lem smo se oprli na stanje, kot je izpričano v sloWNetu in SSKJ: izbrali smo leme z enim ali dvema sopomenskima nizoma (sinsetoma) v sloWNetu in/ali iztočnice z enim ali dvema pomenoma v SSKJ. Znotraj posamezne frekvenčne skupine – na končnem seznamu prevladujejo leme s pogostostjo v razponu med 1000 in 10.000 pojavitev, nekaj pa je tudi redkejših oziroma pogostejših lem, ki smo jih vključili za namene dodatnega testiranja ALLP, pogostejša gesla pa tudi zato, da bi lahko preverili delovanje API skripte za vse relacije v slovnici besednih skic – smo s prekrižanjem preostalih parametrov izdelali seznam s 515 samostalnškimi, 260 glagolskimi, 275 pridevniškimi in 117 prislovnimi lemami.

3.3. Slovnica besednih skic

Za potrebe ALLP iz korpusa Gigafida je bila izdelana nova slovnica besednih skic (ang. sketch grammar), ki izkorišča tudi nekatere elemente, ki so bili v orodje Sketch Engine dodani v novejših različicah. Med njimi so predvsem t. i. direktive[2] (directives) *CONSTRUCTION, *COLLOC in *SEPARATEPAGE. Prva omogoča prepoznavanje skladenjskih struktur brez kolokacij, kar je primerno predvsem za luščenje glagolskih vezljivostnih vzorcev. Druga je namenjena izločanju elementov, ki v LBS spadajo v kategorijo skladenjskih zvez, denimo zveza predlog-samostalnik-predlog (primer: *v primerjavi z, v odnosu do*), tretja pa je namenjena odpiranju relacij s tremi elementi (direktiva *TRINARY) na novi spletni strani, kar omogoča uvedbo natančnejših relacij s predlogi (npr. samostalnik-predlog-samostalnik, glagol-predlog-samostalnik, pridevnik-predlog-samostalnik itd.), kjer po novem lahko upoštevamo tudi sklon predloga, ki v prejšnji slovnici besednih skic ni bil upoštevan zaradi prevelikega števila tako pridobljenih relacij/stolpcev v besedni skici. Nova slovnica je bila izdelana z upoštevanjem vseh struktur, registiriranih v leksikalni bazi v času izdelave, in ima tako bistveno več slovničnih relacij kot slovnica besednih skic, ki je bila uporabljena pri ročni izdelavi LBS. Skupaj je slovničnih relacij 103, število po relacijah je navedeno v tabeli 1.

direktiva	število
SEPARATEPAGE + TRINARY	36
DUAL	23
UNARY	2
CONSTRUCTION	13
CONSTRUCTION +UNARY	6
COLLOC	3
SYMMETRIC	2
brez	18
skupaj	103

Tabela 1: slovnične relacije po direktivah

Kot je razvidno iz tabele, so vse direktive s tremi elementi (*TRINARY) uporabljene v kombinaciji z izpisom na novi strani. Kombinacija direktiv CONSTRUCTION+UNARY je uporabljena v primeru, ko želimo, da nas sistem z izpisom v posebnem stolpcu "Constructions" opozori, da se neka kombinacija pogojev v korpusu pojavlja nadpovprečno pogosto (kar je sicer osnovna funkcija direktive UNARY). S pomočjo te direktive je pri luščenju mogoče tudi avtomatsko generirati opozorila, ki bi jih v klasičnih slovarjih pričakovali v t. i. slovničnih kvalifikatorjih, npr. pogosto zanikano, pogosto v 3. os. ednine itd. V postopku ALLP smo te podatke vključili v LBS v element <oznaka>, ki ima podobno vlogo kot kvalifikatorji. Pri vsaki od slovničnih relacij je naveden tudi podatek, kako se posamezna relacija prevaja v strukture, beležene v leksikalni bazi, s pomočjo katerih je mogoče identificirati neposredno povezavo med relacijo in elementom v leksikalni bazi. Primer:

*DUAL

=S_v_rodil-s/S_s-koga-česa

[2] Direktive določajo, kako program obravnava zapise v vrsticah, ki jim v slovnici besednih skic sledijo.

Struktura, s pomočjo katere luščimo kombinacije samostalnika v kateremkoli sklonu in samostalnika v rodilniku (npr. *delovanje motorja, valovanje morja*) se v leksikalni bazi pojavlja v strukturi SBZ0 sbz2, če je iztočnica jedrni samostalnik, ali v strukturi sbz0 SBZ2, če je iztočnica samostalnik v rodilniku. Ustrezen podatek o povezavi je dodan vsaki relaciji:

```
# LBS-XX ##########
# /1/ <struktura>SBZ0 sbz2</struktura>
# /2/ <struktura>sbz0 SBZ2</struktura>
####################
```

Opisana slovnica besednih skic je namenjena zgolj avtomatskemu luščenju podatkov iz korpusa, saj je za človeškega uporabnika razmeroma težko berljiva zaradi velikega števila relacij in kompleksih poimenovanj relacij.

3.4. GDEX za selekcionirano luščenje korpusnih zgledov

Vključevanje korpusnih zgledov predstavlja v LBS pomemben del človeškemu uporabniku namenjene informacije, saj se z njimi potrjujejo pomenska členitev in definicije, kolokacijske lastnosti besed, njihovo obnašanje v stavčnih vzorcih, tipične besedilne in žanrske rabe, pragmatika ipd. Zato je izbira dobrega korpusnega zgleda, ki naj bi, kot pravita Atkins in Rundell (2008: 458), ustrezal vsaj trem merilom: pristnosti in tipičnosti, informativnosti in razumljivosti, še toliko bolj pomembna. Iskanje takih zgledov pa postaja zaradi čedalje večjih korpusov in posledično velike količine podatkov vse težje in vse bolj zamudno.

Pomoč leksikografom pri iskanju dobrih zgledov predstavlja orodje GDEX (Good Dictionary EXamples; Killgarriff et al., 2008), ki zglede razvršča glede na njihovo kakovost. Ker pa so tipičnost, informativnost in razumljivost težko merljive lastnosti, aplikacija GDEX pri oceni kakovosti zgleda meri predvsem značilnosti, ki so z omenjenimi merili posredno povezane. Sem sodijo zlasti dolžina zgleda, celostavčna oblika, preprosta ali manj kompleksna skladenjska zgradba povedi, prisotnost ali odsotnost redkih besed, spletnih in elektronskih naslovov ipd.

Prva različica orodja GDEX je bila razvita za angleški jezik in uporabljena pri izbiri dodatnih zgledov za kolokacije v spletni postavitvi slovarja Macmillan English Dictionary. Pri projektu SSJ je bila za namene LBS izdelana različica za slovenščino (Kosem et al., 2011), ki je izboljšala angleško in precej olajšala delo leksikografom. Cilj, ki je bil bolj ali manj dosežen, je bil izdelati konfiguracijo, ki bi ponudila vsaj tri dobre zglede med desetimi ponujenimi za vsak kolokator v Besedni skici, pri čemer naj bi bil vsak pomensko ali skladenjsko relevanten podatek v LBS potrjen z najmanj dvema korpusnima zgledoma.

Obstoječa verzija aplikacije GDEX za slovenščino ni bila ustrezna za potrebe ALLP zaradi razlik v konceptu računalniško-leksikografskega dela. Pri običajnem postopku leksikograf s pomočjo korpusnih orodij analizira jezikovne podatke, jih selekcionira in vnese v program za izdelavo slovarjev. Pri postopku ALLP pa se podatki avtomatsko izvozijo iz korpusa neposredno v program za izdelavo slovarjev, kjer jih leksikograf pregleda, selekcionira in uredi. Ker smo z ALLP želeli občutno skrajšati postopek ročnega polnjenja posameznih elementov geselske zgradbe, hkrati pa razbremeniti tudi postopek odstranjevanja nerelevantnih ali neustreznih podatkov, ki se v slovarsko orodje prenesejo zaradi korpusnega šuma, lematizacijskih in drugih bolj ali manj predvidljivih napak, je bil naš cilj izdelati GDEX konfiguracijo, pri kateri bi bili prvi trije ponujeni zgledi že dovolj dobri za pojasnitev predhodno registriranih kolokacij.

Iz izkušenj pri procesu izdelave prvotne konfiguracije GDEX je bilo jasno, da bodo rezultati pri izboru kakovostnih zgledov pri posameznih besednih vrstah različni. Zato smo za vsako besedno vrsto, ki je zastopana v LBS, tj. za samostalnik, glagol, pridevnik in prislov, izdelali samostojno konfiguracijo, pri čemer se konfiguracije niso razlikovale v merilih, naštetih v tabeli 2, temveč v posameznih nastavitvah. Pri določanju nastavitev za posamezno besedno vrsto smo analizirali zglede, ki so bili v LBS že ročno izbrani na podlagi meril dobrih korpusnih zgledov. Na ta način smo dobili izhodiščne statistične vrednosti za klasifikatorje, na podlagi katerih smo izdelali konfiguracijo za vsako besedno vrsto.

- cela poved
- ne vsebuje pojavnico s frekvenco manj kot 3
- poved mora biti daljša od 7 pojavnic
- poved mora biti krajša od 60 pojavnic
- poved ne sme vsebovati ponovitve leme
- vsebuje elektronski ali spletni naslov
- optimalna dolžina (med X in Y pojavnic)
- vsebuje redke leme
- vsebuje pojavnice, daljše od 12 znakov
- število ločil v zgledu (brez vejic)
- število vejic v povedi
- pojavnice z velikimi začetnicami
- pojavnice z mešanimi simboli (npr. črke in številke)
- lastna imena
- zaimki
- položaj leme v povedi
- seznam prepovedanih besed na začetku povedi
- seznam prepovedanih besednih zvez na začetku povedi
- tretji kolokator
- Levenshteinova razdalja[3]

Tabela 2: Hevristika konfiguracij orodja GDEX za slovenščino za ALLP

Drugi del analize za določanje najprimernejših GDEX konfiguracij je vključeval evalvacijo zgledov predhodne konfiguracije v orodju SkE na vzorčnem izboru lem s seznama za ALLP, sledilo je prilagajanje nastavitev oz. izdelava nove različice konfiguracije ter ponovna evalvacija. Postopek smo ponavljali, dokler nismo izoblikovali optimalne končne verzije konfiguracije GDEX za postopek ALLP. Pomemben rezultat tega dela analize je oblikovanje več novih klasifikatorjev, ki jih prvotna verzija GDEX ni vključevala. Zlasti npr. oblikovanje seznama prepovedanih besed ali zvez na

[3] http://en.wikipedia.org/wiki/Levenshtein_distance

začetku povedi in upoštevanje t. i. tretjega kolokatorja. Predvsem zadnje prinaša pri izboru korpusnih zgledov v postopku ALLP dobre rezultate, saj posredno upošteva merilo koligacijske tipičnosti določene kolokacije. Npr. pri kolokaciji *klavrn + podoba* klasifikator višje točkuje zglede s statistično pomembnim tretjim kolokatorjem *kazati*. Izbrana konfiguracija pa posledično ponudi zglede, ki vsebujejo tipično širšo strukturo kolokabilne okolice: *kazati klavrno podobo česa*.

3.5. Priprava API skripte

Prilagoditev slovnice besednih skic in konfiguracije GDEX sta bila predpogoja za pripravo API skripte, ki je zahtevala tudi usklajevanje oz. posodabljanje orodja SkE. API skripta je napisana v programu Python in omogoča luščenje podatkov s povezavo na strežnik, kjer je nameščen SkE, ter določitev ukaznih parametrov, kot so:

- korpus
- lema (za več lem je potrebna datoteka s seznamom)
- slovnična relacija (za več relacij je potrebna datoteka s seznamom)
- GDEX konfiguracija
- število zgledov na kolokator
- število kolokatorjev na slovnično relacijo
- minimalna frekvenca kolokatorja
- minimalna frekvenca slovnične relacije
- minimalna jakost kolokatorja (salience)
- minimalna jakost slovnične relacije (salience).

Za izdelavo API skripte je bilo potrebno pripraviti XML predlogo, ki smo jo nato uporabili pri izvozu podatkov. Da bi bilo avtomatsko izluščene podatke mogoče uvoziti v slovarski program iLex, je bilo potrebno predlogo ustrezno poenotiti z DTD strukturo LBS. Zaradi lažjega pregledovanja izvoženih podatkov smo v DTD dodali atribute pri elementih <kolokacija> in <zgled>, in sicer identifikacijsko številko za kolokator (v oba elementa zaradi možnosti identifikacije povezave med zgledom in kolokatorjem), indeksno številko pojavnice pri elementu <zgled>, kar bi omogočilo identifikacijo zgledov v korpusu, ter zaporedno številko zgleda za vsak kolokator v GDEX-ovi razvrstitvi zgledov.

3.5.1. Določanje parametrov

Prvi test ALLP smo izvedli s privzetimi nastavitvami: 10 kolokatorjev na relacijo, 6 zgledov na kolokator, minimalna jakost relacije ali kolokatorja = 0, minimalna frekvenca kolokatorja = 0, minimalna frekvenca relacije = 25, vendar so prve evalvacije pokazale, da ni mogoče uporabiti enakih nastavitev za vse relacije in kolokatorje, saj je izpis pri nekaterih lemah pokazal veliko nerelevantnih relacij in pripadajočih kolokatorjev, pri drugih pa nekatere relevantne relacije in kolokatorji niso bili zabeleženi. Izkazalo se je tudi, da je izluščenih zgledov za končno urejanje gesla občutno preveč.

Izhodiščne nastavitve smo v nadaljevanju izboljšali tako, da smo iz besednih skic vseh lem z našega seznama pridobili statistične podatke o relacijah in kolokatorjih, nato pa za vsako relacijo (v okviru skupine lem iste besedne vrste) analizirali vrednosti, pri čemer smo iskali optimalne minimalne frekvence in jakosti relacije. Pomagali smo si tudi s podatkom o deležu pojavitev leme v določeni relaciji. Statistično analizo smo kombinirali z ročnim pregledovanjem besednih skic, saj se je pri nekaterih lemah, zlasti tistih, kjer se je relacija pojavljala redkeje, izkazalo, da relacija za luščenje ni relevantna. Dodatna korist ročnega pregledovanja besednih skic je bila identifikacija nekaterih pomanjkljivosti v slovnici besednih skic (npr. napačno opredeljena ali klasificirana relacija), ki smo jih pred izvedbo končnega postopka odpravili.

Pri določanju minimalne vrednosti frekvence in jakosti kolokatorjev smo se oprli na podatke, ki smo jih pridobili z ročnim pregledovanjem besednih skic pri posamezni besedni vrsti in pri različnih slovničnih relacijah. Pri določanju minimalnih statističnih vrednosti na kolokator smo v besedni skici upoštevali kolokatorje, ki so predstavljali še smiselne kombinacije ter uporabili njihove statistične parametre kot osnovo za določitev vrednosti.

Pregled izluščenih podatkov na podlagi izhodiščnih nastavitev je med drugim pokazal, da je nastavitev števila kolokatorjev na slovnično relacijo za končni rezultat zelo pomemben parameter. Če namreč med prvimi desetimi kolokatorji (privzeta nastavitev) ni takih, ki bi presegali minimalno frekvenco in jakost, se relacija pri luščenju ne izpiše, četudi je zelo pogosta. Zato smo minimalno število kolokatorjev na relacijo dvignili na 25, luščenje relevantnih kolokatorjev pa 'prepustili' parametroma za minimalno frekvenco in jakost kolokatorja. Število zgledov na kolokator smo znižali na 3, tudi zaradi tega, ker je evalvacija testnih izpisov pokazala, da je v veliki večini primerov med njimi vsaj en dober zgled (pogosto pa kar vsi trije).

3.6. Od izpisa do gesla

Najboljši pokazatelj učinkovitosti ALLP je čas, v katerem iz izluščenih podatkov izdelamo končno verzijo gesla v LBS. Pri tem je dovolj zgovoren podatek, da leksikograf na podlagi ročne analize korpusnih podatkov (tj. z uporabo Besednih skic) izdela v eni uri slabo četrtino gesla oz. povprečno 0,23 gesla/uro, medtem ko je mogoče na podlagi ALLP izdelati geslo v dveh urah oz. povprečno 0,5 gesla/uro. V enakem času je torej razmerje v prid na podlagi ALLP izdelanih gesel 2 : 1. Ko so leksikalni podatki izluščeni in uvoženi v slovarsko orodje, jih leksikograf pregleda in ustrezno pomensko razčleni oz. združi. Prav tako (za zdaj) ostaja v leksikografovi pristojnosti identifikacija stalnih zvez, frazeoloških enot in pragmatičnih lastnosti pomena. Osnovni vsebinski doprinos leksikografa je še vedno tudi ubesedenje pomenskih indikatorjev in izdelava definicij ter dodajanje stilnih in področnih oznak.[4]

4. Sklep

Velik delež časa pri oblikovanju gesla na podlagi izluščenih podatkov ostaja namenjen razvrščanju in selekcioniranju podatkov, na primer združevanju kolokacij pod posamezne pomene in hkratno razvrščanje ustreznih kolokacijsko povezanih korpusnih zgledov, kopiranju kolokacij in razvrščanju zgledov v primerih, ko se kolokacija pojavi pri več kot enem (pod)pomenu. Poleg

[4] Slovnične oznake, kot denimo pogosta raba tretjeosebne konstrukcije, je, kot rečeno, avtomatično izluščena s kombinacijo direktiv CONSTRUCTION+UNARY.

tega zahteva končna ureditev gesla tudi ustrezno prečrkovanje struktur, kolokacij in skladenjskih zvez. Vendar pa je končna izdelava gesel pokazala, da je vsaj del teh opravil mogoče avtomatizirati in da obstaja še veliko možnosti za izboljšanje postopka. Menimo, da je v prispevku opisani postopek ALLP koristna pridobitev za slovensko leksikografijo, zlasti če upoštevamo stanje v trenutni slovenski leksikografski praksi, ki ne obeta sodobnega slovarskega priročnika v doglednem času. Prikazana metodologija avtomatskega luščenja je jezikovno neodvisna, posamezni parametri na vseh stopnjah ALLP pa so prilagodljivi tudi za druge jezike, kar je pomembno tudi z vidika sodobne leksikografske prakse, kjer se zagovarjajo in preizkušajo metode, ki bi v čim večji meri avtomatizirale slovarsko delo (prim. Rundell in Kilgarriff, 2011).

V prihodnje načrtujemo izboljšavo postopka ALLP, pri čemer nameravamo podrobneje analizirati avtomatsko izluščene podatke in jih primerjati z ročno pridobljenimi v smislu relevantnosti in podrobnosti oz. robustnosti leksikografskega opisa. ALLP nameravamo preizkusiti tudi na pogostejših večpomenskih lemah, predvsem z namenom podrobnejšega testiranja konfiguracij GDEX, saj pri manj pogostih lemah zaradi manjše frekvence kolokatorjev in manjše izbire zgledov učinkovitost orodja dejansko ne pride do izraza.

Posvetili se bomo tudi izboljšavi postopka na ravni različnih nastavitev parametrov API skripte in na ravni izpisa. Pri izpisu izluščenih podatkov v slovarsko orodje nameravamo vključiti postopke, ki bodo dodatno skrajšali čas končnega urejanja gesla. V mislih imamo avtomatsko odstranjevanje kolokatorjev, ki ponudijo same enake zglede (gre za korpusni šum, ki neupravičeno izpostavi določeno besedo kot kolokator), in postavitev leme in/ali kolokatorja pri izpisu v slovarsko orodje v ustrezen sklon, spol in število. Raziskati nameravamo uporabo funkcije gručenja (clustering) kolokacij na podlagi podatkov iz tezavra v orodju SkE ter preizkusiti možnost uporabe funkcije povezav na večbesedne leksikalne enote (MWU links), kar bi, predvidevamo, omogočilo luščenje t. i. razširjenih kolokacij tipa: *[delovno] mesto → [prosto, novo] delovno mesto → [razpisati, objaviti] [prosto, novo] delovno mesto.* Dolgoročni načrti vključujejo pomensko razdvoumljanje s pomočjo podatkov iz sloWNeta in izdelavo geselskih predlog za ALLP na podlagi sistemske polisemije.

S prikazano zasnovo ALLP še zdaleč nismo izčrpali vseh možnosti, ki jih postopek ponuja, vsekakor pa izkušnje in prvi rezultati kažejo, da je z današnjega vidika in prihodnjih trendov v leksikografiji pomembno vlagati v razvoj temeljnih jezikovnih virov, ki so zasnovani z mislijo na avtomatsko izrabo podatkov, in jezikovnih orodij, ki znajo podatke v takšnih jezikovnih virih izkoristiti.

5. Literatura

Atkins, T. B. S., Rundell, M. 2008. *The Oxford Guide to Practical Lexicography*. Oxford: Oxford University Press.

Erlandsen, J. 2004. iLex – new DWS. Third International Workshop on Dictionary Writing systems (DWS 2004). Brno, 6. – 7. september 2004. Dostopno na: http://nlp.fi.muni.cz/dws2004/pres/#15.

Fišer, D. 2009. SloWNet – slovenski semantični leksikon. V Stabej, M. (ur.): *Infrastruktura slovenščine in slovenistike (Obdobja 28)*. Ljubljana: Znanstvena založba Filozofske fakultete. 145–149.

Gantar, P., Krek, S. 2011. Slovene lexical database. V Majchraková, D., Garabík, R. (ur.): *Natural language processing, multilinguality: sixth international conference, Modra, Slovaška, 20-21. oktober 2011*. 72-80.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., Rychly, P. 2008. GDEX: Automatically finding good dictionary examples in a corpus. V Bernal, E., DeCesaris, J. (ur.): Proceedings of the 13th Euralex International Congress. Barcelona: Institut Universitari de Linguistica Aplicada, Universitat Pompeu Fabra. 425-432.

Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D. 2004. The Sketch Engine. V: Williams, G., Vessier, S. (ur.): Proceedings of the 11th Euralex International Congress. Lorient: Universite de Bretagne-Sud. 105-116.

Kilgarriff, A., Rundell, M. 2002. 'Lexical profiling software and its lexicographic applications: a case study'. A. Braasch et al. (ur.) EURALEX 2002 Proceedings. Copenhagen: University of Copenhagen.

Kosem, I., Husák, M., McCarthy, D. 2011. GDEX for Slovene. V Kosem, I., Kosem K. (ur.): *Electronic Lexicography in the 21st Century: New applications for new users, Proceedings of eLex 2011, Bled, 10-12 November 2011*. Ljubljana: Trojina, zavod za uporabno slovenistiko. 151-159.

Krek, S. 2012. New Slovene sketch grammar for automatic extraction of lexical data. SKEW3, tretja mednarodna delavnica orodja Sketch Engine, Brno, Češka, 21-22. marec 2012. Dostopno na: https://trac.sketchengine.co.uk/attachment/wiki/SKEW-3/Program/Krek_SKEW-3.pdf?format=raw.

Lorentzen, H., Theilgaard, L. 2012. Online dictionaries – how do users find them and what do they do once they have? V Vatvedt Fjeld, R., Torjusen, J. M. (ur.): *Proceedings of the 15th EURALEX International Congress, Oslo, 7–11 August 2012*. Oslo: University of Oslo, Department of Linguistics and Scandinavian Studies. 654-660.

Rundell, M., Kilgarriff, A. 2011. Automating the creation of dictionaries: where will it all end? V Meunier, F., De Cock, S., Gilquin, G., Paquot, M. (ur.): *A Taste for Corpora. A tribute to Professor Sylviane Granger*. Amsterdam: John Benjamins.

Izdelava XML-shem za slovarske projekte na primeru nastajajočih tipološko raznovrstnih slovarjev

Nina Ledinek, Andrej Perdih

Inštitut za slovenski jezik Frana Ramovša ZRC SAZU
Novi trg 4, 1000 Ljubljana
{NLedinek, APerdih}@zrc-sazu.si

Povzetek

Prispevek prikazuje dileme, ki so se pojavile pri oblikovanju XML-shem za tipološko raznovrstne slovarske podatkovne baze Inštituta za slovenski jezik Frana Ramovša ZRC SAZU (*Slovar sinonimov slovenskega jezika, Slovar slovenskega knjižnega jezika 16. stoletja, Novi slovar slovenskega jezika, Slovar novejšega besedja slovenskega jezika, Slovenski etimološki slovar, Slovar pravopisno težavnega besedja slovenskega jezika*). Pojasnjuje praktične in tehnične dejavnike, ki na izdelavo XML-shem vplivajo z vidika leksikografskega dela pri vnosu jezikovnih podatkov, opisuje pa tudi tiste dejavnike, ki vplivajo na učinkovito izrabo podatkov iz dokončanih slovarskih podatkovnih baz.

Designing XML Schemas for dictionary projects: the case of emerging typologically varied dictionaries

The article deals with dilemmas that have arisen in the designing of XML Schemas of typologically varied dictionary databases of the Fran Ramovš Institute of the Slovene language (Dictionary of Slovene Synonyms, Dictionary of the Literary Slovene Language of the Sixteenth Century, The New Dictionary of Slovene Language, Dictionary of Newer Standard Slovene Words, Slovene Etymological Dictionary, Dictionary of Less Used Slovene Words). It presents the conceptual, practical and technical factors that influence the designing of XML Schemas from the point of view of lexicographical work, but also describes the factors that determine the efficiency of the use of data from the completed dictionary databases.

1. Uvod

V zadnjih desetletjih so se na področju leksikografije zgodili tehnološki in konceptualni premiki, ki so odločilno vplivali na metodologijo leksikografskega dela ter na dojemanje in uporabo slovarskih priročnikov in drugih sorodnih jezikovnih virov. Morda najodločilneje je sodobno leksikografijo zaznamovalo dejstvo, da leksikografi in uporabniki slovarskih priročnikov ne dojemajo več kot (izhodiščno) knjižnih jezikovnih virov, ampak kot večnamenske razširljive strukturirane računalniško berljive podatkovne baze, v katerih so podatki ustrezno hierarhizirani, (standardno) označeni in medsebojno povezani. Vzajemno s konceptualnimi spremembami se je, ob sočasnem razvoju informacijske tehnologije oz. elektronskih medijev, spremenilo tudi dojemanje slovarskih priročnikov in sorodnih jezikovnih virov – ti namreč niso več namenjeni zgolj tehnološko vedno bolj spretnim in zahtevnim človeškim uporabnikom, ampak jih izkoriščamo kot jezikovne vire tudi pri številnih nalogah procesiranja naravnih jezikov.

V prispevku prikazujemo zadrege in dileme, na katere smo naleteli ali se z njimi trenutno srečujemo pri oblikovanju novih računalniško berljivih slovarskih podatkovnih baz v formatu XML, pa tudi na tiste, ki so povezane s pretvorbo slovarskih virov Inštituta za slovenski jezik Frana Ramovša ZRC SAZU v format XML iz drugih elektronskih formatov. Izkazalo se je, da zaradi raznolikosti slovarskih priročnikov, ki nastajajo na inštitutu (*Slovar sinonimov slovenskega jezika, Slovar slovenskega knjižnega jezika 16. stoletja, Novi slovar slovenskega jezika, Slovar novejšega besedja slovenskega jezika, Slovenski etimološki slovar, Slovar pravopisno težavnega besedja slovenskega jezika*, terminološki slovarji ...), oblikovanje podatkovne baze in XML-sheme za vsak slovarski projekt tako s tehničnega kot tudi leksikografskega vidika prinaša svojevrstne dileme.

2. Standardni format XML in slovarske podatkovne baze

Kot standardni format za zapis slovarskih in drugih jezikovnih podatkovnih baz se je zaradi svoje univerzalnosti in fleksibilnosti uveljavil XML (eXtensible Markup Language). Gre za označevalni jezik, ki je zelo primeren za večnivojsko hierarhično strukturiranje podatkov, torej tudi jezikovnih virov, saj ti navadno vključujejo veliko število hierarhično urejenih podatkovnih tipov. XML odlikuje razmeroma preprosta sintaksa, ker pa metaoznake niso definirane vnaprej, lahko vsebino strukturiramo z metaoznakami, ki so logične in intuitivno razumljive, s čimer dobimo dober nadzor nad logično strukturo podatkovne baze. Datoteke XML so navadne besedilne datoteke s privzetim unikodnim kodiranjem znakov, zato so primerne za dolgoročno shranjevanje podatkov ter njihovo izmenljivost in prenosljivost med različnimi orodji in operacijskimi sistemi.

Eden ključnih in hkrati najzahtevnejših korakov pred samo vzpostavitvijo slovarske podatkovne baze je oblikovanje ustreznega slovarskega koncepta, ki bo služil kot vodilo za oblikovanje vseh predvidenih geselskih sestavkov. Predpostavljena struktura geselskih sestavkov mora biti takšna, da podpira logično in intuitivno strukturiran vnos vseh jeziko(slo)vnih podatkov, ki so za opis posamezne leksikalne enote pomembni, hkrati pa dovolj univerzalna in striktna, da redaktorje usmerja k čim bolj sistematični in konsistentni interpretaciji raznovrstnih jezikovnih podatkov na tak način, da so ti uporabni in razumljivi tudi za končne uporabnike slovarskih priročnikov.

Formalno strukturo slovarske podatkovne baze v formatu XML opisuje t. i. shema, ki jo lahko razumemo kot nekakšno projekcijo formalnih značilnosti mikro- in makrostrukture slovarja, kot ju določa slovarski koncept, v standardni računalniški jezik. Shema določa zlasti, kateri

so možni oz. dovoljeni elementi in atributi slovarske podatkovne baze, kakšna so hierarhična razmerja med njimi in po kakšnem vrstnem redu si elementi sledijo, kakšne so omejitve njihove rabe z vidika sopojavljanja oz. izključevanja, kolikokrat se posamezen element na določenem mestu lahko ponovi. Shema ne nazadnje določa tudi, kakšna sme biti formalna vsebina elementov. Predpisuje, da lahko določen element vključuje le druge elemente ali pa morda le slovarsko besedilo ali številke, ki jih v podatkovno bazo vnese redaktor, omejuje dolžino vnesenega besedila, določa, ali je vsebina specifičnega elementa omejena na seznam vnaprej določenih izbir (npr. na določen nabor besednih vrst, kvalifikatorjev), predpisuje rabo obveznih atributov ipd.

Obstajajo sheme različnih formatov – DTD, XML Schema[1] (.xsd) in RELAX NG (.rng). Format XML-shema, ki je nekoliko bolj fleksibilen od formata DTD, uporabljamo pri opisu slovarskih podatkovnih baz Inštituta za slovenski jezik Frana Ramovša ZRC SAZU, saj njegovo uporabo podpira leksikografski program iLEX,[2] s katerim pripravljamo redakcije slovarjev.[3] Ena temeljnih nalog takšnega leksikografskega programa, poleg dejstva, da redaktorjem omogoča urejanje interpretiranih jezikovnih podatkov in njihovo vizualizacijo, je, da nadzoruje skladnost geselskih sestavkov z določili sheme, tako da opozarja na neustreznosti v formalni vsebini in strukturi segmentov slovarske podatkovne baze. Program, podprt z ustrezno shemo, pripomore k zagotavljanju konsistentnosti slovarske podatkovne baze na formalni ravni, seveda pa ne more preprečiti vsebinskih neustreznosti.

Leksikografi običajno ugotavljajo, da je pri načrtovanju XML-sheme za slovarski projekt smiselno upoštevati številne dejavnike. Ključnega pomena je med drugim, da se konceptualizacija slovarske strukture pri redaktorjih čim bolj neposredno odraža v intuitivno razumljivi hierarhično strukturirani podatkovni strukturi, kot je definirana s shemo – da torej posamezen mikrostrukturni element slovarja, npr. zaglavje, vključuje tiste podelemente, ki jih redaktor vidi v tem razdelku, ne pa npr. v okviru posameznih pomenov. Obenem je treba upoštevati tudi praktični vidik obvladljivosti slovarske strukture pri redaktorjih. Število mikrostrukturnih elementov posameznega slovarja lahko namreč doseže trimestno številko, pri čemer se lahko posamezni podatkovni tipi (npr. razdelki, v katerih je izkazan pomen leksikalnih enot), morda strukturno le nekoliko modificirani, pojavijo na različnih mestih v slovarski strukturi. Pri pripravi sheme se je zato treba vprašati, koliko različnih elementov je v shemo smiselno vključiti (tj. ali vsak slovarski podatek označiti kot svoj element ali pa sorodne podatke obravnavati v okviru istega elementa), da bo podatkovna struktura vsebinsko logična, spominsko za redaktorje ne preveč obremenjujoča, hkrati pa tehnično razmeroma lahko obvladljiva (npr. če bi ob analizi dodatnih jezikovnih podatkov redaktor ugotovil, da je treba strukturo geselskega sestavka spremeniti).

Dileme se pojavljajo tudi glede vprašanja, kako hierarhično globoko je posamezen element praktično umeščati. Hierarhično globlje strukturirane enote so po eni strani primernejše, ker omogočajo večjo segmentacijo jezikovnih podatkov, posledično pa tudi iskanje podatkov glede na kompleksnejše iskalne pogoje in podrobnejše postprocesiranje podatkov, po drugi strani pa velja, da pretirano razvejane podatkovne strukture redaktorjem otežujejo navigacijo po slovarskih geslih, zaradi česar se čas redigiranja geselskih sestavkov podaljša, pozornost redaktorja pa je v večji meri usmerjena k vzpostavljanju ustreznih hierarhičnih razmerij, manj pa k vsebinskim vprašanjem. Odločitev o združevanju več elementov v okviru določenega nadelementa je hkrati treba pretehtati še s tehničnega vidika, saj nanj vpliva tudi leksikografski program. Bistvenega pomena je namreč, kakšne možnosti prikaza podatkov omogoča uporabniški vmesnik in kakšne možnosti iskanja po že vzpostavljeni podatkovni bazi program omogoča. Ob pripravi XML-sheme si je poleg tega treba zastavljati še vprašanja o tem, kako predvideno strukturiranje podatkov vpliva na možnosti rabe jezikovnega vira pri uporabnikih, kakšne so možnosti prikaza podatkov v elektronski ali tiskani obliki glede na vzpostavljeno podatkovno strukturo, ali je omogočeno vzpostavljanje sklicev na različne segmente podatkovne baze in kakšno postprocesiranje baze je predvideno.

V nadaljevanju prispevka na kratko in shematično prikazujemo, kako smo skušali pri pripravi XML-shem za različne slovarske projekte, ki nastajajo na inštitutu za slovenski jezik, omenjene dejavnike (našteti so seveda le nekateri od številnih možnih) prepoznati, jih upoštevati in njihovo součinkovanje čim bolj smiselno uravnotežiti.

Za zapis slovarskih oz. leksikonskih podatkovnih baz so bili po svetu vzpostavljeni različni standardi zapisa, npr. Lexical Markup Framework (LMF),[4] Lexical Interchange Format Standard (LIFT),[5] priporočila iniciative TEI za zapis slovarskih podatkov[6] ipd. Kljub temu da bi bilo podatke večinoma mogoče zapisati v katerem od standardnih formatov, se pri nas za ta korak zaenkrat še nismo odločili. Odločitev je bila posledica presoje, da je tak zapis pri delu slovarskih baz manj smiseln npr. zaradi njihove specifičnosti (*Slovar novejšega besedja slovenskega jezika* je slovar manjšega obsega in je zasnovan predvsem kot dopolnilo *Slovarja slovenskega knjižnega jezika*, ki ni zapisan v skladu z omenjenimi standardi) ali izjemne kompleksnosti slovarske mikrostrukture, zaradi česar bi bila vzpostavitev standardnega zapisa težavna, kljub njegovi fleksibilnosti bi bilo namreč standardu včasih težko slediti (slednje velja zlasti za *Slovar sinonimov slovenskega jezika* in *Slovar slovenskega knjižnega jezika 16. stoletja*). Pri odločitvi smo upoštevali še dejstva, da so številne slovarske baze primarno namenjene (specializiranim) človeškim uporabnikom, v manjši meri pa so uporabne v postopkih procesiranja naravnih jezikov (npr. *Slovenski etimološki slovar*, *Slovar slovenskega knjižnega jezika 16. stoletja*), da so bili koncepti mnogih slovarjev, posledično pa tudi strukturiranje podatkov, oblikovani v času, ko omenjeni

[1] V prispevku je uporabljen poslovenjen zapis XML-shema.
[2] http://www.emp.dk/
[3] Med znanimi leksikografskimi programi so še ABBYY Lingvo Content (http://www.abbyy.com/lingvo_content/), IDM DPS (http://www.idm.fr/products/dictionary_writing_system_dps/27/) in TshwaneLex (http://tshwanedje.com/tshwanelex/), v slovenskem okolju pa je nastala Termania (http://www.termania.net/). IDM DPS uporablja format sheme DTD, iLEX in Termania podpirata XML-shemo, Tshwanelex pa vključuje interni DTD.
[4] http://www.lexicalmarkupframework.org/
[5] http://code.google.com/p/lift-standard/
[6] http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DI.html

standardi še niso bili vzpostavljeni ali splošno razširjeni, poleg tega pa slovarski podatki niso zapisani v formatu XML, zato jih je treba vanj šele pretvoriti, strukturiranje, skladno s standardi, pa je posledično povezano z večjimi finančnimi vložki. Baze so večinoma tudi še v fazi oblikovanja oz. revizije. Deloma je na odločitev vplivala tudi ocena, v kolikšni meri je slovarska podatkovna baza zanimiva z vidika izmenljivosti podatkov in postprocesiranja.

Glede na naštete vidike sta slovarski bazi *Novega slovarja slovenskega jezika* in *Slovarja pravopisno težavnega besedja slovenskega jezika* drugačni, saj sta z vidika izmenljivosti podatkov in predvidenega postprocesiranja bistveno bolj uporabni. Zaenkrat sicer nista oblikovani v katerem od navedenih standardov, sta pa strukturirani na način, ki omogoča relativno enostavno pretvorbo v katerega od njih.

3. Oblikovanje XML-shem za slovarske podatkovne baze

3.1. Slovar sinonimov slovenskega jezika[7]

Slovar sinonimov slovenskega jezika bo prvi specializirani slovarski priročnik za slovenščino, ki bo sistematično izkazoval sinonimna razmerja med leksikalnimi enotami v jeziku. Glede na sklicno naravo slovarja se je pri oblikovanju njegove XML-sheme kot temeljno pojavljalo vprašanje, kako uravnotežiti zahtevo redaktorjev po tem, da bo predpostavljena podatkovna struktura zanje čim bolj logična in intuitivno razumljiva, hkrati pa ustrezna tudi z vidika vzpostavljanja sklicev na različne mikrostrukturne elemente geselskih sestavkov ter predvideni prikaz podatkov v elektronski in tiskani obliki.

Slovarski koncept predvideva geselske sestavke dveh tipov. V dominantnih geselskih sestavkih je poleg osnovnih slovničnih lastnosti v zaglavju za vsako geselsko besedo navedeno, v katerih pomenih in besednih zvezah, katerih skladenjsko jedrni del je, se pojavlja kot nezaznamovan, tj. najbolj običajen leksem za izražanje konkretnega pomena, ob vsakem pomenu in besedni zvezi pa so navedeni še drugi identificirani sinonimni leksemi, ki jih je v različnih okoliščinah mogoče uporabiti za izražanje istega pomena. Slovar prinaša tudi podatke o delnih sinonimih geselske besede v posameznih pomenih, poleg tega pa še poseben sklicni razdelek geselskega sestavka, v katerem je navedeno, v katerih pomenih iztočnica ni dominantni leksem za izražanje določenega pomena, ampak le eden od nedominantnih sinonimov k dominantnemu leksemu, obdelanem v drugem dominantnem geselskem sestavku.

Kazalčni geselski sestavki prinašajo poleg osnovnih slovničnih informacij o iztočnici le sklice na dominantne geselske sestavke, v katerih je iztočnični leksem naveden kot eden od zaznamovanih sinonimov k nezaznamovanemu ali najbolj običajnemu leksemu za izražanje konkretnega pomena. Iztočnica kazalčnega geselskega sestavka je lahko v okviru dominantnega sestavka navedena kot enobesedni sinonim ali pa kot ena od besed v večbesednem sinonimu.

Da bi količino ročnega dela zmanjšali in redakcijski proces pohitrili, smo se odločili, da bodo sklici s kazalčnih geselskih sestavkov na dominantne ob koncu redakcije slovarja na podlagi podatkov v dominantnih geselskih sestavkih samodejno generirani. Z vidika oblikovanja XML-sheme je bila odločitev nekoliko zahtevnejša zaradi zadrege, da med samim redakcijskim procesom ni mogoče z gotovostjo napovedati, ali je konkretna iztočnica izhodišče kazalčnega ali dominantnega geselskega sestavka, saj se strukturni tip konkretnega sestavka lahko določi šele ob redakciji veliko različnih iztočnic. Odločitev o tipu geselskega sestavka, katerega izhodišče je posamezna iztočnica, se namreč vzpostavi na podlagi analize sinonimnih razmerij, ki jih vzpostavljajo vsi leksemi, ki izražajo pomene, ki jih je mogoče izraziti tudi s konkretno iztočnico. Navedeno dejstvo ima neprijetno posledico, da se mikrostruktura oz. strukturni tip posameznega geselskega sestavka lahko spremeni praktično v kateri koli fazi priprave slovarskega besedila.

Pripravljavci XML-sheme so bili zaradi te lastnosti slovarja prisiljeni v iskanje smiselnega ravnovesja med dvema nasprotujočima si težnjama. Po eni strani so bili soočeni z zahtevo, da podatkovne strukture geselskih sestavkov obeh tipov oblikujejo čim bolj enotno in z malo shemskimi elementi, da bi bilo popravkov, če bi redaktor v procesu izdelave slovarja ugotovil, da mora strukturni tip specifičnega geselskega sestavka ali del njegove podatkovne strukture spremeniti, čim manj. Po drugi strani je bilo treba upoštevati vidik sklicevanja. Če bi prevladala odločitev za zelo malo različnih elementov XML-sheme, bi se ti lahko sklicevali na več različnih mest v strukturi dominantnih geselskih sestavkov, kar bi bilo z vidika prikaza možnih tarč sklicev v okviru vmesnika leksikografskega programa in izbire ustreznega tarčnega elementa v pogovornem oknu manj primerno, saj je v tem primeru možnosti za napake več, poleg tega pa bi bilo v veliki meri onemogočeno samodejno vzpostavljanje sklicev. Alternativna možnost bi bila vzpostavitev velikega števila podatkovnih tipov, ki bi bili z vidika sklicevanja zamejeni bolj jasno, vendar bi bilo morebitno spreminjanje strukture določenega geselskega sestavka za redaktorje težavnejše. Pri oblikovanju končne različice XML-sheme smo se zato odločili za kompromisno možnost. Različnih elementov XML-sheme, v katere redaktorji dejansko vpisujejo podatke o sinonimnih razmerjih, je razmeroma malo, vendar pa so ti umeščeni v jasno strukturirane različne nadelemente,[8] ki so redaktorjem logični tudi glede na njihovo konceptualizacijo slovarske strukture, znotraj njih pa so omenjeni elementi na vseh nivojih slovarske strukture hierarhizirani na enak način, zato XML-shema za redaktorje spominsko in tehnično ni preveč obremenjujoča.

Elementom, v katere so vpisani (eno- in večbesedni) zaznamovani sinonimi v dominantnih geselskih sestavkih, bodo ob koncu redakcije slovarja samodejno pripisani atributi z unikatnimi oznakami ID. Skupaj z vsebino nekaterih drugih elementov dominantnih geselskih sestavkov, zlasti neonaglašenih iztočnic, jih bomo nato izvozili v ustrezne, zlasti sklicevalne elemente kazalčnih in dominantnih geselskih sestavkov, na podlagi pripisanih atributov ID pa bodo vzvratno samodejno vzpostavljeni

[7] Natančnejše podatke o oblikovanju XML-sheme za ta slovar prinaša prispevek Ledinek et al. (2012).

[8] Ti se v podatkovni hierarhiji pojavljajo razmeroma visoko, strukturiranje v okviru teh elementov, ki obsega več hierarhičnih podnivojev, pa je praviloma enotno, zato je popravkov ob želji po spreminjanju podatkovne strukture sorazmerno malo.

tudi sklici na (nad)elemente slovarske strukture dominantnih geselskih sestavkov, v katerih so se kot nedominantni sinonimi pojavili. Eno- in večbesedni sinonimi, ki so obravnavani v dominantnih geselskih sestavkih v okviru dopolnilnih razdelkov kot delne sopomenke, se sklicujejo na različne segmente dominantnih geselskih sestavkov (npr. na različne pomene). Ker je z vidika koncepta *Slovarja sinonimov slovenskega jezika* večina iztočnic enopomenskih, hkrati pa večina besednih zvez v slovarju ne nastopa več kot enkrat, bo mogoče tudi večino sklicev z elementov v razdelku delne sopomenke vzpostaviti avtomatsko. Ocenjujemo torej, da bo od nekaj sto tisoč sklicev, ki jih bo slovar prinašal, ročno treba povezati le nekaj tisoč sklicev.

Slovar sinonimov slovenskega jezika je začel nastajati po vzpostavitvi koncepta (Ahlin et al., 2003) brez specializiranih leksikografskih orodij. V programu iLEX, podprtem z opisano XML-shemo, smo slovarsko gradivo pričeli redigirati ob koncu leta 2011, trenutno pa se ukvarjamo z začetno fazo pretvorbe že obstoječega slovarskega gradiva v obsegu približno 30.000 geselskih sestavkov v format XML. Pri pretvorbi se težave pojavljajo zaradi občasne nekonsistentnosti v rabi slogov in ločil, več ročnega dela pa bo v nadaljnjih fazah dela potrebnega zaradi preverjanja ustreznosti segmentacije in lematizacije posameznih segmentov večbesednih enot, saj tudi deli večbesednih enot glede na vzpostavljeni koncept slovarja lahko nastopajo kot iztočnice kazalčnih geselskih sestavkov.

3.2. Slovenski etimološki slovar

Slovenski etimološki slovar je eden od slovarskih projektov, ki so z vidika pretvorbe v standardni format XML in oblikovanja ustrezne XML-sheme zanj porajali najmanj dilem. Podatkovna baza slovarja je bila namreč že izhodiščno oblikovana kot strukturirana podatkovna baza v računalniškem programu Eva[9] ter zelo natančno in premišljeno označena, vendar je bilo strukturiranje in označevanje podatkov vzpostavljeno zlasti z mislijo na možnost ustreznega izpisa slovarskih podatkov in njihovega indeksiranja, zato se v nekaterih segmentih pomembno razlikuje od logike običajnega strukturiranja podatkov v formatu XML.[10] Zaradi omenjenih lastnosti baze se vsemu ročnemu delu pri pretvorbi ni bilo mogoče izogniti.

Že vzpostavljene rešitve v podatkovni bazi so v precejšnji meri pogojevale oblikovanje strukture XML-sheme, ki pa je, glede na običajne slovarske sheme, specifična zlasti zaradi same narave etimoloških jezikovnih podatkov. Zaradi »proznega« zapisa geselskega sestavka etimološkega slovarja, v katerem se rekonstruirane oblike, pomeni besed, tujejezične besede in njihovi pomeni, oznake za jezike, iz katerih izhajajo, in »navadno« pojasnjevalno besedilo med njimi pojavljajo v razmeroma nepredvidljivem zaporedju, se je zdelo pri strukturiranju XML-sheme smiselno uporabiti elemente s t. i. mešano vsebino (mixed content). Z vidika nadaljnje izrabe slovarske baze, zlasti za postprocesiranje, je to lahko problematično, po drugi strani pa je jasno, da bo nadaljnja izraba baze manjša kot pri drugih slovarskih podatkovnih zbirkah. Trenutno predvidevamo izmenljivost podatkov med omenjeno bazo in bazama za *Novi slovar slovenskega jezika* in *Slovar novejšega besedja slovenskega jezika*, mogoča pa bo tudi uporaba podatkov v drugih etimoloških in jezikovnozgodovinskih projektih.

XML-shema, ki vključuje veliko elementov z mešano vsebino, je posledično manj restriktivna, zato je tudi funkcionalnost leksikografskega programa, podprtega s tovrstno shemo, v smislu zagotavljanja konsistentnosti geselskih sestavkov in njihove skladnosti z XML-shemo in opozarjanja na nepravilnosti v hierarhični strukturi baze ter formalni vsebini elementov nekoliko okrnjena.

3.3. Slovar slovenskega knjižnega jezika 16. stoletja

Izdelava *Slovarja slovenskega knjižnega jezika 16. stoletja*[11] kot prvega zgodovinskega slovarja slovenskega jezika je eden ambicioznejših leksikografskih projektov v slovenskem okolju. Pripravljalna dela zanj potekajo že dlje časa, sicer pa je leta 2001 izšel poskusni snopič slovarja (Merše et al., 2001), leta 2011 pa *Besedje slovenskega knjižnega jezika 16. stoletja* (Ahačič et al., 2011), v katerem je skupaj z nekaterimi podatki zbrano besedje, ki bo v nastajajočem slovarju slovarsko obdelano. Slovar bo namenjen nekoliko ožjemu krogu naslovnikov, zlasti tistim, ki se znanstveno ukvarjajo z raziskovanjem zgodovine slovenskega jezika.

Predpostavljena hierarhična podatkovna struktura XML-sheme slovarja je v veliki meri prilagojena slovarski strukturi, kot je bila zasnovana ob oblikovanju poskusnega snopiča. Čeprav je bila naknadno sprejeta odločitev, da se izkazovanje podatkov glede na poskusni snopič v precejšnji meri poenostavi, pa XML-shemo slovarja kljub vsemu zaznamuje izredna kompleksnost.[12] Ta je v največji meri posledica potrebe po natančnem in sistematičnem prikazu rabe leksikalnih enot, ki je bila zaradi nekodificiranosti jezika – zapisani oz. knjižni jezik se je v 16. stoletju šele vzpostavljal in uveljavljal – raznolika in neustaljena. Prav navedeno dejstvo narekuje vzpostavitev logike strukturiranja podatkov, ki se precej razlikuje od strukturiranja elementov v drugih slovarskih podatkovnih bazah. Če namreč želimo celostno in sistematično predstaviti vso variabilnost, ki jo v besedilih izpričane leksikalne enote izkazujejo na pomenski in slovnični ravni, pri čemer je seveda treba vedenje leksikalne enote zaradi nekodificiranosti knjižnega jezika opisati z več vidikov, kot je to običajno v razlagalnih slovarjih (npr. tudi z vidika (ne)regularnosti stranskih oblik in naglasa, zapisa skupaj – narazen, zapisa velike začetnice, stabilnosti besednovrstne kategorije, zapisa posameznih glasov v besedi, pomenske neustaljenosti, opozarjanja na

[9] Avtor programa je dr. Primož Jakopin.

[10] Omeniti velja, da je slovarska podatkovna baza nastajala v času, ko XML še ni bil vzpostavljen kot standardni format za zapis slovarskih podatkovnih baz, zato je opisano neskladje v strukturiranju podatkov seveda pričakovano in upravičeno.

[11] Kot nekakšno izhodiščno različico zgodovinskega slovarja za slovenščino bi bilo mogoče dojemati tudi podatkovno zbirko *Jezikovni viri starejše slovenščine IMP* (http://nl.ijs.si/imp/).

[12] Morda velja zgolj za ilustracijo opozoriti, da ima XML-shema za slovar besedja 16. stoletja kljub siceršnji težnji po sorodnem strukturiranju podatkov kot pri nastajajočem enojezičnem razlagalnem slovarju (*Novi slovar slovenskega jezika*) in težnji po poenostavitvah, kjer je to mogoče, približno 2,5-krat več shemskih elementov.

nepričakovan zapis ali napačen zapis), je treba v hierarhičnem smislu podatkovne strukture oblikovati precej bolj plosko, kot je običajno sicer. V nasprotnem primeru bi bilo kombinatoričnih možnosti toliko, da bi bila shema v tehničnem smislu za redaktorje praktično neobvladljiva – onemogočena bi bila orientacija znotraj geselskih sestavkov, hkrati bi se povečalo število napak v bazi in podaljšal čas izdelave slovarja. Opisano dejstvo seveda pomembno vpliva na možnosti vzpostavitve iskalnih možnosti, ki so redaktorjem in uporabnikom slovarja na voljo, na možnosti postprocesiranja podatkovne baze ipd. – vse te možnosti so nekoliko okrnjene.

Poseben izziv je predstavljalo oblikovanje shemske strukture za oblikoslovno zaglavje, ki prinaša podatke o besednih oblikah. Pri obravnavah sodobnega jezika to ne bi predstavljalo posebnih zadreg, obravnava oblik iz 16. stoletja, ko knjižni jezik še ni bil standardiziran in tako enoten, kot je danes, pa se je izkazala za posebej zapleteno tako z jezikoslovnega kot tudi s tehničnega vidika. Zaradi izredno raznovrstnih, številnih in za ta slovarski projekt specifičnih podatkovnih tipov je bil glavni poudarek pri izdelavi XML-sheme zlasti na čim večji preglednosti in praktičnosti sheme za sestavljavce slovarja, obenem pa je bilo treba predvideti možnost kasnejše dopolnitve sheme, ki ne bi pretirano dodatno zapletla vnosa podatkov, če bi se izkazalo, da niso bili predvideni vsi potrebni podatkovni tipi. Danes obstoječih 552 mest najnižjega hierarhičnega nivoja za vnos besednih oblik je namreč obvladljivih samo v logično urejeni strukturi. V to število niso vključeni elementi za dodatne podatke o variantnosti in posebnosti besednih oblik in njihovega zapisa.

Spodnja slika prikazuje shematični zapis oblikoslovnega zaglavja pridevnika *bel* v formatu XML, ki ustreza XML-shemi. Struktura je zaradi preglednosti skrajšana, hkrati pa ne vključuje nekaterih specifičnih elementov, ki opozarjajo na dodatne lastnosti jezikovnih elementov.

```
<oblikoslovno_zaglavje>
 <pridevniško>
  <nedoločna_obl>
   <m>
    <ednina>
     <imenovalnik>b | é/e<i>j</i>/e/ee/ee | l</imenovalnik>
     <rodilnik>b | é/e/e<i>j</i> | liga</rodilnik>
<!-- ... -->
    </ednina>
    <dvojina>
     <imenovalnik>bela</imenovalnik>
<!-- ... -->
    </dvojina>
    <množina>
     <imenovalnik>b | e/é/e<i>j</i> | li</imenovalnik>
<!-- ... -->
    </množina>
   </m>
  </nedoločna_obl>
  <določna_obl>
<!-- ... -->
  </določna_obl>
 </pridevniško>
</oblikoslovno_zaglavje>
```

Slika 1: Zgled strukture oblikoslovnega zaglavja *Slovarja knjižnega jezika 16. stoletja*

Omeniti velja še eno specifično lastnost XML-sheme slovarja, ki je posledica neustaljenosti rabe obravnavanih enot in dejstva, da slovar izkazuje le v besedilih izpričano rabo. Gre za dejstvo, da so skoraj vsi elementi XML-sheme neobvezni. V tehničnem smislu to zaradi številnih možnih kombinacij elementov postavlja zahteve pred snovalce delovnega in tudi končnega izpisa slovarskih podatkov v elektronski ali knjižni izdaji slovarja. Oblikovanje besedila, njegova segmentacija, različna pomagala za uporabnike in uporabljeni simboli morajo namreč biti taki, da redaktorju in uporabniku omogočajo neovirano orientacijo tudi znotraj najkompleksnejših geselskih sestavkov.

3.4. Slovar novejšega besedja slovenskega jezika

Slovar novejšega besedja slovenskega jezika je priročnik v obsegu približno 6000 geselskih sestavkov s podgesli, ki z besediščem, ki se je v slovenščini dokumentirano pojavilo po izidu *Slovarja slovenskega knjižnega jezika*, omenjeni priročnik dopolnjuje. Gre za slovar z – glede na tip slovarja, tj. enojezični razlagalni slovar – razmeroma preprosto mikrostrukturo. Slovarsko besedilo je bilo izhodiščno oblikovano v slovarski aplikaciji SlovarRed,[13] vendar je bilo zaradi njene neprilagojenosti predpostavljeni slovarski strukturi[14] iz baze v zaključni fazi redakcije ponovno izvoženo in dopolnjeno v urejevalniku besedil. Z vidika pretvorbe v standardni format XML so ravno v sicer konsistentno podatkovno bazo naknadno dodani elementi povzročali največ težav, saj je pri njih zaradi manjših nedoslednosti pri kodiranju stilov in ločil nastalo največ napak, ki jih je bilo treba odpraviti ročno. Končna redakcija slovarskega besedila v formatu XML poteka v programu iLEX.

Zaradi že omenjene relativne preprostosti slovarske mikrostrukture se je pri oblikovanju XML-sheme slovarja pojavljalo razmeroma malo dilem. Glede na nekoliko manjši obseg priročnika smo ocenili, da je podatkovna baza manj zanimiva za postprocesiranje kot drugi priročniki inštituta, zato smo se odločili nekatere slovarske razdelke, zlasti zaglavje, ki vključuje podatke o izgovorjavi, tonemskosti itd., segmentirati nekoliko manj podrobno, kot je to sicer običajno za razlagalne slovarje večjega obsega. Tudi sicer so že vzpostavljene rešitve in uporabljena programska oprema – slovarski vir je bil v format XML pretvorjen v zaključni fazi redakcije – precej vplivale na oblikovanje XML-sheme. Kljub temu da je to sicer manj običajno, so bile npr. številke pomenov in homonimov, ki so običajno razumljene kot del izpisa slovarskega besedila, v podatkovno bazo vključene kot podatkovni tip, saj bi bilo treba v nasprotnem primeru za zagotovitev pravilnosti podatkov v zaključni fazi redakcije v okviru programa iLEX vzpostaviti tudi sistem sklicevanja. Razdelek, v katerem so navedeni podatki o etimologiji, je strukturiran podobno kot razlagalni razdelek *Slovenskega etimološkega slovarja* in etimološki razdelek *Novega slovarja slovenskega jezika*, da bi med obstoječimi slovarskimi bazami omogočili izmenljivost podatkov.

[13] Avtor programa je Tomaž Seliškar, vsebinsko zasnovo zanj je pripravila Borislava Košmrlj-Levačič (2004).

[14] SlovarRed je bil izdelan kot specializirani program za oblikovanje terminoloških slovarjev, ne pa kot univerzalni leksikografski program, ki bi podpiral uporabo XML-formata in vključitev shem za formalni opis podatkovnih baz.

3.5. Drugi slovarji

V letu 2011 je bila oblikovana tudi XML-shema za nastajajoči *Novi slovar slovenskega jezika*, enojezični razlagalni slovar v obsegu približno 70.000 gesel, ki naj bi bil nekoliko manj ambiciozen naslednik *Slovarja slovenskega knjižnega jezika*. Pri njeni pripravi smo se srečali z izzivom, kako vzpostaviti XML-shemo, ki bo omogočala ohranjanje leksikografske tradicije predhodnika v segmentih, ki so se izkazali za dobre in ki so jih uporabniki vajeni, in sicer tudi na ravni izmenljivosti podatkov med obema bazama, hkrati pa omogočala vzpostavitev novih leksikografskih praks, ki so se kot ustrezne potrdile v praksi sodobne, tudi tujejezične leksikografije, pri čemer naj bi bila shema oblikovana čim bolj striktno in preudarno, tj. tako, da v čim večji meri preprečuje nesistematično interpretiranje podatkov, hkrati pa njihovo predstavitev na uporabniku čim bolj prijazen način v elektronski obliki.

Uporabniška prijaznost in nedvoumnost predstavitve podatkov je tudi eno od temeljih načel *Slovarja pravopisno težavnega besedja slovenskega jezika*, katerega XML-shema je v fazi testiranja, da bi lahko že oblikovano slovarsko gradivo v obsegu približno 15.000 geselskih sestavkov v prihodnjih mesecih pretvorili v standardni format XML. Slovarsko gradivo je bilo izhodiščno oblikovano kot relacijska baza v programu Mravljica,[15] zato večjih težav pri pretvorbi ne pričakujemo.

V elektronski obliki kot relacijske baze s pomočjo aplikacije SlovarRed že dobro desetletje nastajajo tudi inštitutski terminološki slovarji, ki pa zaenkrat še niso pretvorjeni v standardni format XML.

4. Zaključek

Tehnološki napredek na področju računalništva je odločilno vplival tudi na leksikografijo na prehodu v novo tisočletje. Zaradi napredka informacijske tehnologije, ki pogojuje tako metodologijo sodobnega leksikografskega dela kot konceptualizacijo nastajajočih slovarskih priročnikov, obenem pa tudi dojemanje nastajajočih jezikovnih virov pri uporabnikih, je postalo nepogrešljivo, da so slovarski priročniki in drugi sorodni jezikovni viri izhodiščno oblikovani kot strojno berljive razširljive hierarhično strukturirane podatkovne baze, zapisane v standardnem formatu, saj lahko le tako zagotavljamo izmenljivost, povezljivost podatkov in njihovo večkratno izrabo, s čimer pripomoremo k učinkovitejšemu redakcijskemu procesu, ob ustrezni ciklični aktualizaciji jezikovnih podatkov pa tudi k aktualnejšim slovarskim priročnikom. Pri oblikovanju XML-shem za slovarske podatkovne baze je treba upoštevati številne vidike (konceptualizacija slovarske strukture, kompleksni načini iskanja podatkov, praktičnost za vnos podatkov, postprocesiranje podatkovne baze, možnosti vzpostavljanja sklicev, prikaz podatkov v elektronski in tiskani obliki, značilnosti leksikografskega programa …), zato delo poraja številne dileme, kljub temu pa je vzpostavitev strojno berljivih slovarskih podatkovnih baz v standardnem formatu XML razumljiv korak na poti sodobne slovenske leksikografije.

[15] Avtor programa je Uroš Parazajda, podatkovno strukturo zanj pa sta zasnovali dr. Helena Dobrovoljc in dr. Nataša Jakop.

5. Literatura

ABBYY Lingvo Content <http://www.abbyy.com/lingvo_content/>.

Ahačič, K., A. Legan Ravnikar, M. Merše, J. Narat, F. Novak, 2011. *Besedje slovenskega knjižnega jezika 16. stoletja*. Ljubljana: Založba ZRC, ZRC SAZU.

Ahlin, M., B. Lazar, Z. Praznik, J. Snoj, 2003. *Slovar sinonimov slovenskega jezika: splošna določila in opis zgradbe slovarskih sestavkov z vzorčno predstavitvijo*. Ljubljana: ZRC SAZU, Založba ZRC SAZU.

Erjavec, T., 2012. The goo300k corpus of historical Slovene. In *Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul.

Hunter, D., J. Rafter, J. Fawcett, E. van der Vlist, D. Ayers, J. Duckett, A. Watt, L. McKinnon, 2007. *Beginning XML*. Indianapolis: Wiley Publishing.

IDM DPS <http://www.idm.fr/products/dictionary_writing_system_dps/27/>.

iLEX <http://www.emp.dk/>.

Jezikovni viri starejše slovenščine IMP <http://nl.ijs.si/imp/>.

Košmrlj-Levačič, B., T. Seliškar, 2004. Uporabniški računalniški program SlovarRed 2.0. In M. Humar (ur.), *Terminologija v času globalizacije*. Ljubljana: Založba ZRC, ZRC SAZU.

Ledinek, N., A. Perdih, 2012. Uporaba XML-formata v leksikografiji na primeru oblikovanja XML-sheme za Slovar sinonimov slovenskega jezika. *Jezikoslovni zapiski*, 18/1:157-176.

Lexical Interchange Format Standard (LIFT) <http://code.google.com/p/lift-standard/>.

Lexical Markup Framework (LMF), ISO-24613:2008 <http://www.lexicalmarkupframework.org/>.

Merše, M., F. Novak, F. Premk, 2001. *Slovar jezika slovenskih protestantskih piscev 16. stoletja : poskusni snopič*. Ljubljana: Založba ZRC, ZRC SAZU.

Smrž, P., 2001. Slovníková data ve formátu XML. In A. Jarošová (ur.), *Slovenčina a čeština v počítačovom spracovaní. Zborník referátov zo seminára. Bratislava 26. – 27. októbra 2001*. Bratislava: Veda.

Snoj, M., 2003. *Slovenski etimološki slovar*. Ljubljana: Modrijan.

Standard XML <http://www.w3.org/standards/xml/>.

TEI Consortium, 2007. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. <www.tei-c.org/release/doc/tei-p5-doc/en/html/>.

Termania <http://www.termania.net>.

Thompson, H. S., D. Beech, M. Maloney, N. Mendelsohn, 2004. *XML Schema Part 1: Structures: W3C Recommendation 28 October 2004*. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>.

TshwaneLex <http://tshwanedje.com/tshwanelex/>.

w3schools.com <http://www.w3schools.com/>.

Building Named Entity Recognition Models for Croatian and Slovene

Nikola Ljubešić, Marija Stupar, Tereza Jurić

Department of Information and Communication Sciences
Faculty of Humanities and Social Sciences
University of Zagreb
Ivana Lučića 3
10000 Zagreb
Croatia
{nljubesi,mstupar,tjuric2}@ffzg.hr

Abstract

The paper presents efforts in developing freely available models for named entity recognition and classification for Croatian and Slovene. Our experiments focus on the most informative set of linguistic features taking into account the availability of language tools for the languages in question. Beside the classic linguistic features, distributional similarity features calculated from large unannotated monolingual corpora are exploited as well. Using distributional information improves the results for 7-8 points in F1 while adding morphological information improves the results for additional 3-4 points in both languages. The best performing models, along with test sets for comparison with future and existing systems and a HunPos part-of-speech model for Croatian are available for download for academic usage.

Izgradnja modelov za prepoznavanje imenskih entitet za hrvaščino in slovenščino

Prispevek predstavi razvoj prostodostopnih modelov za prepoznavanje in klasifikacijo imenskih enot za hrvaški in slovenski jezik. Poskusi se osredotočajo na najbolj informativne jezikoslovne lastnosti, pri čemer upoštevajo dostopnost jezikovnih orodij za jezika. Poleg standardnih jezikoslovnih lastnosti so upoštevane tudi distribucijske lastnosti, ki so bile izračunane iz velikih neoznačenih enojezičnih korpusov. Uporaba distribucijskih lastnosti poboljša rezultate za 7-8 točk v meri F1, uporaba oblikoslovnih informacij pa dodatno za 3-4 točke, in to pri obeh jezikih. Najboljši naučeni model, skupaj s testno množico za primerjavo z obstoječimi in bodočimi sistemi, ter model za oblikoslovno označevanje hrvaščine s programom HunPos so dostopni za prenos za uporabo v znanstvene namene.

1. Introduction

Named entity recognition and classification (NERC), nowadays often called just named entity recognition (NER) is a subtask of the information extraction task. It aims to locate and classify text elements into predefined categories, and is regularly applied in many fields, using statistical or rule-based models. State-of-the-art systems tend to be open domain and language independent.

In this paper we present the process of creating NER models for Croatian and Slovene that we publish for free academic use.

The tool we use to build the models is the Stanford Named Entity Recognizer, nowadays a frequently used tool for NER. It is an implementation of Conditional Random Fields sequence models and is available under GNU GPL licence and free for academic use. (Finkel et al., 2005)

Beside many feature extractors that come with this tool, it is designed to work with the clustering method proposed by (Clark, 2003) which combines standard distributional similarity with morphological similarity to cover infrequent words for which distributional information alone is unreliable.

This paper is structured as follows: in Section 2 we give an overview of related work, in Section 3 we present the datasets used in our research, in Section 4 we give an overview of our experimental setup and in Section 5 we present the results of the experiments.

2. Related work

To our knowledge, there has been some effort in developing NER systems for south Slavic languages mainly in the direction of building rule-based systems.

A rule-based system for Croatian described in (Bekavac, 2005) uses regular grammars for recognition and classification of names over annotated texts. The system contains the module for sentence segmentation, lexicon of common words, specialized lists of names and transducers for automatic recognition of certain word forms.

A statistical approach described in the diploma thesis (Bošnjak, 2007) uses a semi-supervised method based on lists of names and entity extraction system.

For Serbian a rule-based system (Vitas and Pavlović-Lažetić, 2008) shows that there is a great difference between English and Serbian language, as well as all the other Slavic languages which require a more thorough preparation of the system because of the rich inflectional system.

None of the presented systems are available for academic usage which hinders researchers in looking into higher tasks that require NER as a preprocessing step. One of the main intentions of this paper is to improve this situation.

In the process of building a good NER system, features are considered as important as the selection of algorithm for machine learning. The aim is to find an optimal set of features that will ensure the highest system accuracy with minimum complexity in classifier building. Several NER approaches use a very large number of features (Mayfield et al., 2003), but the inclusion of additional features after a

certain point can yield worse results.

In the students' research paper that precedes this research (Filipić et al., 2012) we have identified properties for the Stanford NER system defining feature extractors that seem to work best for Croatian language. In this paper we use these property files and only vary in training and test sets and the usage of distributional information [1].

The only work we are aware of that examines the usage of distributional features in Stanford NER is (Faruqui and Padó, 2010). The paper describes the process of building and optimizing NER models for German and by using distributional features F1 is improved for 6% in-domain and 9% out-of-domain. Our research is considerably inspired by this paper.

3. Corpora

We have built and annotated two corpora, one Croatian and one Slovene. Both copora are built from data taken from specific Internet domains from the Croatian and Slovene web corpora hrWaC and slWaC (Ljubešić and Erjavec, 2011).

The Croatian corpus contains 59,212 tokens taken from four different Internet domains covering two general newspaper portals, nacional.hr and jutarnji.hr, one ICT portal bug.hr and the business news portal poslovni.hr. These data were annotated during a students' project where diversity of data was one of the main points.

The Slovene corpus is almost two thirds the size of the Croatian one containing 37,032 tokens and data from just one general news portal rtvslo.si. While selecting these data the main goal was to build a usable training set with limited annotation capacities.

Beside admitting that these corpora were built opportunistically regarding temporary goals, we want to emphasize that having two corpora of different diversity and size gives us an interesting starting point for our experiments.

The amount of data in both copora is given in Table 1.

corpus	document #	token #
hr	105	59,212
bug.hr	19	9,609
jutarnji.hr	16	9,760
nacional.hr	24	20,583
poslovni.hr	46	19,260
sl	69	37,032
rtvslo.si	69	37,032

Table 1: Size of the corpora used

The corpora were tagged by the IOB2 standard following the CoNLL-2003 annotation guidelines [2] where each row represents a token in the text with its linguistic annotation and designated predefined named entity category. IOB2 labels show whether a word is at the beginning (B), inside (I) or outside (O) of a named entity. Four categories were annotated - location (LOC), organization (ORG), person (PERS) and miscellaneous (MISC).

corpus	NE	LOC	MISC	ORG	PERS
hr	2,647	591	632	818	606
sl	2,491	716	378	311	1,086

Table 2: Number of annotated named entities in both corpora

Since for Slovene there are freely available taggers and this is not the case for Croatian, we manually annotated basic part-of-speech (first letter of the Multext-East MSD) on Croatian data as well since related work shows that these features are useful for the task. Slovene data was MSD tagged and lemmatized with the freely available ToTaLe tagger (Erjavec et al., 2005) trained on JOS corpus data (Erjavec et al., 2010).

To be able to use POS information on unseen Croatian data, we trained a simple model for the HunPos tagger (Halácsy et al., 2007) from the Croatian dataset. We performed a simple test of the resulting model by dividing the Croatian dataset into a training and a test set by the ratio of 9:1. Accuracy obtained on the test set was 95.1%. We publish the tagger trained on all available data along with the NER models and the benchmark datasets. To our knowledge, this is the first freely available part-of-speech tagger for Croatian. We are fully aware of more informative MSD taggers being developed years ago and hope that our approach of publishing most of the available results will speed up the process of other researchers releasing existing data and tools under a permissive license.

The amount of annotated named entities in the Croatian and Slovene corpus is given in Table 2. The expected difference in diversity of the data can be clearly observed from these numbers. First of all, although the Slovene corpus has 37% less textual material, it has just 6% less named entities showing a higher density of named entities one would expect from a straightforward newspaper dataset. Furthermore, when we look at the type of named entities, we can observe that the Slovene dataset contains much more person names and slightly more locations while the Croatian dataset contains more organization names and named entities labeled with the miscellaneous category. These data confirm our assumption that the Croatian dataset is much more diverse and will thereby present a harder task for supervised classification.

A final insight in the features and thereby specificities of the two datasets is given by calculating vocabulary transfer between identical portions of development and test sets. The numbers are given in Table 3. The vocabulary transfer is calculated as the token and type percentage of named entities in the test set being already present in the development set.

Two interesting properties can be observed here. First of all, the Slovene vocabulary transfer is higher than the Croatian one pointing at the expected lower content diversity of Slovene data. Secondly, there is almost no difference

[1] The example property file used in this paper can be retrieved from `http://www.nljubesic.net/upload/ner/ner.prop`

[2] See `http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt`

corpus	token transfer	type transfer
hr	10.7%	10.6%
sl	17.3%	12.4%

Table 3: Vocabulary transfer for both corpora on identical portions of development and test set

between token and type transfer on Croatian data showing that the diversity of named entities is really high since almost none of the named entities from the development set present in the test set appears more than once in the Croatian test set which is not the case on Slovene data.

We divided both corpora into development and test sets by shuffling documents and producing test sets of similar size for both languages. The decision to build test sets of similar size was guided by the idea of publishing those test sets as benchmark datasets for both languages. For that reason the Croatian development set contains 53,142 tokens while the Slovene one contain 29,686 tokens, i.e. 56% of the amount of Croatian data.

For calculating distributional similarity of tokens from large monolingual corpora portions of hrWaC and slWaC web corpora were used. For Croatian we built a 100Mw corpus and for Slovene a 50Mw corpus, both containing data from large news portals.

4. Experimental setup

Since different annotations on Croatian and Slovene data were available, we evaluated different settings for each language. Beside part-of-speech information for both languages, on Slovene data MSD and lemma information was present as well.

On Croatian data we experimented with POS information ("POS"), distributional information ("DISTSIM") calculated from 10Mw, 50Mw and 100Mw corpora while on Slovene data we experimented with POS, MSD ("MSD") and lemma ("LEMMA") information and distributional information obtained from 10Mw and 50Mw copora. Thereby we performed 8 experiments on Croatian data and 11 experiments on Slovene data (we eliminated the experiments varying with availability of lemma information once it proved to be non-informative).

All the experiments were performed on development sets of both datasets via 5-fold-cross-validation that takes into account document borders. By respecting document borders we were trying to keep the vocabulary transfer as low as possible and thereby obtain the most realistic results, i.e. differences between different experimental settings.

Distributional similarity was calculated by using Clark's `cluster_neyessen` tool (Clark, 2003) with default settings (numberStates=5, frequencyCutoff=5, iterations=10). The number of resulting clusters was set on best-performing values in (Faruqui and Padó, 2010), i.e. for 10Mw corpora 100 clusters and for 50Mw and 100Mw corpora 400 clusters were built. First twenty elements of example clusters calculated from the Croatian 100Mw and Slovene 50Mw corpora are given in Table 4. The Croatian cluster contains exclusively country and city names in the locative (or dative) case. The Slovene cluster contains person first names in the nominative case of both Slovene and English origin. These examples show very clearly how the cluster ID can be used as a very informative feature in the supervised training procedure.

njemačkoj rijeci londonu sarajevu osijeku italiji zadru francuskoj haagu austriji parizu dubrovniku vukovaru španjolskoj milanu bruxellesu rimu beču moskvi berlinu
tomaž simon goran martina dejan jan nina tom saša mojca vesna jurij eva nataša maria jernej daniel richard thomas damjan žiga

Table 4: First 20 elements of sample clusters obtained with Clark's tool on the 100Mw Croatian and 50Mw Slovene corpora

After identifying best performing settings on development sets we calculate our final results by training a system on the whole development set and testing it on the left-out test set.

Finally we calculate learning curves for the best performing settings to identify the gain we can expect from annotating more data.

5. Results

The results obtained by 5-fold cross-validation on both development sets are presented for Croatian in Figure 1 and for Slovene in Figure 2. The results of each cross-validation are averaged by calculating the harmonic mean. Regarding the statistical significance of the results, we perform a one-tailed paired t-test over pairs of results we find interesting.

On Croatian results we can observe already in the second experiment that basic morphological information in this simple setting improves F1 for 4.5% ($p = 0,002$). Our third experiment shows that using distributional information obtained from a 10 million token corpus improves the result as much as the part-of-speech information with similar significance ($p = 0.005$). By combining both of these two features we improve our results for 8.5%, highly significantly in comparison to using only one feature ($p < 0.001$). By calculating distributional information on five and ten times more data we get improvements of 2% and 3% when not using part-of-speech information and 1% and 2% improvements when using part-of-speech information. The differences between neighbouring corpus sizes (10 and 50; 50 and 100) are not statistically significant, but the differences between using 10Mw and 100Mw corpora are ($p = 0.007$). We see a steady rise in performance as the unlabeled monolingual corpus size increases motivating us to perform similar calculations on much larger datasets in the future.

The results on Slovene data in the categories present in Croatian data are rather similar backing them up. There are two types of information on Slovene data we did not have for Croatian - MSD and lemma. By using MSD and not only POS information the results do improve for additional 1%, but statistically insignificant ($p = 0.21$). On the contrary, by adding lemma information to the MSD decreases

the result significantly for 5.5% ($p = 0.007$). One could expect such an outcome since lemmatization performs worst on named entities. By adding more distributional information by moving from a 10Mw to a 50Mw corpus we get an improvement even steeper than on Croatian data by getting a 5% improvement, now highly significant ($p < 0.001$). This could be explained by the higher simplicity of this dataset and yield a conclusion that for data from narrower domains additional data sources such as this one give more improvement. We can observe on both datasets that, when using distributional similarity from larger corpora, including additional features like POS or MSD makes the increase in the results lower.

When comparing results on Croatian and Slovene datasets one observes right away that the results on Slovene data are much better although the size of the dataset is under half the size. This can be traced back to the fact that the Slovene dataset has a narrower domain, a higher vocabulary transfer and a higher amount of named entities like person and location which are considered easier to recognize and classify. On the other hand the resulting Croatian module is expected to be more robust and should perform better on different domains.

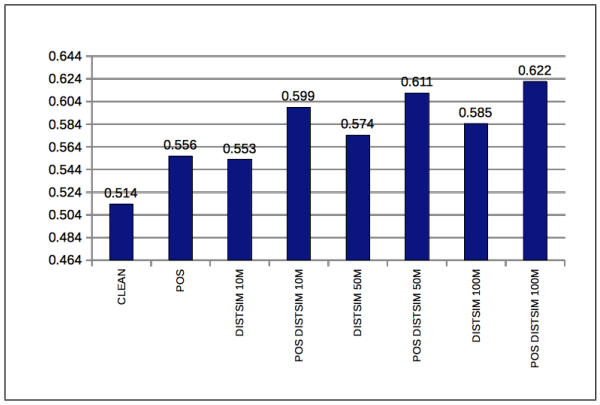


Figure 1: F1 results obtained via 5-fold cross-validation on Croatian development set

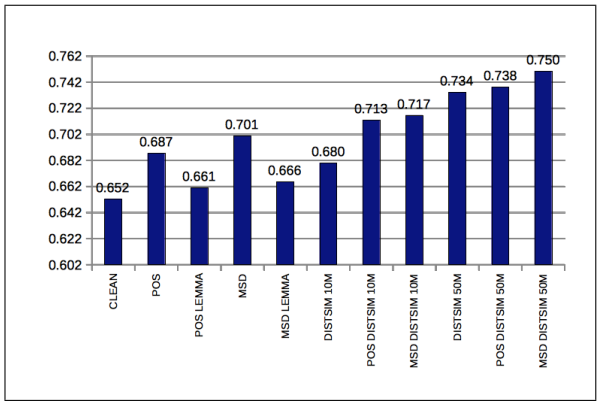


Figure 2: F1 results obtained via 5-fold cross-validation on Slovene development set

We chose two settings per dataset for final testing on the left-out test set. The first one uses distributional information, but leaves out the need for morphological annotation of the data while the second one uses both distributional and morphological information. We present the results of precision, recall, F1, true positives and false positives and negatives by category in Table 5. We consider such an exhaustive data presentation informative since this is the best approximation of the capability of the models we publish alongside this paper.

The number of false negatives shows to be on both datasets and settings higher than the number of false positives with higher percentage than recall as a direct consequence. On Slovene data the best performing categories in reverse order are PERS, LOC, ORG and MISC. On Croatian data LOC tends to perform best, ORG and PERS being a tie and MISC being traditionally the worst category. The somewhat unexpected order of category performance can probably be followed to the wider domain of the Croatian dataset.

hr DISTSIM 100Mw						
Entity	P	R	F1	TP	FP	FN
LOC	0.8049	0.7021	0.7500	33	8	14
MISC	0.7436	0.3867	0.5088	29	10	46
ORG	0.6742	0.6250	0.6486	60	29	36
PERS	0.9032	0.5185	0.6588	28	3	26
Totals	0.7500	0.5515	0.6356	150	50	122
hr POS DISTSIM 100Mw						
Entity	P	R	F1	TP	FP	FN
LOC	0,8293	0,7234	0,7727	34	7	13
MISC	0,7778	0,4667	0,5833	35	10	40
ORG	0,6989	0,6771	0,6878	65	28	31
PERS	0,8500	0,6296	0,7234	34	6	20
Totals	0,7671	0,6176	0,6843	168	51	104
sl DISTSIM 50Mw						
Entity	P	R	F1	TP	FP	FN
LOC	0,7423	0,7273	0,7347	72	25	27
MISC	0,5000	0,2143	0,3000	15	15	55
ORG	0,8947	0,3617	0,5152	17	2	30
PERS	0,8966	0,8509	0,8731	234	27	41
Totals	0,8305	0,6884	0,7528	338	69	153
sl MSD DISTSIM 50Mw						
Entity	P	R	F1	TP	FP	FN
LOC	0,7957	0,7475	0,7708	74	19	25
MISC	0,4688	0,2419	0,3191	15	17	47
ORG	0,8947	0,3617	0,5152	17	2	30
PERS	0,8619	0,8400	0,8508	231	37	44
Totals	0,8180	0,6977	0,7531	337	75	146

Table 5: Test results on the four best performing models (P - precision, R - recall, F1 - F1 measure, TP - true positives, FP - false positives, FN - false negatives)

With the final set of experiments we wanted to examine the learning curves of the best performing approaches to see how much we could benefit in the future by just annotating more data.

The four learning curves were calculated using distributional information and varying the usage of available

morphological information, for Slovene the MSD, and for Croatian part-of-speech information. The curves are calculated by enlarging the training data in ten steps by shuffling the development set data and testing on the test set. The experiment for each training set size was repeated four times to obtain a better estimate of the curve shape. The learning curves are depicted in Figure 3.

The Slovene curve rises much steeper than the Croatian one which is in accordance to all other information pointing to the fact that the Slovene dataset is much easier than the Croatian one. Both learning curves have finished the steepest phase, but are still climbing which shows that the process could further benefit from larger amounts of labeled data. While building the Croatian dataset we actually calculated learning curves during the annotation process to assess if annotating larger amounts of data would prove to be very beneficial. For Slovene data we did an educated guess based on our insights on Croatian data and the fact that this dataset covers a narrower domain.

From these curves no conclusions about the informativeness of morphological information should be drawn like in case of Slovene data where the results not using morphological information seem better than those that use that information. The learning curves are produced by testing the built models on just one dataset while previous results given in Figures 1 and 2 are obtained via cross-validation by evaluating five models built on different data on five different evaluation sets.

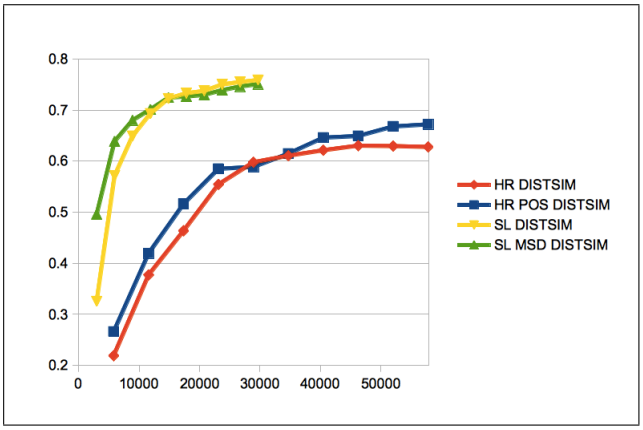


Figure 3: Learning curves calculated on portions of development sets and test sets by using distributional similarity and varying in usage of available morphological information (x axis represents token count, y axis the F1 measure)

6. Conclusion

In this paper we have presented the process of building freely available models for named entity recognition and classification for Croatian and Slovene. We have built two datasets, one for Croatian which is larger and covers a broader domain and one for Slovene which is smaller but covers just the general news domain.

We were searching for the optimal set of features on the development set via five-fold cross-validation. Lemmata have shown to be of no use for a morphologically complex language such as Slovene since lemmatization tends to work worst on word classes such as named entities. On the other hand morphological information such as POS tags or full MSD tags proved to be valuable with the latter being more informative. That type of information improved the F1 measure in a 3-5% window. Clustering tokens from a large monolingual corpus by features such as contextual and morphological properties has proven to be beneficial improving the results by using 10Mw corpora for 3-4%. With clustering results from larger corpora the results continue to improve steadily. Combining both morphological and clustering information proved to be the winning combination with an overall improvement of 10% on datasets of both languages. By omitting morphological information for which some preprocessing is required we still get an improvement of 8%.

We are releasing four best performing models free for academic purpose, two for each language - one that uses morphological annotation, and one that does not require such information. Additionally, we release the two test sets as potential benchmarks for future work on named entity recognition and classification for these two languages. The models and datasets can be found on

- `http://www.nljubesic.net/resources/data/croatian-ner/` for Croatian and
- `http://www.nljubesic.net/resources/data/slovene-ner/` for Slovene.

The HunPos part-of-speech model for Croatian can be obtained from

- `http://www.nljubesic.net/resources/data/croatian-pos-tagger/`.

For the future our plan is to increase the amount of annotated data for training by exploiting semi-supervised approaches. Additionally we plan to calculate distributional similarity on larger corpora and take under consideration variations of the method used in this paper.

Acknowledgement

Research reported in this paper has been supported by the CESAR project within the EU 7th Framework Programme and the ICT Policy Support Programme, grant agreement no. 271022.

7. References

Bekavac, Boško, 2005. *Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima*. Ph.D. thesis, University of Zagreb.

Bošnjak, Matko, 2007. *Strojno prepoznavanje naziva tehnikama strojnog učenja*. Master's thesis, University of Zagreb.

Clark, Alexander, 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary.

Erjavec, Tomaž, Darja Fišer, Simon Krek, and Nina Ledinek, 2010. The JOS Linguistically Tagged Corpus of Slovene. In *International Conference on Language Resources and Evaluation*.

Erjavec, Tomaž, Camelia Ignat, Bruno Poliquen, and Ralf Steinberger, 2005. Massive multilingual corpus compilation: Acquis communautaire and totale. In *The 2nd Language & Technology Conference - Human Language Technologies as a Challenge for Computer Science and Linguistics*. Association for Computing Machinery (ACM) and UAM Fundacja.

Faruqui, Manaal and Sebastian Padó, 2010. Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of the 10th Conference on Natural language processing (KONVENS) 2010*. Saarbrücken, Germany.

Filipić, Lobel, Tereza Jurić, and Marija Stupar, 2012. Strojno prepoznavanje naziva u tekstovima pisanima hrvatskim jezikom. Students' paper awarded with the Rector's award.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning, 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*.

Halácsy, Péter, András Kornai, and Csaba Oravecz, 2007. HunPos: an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics.

Ljubešić, Nikola and Tomaž Erjavec, 2011. hrWaC and slWac: Compiling Web Corpora for Croatian and Slovene. In Ivan Habernal and Václav Matousek (eds.), *Text, Speech and Dialogue - 14th International Conference, TSD 2011, Pilsen, Czech Republic, September 1-5, 2011. Proceedings*, volume 6836 of *Lecture Notes in Computer Science*. Springer.

Mayfield, James, Paul McNamee, and Christine Piatko, 2003. Named entity recognition using hundreds of thousands of features. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03. Stroudsburg, PA, USA: Association for Computational Linguistics.

Vitas, Duško and Gordana Pavlović-Lažetić, 2008. Resources and methods for named entity recognition in Serbian. *INFOTHECA : Journal of Informatics and Librarianship*, 9(1-2).

Luščenje terminoloških kandidatov za slovar odnosov z javnostmi

Nataša Logar Berginc*, Špela Vintar, Špela Arhar Holdt*****

* Univerza v Ljubljani, Fakulteta za družbene vede
Kardeljeva ploščad 5, SI-1000 Ljubljana
natasa.logar@fdv.uni-lj.si
** Univerza v Ljubljani, Filozofska fakulteta
Aškerčeva 2, SI-1000 Ljubljana
spela.vintar@ff.uni-lj.si
*** Trojina, zavod za uporabno slovenistiko
Partizanska cesta 5, SI-4220 Škofja Loka
spela.arhar@trojina.si

Povzetek

V prispevku prikazujemo analizo luščenja eno- in večbesednih terminoloških kandidatov, ki smo ga izvedli za potrebe priprave slovarja odnosov z javnostmi na podlagi korpusa KoRP z luščilnikom LUIZ. Podrobneje se posvečamo dvojemu: (a) izluščenim enobesednim samostalniškim terminološkim kandidatom, katerih seznam primerjamo s pogostostnim seznamom samostalnikov v KoRP in vrednotimo glede na terminološkost, kot sta jo prepoznala dva področna strokovnjaka, ter (b) izluščenim večbesednim nizom z glagolskim jedrom. Nadgrajeno metodo luščenja in izboljšan prikaz rezultatov smo dopolnili še z analizo priklica. Potrdili oz. ugotovili smo, da je v primerjavi s pogostostnim seznamom terminološki potencial enot v zgornjem delu liste izluščenih samostalnikov večji in da imajo izluščeni glagolski besedni nizi predvsem kolokacijsko vrednost, ne pa tudi terminološke. Analiza priklica je pokazala predvsem nizko stopnjo strinjanja med obema področnima strokovnjakoma, sicer pa je bil priklic razmeroma visok.

Term candidate extraction for the dictionary of public relations

The article describes an analysis of automatic term recognition results performed for single- and multi-word terms with the LUIZ term extraction system. The target application of the results is a dictionary of Public Relations and the main resource the KoRP Public Relations Corpus. Our analysis is focused on two segments: (a) single-word noun term candidates, which we compare with the frequency list of nouns from KoRP and evaluate termhood on the basis of the judgements of two domain experts, and (b) multi-word term candidates with verb as headword. In order to better assess the performance of the system and the soundness of our approach we also performed an analysis of recall. Our results show that the terminological relevance of extracted nouns is indeed higher than that of merely frequent nouns, and that verbal phrases only rarely count as proper terms. The analysis of recall shows low inter-annotator agreement, but nevertheless very satisfactory recall levels.

1. Uvod

Korpusno jezikoslovje je z možnostjo meritev zelo različnih lastnosti jezika prineslo opazen premik zlasti v leksikografiji – pa tudi terminografiji (npr. Atkins, Rundell, 2008; Bergenholtz, Tarp, 1995; Biber, Conrad, Reppen, 1998; Čermák, 2011; Halliday et al., 2004; Hanks, 2008; Leech, 1992; McEnery, Wilson, 1996; Pearson, 1998; Schryver, 2003; Sinclair, 2004; Teubert, Krishnamurty, 2007; pri nas npr.: Krek, 2003; Gorjanc, 2005; Gorjanc, Krek, 2005; Gantar, 2007; Gantar, 2009; Logar, 2007; Vintar, 2008). Na osnovi korpusa, ki ima razviden namen, merila gradnje, odločitve snovalcev ter besedilnozvrstno, časovno in drugo zgradbo, lahko področni strokovnjaki in jezikoslovci, ki sodelujejo pri pripravi terminoloških slovarjev ali terminoloških podatkovnih zbirk, svojo interpretacijo, ki bo posledično podana kot jezikovni opis in verjetno tudi predpis, oprejo na podatke, pridobljene iz resnične jezikovne rabe; z umikanjem nuje po introspekciji kot edini možnosti presoje pa se povečata kredibilnost in veljavnost prepoznavanja vseh pojavov – tudi jezikovnih. Pri takem pristopu je ena ključnih prednosti, ki kvalitativno in kvantitativno presega nekorpusno terminografijo, računalniško podprto luščenje terminoloških kandidatov.

Pristopov k luščenju terminoloških kandidatov je več, pri skoraj vseh pa gre za kombinacijo jezikoslovnega znanja o naravi terminov ter izrabo matematičnih lastnosti porazdelitve besed in besednih nizov v korpusih (Vintar, 2008: 100; Vintar, 2009: 346–347 in tam navedena literatura). Za slovenščino je bilo luščenje terminoloških kandidatov preizkušeno že večkrat, npr. za projekt Voice Tran I in II na področju vojaške terminologije ter na področju računalništva in informatike za Islovar (Vintar, Erjavec, 2008; Vintar, 2009; Vintar, 2010: 47); pa tudi pri pripravi slovarja odnosov z javnostmi. Za slednjega smo že predstavili (Logar, Vintar, 2008), da je bilo uspešno zlasti pri pridobivanju večbesednih terminoloških kandidatov, ki je potekalo po kombinaciji statistične in jezikoslovne metode. Na osnovi takrat prepoznanih pomanjkljivosti smo metodo in prikaz rezultatov izboljšali – v nadaljevanju na delu pridobljenih seznamov prikazujemo, na kakšen način in s kakšno uspešnostjo.

2. Eksperiment

Korpus, iz katerega smo luščili terminološke kandidate odnosov z javnostmi, je korpus KoRP. KoRP vsebuje 1,824.699 pojavnic, je enojezični, sinhroni in trenutno statični korpus strokovnih besedil. Od julija 2007 je prosto dostopen na <http://www.korp.fdv.uni-lj.si/>, iskanje po njem pa trenutno poteka še s Konkordančnikom ASP32. Pred tokratnim luščenjem smo ga ponovno označili z najnovejšo različico označevalnika podjetja Amebis, d. o. o. (Romih, Holozan, 2002; Holozan, 2006; Arhar Holdt, 2011: 22–23, 28–29).

S pomočjo luščilnika LUIZ (Vintar, 2010) smo želeli pridobiti dvoje:

a) enobesedne terminološke kandidate: samostalnike, glagole, pridevnike in prislove;

b) večbesedne terminološke kandidate: samostalniške in glagolske besedne zveze.

Tako eno- kot večbesedne kandidate smo luščili s pomočjo oblikoskladenjskih vzorcev (ti so pri enobesednih enotah sestavljeni le iz enega člena) in terminološke uteži, ki se izračuna na podlagi pogostosti pojavitve v specializiranem korpusu v primerjavi s splošnim korpusom ter frazeološke stabilnosti enote. Skupno smo uporabili 39 oblikoskladenjskih vzorcev, od tega 30 s samostalniškim jedrom, 9 z glagolskim (Tabela 1, prvi stolpec). Za tako obsežno število vzorcev smo se odločili zaradi večje zanesljivosti ocene tovrstnega pristopa k pridobivanju terminov v slovenščini, smo pa že pred začetkom luščenja predvidevali, da bo približno polovica vzorcev dala le malo ali celo nič terminološko zanimivega gradiva.

Ker se med izluščenimi kandidati pogosto znajdejo tudi lastna imena, ki pa za slovarske namene v veliki meri niso zanimiva, smo vse enote, ki so vsebovale besede z veliko začetnico, izločili.

3. Analiza rezultatov

Rezultat luščenja so bili seznami, na katerih je bilo skupno 47.007 večbesednih enot (brez lastnih imen; število po vzorcih in primeri so v Tabeli 1) oz. 16.190 enobesednih enot (brez lastnih imen; Tabela 2).

Vzorec	Število kandidatov	Primer
Samostalniške zveze		
1. P S	17.242	*lokalna skupnost*
2. S S	9.362	*vir informacij*
3. S S S	932	*merilo uspešnosti delovanja*
4. P S S	1.670	*uradni vir informacij*
5. S P S	3.160	*dan odprtih vrat*
6. S D S	4.370	*sporočilo za javnost*
7. S D S S	648	*orodje za doseganje ciljev*
8. S D P S	1.174	*odnos z interno javnostjo*
9. P P S	1.398	*celostna grafična podoba*
10. P D S	621	*vodilni v podjetju*
11. R P S	618	*srednje veliko podjetje*
12. S S S S	53	*dvig kakovosti življenja otrok*
13. S P S S	198	*doseganje poslovnih ciljev organizacije*
14. P S S S	130	*osrednje zanimanje svetovne javnosti*
15. P S P S	381	*refleksivni model komunikacijskega menedžmenta*
16. S S P S	321	*model načrtovanja merljivih ciljev*
17. S P P S	245	*model dvosmernih simetričnih odnosov*
18. P P S S	51	*upravljano komunikacijsko ravnanje organizacije*
19. S S D S	982	*strategija odnosov z javnostmi*
20. P S D S	971	*tržni odnosi z javnostmi*
21. R P S S	51	*vnaprej pripravljen predlog vprašanj*
22. R P D S	63	*tesno povezan s teorijo*
23. P D P S	141	*značilen za blagovno znamko*
24. P D S S	100	*potreben za razrešitev konflikta*
25. R P P S	51	*točno določena ciljna javnost*
26. P S in S	445	*medijski čas in prostor*
27. S S in S	365	*trženje izdelkov in storitev*
28. S in S S	434	*mnenje in stališče javnosti*
29. S in P S	367	*čast in dobro ime*
30. P in P S	463	*tiskani in elektronski mediji*
Glagolske zveze		
31. R G	3.032	*pomembno vplivati*
32. G D S	2.208	*odgovoriti na vprašanje*
33. G R	1.602	*delovati neodvisno*
34. D S G	878	*v nadaljevanju predstavljati*
35. G D R	101	*veljati za učinkovito*
36. G kot S	98	*delovati kot posrednik*
37. G kot P	42	*biti kot nov*
38. D R G	14	*od nekdaj spremljati*
39. G kot R	13	*prepoznati kot pomembno*
SKUPAJ	**47.007**	

Tabela 1: Izluščeni večbesedni terminološki kandidati: število po vzorcih in primeri.

Besedna vrsta	Število kandidatov	Primer
1. S	7.379 (z lastnimi imeni: 10.731)	*javnost, odnos, organizacija*
2. P	4.854	*komunikacijski, blagovni, zaposleni*
3. R	1.379	*veliko, pogosto, vedno*
4. G	2.578	*vplivati, sporočiti, komunicirati*
SKUPAJ	**16.190**	

Tabela 2: Izluščeni enobesedni terminološki kandidati: število po besednih vrstah in primeri.

Sezname smo podrobneje pregledali in tako prišli do ocene relevantnosti metode luščenja, dodatno pa izvedli še analizo priklica.

3.1. Enobesedni terminološki kandidati

Predstavili bomo le analizo vrhnjega dela seznama samostalniških terminoloških kandidatov, ki smo ga ocenjevali z dveh vidikov: (a) v primerjavi s pogostostnim seznamom samostalnikov iz KoRP in (b) glede na oceno terminološkosti, ki sta jo dala dva strokovnjaka s področja odnosov z javnostmi.

a) Že ob primerjalnem ogledu zgolj prvih 20 izluščenih samostalnikov in prvih 20 samostalnikov po pogostosti (Tabela 3) opazimo nekaj razlik: pri luščenju so med prvimi dvajsetimi poimenovanja *management*, *deležnik*, *model* in *novinar* (v Tabeli 3 krepki tisk), ki jih na pogostostnem seznamu med prvimi dvajsetimi ni; nasprotno pa so na pogostostnem seznamu samostalnikov višje, tj. do dvajsetega mesta, uvrščeni *človek*, *skupnost* in *stran* (ležeči tisk).

Terminološko luščenje: prvih 20 samostalnikov	**Pogostost: prvih 20 samostalnikov**
javnost	javnost
odnos	odnos
organizacija	organizacija
komuniciranje	podjetje
podjetje	komuniciranje
medij	medij
znamka	znamka
informacija	leto
leto	informacija
cilj	cilj
okolje	primer
management	skupina
področje	okolje
primer	program
skupina	področje
program	*človek*
deležnik	(zaposleni)*
vloga	vloga
model	*skupnost*
novinar	*stran*

Tabela 3: Vrhnji del seznama izluščenih samostalnikov in samostalnikov po pogostosti v KoRP.

* Pri luščenju med prvimi dvajsetimi, vendar pri pridevnikih.

Pregled razlik, ki so se pokazale do stotega mesta tabele, pokaže, da so pri uporabljenem luščenju višje uvrščeni (torej med prvimi stotimi) *dejavnik*, *izvajanje*, *javnost*, *komunikator*, *manager*, *menedžment*, *tveganje*, *uspešnost*, *vedenje* in *zaupanje*; medtem ko so pri pogostostnem seznamu višje uvrščene besede *kot*, *mesto*, *načrt*, *Publica*, *služba*, *str.*, *svet* in *vlada*. Tudi če pogledamo spremembe mest terminoloških kandidatov znotraj stotega mesta, so premiki zanimivi: *management*, ki je pri terminološkem luščenju na 12. mestu, je po pogostosti na 60. mestu, in tako npr. še *deležnik* – 17. (41.), *oglaševanje* – 29. (44.), *manager* – 31. (130.), *marketing* – 36. (59.), *praktik* – 40. (87.), *načrtovanje* – 53. (81.), *komunikator* – 65. (198.), *uspešnost* – 85. (111.), *tveganje* – 97. (122.). Nazadovali pa so pri luščenju samostalniki *človek* – 29. mesto, po pogostosti pa 17. mesto, *stran* – 33. (20.), *sistem* – 94. (64.), *podatek* – 96. (71.) ipd.

b) Strokovnjaka odnosov z javnostmi sta za potrebe priprave terminološkega slovarja pregledala celotni pogostostni seznam samostalnikov, ki je obsegal skoraj 12.500 enot.[1] Če kot načeloma potrjene s pogostostnega seznama vzamemo le tiste, ki so s strani obeh strokovnjakov dobili kljukico (druge oznake so bile še: *nekaj manjka*, *prečrtano*, *vprašaj*), ugotovimo, da jih je med prvimi dvestotimi takih 117. Če na enak način pogledamo tudi seznam terminološko izluščenih samostalniških kandidatov, je rezultat nekoliko boljši: med prvimi dvestotimi izluščenimi je dvojno kljukico dobilo 125 kandidatov. Če preverimo npr. še mesta od 700 do 800 na obeh seznamih, zopet ugotovimo rahlo prednost terminološko izluščenega seznama, in sicer v razmerju 42 : 38. Pri mestih od 1.000 do 1.100 pa je uporabljeno luščenje glede na odločitve obeh strokovnjakov v še večji prednosti: 41 : 28.

Povzamemo lahko, da med vrhnjim delom seznama samostalnikov, ki se kot terminološki kandidati kažejo po luščenju, in vrhnjim delom pogostostnega seznama samostalnikov ni zelo velikih razlik, a so te z vidika ocene terminološkosti vedno v prid terminološko izluščenim seznamom, kar pomeni, da je pri naboru enobesednih terminov za geslovnik bolj smiselno izhajati iz slednjih.

Mesto na seznamu pri terminološkem luščenju	**Terminološki kandidat**
1.	odnos z javnost
4.	korporativen identiteta
5.	odnos z medij
6.	komunikacijski menedžment
10.	uglednosten kapital
11.	lokalen skupnost
13.	ciljen javnost
15.	krizen management
17.	deležniški skupina
22.	posloven komunikator
25.	komunikacijski aktivnost
27.	odnos z zaposleni
31.	komunikacijski kompetenca
32.	komunikacijski program
33.	komunikacijski management

Tabela 4: Prvih 15 izluščenih večbesednih terminoloških kandidatov (luščenje iz leta 2007), potrjenih s strani dveh strokovnjakov odnosov z javnostmi.

3.2. Večbesedni terminološki kandidati

Uspešnost samodejne ekstrakcije večbesednih terminoloških kandidatov smo s pomočjo dveh strokovnjakov s področja odnosov z javnostmi ocenili že pri prvem luščenju, ki smo ga izvedli leta 2007 in v katerega smo zajeli osem samostalniških oblikoskladenjskih vzorcev (prvih osem v Tabeli 1). Za ponazoritev podajamo vrhnji del takratnega seznama, ki prikazuje terminološke kandidate, za katere sta oba strokovnjaka potrdila, da gre za termine odnosov z javnostmi (Tabela 4). Skupno oceno uspešnosti takratnega luščenja lahko strnemo v ugotovitev, da pri prvih 1.000

[1] Tovrstno ocenjevanje terminološkosti zgolj po seznamu, brez vpogleda v sobesedilo, brez posvetovanja s še drugimi področnimi strokovnjaki, pa tudi brez zelo jasnega predstave o končnem slovarju odpira več vprašanj ter ima omejitve in pomanjkljivosti – pri projektu smo ga izvedli tudi zato, da nanje opozorimo in premislimo alternativne rešitve.

enotah na seznamu "v drugi polovici /.../ sicer narašča delež kolokacij, vendar je na tem seznamu veliko terminov" (Logar, Vintar, 2008: 13).[2]

V nadaljevanju izmed vseh 39 na novo luščenih oblikoskladenjskih vzorcev povzemamo le oceno terminološke zanimivosti vzorcev z glagolom, tj. zvez glagola s prislovom, glagolskih predložnih zvez in zvez glagola s podrednim veznikom *kot* (zadnjih devet vzorcev v Tabeli 1; o enobesednih glagolskih terminih, tudi izluščenih, gl. več v Logar, Vintar, 2008: 8–9, 11).

Analiza je pokazala, da le dva od devetih vzorcev dasta nekaj enot, ki bi jih kot celoto lahko vključili kot samostojno geslo v terminološki slovar odnosov z javnostmi, in da je ob teh dveh vzorcih le še eden, ki ima tovrsten potencial – gre za: *R G* in *G R* ter *G D S*. Natančnejši pregled vseh treh je najprej zajel odstranitev primerov z *biti*, *morati*, *želeti* in *jesti* pri *G D S* (kar je s seznama odstranilo 20 % nerelevantnih primerov) ter pri *R G*/*G R* še *moči*, *hoteti*, *smeti*, *imeti*, *začeti*, *postati*, *dobiti*, *iti*; *lahko*, *tako*, *treba*, *mogoče*, *bolj*, *vedno*, *sam*, *potrebno*, *rad*, *nekaj*, *najbolj*, *zato*, *sicer*, pri *G R* pa še dodatno *veliko*, *glede*, *čim* in *vse*. Na ta način so se pri vzorcu *R G* podatki zmanjšali za polovico, pri *G R* pa za 68 %. Potencialnih glagolskih večbesednih terminov je bilo, kot rečeno, le nekaj, in sicer po naši presoji vsi kot zveza glagola *komunicirati* in prislovov *dvosmerno*, *strateško*, *simetrično*, *javno*, *individualno*, *osebno*, *rutinsko* ter *navzven*. Vse ostalo pri teh treh vzorcih (pa tudi pri drugih, čeprav manj) je tako predvsem upoštevanja vreden prikaz kolokacijskega okolja glagolov in samostalnikov, ki so njihov del – če so ti glagoli in samostalniki terminološki, bodo seveda tudi njihovi besedni nizi ob ustrezni pogostosti kot kolokacije prišli v slovar, sicer pa ne, npr.:

– komunicirati*: komunicirati z/s [ljudmi*, *deležniki*, *javnostmi*, *mediji*, *novinarji*, *okoljem*, *organizacijo*, *potrošniki*, *predstavniki /.../*, *tržiščem*, *vlagatelji]*; *[dobro*, *učinkovito*, *nenehno*, *pravilno*, *premalo*, *prepričljivo*, *uspešno] komunicirati*;

– informacija: *[seznaniti*, *razpolagati*, *početi*, *povezati] z informacijo*; *[iti] za informacijo*; *[temeljiti*, *nanašati se] na informacijo*; *[soditi] med informacije*; *[zaupati] glede informacije*; *[predelati] v informacijo*; *[priti] do informacije*.

Lahko torej povzamemo, da so v našem primeru z luščenjem glagolskih vzorcev nastali seznami, ki z vidika nabora samostojnih iztočnic niso pomembno dopolnilo terminografskega dela, z vidika vsebine slovarja (podatkov znotraj iztočnic) pa to vendarle lahko so – zlasti če predpostavljamo, da želimo v slovar vključiti tudi značilno besedilno okolje terminov. Pri slednjem si je sicer mogoče pomagati tudi z naprednimi korpusnimi orodji, ki tovrstne podatke prikazujejo samodejno, tak je npr. program *Besedne skice/Sketch Engine* (Krek, Kilgarriff, 2006). Uporabnost tega in sorodnih orodij za pripravo terminoloških slovarjev bomo preizkusili v nadaljevanju projekta. Analiza je obenem izpostavila seznam glagolov in prislovov, ki bi jih bilo pri eventualnih prihodnjih luščenjih glagolskih zvez smiselno avtomatsko izločiti iz končnega seznama rezultatov in s tem podatke vnaprej selekcionirati; gre denimo za glagole *biti*, *imeti*, *iti*, *dati*, modalne glagole, deloma fazne glagole ipd. ter del prislovov, predvsem tistih, ki so v jeziku zelo pogosti, pomensko pa zelo splošni (*tako*, *zelo*, *lahko*, *vedno* itd.).

3.3. Priklic

Kot je razvidno iz dosedanje razprave, se je natančnost luščenja pri posameznih vzorcih močno razlikovala, različne evalvacije iz preteklih eksperimentov pa kažejo, da se pri osnovnem naboru oblikoskladenjskih vzorcev natančnost luščenja za prvih 100 kandidatov giblje med 70 in 90 % (Vintar in Erjavec, 2008; Vintar, 2010). A medtem ko je merjenje natančnosti razmeroma enostavno, saj moramo zgolj pregledati vrh seznama predlaganih kandidatov in ugotoviti, koliko jih je pravih terminov, je ugotavljanje priklica bistveno težje.

S priklicem pri jezikovnih tehnologijah merimo sposobnost sistema, da v množici podatkov prepozna zadovoljiv odstotek iskanih primerov. Z drugimi besedami: pri našem eksperimentu nas je zanimalo, koliko terminov, ki se v besedilu pojavijo, ostane neizluščenih. V ta namen smo izbrali znanstveni članek s področja odnosov z javnostmi in prosili dva področna strokovnjaka (ne ista strokovnjaka, kot sta pregledovala sezname), da v članku označita vse terminološko relevantne izraze. Iz obeh člankov smo nato izpisali označene termine in opazovali dvoje:

a) v kolikšni meri se mnenji obeh strokovnjakov o terminološkosti ujemata (Tabela 5);

b) v kolikšni meri se nabor "človeško" izbranih terminov ujema s samodejno izluščenim seznamom (Tabela 6).

Prvi strokovnjak	184
Drugi strokovnjak	261
Presek	109
Unija	415
Ujemanje med strokovnjakoma (IAA)	0,26

Tabela 5: Ujemanje pri oceni terminološkosti med obema strokovnjakoma.

Kot vidimo, sta strokovnjaka v istem besedilu označila zelo različno število terminov, saj je presek med njima le 109. Če njuna izbora združimo, dobimo unijo v velikosti 415 terminov, v nadaljevanju pa merimo priklic na obeh seznamih.

	Priklic na preseku	**Priklic na uniji**
Vsi izluščeni terminološki kandidati (63.179)	0,93	0,85
Prvih 10.000	0,84	0,72
Prvih 5.000	0,75	0,63

Tabela 6: Priklic izluščenih terminoloških kandidatov.

[2] Pri tem smo ločili med (a) večbesednim terminom kot stalno zvezo, poimenovanjem in (b) kolokacijo kot leksikalno in/ali pragmatično povezano ponovljivo sopojavitvijo vsaj dveh leksikalnih enot, ki sta med seboj v neposrednem skladenjskem razmerju (Bartsch, 2004, nav. po Heid, 2006: 980), a v ožjem pomenu, tj. kot prosto zvezo. Razmejitev je seveda groba (prim. Vintar, 2003: 74; Erjavec, Vintar, 2004: 104; Logar, Vintar, 2008: 12–14), jo pa ohranjamo tudi v tem prispevku.

Rezultati priklica so dobri. Od 109 terminov ki sta jih označila oba strokovnjaka, jih sistem le 7 ni izluščil in sicer gre v treh primerih za angleške izraze (*issue management*, *press clipping*, *cluster analiza*), v dveh primerih za petbesedne enote, naši vzorci pa ne presegajo dolžine štirih besed (*pragmatična raven odnosov z javnostmi*, *odzivni razvojni model strateškega načrtovanja*), in v enem primeru za vzorec *P P*, česar pri vzorcih nismo predvidevali (*dvosmerni asimetrični*). Kot je pričakovati, priklic pada sorazmerno z "rezanjem" števila kandidatov, vendar dosega pri prvih 5.000 kandidatih (kar je manj kot 10 % celotnega seznama izluščenih) še vedno 0,75 (oziroma 0,63), kar je dober rezultat.

4. Sklep

Pri vsakem ročnem prepoznavanju terminov v besedilu naletimo vsaj na štiri težavne točke, ki so: meja med terminološko in splošno leksiko, razmerje med terminologizacijo in determinologizacijo, termini več strok ter terminološke kolokacije. Zato je še toliko bolj pomembno, da imamo objektiven kazalec večje oz. manjše potencialne terminološkosti. Programu, ki deluje na podlagi statističnih izračunov in vnaprej danega, pri tokratnem eksperimentu precej obsežnega nabora jezikoslovnih vzorcev, smo dali prav tako nalogo: da v korpusu besedil odnosov z javnostmi prepozna strokovno izrazje. S terminološko utežjo, katere izhodišče je razmerje relativnih pogostosti besed v specializiranem korpusu in splošnem korpusu – v našem primeru KoRP in FidaPLUS – smo dobili več seznamov; v vrhu enega od njih so npr. samostalniki *javnost*, *odnos*, *organizacija*, *komuniciranje*, *podjetje*, *medij* in *znamka*. Pred njihovo dokončno vključitvijo v geslovnik bo še vedno potrebna analiza področnih strokovnjakov ter sodelujočih jezikoslovcev, a tak pristop kakovostno vendarle močno presega zgolj individualne in večkrat hipne odločitve posameznikov, ki se iskanja terminov lotijo ročno.

Obe primerjavi, tj. primerjava seznama terminološko izluščenih samostalnikov in pogostostnega seznama samostalnikov ter ocena terminološkosti obeh seznamov s strani strokovnjakov odnosov z javnostmi, sta pokazali prednost izluščenega seznama. Razlike so morda na prvi pogled majhne, vendarle pa so zelo relevantne, saj potrjujejo ravno tisto, kar nas je še posebej zanimalo: občutljivost terminološke uteži. V zvezi s slednjo v nadaljevanju raziskave načrtujemo še en preizkus, in sicer preverbo vpliva strokovnega področja na večjo oz. manjšo uspešnost rezultatov luščenja v razmerju do terminološke uteži. Ta hip se namreč zdi, da je družboslovno področje, kakršno so odnosi z javnostmi, ki obstajajo na presečišču menedžmenta, marketinga in komunikologije (Gruban, 1998: 25) zaradi delne tematske prekrivnosti z referenčnimi korpusi (časopisi in revije, deloma tudi knjige z besedili o aktualnih dogodkih, gospodarstvu, poslu, financah ipd.) za luščenje, ki v uteži vključuje ravno tovrstno primerjavo, večji izziv kot katero drugo specializirano področje ali podpodročje, npr. polimerno inženirstvo ali zaključni procesi v biotehnologiji.

Odločitev za luščenje glagolskih vzorcev, sploh v tolikšnem obsegu, se jezikoslovnemu bralcu morda zdi presenetljiva. Presodili smo, da je glagolom kot pogosto spregledanemu delu terminologije (prim. npr. Žele, 2004: 78: glagoli so "prav zaradi svoje organizacijske vloge v stavčnih povedih povsem netipična besedna vrsta za termine") vredno dati tovrstno pozornost ter z veliko količino podatkov in statistično podprto potrditi ali ovreči njihovo terminološko relevantnost. Z dopuščanjem morebitne specifičnosti katere od strok si dovoljujemo posplošitev, da luščenje glagolskih oblikoskladenjskih vzorcev (za razliko od enobesednih glagolov) za slovenščino ne daje relevantnih seznamov z večbesednimi glagolskimi termini (prim. tudi Arhar Holdt, 2011: 121–125). Ob možnosti opazovanja tipičnega besedilnega okolja glagolov s katerim drugim orodjem je luščenje glagolskih vzorcev bolj smiselno povsem opustiti. Po drugi strani pa velja poudariti, da so bili veliko boljši kot pri glagolskih vzorcih rezultati luščenja vzorcev, ki jih tu nismo posebej obravnavali, tj. vzorcev s samostalniškim jedrom. Tako so npr. vzorci *P S*, *P in P S* ter *P P S* v zgornjem delu seznama v več kot 50 % dali gradivo, ki je neposredno uporabno za slovarski geslovnik. Odprto zaenkrat ostaja vprašanje, ali je smiselno še povečati dolžino vzorcev na pet, šest ali več besed. V naši analizi smo se ustavili pri štiribesednih, vendar tej meji nismo pripisovali dokončnosti. Dve petbesedni enoti sta kot termina prepoznala področna strokovnjaka, kar jasno kaže, da v odnosih z javnostmi taki termini so. Ena pot do njih je povečanje obsega vzorcev luščenja, druga pa analiza besedilnega okolja.

Analiza priklica še potrjuje naše prepričanje, da smo pri izboljšavah luščilnika ter pri širjenju seznama vzorcev na pravi poti, saj je samodejno luščenje pri terminih, ki sta jih označila oba strokovnjaka, doseglo kar 93-odstotni priklic. Po drugi strani pa se skozi nizko ujemanje med obema strokovnjakoma jasno kaže subjektivnost same definicije terminološkosti ter posledično zahtevnost zastavljenih ciljev.

Zahvala

Avtorice se zahvaljujejo anonimnima recenzentoma ter uredniku zbornika za koristne pripombe in predloge. Raziskava, predstavljena v prispevku, je nastala v okviru projekta *Terminološke baze podatkov kot osnova strokovnih znanj: model za sistematizacijo terminologij* (http://www.termis.fdv.uni-lj.si/), ki jo po pogodbi št. 1000-11-274193 financira Javna agencija za raziskovalno dejavnost Republike Slovenije ter sofinancerja Pristop, d. o. o., in Gospodarska zbornica Slovenije. Projekt podpirajo tudi sponzorji: Elektro Ljubljana, d. d., Mercator, d. d., Pošta Slovenije, d. o. o., in Zavarovalnica Maribor, d. d.

Literatura

Arhar Holdt, Š., 2011. *Luščenje besednih zvez iz besedilnega korpusa z uporabo dvodelnih in tridelnih oblikoskladenjskih vzorcev*. Ljubljana: Trojina, zavod za uporabno slovenistiko.

Atkins, B. T. S., Rundell, M., 2008. *The Oxford Guide to Practical Lexicography*. New York: Oxford University Press.

Bergenholtz, H., Tarp, S., ur., 1995. *Manual of Specialised Lexicography*. Amsterdam, Philadelphia: John Benjamins Publishing Company.

Biber, D., Conrad, S., Reppen, R., 1998: *Corpus Linguistics: Investigating Language Structure and Use*. Cambridge: Cambridge University Press.

Čermák, F., ur., 2011. *Korpusová lingvistika*. Praga: Nakladatelství lidové noviny, Ústav Českého národního korpusu.

Gantar, P., 2007. *Stalne besedne zveze v slovenščini: korpusni pristop*. Ljubljana: Založba ZRC, ZRC SAZU.

Gantar, P., 2009. Leksikalna baza: vse, kar ste vedno želeli vedeti o jeziku. *Jezik in slovstvo*, 54(3/4): 69–94.

Gorjanc, V., 2005. *Uvod v korpusno jezikoslovje*. Domžale: Založba Izolit.

Gorjanc, V., Krek, S., ur., 2005. *Študije o korpusnem jezikoslovju*. Ljubljana: Krtina.

Gruban, B., 1998. Izobraževanje v odnosih z javnostmi. V B. Gruban, D. Verčič, F. Zavrl (ur.), *Preskok v odnose z javnostmi*: 25–44. Ljubljana: Pristop.

Halliday, M., et al., 2004. *Lexicology and Corpus Linguistics: An Introduction*. London: Continuum.

Hanks, P., 2008. *Lexicology*. New York: Routledge.

Heid, U., 2006. A model for a multifunctional dictionary of collocations. V *EURALEX*: 979–988.

Holozan, Peter (2006): Dodatne dvoumnosti zaradi popustljivosti analizatorja pri analizi slovenskih stavkov. V T. Erjavec, J. Žganec Gros (ur.), *Jezikovne tehnologije*: 146–149. Ljubljana: Institut Jožef Stefan.

Krek, S., 2003. Sodobna dvojezična leksikografija. *Jezik in slovstvo*, XLII(1): 45–60.

Krek, S., Kilgarriff, A., 2006. *Slovene Word Sketches*. Dostopno prek: <http://nl.ijs.si/is-ltc06/proc/12_Krek.pdf> .

Leech, G., 1992. Corpora and Theories of Linguistic Performance. V J. Svartvik (ur.), *Directions in Corpus Linguistics*: 105–122. Berlin: Mouton de Gruyter.

Logar, N., 2007. *Korpusni pristop k pridobivanju in predstavitvi jezikovnih podatkov v terminoloških slovarjih in terminoloških podatkovnih zbirkah*. Doktorska disertacija. Ljubljana: Filozofska fakulteta.

Logar, N., Vintar, Š., 2008. Korpusni pristop k izdelavi terminoloških slovarjev: od besednih seznamov in konkordanc do samodejnega luščenja izrazja. *Jezik in slovstvo*, LIII(5): 3–17.

McEnery, T., Wilson, A., 1996. *Corpus Linguistics*. Edinbourgh: Edinbourgh University Press.

Pearson, J., 1998. *Terms in Context*. Amsterdam, Philadelphia: John Benjamins Publishing Company.

Romih, M., Holozan, P., 2002. Infrastruktura za razvoj jezikovnih tehnologij – korpus FIDA in sistem ASES. V T. Erjavec, J. Žganec Gros (ur.), *Jezikovne tehnologije*: 166. Ljubljana: Inštitut Jožef Stefan.

Schryver, G. de, 2003. Lexicographers' Dreams in the Electronic-Dictionary Age. *International Journal of Lexicography*, 16(1): 143–199.

Sinclair, J., 2004. *Trust the Text: Language, Corpus and Discourse*. London, New York: Routledge.

Teubert, W., Krishnamurty, R., ur., 2007. *Corpus Linguistics: Critical Concepts in Linguistics*. London, New York: Routledge.

Vintar, Š., 2003. *Uporaba vzporednih korpusov za računalniško podprto ustvarjanje dvojezičnih terminoloških virov*. Doktorska disertacija. Ljubljana: Filozofska fakulteta.

Vintar, Š., 2008. *Terminologija: terminološka veda in računalniško podprta terminografija*. Ljubljana: Znanstvena založba Filozofske fakultete, Oddelek za prevajalstvo.

Vintar, Š., 2009. Samodejno luščenje terminologije – izkušnje in perspektive. V N. Ledinek, M. Žagar Karer, M. Humar (ur.), *Terminologija in sodobna terminografija*: 345–356. Ljubljana: Založba ZRC, ZRC SAZU.

Vintar, Š., 2010. Bilingual Term Recognition Revisited: The Bag-of-Equivalents Term Alignment Approach and its Evaluation. *Terminology*, 16(2): 141–158.

Vintar, Š., Erjavec, T., 2008. iKorpus in luščenje izrazja za Islovar. V T. Erjavec, J. Žganec Gros (ur.), *Jezikovne tehnologije*: 65–69. Ljubljana: Institut Jožef Stefan.

Event and Temporal Relation Extraction from Croatian Newspaper Texts

Mladen Marović, Jan Šnajder, Goran Glavaš

University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
mladen.marovic@gmail.com, {jan.snajder, goran.glavas}@fer.hr

Abstract

Event extraction and temporal relation extraction are the subjects of extensive research, which has been additionally motivated by focused evaluation exercises such as TempEval. In this paper we present the work on supervised event and temporal relation extraction from Croatian newspaper texts. Taking into account the limited availability of linguistic tools for Croatian, we focus our research around simple lexical features. We manually annotated a newspaper corpus or events and temporal relations in Croatian according to the TimeML and TimeBank guidelines. Experimental evaluation yielded promising results: F1 scores of up to 77% for event identification, 48% for event classification, and 51% for temporal relation classification.

Luščenje dogodkov in časovnih relacij iz hrvaških časopisnih besedil

Luščenje dogodkov in časovnih relacij je zelo živahno raziskovalno področje, ki se je še posebej razmahnilo s pojavom skupinskih evalvacijskih pobud, kot je na primer TempEval. V pričujočem prispevku predstavljamo sistem za nadzorovano luščenje dogodkov in časovnih relacij iz hrvaških časopisnih besedil. Glede na to, da je dostopnost jezikovnih orodij za hrvaščino omejena, se v raziskavi osredotočamo zgolj na enostavne leksikalne lastnosti. Pri ročnem označevanju dogodkov in časovnih relacij v korpusu časopisnih besedil smo uporabljali smernice TimeML in TimeBank. Eksperimentalno vrednotenje rezultatov je zelo spodbudno, saj za prepoznavanje dogodkov F1 znaša 77%, za klasifikacijo dogodkov 48% in za klasifikacijo časovnih relacij 51%.

1. Introduction

Event extraction and temporal relation extraction are non-trivial information extraction (IE) tasks that often play an important role in various practical natural language processing (NLP) applications, such as question answering (Saurí et al., 2005) and document summarization (Lee et al., 2003). Event and temporal relation extraction tasks have attracted a lot of attention, in particular within the TempEval evaluation exercises. Event extraction task refers to the classification of words into events and non-events (event identification), or, more commonly, to identifying the events and determining their types (event classification). Temporal relation extraction refers to classification of temporal relations between extracted pairs of events.

In this paper we present the work on supervised event and temporal relation extraction from Croatian newspaper texts. As part of this work, we manually annotated a newspaper corpus following the guidelines from TimeML (Pustejovsky et al., 2003a) and TimeBank (Pustejovsky et al., 2003b). We then evaluated the event and temporal relation extraction performance of several classifiers. Because of the limited availability of linguistic tools for Croatian language, we use mostly lexical features. We achieved promising results, showing that using only simple features can yield satisfactory results for event and temporal relation extraction tasks for Croatian language. To the best of our knowledge, the work described in this paper is the first work on event and temporal relation extraction for the Croatian language, and Slavic languages in general.

The rest of the paper is structured as follows. The next section gives an overview of related research. In Section 3 we describe the corpus annotation. Supervised machine learning for event and temporal relation extraction is described in Section 4. Section 5 describes the experimental setting and discusses the results. Section 6 concludes the paper and suggests future work.

2. Related Work

2.1. Event extraction

Events in sentences were first studied in linguistic literature (Vendler, 1957; Verkuyl, 2005; Pustejovsky, 1991). Different event properties were defined based on event structure, duration, and telicity. Vendler (1957) proposed the distinction between four types of events (states, activities, accomplishments, and achievements), while Pustejovsky (1991) proposed a structural event hierarchy. Work focusing on practical IE and NLP tasks combines statistical and machine learning methods to automatically determine various properties of verbs and events occurring in text. Siegel and McKeown (2000) used machine learning methods based on features such as verb tense, presence of a negation word or absence of a subject to determine the aspectual properties of verbs. Classification of verbs into states and events resulted in an accuracy of 93.9%, while the classification of events into aspectual categories resulted in an accuracy of 74.0%.

To aid further research on event and temporal relation extraction, Pustejovsky et al. (2003a) developed TimeML, a rich specification language for event and temporal expressions in natural language text. In TimeML, events are defined as "situations that *happen* or *occur*" and can either be punctual or last for a period of time. The events are divided into eight classes: OCCURRENCE, PERCEPTION, REPORTING, ASPECTUAL, STATE, I_STATE,

I_ACTION, and MODAL. Based on TimeML, Pustejovsky et al. (2003b) created the TimeBank corpus consisting of 300 documents (60k words) manually annotated for events. Since the introduction of TimeML and TimeBank, a number of event extraction methods have been proposed (Saurí et al., 2005; Boguraev and Ando, 2005; Bethard and Martin, 2006). Boguraev and Ando (2005) additionally performed a quantitative analysis of the TimeBank corpus and criticized its small size and the unbalanced distribution of event classes, which they tried to compensate for with a word profiling technique.

2.2. Temporal relation extraction

The problem of defining events in the context of time was addressed early in linguistic literature (C. Bruce, 1972; Reichenbach, 1980). Allen (1983) introduced interval temporal algebra, which was widely accepted and further improved on by Galton (1990). In interval algebra, temporal relations between two events are defined as relations between the corresponding beginning points and end points.

The study of temporal relations is closely related to event extraction research. In TimeML, Pustejovsky et al. (2003a) introduced eight labels based on Allen's interval algebra for labeling relations between events, as well as relations between events and temporal expressions: *before*, *immediately before*, *includes*, *holds*, *simultaneous*, *identity*, *begins*, and *ends*. These labels were also used to annotate the TimeBank corpus, although with low inter-annotator agreement. Mani et al. (2006) expanded the TimeBank relations using a temporal closure algorithm (Verhagen, 2005), and used machine learning methods to classify the temporal relations. Lapata and Lascarides (2004) chose a somewhat different approach: they selected sentences containing temporal connectives, such as *during*, *when*, *while*, etc., and used a simple probabilistic model to insert the appropriate connectives in place of the removed ones. In their subsequent work, Lapata and Lascarides (2006) determined the mapping between the connectives and the temporal relations defined in TimeML and used this mapping for relation classification.

To encourage further research in temporal text processing, the TempEval (Verhagen et al., 2007) and TempEval-2 (Verhagen et al., 2010) evaluation exercises were organized in 2007 and 2010, respectively. In order to achieve a satisfactory inter-annotator agreement, in TempEval-2 the following reduced subset of relations was used: BEFORE, AFTER, OVERLAP, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER, and VAGUE. Two TempEval-2 tasks are relevant to the work described in this paper: (1) determining the temporal relation between two main events in consecutive sentences, and (2) determining the temporal relation between two events where one event syntactically dominates the other. Diverse systems participated in the competition, achieving F1 scores of up to 58% for the first task and 66% for the second task.

3. Corpus Annotation

The data we used for training and testing of models consists of 230 newspaper articles from the daily Croatian newspaper *Vjesnik* from years 1999–2009. We chose the articles based on their length, type, and topic. The average article length was around 500 tokens (including words and punctuation marks). The chosen topics were daily news from Croatia and the world, sports, politics, and culture. We did not consider articles such as columns, reviews, and other types of opinionated text. The resulting corpus consists of 118,900 tokens (102,830 words), with 26,095 word form types and 10,963 lemma types.

3.1. Events and event classes

In our research an event is a "cover term for situations that *happen* or *occur*," as defined by Pustejovsky et al. (2003a). Because our research focuses on newspaper texts and articles that report about specific instances of events, we introduced three modifications to the TimeML guidelines. First, we consider only *realis* events, i.e., events that are asserted to have already happened, are happening, or will happen (Polanyi and Zaenen, 2006). Events modified by a modal verb are not realis, so we disregard such events. Secondly, we do not consider generic events – events that describe classes of events instead of specific events and are not anchored in time. For instance, the word *playing* in the sentence *Playing with knives is dangerous* is a generic event. Contrary to that, words representing sets of events of the same type that are anchored in time are considered events. For example, the word *matches* in the sentence *The semifinal matches of the Euro 2012* is an event because the matches are all unique events, anchored in time, specific, and well-defined in the context of the Euro 2012 semifinals. Thirdly, in the newspaper domain, we do not consider states as relevant events. For example, the word *knows* in the sentence *He knows a secret* is not an event. However, we do consider changes of states as events. For example, the word *discovered* in the sentence *He discovered a secret* is an event because it refers to a state change, i.e., a transition from the state of not knowing to the state of knowing. Events are considered on a single word basis; each word is classified separately as either event or non-event. We use the following seven event classes:

1. REPORTING – events of narrative character (*reći – say, objaviti – declare*);
2. ASPECTUAL – events that describe the aspect (*početi – begin, završiti – finish*);
3. PERCEPTION – events that describe physical perception (*vidjeti – see, čuti – hear*);
4. I_ACTION – events that introduce another event (*istražiti umorstvo – investigate a murder, pokušati pobjeći – attempt to escape*);
5. OCCURRENCE – single events that happen or occur (*umorstvo – murder, pobjeći – escape*);
6. HALF_GENERIC – words denoting sets of unique events defined in a context (*The semifinal matches of the Euro 2012*);
7. STATE_CHANGE – events that describe the transition from one state to another (*otkriti – discover, prihvatiti – accept*).

Table 1: Event annotation summary

Event Class	Frequency	IAA
OCCURRENCE	6867	0.6537
REPORTING	1303	0.8207
I_ACTION	1124	0.3341
HALF_GENERIC	642	0.2080
STATE_CHANGE	348	0.2349
ASPECTUAL	301	0.4272
PERCEPTION	58	0.3383
Total	10,643	

The first five classes are similar to those defined by Pustejovsky et al. (2003a), whereas HALF_GENERIC and STATE_CHANGE are the newly introduced classes, as described previously.

Some events can belong to more than one class. For example, the event *saw* in the example *He saw the car crash* can be labeled as PERCEPTION, as well as I_ACTION (because of the event *crash*). To avoid inconsistencies, we introduced priority levels for event classes as follows (listed from highest to lowest priority):

1. PERCEPTION, ASPECTUAL, REPORTING;
2. I_ACTION;
3. STATE_CHANGE;
4. OCCURRENCE, HALF_GENERIC.

Classes of the same priority level are mutually exclusive. We established these priority levels based on the frequency of event classes in the TimeBank corpus (Boguraev and Ando, 2005). The priority levels ensure that classes with lower frequencies, such as ASPECTUAL and PERCEPTION, will not be neglected by choosing a more frequent class, such as I_ACTION. The most general event classes – OCCURRENCE and HALF_GENERIC – are assigned the lowest priority level.

Five annotators annotated the corpus. They were given detailed guidelines and performed preliminary annotation in two calibration rounds to improve the inter-annotator agreement (IAA). In each round they annotated a set of ten articles. Following the annotation, they met, discussed borderline cases, and resolved the disagreements. After the second round, the IAA on the event extraction task (computed as the average of the pairwise F1 scores) was 0.7951. The remaining 210 articles were then distributed evenly among the annotators and each article was independently annotated by a single annotator. The summary of event annotation is given in Table 1.

3.2. Temporal relations

Our definition of types of temporal relations is based on Allen's interval algebra (Allen, 1983). Because some of the interval relations seem too specific to be determinable from text, we conjoined some relations with the appropriate more general relations: we conjoined MEET with BEFORE, MEETS_INVERSE with AFTER, STARTS and FINISHES with DURING, and STARTS_INVERSE and FINISHES_INVERSE with COVERS. The resulting relation types are shown in Fig. 1. We used the label NON_DETERMINABLE for relations whose type could not be determined based on the information provided in the text.

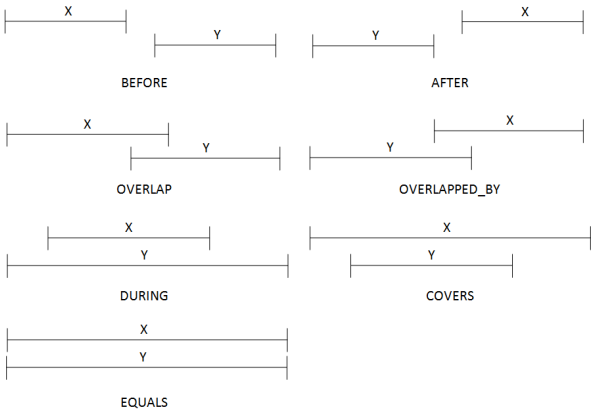


Figure 1: Types of temporal relations

Table 2: Temporal relation annotation summary

Relation Type	Frequency	IAA
BEFORE	4860	0.7660
AFTER	3500	0.8676
EQUALS	1880	0.4968
COVERS	1597	0.5847
DURING	1341	0.5775
NON_DETERMINABLE	763	0.1813
OVERLAP	46	0.0000
OVERLAPPED_BY	24	0.0833
Total	14,011	

After labeling the events, four annotators, who participated in event annotation, annotated the temporal relations between all pairs of events within the same sentence. After two calibration rounds, they achieved the IAA of 0.5855, measured in terms of Cohen's κ (Cohen, 1960) – a moderate agreement according to Landis and Koch (1977). The temporal relation annotation summary is given in Table 2. Relation types OVERLAP and OVERLAPPED_BY occurred rarely, which can be traced down to two causes: (1) a generally small number of such relations in newspaper articles, and (2) the specific nature of such relations that makes them easily confusable with other relations. This was confirmed by the annotators, who reported that they had usually labeled the potential OVERLAP and OVERLAPPED_BY relations as other relations, such as BEFORE and AFTER, because the context did not explicitly indicate that the events were overlapping.

4. Classifiers and Features

For the event extraction task, we experimented with two approaches: binary classification of words into events and non-events (event identification) and multiclass classification of words into one of eight event classes (an additional class was introduced for non-event words). The temporal relation extraction task was framed as multiclass classification of all pairs of events occurring in the same sentence.

For both extraction tasks, we experimented with the following classification algorithms: naive Bayes (NB), k-nearest neighbors (k-NN), and support vector machines (SVM) with a linear kernel. Based on preliminary experiments, $k = 3$ yielded the best results for k-NN. We used *RapidMiner*[1] implementations of naive Bayes and k-NN, and *LibLinear*[2] implementation of the SVM. The baseline for event extraction is a simple classifier that labels each word with its most frequent label in the training data. For relation extraction, a majority class classifier is used as a baseline (with BEFORE being the majority class).

Because the linguistic tools for Croatian are of limited availability, we used mostly lexical features to build the classification models: word, lemma, stem, POS tag, case, number, modality, auxiliary verbs, verb form, verb valence class, negation, and the surrounding words. For lemmatization and (ambiguous) POS tagging we used the semi-automatically acquired morphological lexicon by Šnajder et al. (2008). For verb valence classes we used the Croatian verb valence lexicon *Crovallex* developed by Preradović et al. (2009). Verb form may be either indicative, imperative, conditional, infinitive, or participle. The features we used for temporal relation classification were similar to those used for event extraction. For each event in the pair, we used the following features: word, lemma, stem, POS tag, modality, auxiliary words, *Crovallex* class, and event class. Moreover, we used a binary feature vector indicating which words occurred between the two event words.

5. Experimental Evaluation

We performed two experiments to evaluate event extraction: binary classification (event identification) and multiclass classification (event classification). We evaluated temporal relation classification by determining types of relations between all pairs of events within the same sentence. Performance estimates are obtained using ten-fold cross-validation; the reported results are macro-averaged F1 scores averaged over ten folds.

5.1. Results

Table 3 shows the results for event extraction. All classifiers outperformed the baseline, with SVM performing best in most cases. Event identification achieved the F1 score of up to $77.40 \pm 0.80\%$. As expected, event classification proved to be a more difficult task than event identification, yielding the much lower overall F1 score of $48.04 \pm 3.21\%$. The classifiers performed better on the OCCURRENCE, REPORTING and ASPECTUAL event classes. This is in accordance with the per class inter-annotator agreement (cf. Table 1), indicating that inter-annotator agreement correlates with classifier performance.

The results for temporal relation extraction are given in Table 4. All classifiers outperformed the majority class baseline. SVM performed best, outperforming the naive Bayes and k-NN classifiers by a margin of 12% and 19%, respectively, and yielding the F1 score of $51.16 \pm 2.94\%$. The performance for classes BEFORE, AFTER, DURING, and COVERS is higher than for other classes, which again is in accordance with the per class inter-annotator agreement (cf. Table 1).

5.2. Discussion

The comparison of our event extraction results with the results of others is difficult due to the differences in the annotation schemes. For example, Saurí et al. (2005) achieved the F1 score of 80% for event identification, which is slightly better than the 77% we achieved, but their approach was based on word chunking, whereas we consider events as single words. Similarly, Saurí et al. (2005) report the F1 score of 86% for event classification, which is much better than our 48%, but they used an entirely different set of event classes (states, general occurrences, reporting, intensional, and perception). Moreover, as noticed by Bethard and Martin (2006), their method included a check to determine whether a word occurs as an event in the TimeBank corpus, which resulted in unfair performance estimates. Bethard and Martin (2006) report F1 scores of up to 76% for event extraction and 58% for event classification. Verhagen et al. (2010) achieved somewhat higher F1 scores for Spanish and English: 88% and 80% for event extraction, and 66% and 79% for event classification.

Temporal relation extraction results are also not directly comparable to other reported results because of the differences in the relation types and the pairs of events considered. Verhagen et al. (2010) report F1 scores of 58% and 66% for the tasks relevant to temporal relation extraction. Only the second task can be related to our research because it considers event pairs within the same sentence. However, only specific event pairs are considered, therefore these results are expected to be better than ours.

6. Conclusion and Future Work

Event and temporal relation extraction are widely researched IE tasks, which can be used in various NLP application. In this paper, we studied event and temporal extraction from Croatian newspaper texts. To that end we manually annotated a corpus and used it to evaluate the event and temporal relation extraction performance of several different classifiers. We achieved the F1 scores of 77% for event identification, 48% for event classification, and 51% for temporal relation classification, significantly outperforming the baseline on all three tasks. A direct comparison to the results for English is difficult, nonetheless we consider the results to be satisfactory given that we were using only simple lexical features. We believe that our results are also indicative for other Slavic languages.

There are several directions for further research. First, considering the relatively low inter-annotator agreement,

[1] http://rapid-i.com/content/view/181/190/

[2] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

Table 3: Event extraction performance (% F1)

	Baseline	NB	k-NN	SVM
Event identification	5.82 ± 0.69	68.85 ± 0.63	71.07 ± 1.31	$\mathbf{77.40 \pm 0.80}$
Event classification	0.84 ± 0.20	33.56 ± 1.27	43.63 ± 2.93	$\mathbf{48.04 \pm 3.21}$
OCCURRENCE	3.10 ± 0.76	55.40 ± 2.04	53.33 ± 2.14	$\mathbf{62.34 \pm 1.23}$
REPORTING	0.69 ± 0.87	75.28 ± 1.09	76.44 ± 3.32	$\mathbf{79.91 \pm 2.36}$
ASPECTUAL	0.39 ± 1.24	12.25 ± 2.31	58.42 ± 7.37	$\mathbf{59.21 \pm 5.48}$
PERCEPTION	0.00 ± 0.00	24.66 ± 8.65	50.80 ± 15.64	$\mathbf{56.32 \pm 18.18}$
I_ACTION	0.99 ± 0.90	$\mathbf{28.63 \pm 1.88}$	24.21 ± 4.26	24.28 ± 2.40
STATE_CHANGE	0.72 ± 0.93	$\mathbf{25.11 \pm 3.62}$	23.18 ± 8.04	23.17 ± 6.49
HALF_GENERIC	0.00 ± 0.00	13.59 ± 2.11	18.99 ± 5.70	$\mathbf{31.04 \pm 6.15}$

Table 4: Temporal relation extraction performance (% F1)

	Baseline	NB	k-NN	SVM
Temporal relation classification	6.44 ± 0.01	38.77 ± 1.87	32.17 ± 1.86	$\mathbf{51.16 \pm 2.94}$
BEFORE	51.43 ± 0.05	63.63 ± 1.74	59.61 ± 3.47	$\mathbf{73.12 \pm 0.85}$
AFTER	–	59.35 ± 2.18	56.16 ± 3.83	$\mathbf{71.08 \pm 1.46}$
OVERLAP	–	11.88 ± 11.70	0.00 ± 0.00	$\mathbf{32.07 \pm 19.44}$
OVERLAPPED_BY	–	2.64 ± 2.37	0.00 ± 0.00	$\mathbf{20.67 \pm 23.82}$
DURING	–	55.59 ± 3.14	46.16 ± 4.26	$\mathbf{60.41 \pm 2.89}$
COVERS	–	36.51 ± 2.52	24.49 ± 3.72	$\mathbf{50.83 \pm 3.49}$
EQUALS	–	36.91 ± 3.54	33.87 ± 2.30	$\mathbf{46.01 \pm 3.43}$
NON_DETERMINABLE	–	43.63 ± 3.71	37.05 ± 7.22	$\mathbf{55.11 \pm 8.20}$

further work should be focused on a more detailed analysis of the annotations and the improvement in annotation guidelines. Secondly, the emergence of new linguistic tools for Croatian language presents an opportunity for using more complex features for classification. Finally, we intend to work on methods for relating the events to normalized temporal expressions (TIMEXes) extracted from text, which should aid in the classification of temporal relations between events.

7. Acknowledgments

The authors would like to thank Damir Cvetovac, Latica Čačković, Marijana Marović, and Kristijan Pavlović for manually annotating the corpus. This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia, under Grant 036-1300646-1986.

8. References

J.F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

S. Bethard and J.H. Martin. 2006. Identification of event mentions and their semantic class. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 146–154.

B. Boguraev and R.K. Ando. 2005. Timebank-driven TimeML analysis. *Annotating, Extracting and Reasoning about Time and Events*, (05151).

B. C. Bruce. 1972. A model for temporal references and its application in a question answering program. *Artificial intelligence*, 3:1–25.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

A. Galton. 1990. A critical examination of Allen's theory of action and time. *Artificial Intelligence*, 42(2-3):159–188.

J.R. Landis and G.G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

M. Lapata and A. Lascarides. 2004. Inferring sentence-internal temporal relations. In *Proceedings of HLT-NAACL*, pages 153–160.

M. Lapata and A. Lascarides. 2006. Learning sentence-internal temporal relations. *Journal of Artificial Intelligence Research*, 27(1):85–117.

C.S. Lee, Y.J. Chen, and Z.W. Jian. 2003. Ontology-based fuzzy event extraction agent for Chinese e-news summarization. *Expert Systems with Applications*, 25(3):431–447.

I. Mani, M. Verhagen, B. Wellner, C.M. Lee, and J. Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 753–760.

L. Polanyi and A. Zaenen. 2006. Contextual valence shifters. *Computing attitude and affect in text: Theory and applications*, pages 1–10.

N.M. Preradović, D. Boras, and S. Kisićek. 2009. CROVALLEX: Croatian verb valence lexicon. In *Information Technology Interfaces, 2009. ITI'09. Proceedings of the ITI 2009 31st International Conference on*, pages 533–538. IEEE.

J. Pustejovsky, J. Castano, R. Ingria, R. Sauri, R. Gaizauskas, A. Setzer, G. Katz, and D. Radev. 2003a. TimeML: Robust specification of event and temporal expressions in text. *New Directions in Question Answering*, 2003:28–34.

J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. 2003b. The TimeBank corpus. In *Corpus Linguistics*, volume 2003, page 40.

J. Pustejovsky. 1991. The syntax of event structure. *Cognition*, 41(1):47–81.

H. Reichenbach. 1980. *Elements of symbolic logic*. Dover Publications.

R. Saurí, R. Knippen, M. Verhagen, and J. Pustejovsky. 2005. Evita: a robust event recognizer for QA systems. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 700–707.

E.V. Siegel and K.R. McKeown. 2000. Learning methods to combine linguistic indicators: Improving aspectual classification and revealing linguistic insights. *Computational Linguistics*, 26(4):595–628.

Z. Vendler. 1957. Verbs and times. *The philosophical review*, 66(2):143–160.

M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80.

M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62.

M. Verhagen. 2005. Temporal closure in an annotation environment. *Language Resources and Evaluation*, 39(2):211–241.

H. Verkuyl. 2005. Aspectual composition: Surveying the ingredients. *Perspectives on aspect*, pages 19–39.

J. Šnajder, B. Dalbelo Bašić, and M. Tadić. 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing and Management*, 44(5):1720–1731.

Korpusna analiza slovenskega deležja v različnih besedilnih tipih

Tamara Mikolič Južnič

Oddelek za prevajalstvo, Filozofska fakulteta, Univerza v Ljubljani
Aškerčeva 2, Ljubljana
tamara.mikolic@guest.arnes.si

Povzetek

Prispevek se osredotoča na raziskavo rabe deležja v različnih besedilnih tipih v slovenskem jeziku. O splošno uveljavljenem mnenju o redkosti slovenskega deležja govorijo številni avtorji, vendar so dejanske, s konkretnimi podatki podprte raziskave na tem področju izjemno redke. V tem prispevku želimo ugotoviti, kako pogosto se deležje pojavlja v nekaterih slovenskih neliterarnih besedilnih tipih, kot so znanstvena besedila (humanistično-družboslovna ter naravoslovna) in strokovna pravna besedila (zakoni) ter v korpusu splošne slovenščine, pa tudi v leposlovnih besedilih. Pri zbiranju podatkov so bili uporabljeni različni razpoložljivi korpusi, kot je korpus JOS 100K, korpus Spook in ad hoc korpus neliterarnih besedil. Ob zbiranju podatkov smo preverili tudi uspešnost oblikoskladenjskega označevanja projekta JOS glede deležij.

Corpus Analysis of the Slovene Participle in various text types

The paper focuses on a study of the use of the Slovene participle in various text types. Several authors have pointed out that the participle is rarely used, but actual research based on concrete data is extremely rare in this field. Here we want to find out how frequent the participle is in certain Slovene non-literary text types such as scientific texts (from the fields of the arts and humanities and the natural sciences) and legal texts (laws) and in a reference corpus of Slovene, as well as in literary texts. Several different corpora were used in collecting data (JOS 100K, Spook and an ad hoc corpus of non-literary texts). When collecting the data, we also verified how successful the morphosyntactic tagging system of the JOS project is with respect to participles.

1. Uvod

Slovensko deležje, tako kot nekatere druge slovenske neosebne glagolske oblike, nima dobrega slovesa med uporabniki jezika: Žagar (1998: 41) ugotavlja, da njihova raba peša, Schlamberger Brezar (2005: 127) pa vidi rabo deležja kot oddaljeno od naravne, zlasti govorne rabe, ter jo povezuje s pridihom arhaičnosti. Dejansko pa prepričanje, da se deležje dandanes ne uporablja (naj ne bi uporabljalo?), zaenkrat še ni dobilo dokončne znanstvene potrditve, saj se obstoječe študije osredotočajo bodisi na zgodovinski vidik (Jesenšek 1998) bodisi na posamezne besedilne tipe (prim. naslednji razdelek).

Z vidika zastopanosti določenega jezikovnega elementa v jeziku nasploh ali v posameznih žanrih oz. področjih, se korpusi samoumevno ponujajo kot odlično raziskovalno orodje. Slovenščina je v tem smislu privilegirana, saj obstaja že cela vrsta prosto dostopnih korpusov in orodij za gradnjo lastnih korpusov, ki raziskovalcem omogočajo relativno hitro in enostavno pregledovanje velikih količin podatkov.

Tako želimo v pričujoči raziskavi na podlagi različnih korpusov preveriti, koliko in kako se deležje pojavlja v slovenskih pisnih besedilih, in sicer zlasti v določenih neleposlovnih žanrih v primerjavi s splošnim jezikom oz. z leposlovjem. Pričakujemo, da se deležje v nobenem besedilnem tipu ne bo pojavljalo pogosto; po drugi strani pa pričakujemo tudi, da se bodo med različnimi besedilnimi tipi pojavljale konkretne razlike, saj predhodne raziskave (prim. Mikolič Južnič, v tisku) nakazujejo, da je pojavljanje deležja do določene mere odvisno od besedilnega tipa. Poleg tega skozi raziskavo nameravamo opazovati, kako natančno so korpusi oblikoskladenjsko označeni oz. ali so v izbranih korpusih, označenih večinoma z avtomatskimi orodji, deležja pravilno kategorizirana.

2. Slovenska deležja

Deležje ima v slovenskem jeziku dvojno naravo: je glagolska oblika, ki meji na prislov. V skladu s Toporišičem (1991: 339), deležja na *-č* in *-aje* izražajo »glagolsko dejanje, ki se vrši hkrati s kakim drugim dejanjem«, deležje na *-e* pa se »večinoma rabi kot prislov načina«. Deležijski polstavek, ki deležje vsebuje, »izraža glagolsko dejanje, ki poteka hkrati z drugim dejanjem« (Žagar 2001: 291). Jesenšek (1998: 41) navaja, da deležja najdemo zlasti v umetnostnih in strokovnih besedilih, vendar pa da njihova raba upada, če pa se pojavljajo, izražajo istodobnost dogajanja z glagolskim dogodkom v glavnem stavku. Po drugi strani Mezeg (2011: 323) ugotavlja, da se deležja na *-č* pojavljajo v nezanemarljivem številu tako v publicističnih kot tudi (še pogosteje) v literarnih prevodih iz francoščine, Schlamberger Brezar (2005) pa ugotavlja povišano prisotnost deležijskih stavkov v prevodih političnih besedilih iz francoščine.

Deležja prepoznamo po značilnih obrazilih *-oč* (npr. *čakajoč*, primer 1), *-eč* (*hodeč*, primer 2), *-aje* (*upoštevaje*, primer 3) in *-(v)ši* (*začenši,* primer 4, *vštevši*, primer 5), poleg teh pa tudi *-e* (*miže, smeje, sede*, primeri 6, 7, 8), vendar so taka deležja pogosto kategorizirana kot prislovi z glagolsko etimologijo.[1]

(1) *Čakajoč*, da obiskovalci opravijo svoj ritual …
(2) Modno pisto sem zapustil *hodeč* nazaj …
(3) *Upoštevaje* okoliščine, je bil jedilni kot …
(4) … bil pred tem, *začenši* z letom 1961, sekretar …
(5) … Jackson pa je, *vštevši* še chicaške uspehe, nanizal …
(6) Časopis bi lahko delila *miže* …
(7) … žensko, ki je *smeje* se zapuščala prizorišče …

[1] Navedeni primeri so bili naključno izbrani iz korpusa Fidaplus.

(8) …je življenje *sedé*, brez hoje in brez dela ali športa nezdravo, …

V pričujočo raziskavo so vključene vse navedene oblike, saj nas po eni strani zanima celovit pregled rabe deležja v obravnavanih besedilnih tipih, po drugi strani pa je dejansko praktično vse primere takih 'glagolskih' prislovov mogoče pretvoriti v osebne glagolske oblike, tako kot to lahko naredimo za 'prava' deležja, kar je dodaten dokaz, da gre v bistvu za glagolske oblike in ne prislove.[2]

Iz omenjenih raziskav lahko sklepamo, da je raba deležja in njegova pogostnost do določene mere odvisna od besedilnega tipa, s katerim imamo opravka, zato želimo v pričujoči raziskavi s pomočjo korpusov, opisanih v naslednjem razdelku, ter s primerjavo s podatki, pridobljenimi v predhodni raziskavi (Mikolič Južnič, v tisku), ugotoviti, ali naši podatki dejansko potrjujejo to odvisnost.

3. Korpusi in metode

3.1. Korpusi

Ker želimo v raziskavo vključiti različne besedilne tipe ter predstaviti, kako se raba deležja spreminja v nekaterih različicah slovenskega jezika, se kot samoumevni vir ponuja korpus Fidaplus, ki je uravnotežen, izredno zajeten referenčni korpus (621 milijonov besed) in do neke mere omogoča izbiro prenosnika in zvrsti besedil, ki jih vključuje. Vendar pa ta korpus prinaša določene težave, ki jih v okviru te raziskave ni bilo mogoče rešiti: po eni strani so podatki o zvrstnosti besedil oz. s tem povezane možnosti izbire podkorpusov preveč omejene, da bi bile dejansko uporabne za natančnejše analize; po drugi strani v korpusu deležja niso označena s svojo lastno kodo MSD (prim. Arhar 2007), kar bistveno oteži luščenje primerov, saj druge metode (podobne tistim, opisanim v naslednjem razdelku) prinašajo preveliko število šuma (oz. nezaželenih zadetkov) in s tem za ročno analizo neobvladljivo število primerov.

Zaradi navedenih razlogov smo se odločili, da uporabimo drug korpus (ali bolje korpuse). Kot kontrolni korpus splošne slovenščine smo uporabili JOS 100K, korpus s 100.000 pojavnicami, ki vsebuje vzorce referenčnega korpusa Fidaplus s podrobno ročno preverjenimi jezikoslovnimi oznakami. Poleg tega smo uporabili specializirani korpus znanstvenih in strokovnih besedil, ki je nastal v sklopu predhodne raziskave o pojavljanju nominalizacije v izbranih slovenskih besedilnih tipih (Mikolič Južnič 2011). Korpus skupno šteje približno 870.000 pojavnic ter združuje znanstvena besedila s področja humanistike in družboslovja na eni ter naravoslovja na drugi strani in strokovna besedila s področja slovenske zakonodaje. Podrobna sestava korpusa je navedena v Tabeli 1. Korpus je tudi jezikoslovno označen, in sicer s pomočjo spletnega servisa projekta JOS (Erjavec idr. 2010) oz. orodjem ToTaLe.

Podatke, pridobljene po metodah, opisanih v nadaljevanju, smo primerjali tudi s podatki iz predhodne raziskave (Mikolič Južnič, v tisku), opravljene na osnovi korpusa Spook (Vintar 2009, v tisku), ki združuje izvirna ter prevedena slovenska leposlovna besedila ter izvirnike danih prevodov v štirih jezikih (angleščini, francoščini, nemščini in italijanščini).[3] Na ta način smo želeli omogočiti širši vpogled v rabo deležja v slovenskem jeziku.

3.2. Metode

Po izboru korpusov je bil prvi korak analize luščenje primerov rabe deležja. Korpus JOS 100K omogoča ustvarjanje konkordanc v lastnem spletnem iskalniku, za brskanje po specializiranem korpusu znanstvenih in strokovnih besedil pa smo uporabili Sketch Engine (Krek in Kilgarriff 2006), pri čemer smo predhodno korpus oblikoskladenjsko označili, kot smo omenili zgoraj.

Ker sta izbrana korpusa torej označena z jezikovnimi informacijami (npr. glede besedne vrste), v okviru katerih imajo deležja svojo specifično kodo, smo najprej izluščili primere z iskanjem te kode. A ker pri avtomatskem označevanju včasih lahko prihaja tudi do napak (narobe označenih besed, in sicer bodisi besed, ki jim je koda za deležje napačno pripisana, bodisi takih, ki deležja so, vendar niso označena kot taka), smo podatke želeli preveriti tudi s kombinirano avtomatsko in ročno metodo, oz. z iskanjem po obrazilih. Tako smo preverili prisotnost pripon *-eč, -oč, -aje, -ši,* poleg tega pa tudi *-če, -de, -te* in *-eje*.[4]

Ko smo avtomatsko pridobili vse primere, v katerih se pojavljajo dane pripone, je bilo treba odstraniti šum, tj. vse tiste primere, kjer se naključne besede končajo z identičnimi črkami kot naša obrazila, vendar ne gre za deležja (npr. *več* za *-eč*). Tega dela smo se lotili v dveh fazah: v prvi smo odstranili vse naključno podobne besede, nato pa posebej razločili še med deležji in deležniki, saj imata slednji dve kategoriji pogosto identični obliki, ki seveda nista uporabljeni v enakih okoliščinah. Na ta način smo dobili seznam praktično vseh deležij, ki se pojavljajo v danih besedilih.[5]

Dobljene rezultate smo končno primerjali tudi s predhodno raziskavo o deležjih v literarnih (izvirnih in prevedenih) besedilih in podatki, ki so bili pridobljeni s

[2] Izjema je beseda *glede* (npr. *Prečne pregrade razporedimo glede na izvedbo notranjosti vlaka;* vir: Fidaplus). Čeprav bi pogosto v primerih, ki vsebujejo *glede*, lahko besedno zvezo prav tako pretvorili v odvisni stavek, to ni vedno mogoče oz. smiselno. Poleg tega ima beseda izjemno veliko pogostnost (preko 100.000 primerov v Fidaplus) in bi izkrivila sliko o splošni rabi deležja v jeziku.

[3] Dejansko so bili v dani raziskavi bili uporabljeni samo prevodi iz italijanskega jezika.

[4] Čeprav se določena obrazila v obravnavanih korpusih pravzaprav ne pojavijo niti enkrat (prim. razdelek 4.), jih vseeno navajamo, in sicer iz dveh razlogov: a) ker s tvorbenega in slovarskega vidika kot možne oblike obstajajo in smo njihovo prisotnost zato preverjali; in b) ker smo želeli doseči čim večjo primerljivost z rezultati, pridobljenimi iz literarnih besedil (prim. razdelek 4.3.).

[5] Mogoče je, da se je kako posamezno deležje izmuznilo, saj zaradi neizmerno velikega šuma, ki bi se pojavil ob iskanju z obrazilom *–e*, nismo pregledali vseh mogočih kombinacij deležij na *-e*, temveč le določene, ki so se predhodno izkazale za bolj pogoste, vendar s precejšnjo mero gotovosti lahko trdimo, da smo jih zajeli zelo veliko večino (tudi z ozirom na dejstvo, da med deležji, ki smo jih izluščili s pomočjo kod MSD, praktično ni takih, ki jih ne bi zajeli z obrazili).

Vrsta besedila	Izbrana področja (in viri)	Število pojavnic
Znanstvena besedila	Humanistika in družboslovje	415.000
	Slavistična revija	81.399
	Sodobna pedagogika	74.355
	Revija za kriminalistiko in kriminologijo	109.788
	Geografski vestnik	53.611
	Etnolog	95.847
	Naravoslovje	304.744
	Zobozdravstveni vestnik	55.456
	Medicinski razgledi	72.178
	Acta biologica slovenica	52.780
	Geodetski vestnik	55.364
	Farmacevtski vestnik	68.966
Strokovna besedila	Slovenska zakonodaja	154.088
	Stanovanjski zakon SZ-1	
	Zakon o društvih ZDRu-1	
	Zakon o verski svobodi ZVS	
	Zakon o volilni in referendumski kampanji ZVRK	
	Zakon o javnih zbiranjih ZJZ	
	Zakon o splošnem upravnem postopku ZUP	
	Zakon o upravnih taksah ZUT	
	Kazenski zakonik KZ-1	
	SKUPAJ POJAVNIC	873.832

Tabela 1: Podrobna zgradba specializiranega korpusa znanstvenih in strokovnih besedil ter število pojavnic v posameznem podkorpusu (povzeto po Mikolič Južnič 2011: 323).

pomočjo korpusa Spook, kot smo že omenili (Mikolič Južnič, v tisku). Ker je bila ta raziskava izvedena z enako metodologijo in smo rezultate normalizirali na 500.000 pojavnic, so rezultati obeh analiz primerljivi (čeprav je seveda normaliziranje navzgor pri tako majhnih številkah do neke mere tvegano. Sicer pa gre v Mikolič Južnič (v tisku) za širšo analizo, ki upošteva tudi določene aspekte z vidika prevajanja, zato smo povzeli le del podatkov, ki je relevanten za pričujočo analizo (navedeni so v razdelku 4.3.)

4. Rezultati in diskusija

4.1. Iskanje s kodami MSD

Tako spletni konkordančnik korpusa JOS 100K kot tudi Sketch Engine omogočata iskanje po oblikoskladenjskih oznakah besed v korpusih. Ker je bila pri korpusu JOS 100K uporabljena tabela oznak JOS in je bil specializirani korpus, ki smo ga uporabili, označen z orodjem ToTaLe, je bilo iskanje izvedeno z enakim iskalnim pogojem. Rezultati te analize so povzeti v Tabeli 2.

Podatki o pojavitvah deležij, zbrani po kodah MSD, so zaradi različne velikosti korpusov in lažje primerljivosti preračunani na 500.000 pojavnic. Na prvi pogled lahko opazimo, da so razlike med posameznimi korpusi precejšnje, vendar se bomo temu podrobneje posvetili v nadaljevanju, ko bomo pregledali pojavitve tudi po posameznih obrazilih. Pred tem nas je zanimalo tudi, katera obrazila se pojavljajo pri zgoraj navedenih deležjih: ti podatki so zbrani v Tabeli 3, v kateri smo prav tako v oklepajih navedli normalizirane podatke na 500.000 pojavnic.

Korpus		**Absolutno št. deležij**	**Št. deležij na 500.000 pojavnic**
JOS 100K		18	90
Znanstvena besedila		58	40,3
	Humanistika/ družboslovje	*43*	*29,9*
	Naravoslovje	*15*	*10,4*
Strokovna besedila	Zakonodaja	4	13,0

Tabela 2: Pojavitve deležja v treh obravnavanih besedilnih tipih (iskanje po kodah MSD)

		Znanstvena besedila		**Strokovna besedila**
	JOS 100K	**Humanistika/ družboslovje**	**Naravoslovje**	**Zakonodaja**
-eč	2 (10)	5 (3,5)	3 (2,1)	1 (3,2)
-oč	9 (45)	31 (21,5)	10 (6,9)	3 (9,7)
-aje	2 (10)	3 (2,1)	0	0
-ši	1 (5)	3 (2,1)	2 (1,4)	0
-če	3 (15)	0	0	0
-te	0	1 (0,7)	0	0
-de	1 (5)	0	0	0
-eje	0	0	0	0

Tabela 3: Pojavitve posameznih obrazil ob iskanju s kodami MSD (podatki, preračunani na 500.000 pojavnic, v oklepaju)

Preden se lotimo interpretiranja podatkov, ki smo jih pridobili, si oglejmo še rezultate iskanja s posameznimi obrazili.

4.2. Iskanje z obrazili

Iskanje z obrazili je seveda malo bolj zamudno zaradi nujnega ročnega pregledovanja primerov, vendar so rezultati pogosto vredni dodatnega truda.

Preverili smo vsa zgoraj navedena obrazila (*-ečˊ, -očˊ, -aje, -ši, -če, -de, -te* in *-eje*), končne rezultate pa strnili v tabeli 4, v kateri so v zadnjem stolpcu navedeni tudi rezultati, preračunani na 500.000 pojavnic.

Korpus		**Absolutno št. deležij**	**Št. deležij na 500.000 pojavnic**
JOS 100K		23	115
Znanstvena besedila		83	57,7
	Humanistika/ družboslovje	*61*	*42,4*
	Naravoslovje	*22*	*15,3*
Strokovna besedila	Zakonodaja	15	46,7

Tabela 4: Pojavitve deležja v treh obravnavanih besedilnih tipih (iskanje po obrazilih)

Če primerjamo Tabelo 4 s Tabelo 2, takoj opazimo, da podatki ne sovpadajo popolnoma, oziroma da so številke v Tabeli 4 višje. Pred natančnejšo diskusijo si lahko ogledamo še podrobnejšo sliko za posamezna obrazila, ki jo navajamo v Tabeli 5, in sicer v absolutnih ter normaliziranih številkah, preračunanih na 500.000 pojavnic (v oklepaju).

		Znanstvena besedila		**Strokovna besedila**
	JOS 100K	**Humanistika/ družboslovje**	**Naravoslovje**	**Zakonodaja**
-ečˊ	3 (15)	6 (4,2)	3 (2,1)	1 (3,2)
-očˊ	9 (45)	43 (29,8)	15 (10,4)	5 (16,2)
-aje	2 (10)	3 (2,1)	0	1 (3,2)
-ši	1 (5)	2 (1,4)	2 (1,4)	0
-če	8 (40)	4 (2,8)	2 (1,4)	8 (26,0)
-te	0	3 (2,1)	0	0
-de	1 (5)	0	0	0
-eje	0	0	0	0

Tabela 5: Pojavitve posameznih obrazil ob iskanju z obrazili (podatki, preračunani na 500.000 pojavnic, v oklepaju)

4.2.1. Primerjava med iskanjem s kodami MSD in z obrazili

V zgornjih tabelah navedeni podatki nam lahko odgovorijo na različna vprašanja. Če se najprej osredotočimo na primerjavo med pridobivanjem podatkov s kodami MSD in z obrazili, vidimo, da prihaja do konkretnih razlik med obema metodama analize: v vseh primerih je pri analizi s kodami MSD prišlo do izgube podatkov ali, z drugimi besedami, vsa dejansko prisotna deležja niso bila kategorizirana kot taka. Najmanjša razlika se pojavlja pri korpusu JOS 100K, kjer kode MSD niso zajele 11,7 odstotkov najdenih deležij. Pri znanstvenih besedilih je bilo napačno označenih 30 odstotkov deležij, pri strokovnih pravnih besedilih pa celo 73,3 odstotkov. Tabeli 3 in 5 nam kažeta, kako do razlik prihaja tako pri enem in istem obrazilu (pri pravnih besedilih, na primer, kode MSD označujejo tri primere na *-očˊ,* medtem ko jih je prisotnih pet. Pravilno je na primer kategorizirano deležje v primeru (9), napačno pa v primeru (10).[6]

(9) … šest mesecev in ne daljši od dveh let, *računajočˊ* od dne pravnomočnosti sodbe.
(10) … kadar v upravnih stvareh, neposredno *uporabljajočˊ* predpise, odločajo o pravicah …

Razlogi za napačno označevanje ležijo gotovo v skladenjskem okolju deležij, ki so navadno zaradi bližine besede, kategorizirane kot samostalnik, zmotno označena kot deležniki (seveda do napak prihaja skoraj vedno ravno tam, kjer so oblike deležnika in deležja identične). Sklepamo torej, da ima uporabljeni sistem za označevanje določene sistemske težave z razdvoumljanjem med tema dvema glagolskima oblikama, ki se lahko pojavljata v na videz podobnih skladenjskih strukturah.

Zaključimo lahko, da nam avtomatska analiza deležij s pomočjo kod MSD sicer lahko nudi podatke o tem, ali so deležja prisotna v določenem korpusu besedil ali ne, vendar za natančnejše kvantitativne analize podatki žal niso dovolj zanesljivi.

4.2.2. Primerjava pojavljanja deležja v obravnavnih besedilnih tipih

V nadaljevanju se vrnimo na osnovno vprašanje, ki ga obravnavamo v pričujočem prispevku, pojavljanje deležja v izbranih besedilnih tipih. Pri tem se bomo oprli na podatke v Tabelah 4 in 5, ki zajemata praktično celotno število deležij, prisotnih v analiziranih korpusih, in bomo opazovali normirane podatke na 500.000 pojavnic.

Največ deležij je zabeleženih v korpusu splošne slovenščine JOS 100K (115 primerov na 500.000 pojavnic); čeprav gre podatke jemati z dozo rezerve zaradi majhnega obsega korpusa, vsekakor nudijo osnovno idejo o pogostnosti deležja v splošnem (pisnem) jeziku. V ostalih dveh besedilnih tipih je deležij bistveno manj (pol manj v znanstvenih besedilih in celo 60 odstotkov manj v strokovnih pravnih besedilih).

Zanimiva je tudi velika razlika med obema analiziranima tipoma znanstvenih besedil: veliko bolj pogosto se deležje namreč pojavlja v humanističnih in družboslovnih znanstvenih prispevkih (42,4 primerov na 500.000 pojavnic) kot v naravoslovnih (15,3 primerov na 500.000 pojavnic). Očitno gre pri zadnjem tipu besedil skorajda za izogibanje tej obliki: skoraj vsi obravnavani primeri imajo obrazilo *-očˊ* in večinoma gre za primere, v katerih so uporabljene ustaljene fraze, ko vidimo v

[6] Oba primera sta vzeta iz korpusa strokovnih pravnih besedil.

primerih (11) in (12), ne pa za deležja v njihovi klasični glagolski funkciji v polstavkih, kot v primeru (13).

(11) V preteklih desetih letih je bil tako *rekoč* v vseh državah v regiji …
(12) Za Slovensko populacijo imamo, *zahvaljujoč* že omenjenim raziskavam …
(13) *Zavedajoč* se problema, mnogi iščejo ustrezne …

V družboslovnih in humanističnih besedilih je raznolikost uporabljenih deležij večja; čeprav tudi pri teh izrazito prevladujejo deležja na *-oč*. Najdemo primere z vsemi obrazili razen *-de* in *-eje*. In če so med primeri z obrazilom *-oč* v naravoslovnih besedilih samo štirje primeri od 15 taki, kjer deležje ni ustaljena fraza, temveč živa oblika, je takih deležij v humanističnih in družboslovnih besedil več kot polovica (23 primerov od 43 deležij na *-oč*). Torej ne gre samo za razliko v pogostnosti, marveč tudi v raznovrstnosti deležij.

V strokovnih pravnih besedilih kljub temu, da se na splošno deležja precej redko pojavljajo (46,7 primerov na 500.000 pojavnic), ne najdemo ustaljenih fraz, podobnih zgoraj navedenim, temveč izključno različna deležja v 'pravih' polstavkih.

Tudi v korpusu splošnih besedil JOS 100K je opazna določena raznolikost, tako glede prisotnih obrazil kot tudi glede posameznih deležij, ki se pojavljajo.

Če podatke pogledamo še transverzalno, opazimo, da se med obrazili skoraj povsod na prvem mestu pojavlja *-oč* (razen pri strokovnih pravnih besedilih, kjer je na drugem mestu, za *-če*), ostala obrazila pa se (spet z izjemo pravnih besedil in tokrat tudi splošne slovenščine) pojavljajo neprimerno redkeje, če sploh.

Mogoče je, da se priponi *-oč* in *-če* na splošno najpogosteje pojavljata tudi zaradi podobnosti oblike z deležniki, ki se vsekakor zdijo veliko bolj prisotni in uporabni/uporabljeni od deležij:[7] prisotnost in sprejemljivost enake formalne oblike bi lahko pogojevala večjo sprejemljivost druge, vsaj v določenih besedilnih tipih.

Zbrani podatki torej nakazujejo, da se deležje precej različno uporablja v različnih besedilnih tipih in da je pravzaprav veliko pogostejše v splošnem pisnem jeziku kot v znanstvenem ali strokovnem pravnem jeziku, v nasprotju z nekaterimi teoretičnimi predpostavkami, izraženimi v uvodnih delih prispevka. Poleg tega se izkazuje tudi razlika med dvema analiziranima tipoma znanstvenih besedil, saj je v našem humanistično-družboslovnem podkorpusu skoraj trikrat več deležij kot v naravoslovnem. Besedila s teh dveh področij se gotovo zelo razlikujejo po slogu in načinu argumentiranja, zato ta razlika ni posebno presenetljiva. Pravzaprav se celo odlično ujema s podatki o deležjih v literarnih besedilih, ki jih navajamo v nadaljevanju.

[7] Dejansko o tem, kolikor je znano, ne obstaja nobena študija, gre preprosto za kombinacijo površinskega opazovanja korpusov in osebnega intuitivnega mnenja. Sicer pa je primerjava z deležniki omenjena tako zaradi površinske podobnosti oblik kot tudi zaradi dejstva, da gre v obeh primerih za neosebne glagolske oblike, čeprav je njihova pogostnost v jeziku seveda medsebojno neodvisna.

4.3. Primerjava pojavljanja deležja v analiziranih neliterarnih korpusih in v literarnih besedilih

Za konec smo dobljene rezultate primerjali s podobno raziskavo, izvedeno na korpusu izvirnih in prevedenih literarnih besedil Spook (Mikolič Južnič, v tisku). Kot omenjeno zgoraj, gre za raziskavo, ki se posveča tudi vplivom izvirnega jezika (v tem primeru italijanščine) na jezik literarnih prevodov, vendar se na tem mestu omejujemo le na pojavljanje deležja v slovenskih besedilih kot takih.

Izkaže se, da je število deležij v literarnih besedilih, tako v izvirnih kot v prevedenih, na splošno veliko večje kot v zgoraj omenjenih splošnih, znanstvenih ali strokovnih besedilih. Podrobni podatki o številu pojavnic v uporabljenih korpusih in o najdenih deležjih so podani v Tabeli 6.

	Izvirna literarna besedila	**Prevedena literarna besedila**
Št. pojavnic	*1.454.275*	*478.591*
-oč	337 (115,9)	35 (36,6)
-eč	137 (47,1)	34 (35,5)
-aje	70 (24,1)	14 (14,6)
-ši	4 (1,4)	0
-če	401 (137,9)	87 (90,9)
-de	345 (118,6)	47 (49,1)
-te	52 (17,9)	7 (7,3)
-eje	4 (1,4)	6 (6,3)
Skupno	1.350 (464,2)	230 (240,3)

Tabela 6: Prisotnost deležja v izvirnih in prevedenih literarnih besedilih (povzeto po Mikolič Južnič, v tisku; v oklepajih so rezultati, normalizirani na 500.000 pojavnic)

Če vzamemo povprečje pojavljanja deležja v vseh obravnavanih literarnih besedilih, se številka giblje okoli 352,25 primerov na 500.000 pojavnic, kar je več kot trikrat toliko kot v splošnih besedilih korpusa JOS 100K, kjer je bila pogostnost deležja v naši raziskavi največja. Pomenljivo je dejstvo, da se deležje pogosteje (skoraj dvakrat pogosteje) pojavlja v izvirnih literarnih besedilih kot v prevedenih delih, in to kljub dejstvu, da je italijanščina bogata z neosebnimi glagolskimi strukturami in bi lahko s tega vidika pričakovali kako interferenco v prid povečanega števila deležij (kot npr. opaža Schlamberger Brezar 2005 za prevode političnih besedil iz francoščine), vendar podatki tega ne potrjujejo.[8]

Razlog za tako izrazito različno pojavnost morda lahko iščemo v odnosu do deležja, ki ga imajo govorci slovenščine. O tem, koliko je deležje prisotno v spontanem govoru oz. drugih, s pravili manj obremenjenih besedilnih tipih, ni podatkov, vendar smo govorci večinoma vsi zrasli z idejami, ki jih širijo obstoječe slovnice (npr. Toporišič 1991), o tem, da je deležje zastarelo, nerodno, malo uporabno, tako da ni nič kaj

[8] Smotrno bi bilo preveriti, ali se podobno dogaja tudi v literarnih prevodih iz drugih jezikov, vendar naj bo to predmet prihodnjega raziskovanja.

presenetljivo, da se mu pri tvorjenju besedil marsikdo zavestno ali podzavestno izogiba. Le pri kreativnem pisanju – literaturi – si tvorci besedil očitno suvereno upajo posegati po tem slovničnem orodju slovenščine. Zdi se, da bolj kot je besedilo oddaljeno od te kreativnosti, bolj se deležje izgublja: v slogu humanistično-družboslovnih znanstvenih besedil, na primer, pogosto lahko zasledimo splošne ambicije po leporečju in posledično poseganje tudi po nevsakdanjih jezikovnih oblikah (v našem primeru deležjih), redkeje pa so take težnje prisotne v naravoslovnih ali strokovnih pravnih besedilih. Po drugi strani se prevajalci pri prevajanju literarnih besedil morajo soočiti s številnimi težavami, ne nazadnje z izogibanjem negativnemu transferu in s tem povezani morebitni posledični hiperkorektnosti, zaradi katere je število deležij dejansko bistveno manjše kot v izvirni slovenski literaturi. Tako se literarna dela, prevedena iz italijanščine, dejansko po številu prisotnih deležij postavljajo nekako vmes med splošni in literarni jezik ter na ta način deloma sledijo tradicionalnemu slovničarskemu videnju deležja, deloma se prilagajajo živemu vplivu slovenske literature, deloma pa verjetno tudi drugim literaturam, kjer so podobne slovnične strukture pogostejše kot v našem jeziku.

5. Sklep

Deležje se je izkazalo kot zanimiva tema za korpusno raziskovanje, kljub temu da se samodejno označevanje besedne vrste v tem primeru ni najbolje obneslo, saj je bilo iskanje po obrazilih kot preprostih zaporedjih črk precej bolj uspešno.

Raziskava je ponudila zanimiv vpogled v pojavljanje deležja v različnih besedilnih tipih slovenskega jezika. Tudi če gre precejšen del pojavitev deležja pripisati stalnim besednim zvezam, pri katerih ni prisoten pridih starinskosti, se namreč izkaže, da se ta glagolska oblika v jeziku ne pojavlja enakomerno, marveč se zdi vezana na tiste žanre, kjer avtorji čutijo večjo kreativno svobodo in jezikovno samozavest, to je v izvirnih literarnih besedilih. Besedila, v katerih se tega elementa kreativnosti ne pričakuje, kot so na primer znanstvena besedila s področja naravoslovja ali strokovna pravna besedila, vsebujejo izrazito manj deležij (tudi 30-krat manj) od literarnih besedil.

Veliko vprašanj glede deležja ostaja še nerešenih in torej odprtih za nadaljnje raziskovanje: besedilnih tipov, v katerih bi lahko iskali prisotnost deležja, ostaja še mnogo, na prvem mestu pa so zagotovo različne tipologije govorjenega jezika; poleg tega je nujno tudi preverjanje, kako se deležje pojavlja v različnih tipih literarnih besedil in ali prihaja do razlik pri prevodih iz različnih tujih jezikov.

Kar zadeva označevanje slovenskih besedil s kodami MSD pa bi bilo prav glede razdvoumljanja med deležji in deležniki potrebno vložiti še precej truda, preden bi korpusi omogočali dovolj zanesljivo samodejno iskanje tovrstnih elementov.

6. Literatura

Arhar, Š. (2007). *Kaj početi z referenčnim korpusom Fidaplus.* Ljubljana: Filozofska fakulteta. http://www.fidaplus.net/Files/Kaj%20poceti%20s%20korpusom%20FidaPLUS_navadna%20dvostranska.pdf, dostop 31.07. 2012.

Erjavec, T., D. Fišer, S. Krek, in N. Ledinek (2010). The JOS Linguistically Tagged Corpus of Slovene. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). Malta.* http://www.lrec-conf.org/lrec2010/, dostop 31.07.2012. 1806–1809.

Fidaplus. Referenčni korpus slovenskega jezika. http://www.fidaplus.net/, dostop 31.07.2012.

Jesenšek, M. (1998). *Deležniki in deležja na -č in -ši. Razširjenost oblik v slovenskem knjižnem jeziku 19. stoletja.* Maribor: Slavistično društvo Maribor.

Korpus JOS 100K. http://nl.ijs.si/jos/, dostop 31.07.2012.

Krek, S. in A. Kilgarriff (2006). Slovene Word Sketches. V T. Erjavec in J. Žganec Gros (ur.), *Zbornik 5. slovenske in 1. mednarodne konference Jezikovne tehnologije 2006*. Ljubljana: Institut Jožef Štefan. 62-67

Mezeg, A. (2011). *Korpusno podprta analiza francoskih polstavkov in njihovih prevedkov v slovenščino.* Doktorska disertacija, Univerza v Ljubljani.

Mikolič Južnič, T. (2011). Vpliv besedilnih tipov na pojavljanje nominalizacije v slovenščini: korpusna raziskava. V S. Kranjc (ur.), *Meddisciplinarnost v slovenistiki*. Ljubljana: Znanstvena založba Filozofske fakultete. 321-327.

Mikolič Južnič, T. (v tisku). Neosebne glagolske oblike v prevodni in izvirni slovenščini: primer deležja. V Š. Vintar (ur.), *Slovenski prevodi skozi korpusno prizmo*. Ljubljana: Znanstvena založba Filozofske fakultete.

Schlamberger Brezar, M. (2005). Politična besedila kot tip besedil in postopki prevajanja stalnih formul. V N. Kocijančič Pokorn, E. Prunch in A. Riccardi (ur.), *Beyond Equivalence – Jenseits der Äquivalenz – Oltre l'equivalenza – Onkraj ekvivalence*. Gradec: Institut für Theoretische und Angewandte Translationswissenschaft. 121-135.

Sketch Engine. http://www.sketchengine.co.uk/, dostop 31.07.2012.

Toporišič, J. (1991). *Slovenska slovnica.* Maribor: Obzorja.

Vintar, Š. (2009). Slovenski prevodoslovni korpus. V M. Stabej (ur.), *Infrastruktura slovenščine in slovenistike*. Ljubljana: Znanstvena založba Filozofske fakultete. 385-391

Vintar, Š. (ur.) (v tisku). *Slovenski prevodi skozi korpusno prizmo*. Ljubljana: Znanstvena založba Filozofske fakultete.

Žagar, F. (2001). *Slovenska slovnica in vadnica*. Maribor: Obzorja.

Identifying Fear Related Content in Croatian Texts

Lucia Načinović*, Benedikt Perak† Ana Meštrović*, Sanda Martinčić-Ipšić*

* Department of Informatics, University of Rijeka,
Omladinska 14, 51000 Rijeka, Croatia
{lnacinovic, amestrovic, smarti}@inf.uniri.hr

† Department of Cultural Studies, University of Rijeka,
Slavka Krautzeka bb, 51000 Rijeka, Croatia
bperak@ffri.hr

Abstract

This paper presents the initial work for the task of identifying the emotion of FEAR in texts written in the Croatian language. For the purpose of this analysis, text articles, blogs and online comments were collected from specific Croatian websites and classified into two categories: "*Fear present*" and "*Fear not present*". In the process of classification, supervised machine learning method based on Naïve Bayes algorithm was used. We experimented with different sets of features to compare their impact on the accuracies of the learnt classifiers and to determine which set of features yields better result (accuracy). The first set of features was constructed on a semantic model of embodied metonymic domains of fear derived from the cognitive semantic study of metonymic constructions of fear in Croatian. The second set of features contained only lexical concepts of fear, its synonyms and direct hyponyms. The third set is a combination of both sets. The results are presented in terms of accuracies, indicating that both embodied metonymic and lexically expressed features can be relevant for the emotion identification in texts.

Identifikacija s strahom povezanih vsebin v hrvaških besedilih

V članku je predstavljena začetno delo pri nalogi identifikacije čustva " strah" v besedilih, zapisanih v hrvaškem jeziku. Za namen te analize smo s hrvaških spletnih strani zbrali članke, bloge in spletne komentarje ter jih razvrščali v dve kategoriji: "Prisotnost strahu" in "Odsotnost strahu". Pri razvrščanju smo uporabili nadzorovani postopek strojnega učenja, ki temelji na naivnem Bayesu. Preizkušali smo različne nabore značilk in domen in opazovali njihov vpliv na natančnost naučenih razvrščevalnikov, da bi določili nabor značilk, ki daje najboljše rezultate razvrščanja. Prvi nabor značilk smo razvili na podlagi semantičnega modela vgrajenih metonimičnih domen strahu, ki izvirajo iz kognitivne semantične raziskave strahu v hrvaščini. Drugi nabor značilk je vseboval le leksikalne koncepte za strah, njegove sinonime in hiponime. Tretji nabor značilk pa je združeval oba prejšnja nabora. Dobljeni rezultati razvrščanja izraženi z mero pravilnosti, kažejo da sta obe metonimično in leksično zasnovane značilkerelevantne za identifikacijo čustev v besedilih.

1. Introduction

This paper presents research in the field of sentiment analysis. Sentiment analysis is a computer aided process of identifying different affective states within a particular segment of specific text corpora.

From the epistemological standpoint it can be said that sentiment analysis of a particular emotional category is a lost cause, for how can a machine detect emotions of another when it does not have emotions of its own? However, from the perspective of cognitive science one could argue that the process of emotional analysis in humans is not that dissimilar from those used in computers. After all, we as humans learn to acquire different emotional words for affective states, establishing connections between symbolic labels and our psychological, physiological states and behavioural traits (Davidson et al., 2003; Lewis et al., 2008). Furthermore, we learn how to express them appropriately in communication and elicit those states in others (Fussell, 2002; Barrett et al., 2007).

Categorization and meaning of emotions is complex and dynamic informational process emerging from the interaction of neurobiological, cognitive and symbolic structures (Barrett, 2011; Damasio, 1999). With so much lacking in comparison to human neurobiological structure, it would be irrational to demand from machines (on this level of technology) to *feel* emotions or to *recognize* emotional categories. On the other hand, to *analyze* and *identify* emotional categories on the basis of cognitive networks expressed in the linguistic symbolic structures is a feasible task. Analysis of emotional content in texts can therefore be reduced to the identification of emotional conceptual schemas. The need for conceptual organization of emotions is necessary because the epistemological nature of affective phenomena is highly individual. Therefore, the embodied feeling of a certain emotion is always purely subjective: no one can feel emotion of the other. Insuring incommensurability of affective experience enables cultural relativity in emotional categorization and lexicalization (Wierzbicka, 1999; Boster, 2005).

According to the FrameNet Project (Baker et al., 1998), based on Fillmore's frame semantics, core frames of the lexical concept FEAR are: (1) Experiencer - a person or a sentient entity that experiences or feels emotions. (2) Expressor - a body part, gesture, or other expression of the Experiencer that reflects his or her emotional state. (3) State - an abstract noun that describes a more lasting experience by the Experiencer. (4) Stimulus - a person, event, or state of affairs that evokes emotional response in the Experiencer. (5) Topic - a general area in which the emotion occurs.

Using theoretical framework of Cognitive Semantics, we modelled the identification of emotional category FEAR with reference to the embodied Expressor frame and its related metonymic domains, as well as using related lexical concepts of the State frame (Perak, 2011). One of the motivations for this study was to see whether embodied metonymic domains of fear could be relevant features for the identification of fear related content.

Related work on emotion (fear) identification was reported in many recent papers. In (Strapparava & Mihalcea, 2008) the identification of all major emotions from news headlines and blogs was based on WordNet-Affect. Blogs were also used in work by (Aman & Szpakowicz, 2007). Mohammad (Mohammad, 2012) reported on fear/no fear classification of each sentence in the newspaper headlines and blogs comparing the ngram and emotion based lexicon features. Tao (Tao, 2004) detected emotions from text based on lexicon consisting of emotional keywords, modifier words and metaphor words. The extensive work on feature selection for sentiment analysis is reported in (Duric & Song, 2011). The first research regarding sentiment classification in Croatian texts is reported in (Agić et al., 2010).

The process of text classification using Naïve Bayes and the framework for this work is described in section 2. The procedure of document collection is given in section 3. Feature sets extracted from document collection are presented in section 4. The results of the conducted experiment are given in section 5. The paper ends with some concluding remarks.

2. Text classification with Naïve Bayes

In our study, we tried to automatically identify the emotion of FEAR in specific corpora of the Croatian language. The initial task of this research was to classify articles, blogs and comments collected from web-sites into two categories: "*Fear present*" and "*Fear not present*" using supervised machine learning classification method based on Naïve Bayes algorithm. We experimented with different sets of features to compare their impact on the accuracies of the learnt classifiers and to determine which set of features yields better result (accuracy).

There are various approaches to text classification ranging from hand-written rules to unsupervised and supervised automatic machine learning techniques (Manning, 2009). The Naïve Bayes classifier relies on a simple representation of a document as a "bag of words". Another assumption that Naïve Bayes classifier entails is that the feature probabilities are independent of each other given the class. For this initial experiment we used Naïve Bayes classifier.

In text classification, we are given a set of documents d and a fixed set of classes $C = \{c_1, c_2, \ldots, c_n\}$. Classes are also called categories or labels. In supervised classification, documents are represented by feature sets which capture the basic information about each input (document) that should be classified. The classes are human defined. The training set consists of m hand-labelled documents with the corresponding class annotations $(d_1,c_1),\ldots,(d_m,c_m)$.

The probability of a document d being in class c is computed as (Manning et al., 2009):

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c. $P(c)$ is the prior probability of a document occurring in class c.

The goal of text classification is to find the best class for the document. In Naïve Bayes classification, the best class is the most likely or maximum a posteriori class c_{map} (Manning et al., 2009):

$$c_{map} = \underset{c \in C}{\operatorname{argmax}}\, \hat{P}(c|d) = \underset{c \in C}{\operatorname{argmax}}\, \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c)$$

We do not know the true values of parameters $P(c)$ and $P(t_k|c)$ but we use their estimates $\hat{P}(c)$ and $\hat{P}(t_k|c)$. We estimate the prior probablity by the formula:

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

where N_c is the number of documents in class c and N_{doc} is the total number of documents.

The conditional probability $\hat{P}(t_i|c_j)$ for each term (feature) in class is estimated as the fraction of times term t_i appears in the set of all terms (V) in documents of the class c_j:

$$\hat{P}(t_i|c_j) = \frac{count(t_i, c_j)}{\sum_{t \in V} count(t, c_j)}$$

3. Document collection

For the purpose of our research, documents from various online information sources were collected. We collected a total of 3218 articles (5534367 tokens). The selected web-sources included 4 religious and 5 political portals, 6 blog spaces, 3 web-pages with comments and 4 columns from the daily newspapers (shown in Table 1).

Religion	
	http://www.glas-koncila.hr/
	http://www.hbk.hr/
	http://www.hrvatskipravoslavci.com/
Politics	
	http://www.vlada.hr/
	http://mrak.org/
	http://novapolitika.blog.hr/
	http://pollitika.com/ptracker
	http://www.ivangrubisic.com/
Blogs	
	http://www.blog.hr/
	http://www.bloger.hr/index.aspx
	http://www.monitor.hr/vijesti/kategorija/blogovi/
	http://www.jutarnji.hr/komentari/blogovi/
	http://blog.vecernji.hr/
	http://www.slobodnadalmacija.hr/Blogeri/tabid/54/Default.aspx
Comments	
	http://www.novilist.hr/Komentari
	http://www.index.hr/vijesti/komentatori/
	http://www.jutarnji.hr/komentari/komentari_sub/
Columns	
	http://www.jutarnji.hr/komentari/kolumne/
	http://www.vecernji.hr/kolumne/
	http://www.glas-slavonije.hr/kolumne.asp
	http://www.dubrovacki.hr/pregled/kolumne

Table 1: Web-sources used in document collection

Abovementioned web-pages and respective topics were chosen because their genre and literary style indicated that subjective account of emotions (particularly fear) would be expressed to a greater extent than in the articles from any randomly chosen web-pages. All articles are written in the Croatian language.

Out of 3218 articles, 1507 articles were manually annotated into the categories "*Fear present*" and "*Fear*

not present" by 23 annotators. They were instructed to annotate the articles according to their subjective judgment whether there is fear related content present in text or not. Out of 1507 annotated articles, there were 1263 articles categorized into the category "*Fear not present*" and 244 articles were categorized into the category "*Fear present*" (statistics shown in Table 2).

Category	Number of articles	Tokens
Fear present	244	271966
Fear NOT present	1263	2130275
Total number of articles	1507	2402241

Table 2: Statistics of the document collection used in the research

4. Feature selection

The process of sentiment analysis is designed by selection of salient features. According to (Duric & Song, 2011), the criteria that are useful in selecting salient features for sentiment analysis include: a) features should be expressive enough to add useful information to the classification process, b) all features together should form a broad and comprehensive viewpoint of the entire corpus, c) features should be as domain-dependent as possible, d) features must be frequent enough and e) features should be discriminative enough.

In our classification procedure of fear related content we experimented with three different sets of features. First set of features contained only words that represent embodied metonymical domains of fear, i.e. bodily features of experiencing and expressing fear such as *tresti* (en. *tremble), blijed* (*en. pale*), *hladan* (*en. cold*), *znoj* (*en. sweat*). The embodied metonymical domains were provided by corpus based research of metaphoric and metonymic emotional conceptualization of fear in the Croatian language (Perak, 2011). In this work, Perak identified 2231 metonymical constructions that profile an embodied emotional model of fear. Out of 2231 metonymical constructions 60 words were directly associated with physical manifestation of fear. This result was used as the initial feature set for the automatic fear identification of Croatian texts presented in this paper. The list of 60 words was morphologically expanded using Croatian Morphological Lexicon (Tadić & Fulgosi, 2003) and manually verified. Finally, the first feature set "*Physical manifestation*" contained a list of 505 words related to the physical manifestation of fear.

The other set of features contained only lexical concepts of fear, its synonyms and direct hyponyms such as *strah* (en. fear), *prestrašiti* (en. scare), *horor* (en. horror), etc. which were extracted from the English WordNet and translated to Croatian. The resulting set contained 270 words which formed the second feature set "*Lexically expressed fear*".

The third set of features labelled "*Combination*" with 775 words is the union of all words from the first and the second feature sets.

5. Experiment

In our initial experiment, the Naïve Bayes classifier was trained on the document collection (described in section 3) using NLTK - Natural Language Toolkit (Bird et al., 2012). For each feature set described in section 4, 10 runs of Naïve Bayes training/testing was performed. For each run the document collection was randomly divided into training and test set in the ratio 9:1. The accuracy was computed as the average accuracy of ten different runs for each feature set. The results are shown in Table 3.

Feature set used in classification	Accuracy
Physical manifestation	0.8
Lexically expressed fear	0.83
Combination	0.81

Table 3: Accuracies for three different features sets

We obtained the best accuracy with the feature set that contains words that lexically express fear (0.83). The accuracies of the other two classifiers were slightly lower.

During the first experiment, we also identified the most informative features, i.e. features that have the biggest ratios of conditional probabilities. For each feature set, 30 most informative features were selected in order to have feature sets of the same size. Then, the procedure of classifier training and testing was repeated with the 30 most informative features from each feature set in order to compare the performance of reduced feature sets of the same size.

The results with the 30 most informative features are shown in Table 4.

Feature sets reduced to 30 most informative features	Accuracy
Physical manifestation	0.8
Lexically expressed fear	0.82
Combination	0.83

Table 4: Accuracies for reduced feature sets

Ten most informative features in *"Physical manifestation"* feature set are: cold *hladnu*, limbs *udovi*, face *licem*, pale *blijedi*, small *malima*, green *zeleni*, bitter *gorko*, green *zelena*, limb *ud*, fat *debelim.*

Ten most informative in *"Lexically expressed fear"* feature set are: alert *uzbunu*, horror *grozote*, alert *uzbuna*, chill *jeza*, panic *panika*, fear *straha*, terror *terora*, fears *strahove*, panic *panici*, afraid *boji.*

Ten most informative features in *"Combination"* feature set are: alert *uzbunu*, bitter *gorka*, horrors *grozote*, limbs *udovi*, face *licem*, pale *blijedi*, seen *vidljivi*, green *zeleni*, panic *panika*, alert *uzbuna*.

The best results of identification were achieved with the 30 most informative features from the combined feature set (embodied metonymic profiles/features and explicitly lexically expressed fear). So far, the results support the usage of physically expressed features in fear identification.

The performance comparison of "*Physical manifestation"* and "*Lexically expressed fear*" feature sets shows that the embodied profiles of the physiological processes of perception, representation and reaction can be relevant features for the identification and elicitation of the concept fear even without explicit lexicalization of the

category. The same principle should be considered for identification of other emotions in Croatian texts as well.

6. Conclusion and future work

In this work the articles, blogs and comments collected from the Croatian web-sites were classified into two categories: "*Fear present*" and "*Fear not present*". Supervised machine learning classification method based on Naïve Bayes algorithm was used. This classification process was designed to automatically determine whether there is fear related content in the text or not. We experimented with different sets of features in order to compare their impact on the accuracies of the learnt classifiers and to determine which set of features yields better result in terms of achieved accuracy. The experiment that we conducted was the initial attempt in fear identification in Croatian texts.

The best accuracy was achieved with the feature set that contains words that lexically express fear: In the first experiment, the performances of the feature sets could not be compared since there were far more features in the third set ("*Combination"*) which comprises of all the features from the first ("*Physical manifestation"*) and the second feature set ("*Lexically expressed fear"*). The second experiment was conducted with the purpose of comparing accuracies depending on feature sets of the same size. In the reduced feature set experiment the best accuracy is obtained with combined feature set (embodied metonymic profiles/features and explicitly lexically expressed fear).

The results encourage further research in combining cognitive interpretation of various sensory-motor, visceral, causative or culturally related domains that lead to a particular kind of affective experience with the lexicon of semantically related words of the emotional category for emotion recognition in Croatian texts. The first step will be verifying the corpus annotation by repeating the annotation of the same articles by more annotators in order to obtain the agreement of the annotators. We also plan to lemmatize corpus. Afterwards, the experiments including other algorithms and feature sets are planned. Additional features such as the most frequent phrases and derived metonymical and metaphorical constructs will be considered as well. We would also like to extend the research by experimenting with the identification of other emotions in order to identify different types of emotional content in Croatian texts.

7. References

Agić. Ž., Ljubešić. N., Tadić. M., 2010. *Towards Sentiment Analysis of Financial Texts in Croatian*. Proc. 7th International Conference on Language Resources and Evaluation. Valletta: 1164-1167

Aman. S., Szpakowicz. S., 2007. *Identifying Expressions of Emotion in Text*, In Proc. 10th International Conference on Text, Speech and Dialogue. Plzen, Czech Republic. LNCS 4629. Springer. 196-205.

Baker. C.F., Fillmore. J., Lowe. J.B., 1998. *The Berkeley FrameNet Project*. In Proc. 36th Annual Meeting of the Association for Computational Linguistics (ACL '98),. Association for Computational Linguistics. 86-90.

Barrett Feldman. L., 2011. Constructing emotion. *Psychological Topics*, 20/2: 359-380.

Barrett Feldman. L., Lindquist, K., & Gendron, M. 2007. Language as a context for emotion perception. *Trends in Cognitive Sciences*, 11, 327-332.

Bird. S., Klein. E., Loper. E., 2012. *Natural Language Processing with Python*. O'Reilly.

Boster. J., 2005. Emotion categories across languages. In: Cohen, H. and Lefebvre, C. (ed.) *Handbook of categorization in Cognitive science*: 187-222. Amsterdam, NL: Elsevier.

Damasio, A., 1999. *The feeling of what happens body and emotion in the making of consciousness*. New York. Harcourt.

Davidson. R.J., Scherer, K., Goldsmith, H., 2003. *Handbook of Affective Sciences*. New York. Oxford University Press.

Duric. A., Song. F., 2011. *Feature Selection for Sentiment Analysis Based on Content and Syntax Models*. Proc. 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, 96-103.

Fussell, S., 2002. *The Verbal Communication of Emotions. Interdisciplinary Perspectives*. Mahwah, NJ: Lawrence Erlbaum.

Manning. D.C., Raghvan. P., Schütze. H., 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

Mohammad. S., 2012. *Portable Features for Classifying Emotional Text*, In Proc. of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: HLT.Montreal,Canada.

Lewis, M., Haviland-Jones, J., Barrett Feldman, L., 2008. *Handbook of Emotions.* New York: The Guilford Press.

Perak. B., 2011. *The role of Embodied Cognition in Conceptualization of the Emotional Categories.* Context, Review for Comparative Literature and Cultural Research. 9; 193-1-212-20

Strapparava. C., Mihalcea. R., 2008. *Learning to identify emotions in text*. In Proc. ACM symposium on Applied computing. New York. 1556-1560.

Tadić. M., Fulgosi. S., 2003 *Building the Croatian Morphological Lexicon*. Proc. EACL2003 Workshop on Morphological Processing of Slavic Languages. Budapest. 41-46.

Tao. J., 2004. *Context Based Emotion Detection from Text Input*. 8th International Conference on Spoken Language Processing, ICSLP2004. Jeju. 1337-1340.

Wierzbicka, A. (1999) Emotions across languages and cultures. Cambridge, UK: Cambridge University Press.

A Web Service Implementation of Linguistic Annotation for Slovene and English

Senja Pollak[1,2], Nejc Trdin[1,3], Anže Vavpetič[1,3] and Tomaž Erjavec[1,3]
[1]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2]Faculty of Arts, Aškerčeva 2, 1000 Ljubljana
[3]International Postgraduate School Jožef Stefan, Jamova cesta 39, 1000 Ljubljana, Slovenia
{senja.pollak, nejc.trdin, anze.vavpetic, tomaz.erjavec}@ijs.si

Abstract

This paper presents a web service for automatic linguistic annotation of Slovene and English texts. The texts are tokenised, morphosyntactically tagged and lemmatised by the ToTrTaLe annotation tool, while the web service for this annotation is made available in the Orange4WS and the ClowdFlows workflow construction environments. The workflows enable the users to apply the annotation tool as an elementary constituent for other natural language processing workflows. The user can upload the text(s) in different formats (TXT, DOC, DOCX, PDF, ZIP), convert them to plain text and annotate them with ToTrTaLe. The paper also proposes several improvements of the ToTrTaLe tool based on the identification of various types of errors of the existing implementation, and implements these improvements as a post-processing step in the workflow.

Implementacija spletnega servisa za jezikoslovno označevanje slovenskega in angleškega jezika

Prispevek predstavi implementacijo spletnega servisa za jezikoslovno označevanje slovenskega in angleškega jezika. Program ToTrTaLe je kot spletni servis uporabnikom na voljo v okoljih za izgradnjo delotokov Orange4WS in ClowdFlows in omogoča tokenizacijo, oblikoskladenjsko označevanje in lematizacijo besedil. Delotoki omogočajo uporabniku, da uporabi jezikoslovno označevanje kot elementarni gradnik pri večjih delotokih za analizo besedil. Uporabnik lahko naloži besedila v različnih formatih (TXT, DOC, DOCX, PDF, ZIP), jih pretvori v navadno besedilo in jih označi s ToTrTaLe. Prispevek tudi predlaga več izboljšav za ToTrTaLe, ki temeljijo na identifikaciji različnih vrst napak obstoječe implementacije, in jih implementira kot dodaten korak delotoka.

1. Introduction

In corpus linguistics, part-of-speech tagging (PoS tagging), also called word-level grammatical tagging, is the process of marking up word tokens in a text (corpus) as corresponding to a particular part of speech, based on the lexicon giving the possible PoS tags of the word and the context in which the word appears. PoS-tagging algorithms fall into two groups: rule-based taggers and statistical taggers where the PoS tags are learned from a manually annotated text corpus. For languages with rich inflection, such as Slovene, it is better to speak of morphosyntactic annotations or descriptions (MSDs) rather than PoS tags, as such MSDs contain much more information than do PoS tags. For example, the PoS tagsets for English have typically from 20 – 60 different tags, while Slovene has over 1,000 MSDs.

This paper focuses on a particular tool for automatic morphosyntactic tagging, named ToTrTaLe (Erjavec et al., 2011). A brief description of ToTrTaLe is presented in Section 2. As one of the main contributions of this work is the implementation of ToTrTaLe as a web service which can be used as an ingredient of complex NLP workflows, we first motivate this work in Section 3 by a short introduction to web services, workflows and by presenting two specific workflow construction environments Orange4WS and ClowdFlows. The main contributions of this research are presented in Sections 4 and 5. Section 4 presents the implementation of the ToTrTaLe analyser as a web service in two service-oriented workflow construction and management platforms Orange4WS and ClowdFlows. Section 5 presents the proposed improvements of the ToTrTaLe tool based on the identification of several types of errors of the existing implementation. These error corrections are implemented as a part of our web-service.

2. The ToTrTaLe annotation tool

ToTaLe (Erjavec et al., 2005) is short for Tokenisation, Tagging and Lemmatisation and is the name of a script implementing a pipeline architecture comprising these three processing steps. While the tool makes some language specific assumption, they are rather broad, such as that text tokens are (typically) separated by space; otherwise, the tool itself is language independent and relies on external language resources. The tool is written in Perl and is reasonably fast. The greatest speed bottleneck is the tool start-up, mostly the result of the lemmatisation module, which for Slovene contains thousands of rules and exceptions.

In the context of the JOS project (Erjavec et al., 2010) the tool was re-trained for Slovene and made available as a Web application at http://nl.ijs.si/jos/analyse/. It allows pasting the text to be annotated into the form or uploading a plain-text UTF-8 file and either have the annotated text displayed or downloaded as a ZIP file.

The tool (although not the Web application) has been recently extended with another module, Transcription, and the new edition is called ToTrTaLe (Erjavec, 2011). The transcription step is used for modernising historical language (or, in fact, any non-standard language), and the tool was used as the first step in the annotation of a reference corpus of historical Slovene (Erjavec, 2012a). And additional extension of ToTrTaLe is the ability to process heavily annotated XML document conformant to the Text Encoding Initiative Guidelines (TEI, 2007).

The Web service presented in this paper uses To(Tr)TaLe with models for Slovene and English, but as the historical language models are not as mature as those for contemporary language, this extra functionality is not discussed here further. In the rest of this section we present the main modules of To(Tr)TaLe and also their models for Slovene and English.

2.1. The tokenisation module

The multilingual tokenisation module mlToken[1] is written in Perl and in addition to splitting the input string into tokens, it also assigns to each token its token type, e.g. XML tag, sentence final punctuation, digit, abbreviation, URL, etc. and preserves (subject to a flag) white-space, so that the input can be reconstituted from the output.

The tokeniser can be fine-tuned by putting punctuation into various classes (e.g. word-breaking vs. non-breaking) and also uses several language-dependent resource files: a list of abbreviations ("words" ending in period, which is a part of the token and does not necessarily end a sentence); a list of multi-word units (tokens consisting of several space-separated "words"); and a list of (right or left) clitics, i.e. cases where one "word" should be treated as several tokens. Such resource files allow for various options to be expressed, although not all, as will be discussed in section 5.

The tokenisation resources for Slovene and English were developed by hand, and cover most typical exceptions in both languages.

2.2. Tagging

For tagging words in the text with their context disambiguated PoS tags (or, better, morphosyntactic annotations) we use TnT (Brants, 2000), a fast and robust tri-gram tagger.

For Slovene, the tagger has been trained on jos1M, the 1 million word JOS corpus of contemporary Slovene (Erjavec et al., 2010), and is also given a large background lexicon extracted from the 600 million word FidaPLUS reference corpus of contemporary Slovene (Arhar Holdt and Gorjanc, 2007). The English model was trained on the MULTEXT-East corpus (Erjavec, 2012b), namely the novel "1984". This is of course a very small corpus, so the resulting model is not very good. However, it does have the advantage of using the MULTEXT-East tagset, which is compatible with the JOS one.

2.3. Lemmatisation

For lemmatisation To(Tr)TaLe uses CLOG (Erjavec and Džeroski, 2004), which implements a machine learning approach to the automatic lemmatisation of (unknown) words. CLOG learns on the basis of input examples (pairs word-form/lemma, where each morphosyntactic tag is learnt separately) a first-order decision list, essentially a sequence of if-then-else clauses, where the defined operation is string concatenation. The learnt structures are Prolog programs but in order to minimise interface issues we made a converter from the Prolog program into one in Perl.

The Slovene lemmatiser was trained on a lexicon extracted from the jos1M corpus, and the lemmatisation of contemporary language is reasonably accurate, with 92% on unknown words. However the learnt model, given that there are 2,000 separate classes, is quite large: the Perl rules have about 2MB, which makes loading the lemmatiser slow.

The English model was trained on the English MULTEXT-East corpus, which has about 15,000 lemmas and produces a reasonably good model, especially as English is fairly simple to lemmatise.

[1] mlToken was written in 2005 by Camelia Ignat, then working at the EU Joint Research Centre in Ispra, Italy.

3. Web services and workflows

A Web service is a method of communication between two electronic devices over the web. The W3C defines a Web service as "a software system designed to support interoperable machine-to-machine interaction over a network". A Web service's functionalities are described in a machine-processable format i.e., the Web Services Description Language, known by the acronym WSDL. Other systems interact with the Web service in a manner prescribed by its description using SOAP XML messages, typically conveyed using HTTP in conjunction with other Web-related standards. The W3C also states that we can identify two major classes of Web services, REST-compliant Web services, in which the primary purpose of the service is to manipulate XML representations of Web resources using a uniform set of "stateless" operations, and arbitrary Web services in which the service may expose an arbitrary set of operations.

3.1. Workflow construction platforms

Main data mining environments that allow for workflow composition and execution, implementing the visual programming paradigm, include Weka (Witten et al., 2011), Orange (Demšar et al., 2004), KNIME (Berthold et al., 2007) and RapidMiner (Mierswa et al., 2006). The most important common feature is the implementation of a workflow canvas where workflows can be constructed using simple drag, drop and connect operations on the available components, implemented as graphical units named widgets. This feature makes the platforms suitable for use also by non-experts due to the representation of complex procedures as relatively simple sequences of elementary processing steps (workflow components implemented as widgets).

In this work, we use two recently developed service-oriented environments for data mining workflow construction and execution: Orange4WS and ClowdFlows.

The first platform Orange4WS (Podpečan et al., 2012) is distinguished from other main data mining platforms by its capacity of including web services into data mining workflows, allowing for distributed processing. Such a service-oriented architecture has already been employed in Taverna (Hull et al., 2006), a popular platform for biological workflow composition and execution. Using processing components implemented as web services enables remote execution, parallelisation, and high availability by default. A service-oriented architecture supports not only distributed processing but also distributed development.

The second platform ClowdFlows (Kranjc et al., 2012) is distinguished from other main data mining platforms by its capacity of workflow sharing. Sharing of workflows has previously been implemented through the myExperiment website of Taverna (Hull et al., 2006). This website allows the users to publicly upload their workflows so that they are made available to a wider audience. Furthermore, publishing a link to a certain workflow in a research paper allows for simpler dissemination of scientific results. However, the users who wish to view or execute these workflows are still

required to install the specific software in which the workflows were designed and implemented. On the other hand, the ClowdFlows platform implements the described features also with one major advantage. ClowdFlows requires no installation and can be run on any device with an internet connection, using any modern web browser. The ClowdFlows platform is described in more detail below.

3.2. The ClowdFlows platform

ClowdFlows is implemented as a cloud-based application that takes the processing load from the client's machine and moves it to remote servers where experiments can be run with or without user supervision. The user does not need to perform any specific installation. ClowdFlows consists of the workflow editor (the graphical user interface, as shown in Figure 1) and the server-side application which handles the execution of the workflows and hosts a number of publicly available workflows.

The workflow editor consists of a workflow canvas and a widget repository, where widgets represent embedded chunks of software code. The widgets are separated into categories for easier browsing and selection and the repository includes a wide range of readily available widgets. Our NLP processing modules have also been implemented as such widgets.

By using ClowdFlows we were able to make our NLP workflow public, so that anyone can use and execute it. The workflow is exposed by a unique URL, which can be accessed from any modern Web browser. Whenever the user opens a public workflow, a copy of this workflow appears in her private workflow repository. The user can execute the workflow and view its results or expand it by adding or removing widgets. Any user can therefore use ToTrTaLe as a pre-processing step in any other NLP workflow.

4. ToTrTaLe web service implementation

In this section we present the services we implemented and also some details regarding the implementations. All services were implemented in the Python programming language, using Orange4WS API and additional freeware software packages. Services are currently adapted to run on Unix-like operation systems, but are easily transferable to other operation systems.

4.1. Implemented web services

We implemented two web services, which constitute the main implementation part of this work. The first service converts the files to plain text. The second service uses ToTrTaLe to annotate the texts.

4.1.1. Converting input data

The first service parses the input data and converts it into plain text. The input corpus can be uploaded in various formats, either as a single file or as several files compressed in a ZIP file. The supported formats are PDF, DOC, DOCX, TXT and HTML, the latter passed to the service in the form of an URL.

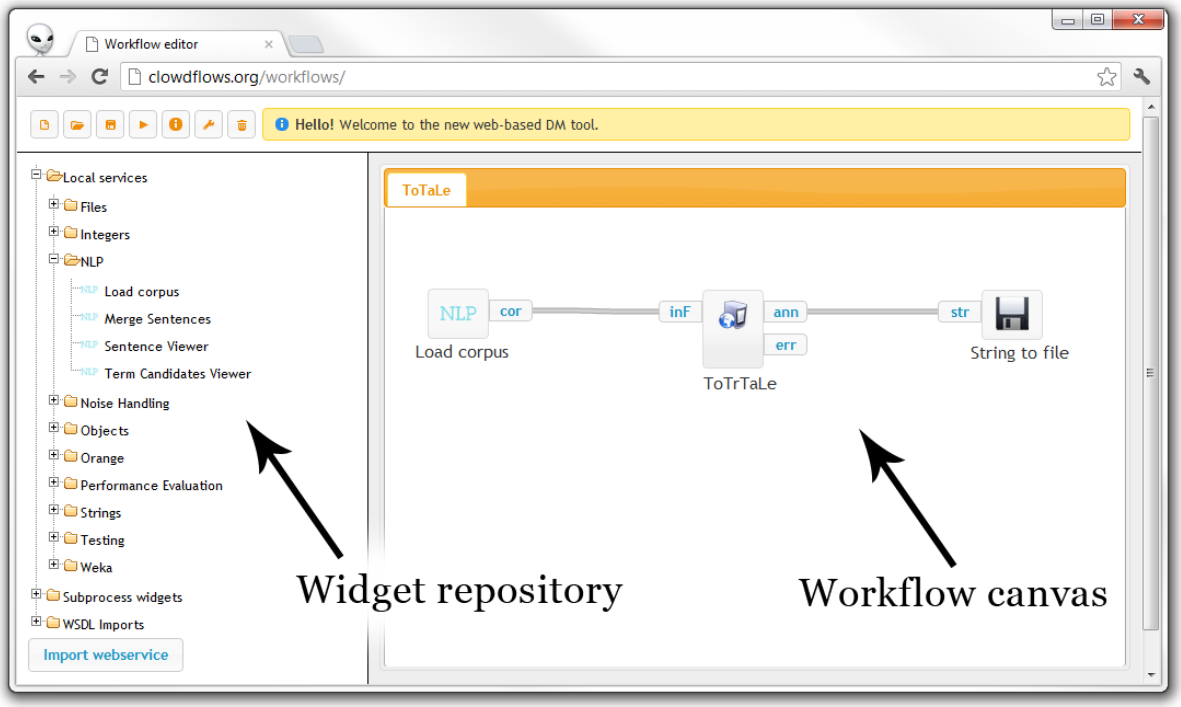


Figure 1. A screenshot of the ClowdFlows workflow editor in the Google Chrome browser and the ToTrTaLe workflow, available at http://clowdflows.org/workflow/228/.

Based on the file type, the program chooses the correct converter:

- If the document is an HTML document, its URL is written in the document variable and the document is assumed to contain only plain text. The web service then downloads the document via the given URL in plain text.
- DOCX Microsoft Word documents are essentially compressed ZIP files containing the parts of the document in XML.
- DOC Microsoft Word files are converted using an external tool, wvText (Lachowicz et al., 2006), which transforms the file into plain text.
- PDF files are converted with the Python pdfminer library (Shinyama, 2010).
- If the file name ends with TXT, then the file is assumed to be already in plain UTF-8 text.
- ZIP files are extracted into a flat directory and converted to a file with XML elements containing the plain-text of the individual files.

The resulting text representation is then sent through several regular expression filters, in order to further normalize the text. For instance, white space is normalised.

The final step involves sending the data. At each step of the web service process, errors are accumulated in the error output variable.

4.1.2. ToTrTaLe web service

The second web-service implements the ToTrTaLe annotation tool and also supports post-processing which corrects some systematic errors, which are further described in Section 5. The parameters of this web service are: the document, the language of the text (English, Slovene or historical Slovene), if we want post-processing, and if we want the output in XML format or as plain text.

The local ToTrTaLe service is then run, the output is written into the output variable, and the possible errors are passed to the error variable. Additionally, the input parameter for post-processing defines if the post–processing scripts are run on the text. The post-processing scripts are Perl implementations of corrections for tagging mistakes described in Section 5.

Finally, the output string variable is passed on to the output of the web service.

4.2. Implemented widgets

Apart from the web services we also needed to adapt some platform specific widgets to successfully use the web services. These widgets, not exposed as web services, are run locally; in the case of Orange4WS they are executed on the user's machine, whereas in the case of ClowdFlows they are executed on the server hosting the ClowdFlows application.

Orange4WS and ClowdFlows can automatically construct widgets for web services. They identify the inputs and the outputs of the web service from the service's WSDL description. Nevertheless, an additional functionality was required to adequately support the user in using the web services. Both in Orange4WS as well as in ClowdFlows, we implemented a widget called "Load Corpus" that opens a corpus in one of the formats supported by the web service for parsing input data, as well as internally calls the web service for converting input data.

4.3. Example workflow

The widgets implementing the existing software components are shown in Figure 1 and Figure 2. Figures show that the implementation of web-services is platform-independent.

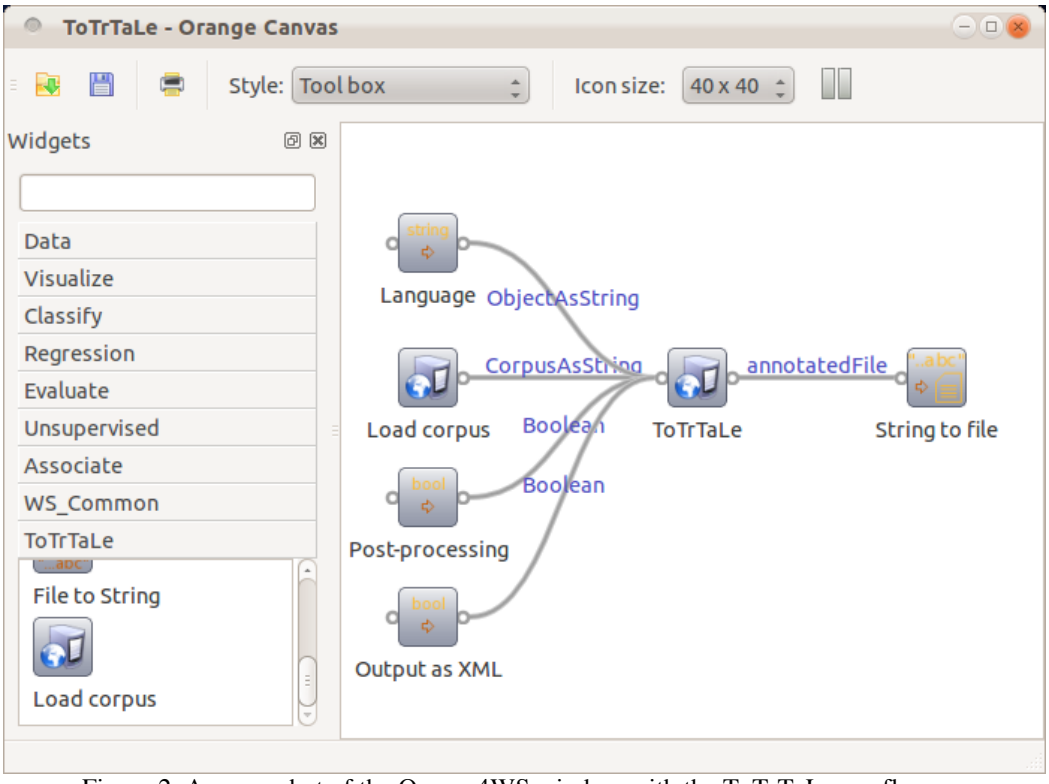


Figure 2. A screenshot of the Orange4WS window with the ToTrTaLe worflow.

In both figures the same workflow is represented. Figure 1 shows the workflow in the ClowdFlows platform and Figure 2 shows the workflow in the Orange4WS platform.

On the left side of both figures, there is a widget repository, and the right side is intended for the construction of workflows – the canvas. Apart from our web service widgets, there are some general-purpose widgets (e.g., file reading, file writing, construction of strings). The purpose of both workflows is essentially the same: they accept a file and read the file. Then the file is parsed from its original form into the plain text representation of the file. After the parsing of the file, the plain text representation is input into the ToTrTaLe web service. The service returns the annotated file in the plain text representation according to other input parameters. The final file can be viewed in the right most widget (String to file) of the corresponding workflow.

There is also a minor difference in the workflows presented in Figures 1 and 2. The difference is that the Orange4WS workflow has more widgets than the ClowdFlows workflow. This is due to the fact that widgets for Orange4WS were implemented to accept input data from other widgets (String widget, Boolean widget, etc.), whereas the widgets for ClowdFlows were implemented to accept inputs directly as parameters (by double clicking on the widget).

The sample output produced by either of the two workflows is shown in Figure 3. The figure clearly shows the function of each token, the sentence splitter tags and also morphosyntactic annotation of each token. The final output is in the form of plain text, where the input to the workflow was a Slovene PDF file.

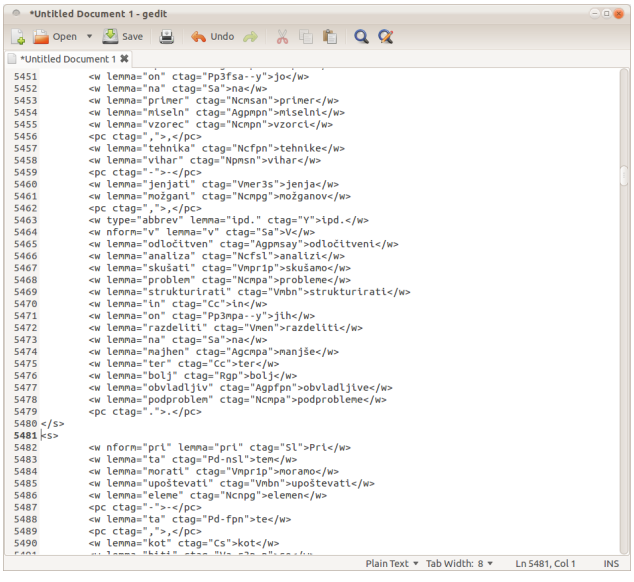


Figure 3. A sample output from the ToTrTaLe web-service, annotating sentences and tokens, with lemmas and MSD corpus tags on words.

5. Analysis of tagging mistakes

In this section we present the observed ToTrTaLe mistakes, mainly focusing on Slovene. The corpus used for analysis contains the papers of the Proceedings of the past seven Language Technology conferences. The construction of the corpus is described in Smailović and Pollak (2011).

The majority of the described mistakes are currently handled in an optional post-processing step, but can be taken into consideration in future versions of ToTrTaLe, by improving tokenisation rules or changing the tokeniser, re-training the tagger with larger and better corpora and lexica, and improving the lemmatisation models or learner.

5.1. Incorrect sentence segmentation

Errors in sentence segmentation originate mostly from the processing of abbreviations. Since the analysed examples were taken from academic texts, specific abbreviations, leading to incorrect separation of sentences, are frequent. The abbreviations that should be added to the abbreviation list for ToTrTaLe are e.g. "*et al.*", "*in sod.*", "*cca.*". On the other hand there are abbreviations after which ToTrTaLe should end the sentence, but doesn't. Checking if there is an upper case letter following the abbreviation would, in most cases, solve this mistake. Examples include the measures "*KB*", "*MB*", "*GB*", and "*ipd.*", "*itd.*", "*etc.*".

5.2. Incorrect morphosyntactic annotations

The tagging also at times makes mistakes, and in some cases these mistakes occur systematically. One example is in subject complement structures. For instance "*Kot podatkovne strukture so semantične mreže usmerjeni grafi.*" [*As data structures semantic networks are directed graphs.*] the nominative plural feminine "*semantične mreže*" [*semantic networks*] is wrongly annotated as singular genitive feminine. Another frequent type of mistake, easy to correct, is unrecognized gender/number/case agreement between adjective and noun in noun phrases. For example, "*Na eni strani imamo semantične leksikone …*" [*On the one hand we have the semantic lexicons…*], "*semantične*" [*semantic*] is assigned a feminine plural nominative MSD, while "*leksikone*" [*lexicons*] is attributed a masculine plural accusative tag. Next, in several examples, "*sta*" (second person, dual form of verb "*to be*") is tagged as a noun. Even if "*STA*" can be used as an abbreviation (when written with capital letters), it is much more frequent as the word-form of the auxiliary verb.

5.3. Incorrect lemmatisation

Besides the most common error of wrong lemmatisation of individual words (e.g. "*hipernimija*" being lemmatised as "*hipernimi*" [hypernyms] and not as "*hipernimija*" [*hypernymy*]), there are systematic errors when lemmatising Slovene adjectives in comparative and superlative form, where the base form is not chosen as a lemma. Last but not least, there are typographic mistakes in the original text and of end-of-line split words.

6. Conclusions and further work

In this paper we presented the ToTrTaLe web service and demonstrated how it can be used in workflows in two service-oriented data mining platforms – Orange4WS and ClowdFlows. Together with the ToTrTaLe web service, we developed a series of widgets (workflow components) for pre-processing the text, consisting of reading the text corpus files in various formats, tokenising the text, lemmatising and morphosyntactically annotating it, as well as adding the sentence boundaries, followed by a post-processing widget for error correction.

Before starting this work, the To(Tr)TaLe tool has already existed as a web application for Slovene, where the user was able to upload and add the text, but the novelty is that a web service implementation now enables the user to use ToTrTaLe as a part for various other NLP applications. The presented web service has already been incorporated in the term and definition extraction workflow[2] (Pollak et al. 2012).

Acknowledgements

We are grateful to Vid Podpečan and Janez Kranjc for their support and for enabling us to include the developed widgets into Orange4WS and ClowdFlows, respectively. This work was partially supported by the Slovene Research Agency and the FP7 European Commission projects "Machine understanding for interactive storytelling" (MUSE, grant agreement no: 296703) and "Large scale information extraction and integration infrastructure for supporting financial decision making" (FIRST, grant agreement 257928).

References

Špela Arhar Holdt and Vojko Gorjanc (2007). Korpus FidaPLUS: nova generacija slovenskega referenčnega korpusa. *Jezik in slovstvo*, 52(2): 95–110.

Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel and Bernd Wiswedel (2007). KNIME: The Konstanz Information Miner. In Preisach, C., Burkhardt, H., Schmidt-Thieme, L., Decker, R., (eds.): *GfKl. Studies in Classification, Data Analysis, and Knowledge Organization*, Springer, pp. 319–326.

Thorsten Brants (2000). TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the 6h Applied Natural Language Processing Conference* (ANLP 2000), Seattle, WA, pp. 224–231.

Janez Demšar, Blaž Zupan, Gregor Leban and Tomaž Curk (2004). Orange: From experimental machine learning to interactive data mining. In Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.): *Proceedings of ECML/PKDD-2004*. Springer LNCS Volume 3202, pp. 537–539.

Tomaž Erjavec (2011). Automatic linguistic annotation of historical language: ToTrTaLe and XIX century Slovene. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, ACL.

Tomaž Erjavec (2012a). The goo300k corpus of historical Slovene. In *Proceedings of the 8th conference on Language Resources and Evaluation (LREC'12)*, Istambul. European Language Resources Association (ELRA).

Tomaž Erjavec (2012b). MULTEXT-East: morphosyntactic resources for Central and Eastern European languages. *Language resources and evaluation*, 46(1): 131–142.

Tomaž Erjavec and Sašo Džeroski (2004). Machine Learning of Language Structure: Lemmatising Unknown Slovene Words. *Applied Artificial Intelligence*, 18(1):17–41.

Tomaž Erjavec, Darja Fišer, Simon Krek and Nina Ledinek (2010). The JOS linguistically tagged corpus of Slovene. In *Proceedings of the 7th International Conference on Language Resources and Evaluations*, LREC 2010, Valletta, Malta, pp. 1806-1809.

Tomaž Erjavec, Camelia Ignat, Bruno Pouliquen and Ralf Steinberger (2005). Massive Multi-Lingual Corpus Compilation: Acquis Communautaire and ToTaLe. In *Proceedings of the 2nd Language & Technology Conference*, April 21-23, 2005, Poznan, Poland, pp. 32–36.

Duncan Hull, Katy Wolstencroft, Robert Stevens, Carole Goble, Matthew R. Pocock, Peter Li and Thomas M. Oinn (2006). Taverna: A tool for building and running workflows of services. *Nucleic Acids Research 34* (Web-Server-Issue): 729–732.

Dom Lachowicz and Caolán McNamara (2006). *wvWare, library for converting Word document*. http://wvware.sourceforge.net/, accessed in August 2012.

Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz and Timm Euler (2006). YALE: rapid prototyping for complex data mining tasks. In Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopulos, D. (eds.): *Proceedings of KDD-2006*, ACM , pp. 935–940.

Janez Kranjc, Vid Podpečan, and Nada Lavrač (2012). ClowdFlows: A cloud-based scientific workflow platform. In *Proceedings of ECML/PKDD-2012*. September 24-28, 2012, Bristol, UK, *Springer LNCS* (in press).

Vid Podpečan, Monika Žakova and Nada Lavrač (2012). Orange4ws environment for service-oriented data mining. *The Computer Journal* (2012), 55(1): 82–98.

Senja Pollak, Anže Vavpetič, Janez Kranjc, Nada Lavrač and Špela Vintar (2012). In J. Jancsary (ed.): *Proceedings of the 11th Conference on Natural Language Processing* (KONVENS 2012), September 19-21, 2012, Vienna, Austria, pp. 53–60.

Yusuke Shinyama (2010). PDFMiner http://www.unixuser.org/~euske/python/pdfminer/index.html, accessed in August 2012.

Jasmina Smailović and Senja Pollak (2011). Semi-automated construction of a topic ontology from research papers in the domain of language technologies. In *Proceedings of the 5th Language & Technology Conference*, November 25–27, 2011, Poznan, Poland, pp. 121–125.

TEI Consortium (2007). *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. http://www.tei-c.org/Guidelines/P5/

Ian H. Witten, Eibe Frank and Mark Hall (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Third Edition. Morgan Kaufmann.

[2] http://clowdflows.org/workflow/76/

Termania – prosto dostopni spletni slovarski portal

Miro Romih *, Simon Krek †,

*†Amebis, d. o. o., Kamnik
Bakovnik 3, 1241 Kamnik
miro.romih@amebis.si, simon.krek@amebis.si
† Odsek za tehnologije znanja, Institut Jožef Stefan«
Jamova 39, 1000 Ljubljana

Povzetek

V prispevku predstavljamo prosto dostopni spletni portal Termania, ki je namenjen iskanju po slovarskih zbirkah ter izdelavi in urejanju slovarskih gesel. Cilj portala je postati centralno mesto zbiranja predvsem terminoloških in tudi drugih podatkov slovarske narave za slovenščino, v kasnejši fazi pa tudi za druge jezike. Ciljna publika portala so vsi uporabniki spleta, zato je posebna pozornost namenjena uporabniški prijaznosti orodij, ta osnovna naravnanost pa hkrati omogoča tudi rabo zahtevnejših funkcij, kot je uredniški nadzor nad vsebino, uporabo jezikovnih tehnologij za pridobivanje leksikalnih podatkov iz besedilnih korpusov in druge napredne funkcije.

Termania - freely available online lexical portal

The paper describes Termania, a free dictionary portal with a search engine and an online dictionary editor. Termania is designed to become a central exchange place for terminological and other lexicographic data for Slovene and subsequently for other languages. The portal is aimed at general web users and thus the primary goal of its design is user-friendliness which on the other hand does not prevent the implementation of more advanced features such as editorial management system, the use of language technologies for language data extraction from text corpora and similar.

1. Uvod

Splošni, terminološki in drugi slovarji oz. splošneje rečeno podatkovne zbirke leksikalne in terminološke narave se vse bolj selijo iz tradicionalne knjižne oblike v elektronske medije, pri čemer že nekaj časa postaja zastarel tudi njihov prvi statični elektronski medij – CD-ROM ali DVD (Krek, 2003; Arhar in Krek, 2010). Razmah interaktivne rabe spleta, ki ga po eni strani uteleša izjemen uspeh Wikipedije in socialnih omrežij, po drugi pa selitev mnogih, v precejšnji meri brezplačnih storitev na splet, kot so Googlova ali Microsoftova orodja – iskalniki, odjemalci za elektronsko pošto, prevajalniki itd., narekuje tudi bodoča pričakovanja uporabnikov spleta in hkrati z dostopnostjo istih vsebin tudi uporabnikov mobilne telefonije.

Hkrati z možnostjo interaktivne rabe spleta je nastalo veliko število portalov, ki ponujajo prost dostop do svojih leksikografskih ali terminoloških vsebin, tudi z možnostjo urejanja vsebine (Wictionary, Urban Dictionary itd.) in to je posledično privedlo do tega, da so vsaj pri angleškem jeziku, pa tudi pri drugih jezikih, mnogi lastniki kvalitetnih slovarskih vsebin svoje slovarje dali na splet v brezplačni obliki, ta oblika pa je postala privzeta, pričakovana in uporabljana s strani večine uporabnikov.

V splošnem trendu se je kot glavna ovira izmenjave pri leksikografskih ali terminoloških podatkih za razliko od enciklopedičnih podatkov, izkazala relativno večja zadrega s konsistentnostjo, če velika množica ustvarja enotno podatkovno zbirko. Eden od možnih odgovorov na to zadrego je portal, ki dopušča hkratni obstoj velike množice v sebi zaključenih leksikografskih ali terminoloških baz, ki nimajo enotne strukture, imajo pa na drugi strani enoten način prikaza in iskanja po celoti podatkov, ki so na voljo.

S tem namenom je bil oblikovan portal Termania, ki ga predstavljamo v nadaljevanju prispevka.

2. Namen članka

Namena članka je predstavitev prosto dostopnega slovarskega portala Termania, ki ga je razvilo podjetje Amebis v sodelovanju z zavodom Trojina, dostopen pa je na naslovu www.termania.net. Vsebuje iskalnik in enostaven, a hkrati tudi vsestransko zmogljiv sistem za interaktivno urejanje slovarjev, v bližnji prihodnosti pa še sistem za korpusno podporo, forum ter še nekatere možnosti, ki bodo izdelovalcem in uporabnikom omogočale maksimalno uporabniško izkušnjo pri delu s slovarji. Portal je namenjen predvsem splošnim uporabnikom, brez specializiranega znanja s področja računalništva ali leksikografije, ki jim je skupna želja po izmenjavi terminološkega ali splošnega jezikovnega znanja – bodisi prevodov v dvo- ali večjezikovnem okolju, bodisi definicij v enojezikovnem kontekstu. Seveda pa je ambicija portala hkrati zadostiti tudi osnovnim potrebam profesionalnih uporabnikov, tako tistih, ki slovarje le uporabljajo (npr. prevajalci), kot tistih, ki slovarje soustvarjajo.

3. Slovarski spletni iskalnik

Iskanje je osnovna funkcija slovarskega portala Termania, ki omogoča uporabo informacij, vsebovanih v dostopnih slovarskih zbirkah. Prav hkratno iskanje po vseh (izbranih) slovarjih, ne glede na njihovo strukturo, je ena od bistvenih prednosti pred drugimi slovarskimi portali.

3.1. Osnovno iskanje

Osnovno iskanje omogoča enostavno in hitro poizvedovanje po slovarjih, dostopnih na portalu Termania, brez dodatnih iskalnih pogojev, ki pa kljub temu daje najboljše možne zadetke, prilagojene večini uporabnikov.

Osnovna iskalna stran je enostavna, primerljiva s klasičnimi spletnimi iskalniki, z iskalnim oknom na sredini in nekaterimi dodatnimi možnostmi: povezava na tipkovnico za vnos posebnih znakov in povezava na napredno iskanje. Osnovna stran, kakor tudi večina ostalih strani, vsebuje še povezave na druge dele portala: strani za delo s slovarji, registracija in prijava, izbira jezika vmesnika, oglaševanje in informacije o portalu.

3.2. Napredno iskanje

S pomočjo naprednega iskanja ima uporabnik možnost omejiti iskanje na določeno lastnost, kot je element slovarja (iztočnica, prevod, drugo), jezik, področje ali slovar.

Tako pri osnovnem, kot pri naprednem iskanju, lahko poleg iskanja ene besede iščemo tudi po več besedah (besednih zvezah), uporabljamo pa lahko tudi posebne znake, s pomočjo katerih lahko še dodatno razširimo ali omejimo iskanje. Za razširitev lahko uporabimo znaka »?« in »*«, ki nadomestita en znak oz. poljubno število znakov, iskanje pa lahko omejimo z narekovaji »""«, ki določajo točno določen vrstni red besed, ki stojijo skupaj.

3.3. Prikaz zadetkov

Rezultati iskanja se prikažejo glede na iskalni pogoj. Izpišejo se skrajšana gesla, oz. le tisti podatki, ki uporabnika običajno najbolj zanimajo: iztočnica, prevod(i) in definicija, vsi ostali podatki pa le izjemoma glede na strukturo posameznega slovarja.

Razvrstitev zadetkov je bistvenega pomena za uporabnost portala, kot se je to npr. pokazalo tudi pri običajnih spletnih iskalnikih. Pri razvrščanju zadetkov se zato upošteva več kriterijev:

- izbrani jezik vmesnika (uporabniki različnih jezikovnih vmesnikov pričakujejo različen vrstni red zadetkov);
- mesto zadetka v geselskem članku (iztočnica, prevod, drugo);
- teža oz. pomembnost slovarjev ali posameznih gesel;
- abecedni vrstni red.

Na vrstni red zadetkov lahko dinamično vplivajo tudi sami uporabniki s pomočjo dodanih mehanizmov ocenjevanja posameznih gesel in/ali slovarjev.

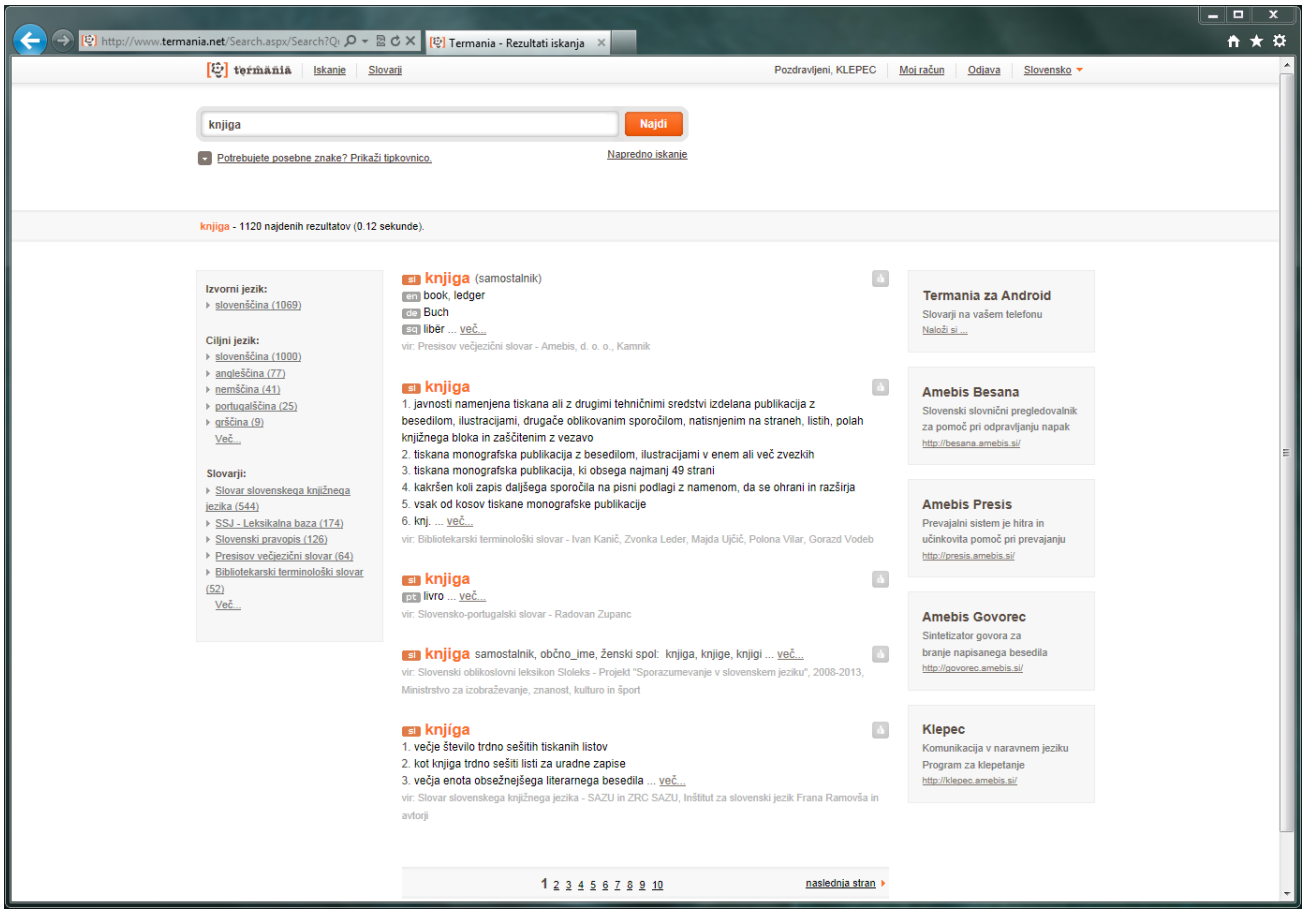
Ob seznamu zadetkov se na levi strani prikaže tudi filter, s pomočjo katerega lahko hitro in enostavno skrčimo nabor zadetkov po treh kategorijah. Izvorni jezik določi jezik iskalnega niza, ciljni jezik določi jezik, ki ga vsebujejo zadetki, te pa lahko dodatno omejimo tudi na izbrane slovarje.

3.4. Prikaz gesla

Na zahtevo lahko uporabnik vidi celotno geslo. To se izpiše na način, ki ga glede na strukturo in vsebino določa vsak slovar posebej. Zaradi enotnejšega izgleda so določeni le osnovni elementi prikaza, kot so uporabljena pisava, velikosti in barve.

3.5. O slovarju

Za vsak dostopen slovar je mogoče videti osnovne informacije, kot so npr. lastnik, področje, datum kreiranja, datum zadnje spremembe in trenutno število gesel, lahko pa tudi podatke o zgodovini nastanka, celotni uredniški ekipi, podrobnosti o strukturi gesel ipd. Vse te in podobne



informacije so poleg same vsebine gesel za uporabnike lahko zelo pomembne za ustvarjanje celovite podobe o posameznem slovarju.

4. Ustvarjanje in urejanje slovarjev

Možnost kreiranja in urejanja slovarjev je tisti del portala Termania, ki je namenjen avtorjem in urednikom slovarjev. Delo s slovarji je zasnovano tako, da ga lahko uporabljajo tako običajni uporabniki portala, ki jim je urejanje slovarjev le hobi, kot tudi profesionalni leksikografi, ki pa bi jim za resnejše urejanje slovarjev prav prišle še številne dodatne funkcije, ki trenutno še niso vgrajene.

4.1. Seznam slovarjev

Slovarji, do katerih lahko dostopa uporabnik, so razporejeni v tri skupine. »Moji« so tisti, v katerih je uporabnik del uredniške ekipe, »Vsi« so vsi za iskanje odprti slovarji (ne pa tudi urejanje), »Priljubljeni« so tisti, ki jih uporabnik iz prvih dveh skupin sam izbere za njegove priljubljene. Preko kateregakoli od teh treh seznamov lahko uporabnik dostopa do informacij o slovarjih, do njihovih nastavitev in do urejanja gesel, če ima za to seveda pravico. To mu lahko dodeli glavni urednik ali administrator slovarja.

Za zdaj so vsi slovarji v sistemu narejeni po meri, kmalu pa bo dodan poseben čarovnik, s pomočjo katerega bodo registrirani uporabniki lahko sami ustvarjali nove slovarje v nekaj enostavnih korakih. Potrebno bo le izbrati ustrezno predlogo, ter določiti ime in področje.

Konec julija 2012 je bilo javno dostopnih 21 slovarjev: Amebisov slovar rim, Amebisov slovar sopomenk, Besedišče slovenskega jezika, Bibliotekarski terminološki slovar, Grško-slovenski slovar, Latinsko-slovensko-angleška slovarska zbirka pojmov iz srednjeveških notarskih knjig, Nemško-slovenski avtomobilistični slovar, Portugalsko-slovenski slovar, Presisov večjezični slovar, Slovar družboslovne informatike, Slovar Izolskega slenga, Slovar krajšav, Slovar slovenskega knjižnega jezika, Slovenski oblikoslovni leksikon Sloleks, Slovenski pravopis, Slovensko-nemški slovar, Slovensko-portugalski slovar, Slovensko-ukrajinski slovar, Taksonomski slovar, Ukrajinsko-slovenski slovar in Vezljivostni slovar slovenskih glagolov, v pripravi pa je bilo že 8 novih, med njimi tudi Turistični slovensko-angleški terminološki slovar, Slovensko-angleški glosar jezikoslovnega izrazja in Slovenski medicinski slovar.

4.2. Nastavitve slovarja

Nekatere nastavitve slovarja lahko uporabnik kasneje tudi spreminja oz. dopolnjuje. Tako npr. lahko dodaja nove sodelavce, spreminja njihove pravice pri urejanju slovarja, spremeni ime, področje slovarja ali njegov izgled in dodaja nove znake v (slovarsko) tipkovnico.

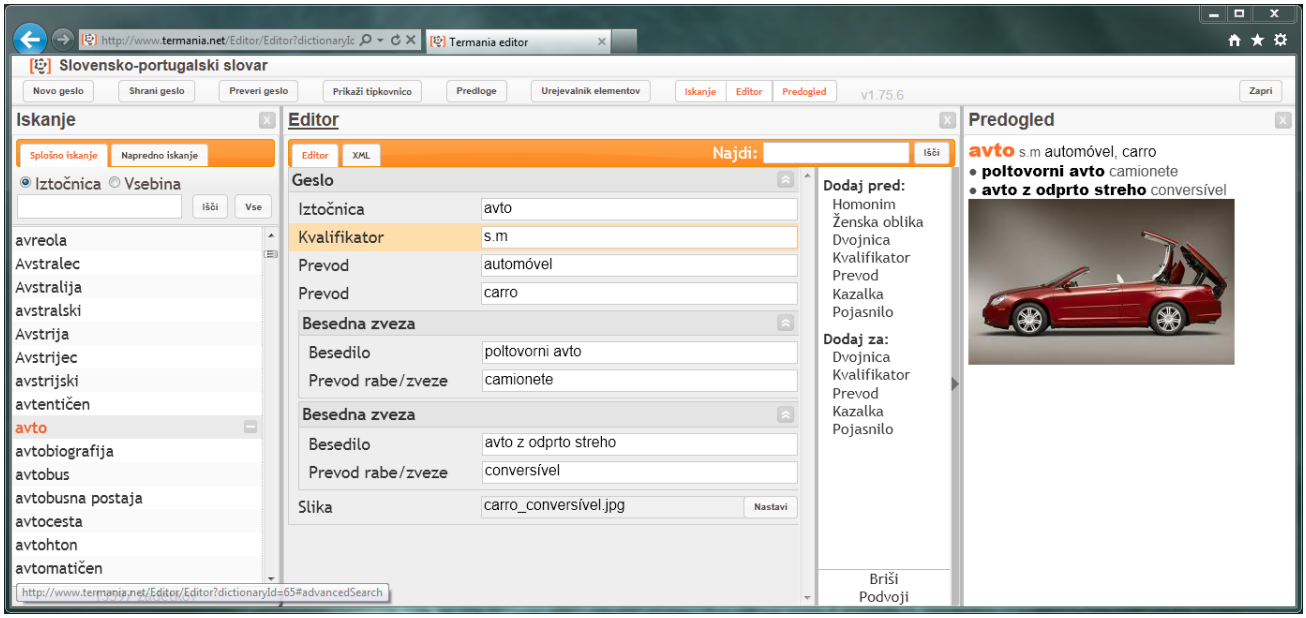
4.3. Urejanje gesel

Urejevalnik je namenjen dodajanju, spreminjanju in brisanju slovarskih gesel. Okno urejevalnika je razdeljeno na tri glavne dele:

- osrednji del, namenjen oblikovanju strukture gesla in vnosu vsebine. Na voljo sta dva načina vnosa: s pomočjo vnosnih polj/mask, in tekstovni XML način. Oba vnosa se preverjata s pomočjo ustrezne XML sheme, ki jo predhodno definira uporabnik;
- iskalni del, kjer uporabnik vnese osnovni ali zahtevnejši iskalni pogoj in s pomočjo seznama zadetkov izbira posamezna gesla za urejanje;
- predogled, kjer se geslo, ki ga trenutno urejamo, prikazuje v končni obliki.

Seveda je mogoče poljubno spreminjati tudi položaj in velikost posameznih delov (oken) urejevalnika – povečevati, pomanjševati ali povsem skriti, ter na ta način delovno površino povsem prilagoditi svojim potrebam in velikosti ekrana.

Pri vnosu s pomočjo vnosnih polj/mask se njihova velikost (višina) samodejno prilagaja glede na količino besedila, ki ga element vsebuje, kar omogoča pregledno, enostavno in hitro urejanje posameznih elementov gesla. Pri izbiri elementa se na desni strani pokaže seznam elementov, ki jih lahko izbranemu elementu dodamo, bodisi pred njega, v njega ali za njega, odvisno od slovarske sheme. Poleg elementov je mogoče tudi dodajanje predlog, ki jih lahko uporabnik definira sam.



Za zahtevnejše in bolj vešče uporabnike je na voljo tudi možnost neposrednega XML urejanja slovarskih gesel, ki omogoča hitrejše in prožnejše delo.

Dodane so še nekatere funkcionalnosti, ki dodatno olajšajo urejanje. Med njimi je prikaz izbrane virtualne tipkovnice, s pomočjo katere je omogočen hiter vnos posebnih znakov, ki jih na tipkovnici običajno ni. Potem je tukaj možnost urejanja prikaza elementov v predogledu, s pomočjo katerega si uporabnik po lastni želji prilagodi velikost in barve izpisanih elementov. Ne nazadnje je dodana tudi funkcionalnost ustvarjanja in urejanja lastnih predlog, ki se je pokazala še kot posebej koristna pri učinkovitem ustvarjanju in urejanju gesel.

Urejevalniku bodo sčasoma dodane še nekatere druge splošno koristne funkcije, kar bo urejanje na portalu Termania še dodatno obogatilo. Take funkcije so npr. zgodovina in primerjava sprememb urejanja, izpis frekvenčnih spiskov vsebine izbranih elementov ter izvoz (tiskanje) gesel, kjer bo uporabnik lahko izvozil (natisnil) izbrana gesla oz. slovar v celoti.

5. Druge funkcije

Poleg obeh glavnih delov (»iskanje« in »slovarji«) portal Termania vsebuje še nekatere dodatne funkcionalnosti, potrebne ali koristne za učinkovito delo s slovarji.

5.1. Registracija in prijava

Iskanje po slovarjih in urejanje tistih slovarjev, ki so s strani njihovih lastnikov označeni kot javno dostopni, so dostopni vsakemu, tudi neregistriranemu uporabniku. Večino ostalih funkcionalnosti pa zahteva registracijo oz. prijavo uporabnika z uporabniškim imenom in geslom, saj je le na ta način mogoče zagotoviti ustrezno varstvo podatkov in kontrolirano omejen dostop do njih.

5.2. Oglaševanje

Za vzdrževanje in delovanje portala so seveda potrebna določena sredstva, ki jih ob brezplačnosti vsebin lahko pridobimo tudi s pomočjo oglaševanja, zato smo v portal vgradili tudi prikaz (izključno besedilnih) oglasov, ki se lahko spreminjajo glede na prikazane slovarske podatke.

5.3. O portalu

Uporabniki vse funkcionalnosti portala Termania uporabljajo v skladu s splošnimi pogoji uporabe, kjer so določene obveznosti in pravice uporabnika na eni strani, ter ponudnika storitve na drugi strani. Za večje in pomembnejše uporabnike, ali pa take, ki imajo posebne zahteve in želje, pa je mogoče skleniti tudi posebno pogodbo o uporabi.

6. Zaključek

Portal Termania omogoča brezplačno hkratno iskanje in interaktivno urejanje strukturno povsem različnih slovarjev na enem samem mestu. Slovarskega portala s primerljivimi lastnostmi v svetu še ni. Nekateri portali sicer omogočajo hkratno iskanje po različnih slovarjih, vendar ne omogočajo njihovega interaktivnega urejanja, drugi omogočajo urejanje, vendar ne hkratnega iskanja, tretji pa omogočajo iskanje in urejanje po strukturi enakih, ne pa tudi različnih slovarjev.

Opisane lastnosti omogočajo, da na portalu Termania lahko brez omejitev združimo številne obstoječe slovarje z novimi, ter na enem mestu uporabnikom omogočimo dostop do različnih slovarskih informacij, avtorjem pa dajemo orodje za enostavno gradnjo slovarjev po njihovih željah.

Namen portala Termania je tudi zapolniti vrzel, ki je po zapiranju slovarskih oddelkov številnih založb nastala med ustvarjalci in uporabniki slovarjev, izkoristiti razširjenost in zmogljivosti spleta, ter postati centralno mesto zbiranja in nastajanja predvsem terminoloških slovarjev. Mesto, kjer bi o slovarjih, pa tudi posameznih geslih potekale razprave, se izmenjevale informacije in zbirali komentarji uporabnikov, vse z namenom, da bi bili v njih zbrani podatki čim boljši in uporabnejši.

Za zdaj je vmesnik v slovenskem in angleškem jeziku, kasneje pa bo portal razširjen z večjezikovnim uporabniškim vmesnikom, z namenom širitve na druga jezikovna področja.

Glavna usmeritev portala je ponujanje podatkov pod pogoji »Creative Commons Attribution Non-Commercial Share Alike«, čeprav bo mogoče vključevati tudi druge licenčne modele po željah pomembnejših ponudnikov slovarskih vsebin.

7. Literatura

Arhar, Š., Krek, S., 2010. Slovenski besedilni korpusi: kako v razred?, *Sodobna pedagogika*, številka 1, februar 2010, 224-241.

Krek, S., 2003. Jezikovni priročniki in novi mediji, *Jezik in slovstvo*, letnik 48, št. 3–4, maj–avg. 2003, 29–46.

Izdelava slovensko-srbskega vzporednega korpusa podnapisov za razvoj strojnega prevajanja v projektu SUMAT

Mirjam Sepesy Maučec, Marko Presker, Danilo Zimšek, Matej Rojc, Damjan Vlaj, Darinka Verdonik, Zdravko Kačič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova 17, SI-2000 Maribor
mirjam.sepesy@uni-mb.si, marko.presker@uni-mb.si, danilo.zimsek@uni-mb.si, matej.rojc@uni-mb.si, damjan.vlaj@uni-mb.si, darinka.verdonik@uni-mb.si, kacic@uni-mb.si

Povzetek

Prispevek predstavlja izkušnje pri izdelavi vzporednega korpusa podnapisov, namenjenega za učenje modelov strojnega prevajanja. Opisane so značilnosti izvornega gradiva, procesi predpriprave gradiva, ki vključujejo pretvorbe v enoten format in enotno kodiranje, identifikacijo jezika in poravnavanje datotek, tokenizacijo in razcep na povedi, ter postopki poravnavanja ter rezultati evalvacije poravnanih podnapisov in povedi. Vzporedni korpus podnapisov za srbščino in slovenščino je bil razvit v okviru evropskega FP7 projekta SUMAT, katerega cilj je razvoj spletne aplikacije za strojno prevajanje podnapisov.

Building the parallel Slovene-Serbian corpus of subtitles for machine translation in the SUMAT project

The paper describes experiences in building parallel corpus of subtitles, aimed for usage in machine translation. We describe characteristics of the source data, pre-processing (which includes conversion to common format and common coding, language identification and file alignment, tokenization and sentence splitting), subtitle and sentence alignment, and results of alignment evaluation. The parallel corpus of Slovene-Serbian subtitles was developed within the FP7 EU-funded project, named SUMAT, which aims to develop an online service for machine translation of subtitles.

1. Uvod

Podnaslavljanje je priljubljen način za posredovanje tujejezičnih multimedijskih vsebin v veliko evropskih državah in za večino žanrov. Trenutna evropska politika (European Commission, 2010) podpira podnaslavljanje v javnih televizijskih mrežah in posledično se je potreba po podnaslavljanju v avdiovizualni industriji v preteklih letih povečala (MCG, 2007).

Hkrati se podnaslavljanje srečuje s pomembnimi problemi, kot so visoki stroški, časovna potratnost in posledično vprašanje kvalitete podnapisov. Določene raziskave (npr. Volk, 2008; de Sousa et al., 2011) nakazujejo, da bi lahko v prevajanje podnapisov uspešno vključili strojno prevajanje in tako pomagali rešiti navedene probleme.

Trenutno ne obstajajo orodja, ki bi zagotavljala avtomatsko podnaslavljanje gradiv v tujem jeziku. Ena osnovnih ovir je pomanjkanje ustreznih vzporednih korpusov, potrebnih za razvoj modelov za strojno prevajanje. Eden redkih dostopnih korpusov je OPUS OpenSubtitle corpus (Tiedemann, 2009), ki pokriva precej evropskih jezikov, problem pa je, da temelji na odprto dostopnih prevodih s spleta, za katere ni nobenega zagotovila o njihovi kvaliteti.

Po drugi strani so profesionalni prevodi podnapisov večinoma last podjetij, ki se ukvarjajo s podnaslavljanjem in ki praviloma skrbno ščitijo te vire, zato je do njih izredno težko dostopati. Prav tako so formati teh podnapisov zelo različni in nekateri od njih lastniški, npr. Softelovi .o32, .x32 in .s32, Screenov .890, Poliscriptov .pac ali EZTitlesov .ezt.

Z namenom, da se izdela spletno aplikacijo za podnaslavljanje za različne evropske jezike, se je v letu 2011 začel evropski projekt SUMAT[1] (An Online Service for SUbtitling by MAchine Translation), katerega partner je poleg podjetij, ki se ukvarjajo s podnapisi, ter tujih raziskovalnih institutov tudi Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko. V projektu bodo pokriti naslednji jezikovni pari: nemško, francosko, špansko, nizozemsko, švedsko in portugalsko v povezavi z angleščino ter slovensko v povezavi s srbščino[2]. Spletna aplikacija bo komercialni produkt[3], za nekomercialno rabo bo odprta le v okrnjeni funkcionalnosti.

Eden od pomembnih korakov pri izdelavi sistema za strojno prevajanje podnapisov je izdelava vzporednega korpusa podnapisov, potrebnega za učenje prevajalnika. Namen tega prispevka je predstaviti izkušnje pri izdelavi slovensko-srbskega vzporednega korpusa podnapisov v okviru projekta SUMAT. Poravnavanje podnapisov ima v primerjavi s poravnavanjem drugih besedil v vzporedni korpus kar nekaj posebnosti, kot so različni formati izvornih datotek, jezikovne posebnosti prevajanja podnapisov, ki je znano po mnogih redukcijah, preklapljanje med povedmi in podnapisi (podnapis namreč ne sovpada s povedjo) itn.

Članek v nadaljevanju predstavlja značilnosti izvornega gradiva v drugem poglavju, v tretjem poglavju

[1] http://www.sumat-project.eu; projekt financira EU po pogodbi ICT-PSP-270919.

[2] Za slovenski jezik pri prevajalskih podjetjih ni bilo ustrezne količine angleških podnapisov, uporaba para slovenščina-srbščina pa je predvidena predvsem kot posredno prevajanje med pari angleščina-slovenščina in angleščina-srbščina, kadar za posameznega od teh parov že obstaja prevod.

[3] Komercialna narava končnega produkta je med drugim pogojena z vrsto evropskega programa, v okviru katerega je projekt sofinanciran, to je ICT PSP (http://ec.europa.eu/information_society/activities/ict_psp/about/index_en.htm).

potrebne korake predpriprave gradiva ter v četrtem poglavju izkušnje iz poravnavanja in rezultate evalvacije.

2. Izvorno gradivo

Izvorno gradivo so iz svojih arhivov posredovala tri mednarodna podjetja, ki so specializirana za prevajanje podnapisov. Podjetja so zagotavljala, da gre za visoko kvalitetne podnapise, saj je vsak prevod pregledan na več nivojih, preden je posredovan naročniku.

Prejšnji eksperimenti (Volk, 2008) so pokazali, da je kvaliteta avtomatskih prevodov močno odvisna od žanra filma. Ločevanje gradiva po žanrih je temeljilo na klasifikaciji gradiva pri gradivodajalcih, tako da so že ti posredovali datoteke ločene po posameznih žanrih. V osnovi so bili žanri razdeljeni na tiste, ki izhajajo iz vnaprej napisanih scenarijev (ang. scripted), kot so na primer dokumentarni filmi, serije, novice, in tiste, ki predstavljajo spontano govorjene žanre (ang. unscripted), na primer pogovorne oddaje, intervjuji itd. Skupaj so bili žanri razdeljeni na 22 domen.

Poleg datotek s prevodi smo zbirali tudi samo enojezične datoteke, saj je pomembna komponenta prevajalnika tudi jezikovni model.

Posredovanje datotek je potekalo prek FTP-strežnika.

Za jezikovni par slovenščina-srbščina je bilo zbranih 825 datotek, ki so obsegale skupno 169.654 podnapisov v slovenščini in 219.139 podnapisov v srbščini. Datoteke so pripadale naslednjim žanrom: novice, serije in dokumentarni filmi. Več kot pol datotek za par slovenščina-srbščina ni imelo opredeljenega žanra in so bile kategorizirane kot drugo.

Že začetna statistična analiza je pokazala, da imajo v povprečju srbske datoteke več ponapisov kot slovenske. Pozneje se je tudi pokazalo, da datoteke slovensko-srbskega jezikovnega para niso nastale z navzkrižnim prevajanjem, ampak sta slovenski in srbski prevod nastala neodvisno drug od drugega, in sicer na podlagi avdio datoteke (ne na podlagi pisnega scenarija!) v angleščini. To dejstvo je prineslo številne težave v nadaljnji obdelavi gradiva, tako da kljub sicer jezikovno brezhibnemu materialu zbrano gradivo ni zagotavljalo želenih lastnosti za učenje strojnega prevajanja. Tudi količina zbranega gradiva za slovenščino in srbščino je bila manjša kot za druge jezikovne pare.

Televizijski podnaslovi pa so tudi sicer jezikovno nekoliko posebni. Podnaslavljanje je namreč zaznamovano z omejitvami prostora (največ dve vrstici) in časa, zapisovanjem posebnosti govora, prisotnostjo slike in s tem, da pri prevajanju ni nujno osnova izvorni tekst (scenarij), ampak je lahko to tudi zvočni/video posnetek. Zato je pri podnaslavljanju, kot pravi Kovačič (1996: 298), vprašanje ne samo, kako prevesti, ampak najprej kaj prevesti in kaj izpustiti. Redukcije, zgoščevanja, parafraziranje, tako na besedni kot na stavčni ravni, vse to je zelo pogost element podnaslavljanja. V gradivu projekta SUMAT so zato zelo pogosti prevodni pari takšni:

SL: *Naslednjega kar testirajte.*
SR: *Sledeći put kada budem našla dečka, odvešću ga tamo.*

ali:

SL: *Izjavljam, da se bom upokojil.*
SR: *Sada zvanično podnosim ostavku!*

Vse te značilnosti predstavljajo težavo pri poravnavanju in tudi pri nadaljnji uporabi gradiva za učenje modelov strojnega prevajanja.

3. Predpriprava gradiva

Predpriprave izvornega gradiva vključujejo naslednje korake: pretvorbe v enoten format in enotno kodiranje znakov, identifikacijo jezika v datotekah, poravnavanje datotek, tokenizacijo in razcep po povedih.

3.1. Pretvorbe v enoten format

Za učenje prevajanja je v projektu SUMAT uporabljeno orodje MOSES (Koehn idr., 2007), ki zahteva vhodne datoteke v formatu txt. To je bil posledično ciljni format datotek tudi v korpusih SUMAT.

Izvorne datoteke so bile pretežno v formatu pac in različnih izpeljankah formata txt, zato je bil razvit namenski program, ki je vse datoteke pretvoril v enoten txt-format. Največ težav pri pretvorbi je bilo z datotekami v formatu pac, ki je lastniški.

Za kodiranje znakov je bil sprejet dogovor, da bo enoten format UTF-8. Partnerji v projektu so izdelali namenski program za detekcijo in pretvorbo kodiranja. V nekaterih slovensko-srbskih datotekah smo kljub temu zaznali, da imajo črke c, č in ć ter d in đ enake kode. Te datoteke so bile iz korpusa izločene, saj popravljanje te napake ni izvedljivo na preprost način.

Po teh pretvorbah je bila vsebina datotek predstavljena na način, kot prikazuje slika 1. Vsak podnapis ima zaporedno številko, sledita mu časovni kodi za začetek in konec. Besedilo podnapisa je zapisano v eni ali dveh vrsticah.

```
0003    00:00:18:05    00:00:25:15
Ne vem . Ne trdim ,
da vse vem ali da se dobro poznam .

0004    00:00:25:21    00:00:29:21
Vsak dan se spoznavam . O nekom
ne moreš imeti napačne predstave .
```

Slika 1: Izsek iz datoteke v poenotenem formatu

3.2. Identifikacija jezika in poravnavanje datotek

Prenos datotek iz arhivov podjetij na projektni strežnik je bil izveden ročno. To pomeni, da napake pri prenosu niso izključene. Čeprav je del imena datoteke tudi koda jezika, smo jezik v dokumentu identificirali s programom Lingua:Ident (http://search.cpan.org/\~mpiotr/Lingua-Ident-1.6/Ident.pm), ki temelji na verjetnostnem algoritmu na osnovi trigramov črk. Program je bil učen na korpusu OpenSubtitle v.2 za srbščino (Tiedemann, 2009) in Europarl v.6 za slovenščino (Koehn, 2005). S pomočjo omenjenega programa smo iz korpusa uspešno izločili dve napačni datoteki.

Če so datoteke s prevodi generirane iz iste predloge, je upravičeno pričakovati, da se časovne kode v datotekah ujemajo, zato smo v prvem koraku razvili program za poravnavanje datotek na osnovi podobnosti časovnih kod. Program temelji na dinamičnem programiranju in pri primerjavi časovnih kod upošteva vnaprej definirano odstopanje. V projektu je bil sprejet dogovor, da je

dovoljeno odstopanje do 1 sekunde. Vendar program pri slovensko-srbskem jezikovnem paru ni bil uspešen, saj je zaznal le 8 parov datotek od 380. Razlog je bil, da so slovenski in srbski prevodi v SUMAT-ovem gradivu nastajali neodvisno drug od drugega in se časovne kode niso ujemale.

Poravnavanje datotek smo tako izvedli na osnovi imen datotek. Analiza imen je namreč pokazala, da imajo datoteke, ki so pari, v delu imena enako kodo. Tabela 1 kaže primer takih parov datotek.

Datoteka s slovenskim prevodom	**Datoteka s pripadajočim srbskim prevodom**
Glamour's 50 Biggest Fashion Do's and Don'ts_101_Glamour's 50 Biggest Fashion Do's and Don'ts_GBFD0101A_SLV.PAC	GBFD_101_Glamour's 50 Biggest Fashion Do's and Don'ts_GBFD0101A_SRP.PAC
TOO YOUNG TO KILL 15 SHOCKING CRIMES_TOO YOUNG TO KILL 15 SHOCKING CRIMES_101_TYKA0101A-SLV_NEW.pac	TYK_Too Young to Kill - 15 Shocking Crimes_TYKA0101A-SRP_NEW.PAC

Tabela 1: Poravnavanje datotek na osnovi imen datotek

Datoteke, pri katerih v imenu nismo avtomatsko zaznali skupnega niza znakov, je bilo treba poravnati ročno. Tako smo dobili na koncu 380 parov datotek.

3.3. Tokenizacija in razcep po povedih

Prvi korak na vhodnem tekstu predstavlja tokenizacija. Vhod v modulu za tokenizacijo je besedilo v standardu UTF-8.

Vse pomenske enote besedila smo opisali z uporabo regularnih izrazov, tudi morebitne okrajšave in akronime. Detekcija okrajšav in akronimov je podprta z naborom okrajšav in akronimov, predstavljenim v obliki končnega stroja (Finite State Machine – FSM), ki je bil sestavljen na osnovi korpusa FidaPLUS (www.fidaplus.net).

Glavni del tokenizatorja je končni stroj, ki smo ga uporabili v vlogi procesa tokenizacije (prim. Rojc, 2007). Osnovno abecedo smo najprej razširili z znaki UTF-8, npr.: alphabet1 [A-Za-z\\0-\x7F\\xC2-\xDF\\x80-\xBF\^]. Modul tokenizacije je izveden v obliki povezanega seznama (linked list – dequeue), saj mora povezovati in obdelovati več virov informacij med procesom tokenizacije. Tudi na tem nivoju je bilo treba poskrbeti za podporo procesiranju nizov UTF-8.

Proces tokenizacije se izvaja znotraj zanke, vse dokler imamo na vhodu besedilo. Tokeni se med procesiranjem shranjujejo v povezani seznam, dokler ne zaznamo konec povedi. V okviru projekta smo predvideli naslednje tipe tokenov: ločila, besede, akronime, glavne številke, vrstilne številke, decimalna števila itd. Normalizacija tokenov v projektu ni bila potrebna, zato smo jo onemogočili.

```
0       00:00:57,007    00:00:59,924
Peter, general prihaja.

1       00:01:00,051    00:01:02,421
Kako gre? -Ne preveč dobro.
0       00:00:56,484    00:00:59,444
Piter, general dolazi.

1       00:00:59,524    00:01:01,964
Kako ide sada? -Ne baš dobro, gospodine.
```

(a)

```
0 00:00:57:00 00:00:59:23
Peter , general prihaja .

1 00:01:00:01 00:01:02:10
Kako gre ? -Ne preveč dobro .
0 00:00:56:12 00:00:59:11
Piter , general dolazi .

1 00:00:59:13 00:01:01:24
Kako ide sada ? -Ne baš dobro , gospodine .
```

(b)

```
peter , general prihaja .
kako gre ?
-ne preveč dobro .
piter , general dolazi .
kako ide sada ?
-ne baš dobro , gospodine .
```

(c)

Tabela 2: Vhod za tokenizacijo (a), rezultat po tokenizaciji (b) in razcep na povedi (c)

Smo pa v mehanizem tokenizacije vključili posebna tokena za označevanje konca povedi in konca vhodnega besedila: EOS in EOF. Token EOS predstavlja samo možen konec povedi, za končno potrditev konca se v naslednjem koraku preveri tudi širši kontekst. Po procesiranju vsake povedi vhodnega besedila se preverja tudi, ali je že nastopil token za konec vhodnega besedila. Če še ni, se nadaljuje proces tokenizacije na naslednji povedi. Predstavljeni mehanizem tokenizacije in razcep po povedih se da preprosto izvesti z uporabo povezanega seznama in pripadajočih funkcij: *gettoken()*, *pushtoken()* in *Fill*(). Funkcija *Fill()* tako uporablja končni stroj in kopiči tokene v povezanem seznamu, jih analizira, opazuje konktest itd., kar je v veliko pomoč v procesu razcepa na povedi. Prvi korak določanja konca povedi izvaja sam končni stroj (označi možen token EOS). Na nivoju povezanega seznama pa nato ob upoštevanju desnega in levega konteksta podamo končno odločitev. Za končni stroj so možni nastopi konca povedi (EOS) na danem vhodnem besedilu: ločila (.!?...), kombinacija znakov \n\n, če se hkrati naslednji token začne z veliko črko ter če token z ločilom (.) ni okrajšava ali akronim. Problem v tem primeru predstavljajo npr. lastna imena, kar smo reševali z obsežno zbirko lastnih imen (Onomastica, slovar LC-STAR (dostopna tudi prek ELDE)).

Vhodne datoteke so vključevale tudi specifične tokene, ki jih je bilo treba ustrezno detektirati in procesirati ter prikazati na izhodu tokenizacije, zlasti številčenja segmentov in časovne kode. Tako smo morali generirati več formatov izhodov. Za en del korpusa je bilo treba to informacijo izločiti, za drugi del korpusa pa ohraniti in ustrezno dodati označenim povedim. Tudi te specifične tokene smo opisali z uporabo regularnih izrazov. Primer vhoda/izhoda modula za tokenizacijo nazorneje prikazujemo v tabeli 2.

Procesu tokenizacije je sledila še pretvorba v male črke.

4. Poravnavanje

Na voljo so različna orodja za avtomatsko poravnavanje besedil po povedih. Gale-Churchev poravnalnik (1993) temelji na verjetnosti, izračunani za vsak par povedi glede na dolžino povedi (število znakov). Uporabljen je bil npr. za poravnavanje korpusov Europarl (Koehn, 2005) in JRC-Acquis (p://langtech.jrc.it/JRC-Acquis.html). Moorov (2002) dvojezični poravnalnik kombinira dolžino povedi in število besed in je uspešen predvsem pri čim daljših korpusih. Hunalign (Varga et al., 2005) kombinira dolžino povedi in slovar, če ni slovarja, pa ga nadomesti z verjetnostmi, izračunanimi na podlagi korpusa. Deluje samo na korpusih z do 20.000 povedmi, daljše korpuse pa razbije na manjše dele. Gargantua (Braune, Fraser, 2010) podobno kot Moorov poravnalnik temelji na podobnosti povedi, razlike so v iskalnih strategijah in klestenju iskalnega prostora. Bleualign (Sennrich, Volk, 2010) uporablja strojni prevajalnik izvornega teksta, zato je manj primeren za uporabo, če tega ni na voljo. Podrobneje primerjajo navedena orodja Abdul-Rauf idr. (2010) in ugotavljajo, da se najbolje obnesejo Bleualign, Gargantua in Hunalign. Ker je za prvega potreben strojni prevajalnik, za drugega pa večji korpus, smo se v projektu SUMAT odločili za poravnalnik Hunalign.

Pri poravnavanju korpusa podnapisov imamo na izbiro dve osnovni enoti poravnavanja: podnapis ali poved. Iz teorije statističnega prevajanja vemo, da so za algoritem učenja primernejše krajše enote. V splošnem so podnapisi krajši od povedi, kakršne so sicer značilne za pisna besedila. Toda če podnapise združimo in razcepimo po povedih, ni nujno tako, saj so posamezne povedi v podnapisih pogosto samo eno- ali dvobesedne fraze, ki so značilne za govorno komunikacijo. Analiza (glej tabelo 3) učnega korpusa SUMAT je pokazala, da so povedi v njem v povprečju krajše od podnapisov. Prav tako so v povprečju povedi/podnapisi v srbskem delu korpusa daljši.

	Slovenščina	Srbščina
dolžina povedi	6,5	6,8
dolžina podnapisa	8,2	8,5

Tabela 3: Povprečne dolžine povedi in podnapisov v učnem korpusu SUMAT

Korpus smo ločeno poravnali po povedih in po podnapisih, da bomo lahko v nadaljevanju primerjali uspešnosti prevajalnih sistemov na osnovi povedi in podnapisov.

4.1. Poravnavanje na osnovi besedila in na osnovi časovnih kod

Za poravnavanje povedi in podnapisov smo najprej uporabili pristop poravnave na osnovi besedila in besedilnih značilnosti s pomočjo orodja Hunalign (Varga et al., 2005). Poravnavanje poteka v več zaporednih iteracijah. Orodje tvori matrike poravnav in izračuna uteži za te poravnave. Te uteži temeljijo na podobnosti dolžine enote in morebitni prisotnosti različnih besed v slovarju. Orodje lahko, če slovarja ne vključimo v postopek poravnave, samo generira slovar, ki ga uporabi v kasnejših iteracijah. V našem primeru se je pokazalo, da generirani slovar ni preveč uporaben, vsako vključevanje lastnih slovarjev pa je rezultat poravnave le še poslabšalo. Primer poravnave na podlagi besedila je prikazan v tabeli 4.

Drugi pristop temelji na podlagi poravnavanja časovnih kod podnapisov. Pri poravnavanju povedi smo sami tvorili časovne kode začetka in konca povedi na podlagi števila besed v povedi. Pri poravnavanju podnapisov, in posledično tudi pri poravnavanju povedi, se je pokazalo, da je poravnavanje zelo odvisno od tolerance (tj. odstopanja časovnih kod), ki smo jo nastavili za še dopustno, saj nekatera prevajalska podjetja ne uporabljajo predlog, kar pomeni, da poleg prevoda spreminjajo tudi začetne in končne časovne kode. Premajhna toleranca pomeni premalo poravnanega materiala, prevelika toleranca pa privede do nepravilnosti, saj dopušča, da se kratke povedi ne poravnajo oz. da se po nepotrebnem dodajo poravnavi. Z našo skripto smo lahko zaznali poravnave 1:1 ali 1:N.

4.2. Problemi pri poravnavanju

Pri poravnavanju korpusa smo naleteli na številne težave. Izvirajo predvsem iz tega, da so prevajalci spreminjali časovne kode in razbijali podnapise na različne dele.

Ena od težav, ki se je pogosto pojavljala, je povezana z zelo kratkimi podnapisi, ki so predvsem v srbskem delu korpusa vključeni, v slovenskem delu korpusa pa izpuščeni. Problem je prikazan na primeru v tabeli 4.

Slovenski prevod	Srbski prevod
0020 00:02:01:11 00:02:06:11 Kako se počutiš v središču ? Je naporno ? Kako se spopadaš s tem ?	0029 00:02:01:20 00:02:05:11 Kako se osećaš zbog ovolike pažnje koju dobijaš ? Zar nije ludo ? 0030 00:02:05:14 00:02:06:15 Jeste ! 0031 00:02:06:19 00:02:07:24 Da li se dobro nosiš sa tim ?

Tabela 4: Prevodi – primer kratkih odgovorov

V srbskem delu korpusa vidimo, da 30. podnapis predstavlja odgovor, ki ga v slovenskem delu korpusa ne najdemo. Če opazujemo časovne kode, pa vidimo, da bi 20. slovenski podnapis poravnali z 29. in 30. srbskim podnapisom, 31. pa bi ostal neporavnan.

Nadaljnja težava je bila, da so povedi v obeh prevodih različno dolge. Slovenski prevodi imajo v povprečju krajše povedi kot srbski prevodi, zato je tudi časovni interval temu primerno krajši. Na primeru v tabeli 5 je prikazano, kaj to pomeni za poravnavanje.

Slovenski prevod	Srbski prevod
00:20:56:15 00:20:59:06 Je to človek ?	00:20:57:04 00:21:01:18 Je li to čovek tamo ili nešto slično ?

Tabela 5: Prevodi – primer različno dolgih povedi

Da bi povedi poravnali, bi potrebovali toleranco vsaj 2 sekund in 12 okvirov, kar pomeni skupaj 2,48 sekunde. Tako velika toleranca pa ni več smiselna, saj je predolga, nekatere povedi so celo krajše od tega.

Poleg navedenih težav so se pojavljale še druge, na primer neoznačena menjava govorca, manjkajoči sklop podnapisov, časovno zamaknjene datoteke ali časovno raztegnjene datoteke ipd.

4.3. Postopek evalvacije

Za evalvacijo poravnavanja je bil iz celotnega korpusa izločen testni nabor, in sicer za vsak jezikovni par 1.000 podnapisov in pripadajočih povedi. Material je bil izbran tako, da je odražal delež zastopanosti posameznih žanrov iz celotnega korpusa, in sicer:

- za evalvacijo poravnavanja povedi smo uporabili 50 vzporednih slovensko-srbskih zaporednih podnapisov, razcepljenih po povedih, iz 10 različnih parov datotek,
- za evalvacijo poravnavanja podnapisov smo uporabili 50 vzporednih slovensko-srbskih zaporednih podnapisov iz 10 različnih parov datotek.

Ročno poravnavanje testnega nabora po povedih in po podnapisih je izvajalo 6 oseb. Navodilo je bilo, da se poravnajo samo tiste povedi in podnapisi, ki jih je mogoče smiselno poravnati; če določen podnapis ali poved v drugem jeziku ni imel para, je ostal neporavnan. Pri ročni poravnavi smo tvorili dva tipa datotek, ene so vsebovale slovenske in druge vzporedne srbske podnapise. Neporavnani podnapisi so bili izločeni. Če je bil istopomenski podnapis v slovenskem jeziku predstavljen v eni vrstici, v srbskem jeziku pa v dveh, smo podnapis v srbskem jeziku predstavili v eni vrstici ter dodali tri znake "~~~" med združenima vrsticama. Na ta način smo lahko uporabili skripte, ki smo jih uporabljali za postprocesiranje poravnav, ki smo jih dobili s pomočjo programa Hunalign. Tabela 6 prikazuje primer takšne ročne poravnave.

Slovenski podnapis
Izvorni
Te pokličeva pozneje . Rada vaju imam !
Je to avtobus za žuranje ?
Ročno poravnan
Te pokličeva pozneje . Rada vaju imam !
Je to avtobus za žuranje ?

Srbski podnapis
Izvorni
Zvaćemo te kasnije .
Volim vas .
Da li je ovo autobus za zabavu ?
Ročno poravnan
Zvaćemo te kasnije . ~~~ Volim vas .
Da li je ovo autobus za zabavu ?

Tabela 6: Primer ročne poravnave podnapisov

Za boljši pregled nad poravnavami in lažjo evalvacijo smo ustvarili še tretjo datoteko, v kateri so bile oštevilčene vrstice podnapisov in usklajene glede na to, katere vrstice so dejanski vzporedni pari prevodov. Primer vsebine takšne datoteke je podan v tabeli 7.

Za slovenske podnapise	**Za srbske podnapise**
1	1
2	3

Tabela 7: Označevanje začetkov vrstic vzporednih prevodov

Številke v tabeli 7 pomenijo, da je vrstica 1 v datoteki s srbskim prevodom prevod vrstice 1 v datoteki s slovenskim prevodom, vrstica 2 v datoteki s srbskim prevodom nima prevoda v slovenskem delu korpusa, vrstica 3 v datoteki s srbskim prevodom pa je prevod vrstice 2 v datoteki s slovenskim prevodom.

4.4. Rezultati evalvacije

Po opravljeni avtomatski poravnavi in ročno poravnanem testnem naboru smo ocenili avtomatsko poravnavo testnega nabora. Ocene smo dodelili po naslednjih formulah:

- pravilnost = tp + tn / (tp + tn + fp + fn),
- natančnost = tp / (tp + fp),
- priklic = tp / (tp + fn),

pri čemer predstavlja tp število pravilno poravnanih enot, tn število pravilno neporavnanih enot, fp število napačno poravnanih enot in fn število napačno neporavnanih enot. Po opravljenem poravnavanju smo izvedli še primerjavo uspešnosti različnih načinov poravnavanj. Rezultate evalvacije prikazuje tabela 8.

Enota	**Poravnava**	**Pravilnost**	**Natančnost**	**Priklic**
poved	časovne kode	45 %	81 %	43 %
poved	besedilo	74 %	74 %	99 %
podnapis	časovne kode	47 %	74 %	49 %
podnapis	besedilo	51 %	54 %	78 %

Tabela 8: Uspešnost poravnavanja korpusa SUMAT

Rezultati evalvacije kažejo nizko uspešnost poravnavanja. Razlog je v gradivu. To smo potrdili z eksperimentom, v katerem smo z algoritmom poravnavanja na osnovi časovnih kod poravnavali korpus OpenSubtitle v.2. Poravnave povedi v tem korpusu so zapisane tudi kot dodatna informacija. Oboje smo primerjali z ročno narejenimi poravnavami. Rezultate prikazuje tabela 9. Poravnave, zapisane v korpusu, smo poimenovali OPUS (tj. ime projekta, v katerem je korpus nastal). Iz tabele 9 je razvidno, da daje naš algoritem poravnavanja na osnovi časovnih kod boljše rezultate kot algoritem, uporabljen pri gradnji korpusa OpenSub. S tem je bila potrjena pravilnost metod, implementiranih v našem algoritmu.

Enota	**Poravnava**	**Pravilnost**	**Natančnost**	**Priklic**
poved	časovne kode	93 %	94 %	98 %
poved	OPUS	85 %	87 %	96 %
podnapis	časovne kode	73 %	81 %	85 %

Tabela 9: Uspešnost poravnavanja korpusa OpenSubtitle v.2

V naslednjem koraku smo se odločali, kateri način poravnavanja ohraniti v ciljnem korpusu. Če primerjamo uspešnost obeh načinov poravnavanj na povedih, vidimo, da je le natančnost poravnavanja na osnovi časovnih kod

večja, priklic in pravilnost pa sta bistveno slabša. Odločili smo se, da v primeru korpusa poravnanih povedi ohranimo poravnave, ki jih je tvorilo poravnavanje na osnovi besedila. V primeru poravnavanja podnapisov razlika v pravilnosti obeh postopkov ni tako izrazita, natančnost pa kaže spet v prid poravnavanja na osnovi časovnih kod. V tem primeru smo se odločili, da korpus poravnanih podnapisov sestavimo iz poravnav, ki jih je tvorilo poravnavanje na osnovi časovnih kod.

V tabeli 10 je podana velikost končnega korpusa SUMAT, ki obsega okrog 110.000 poravnanih enot. Primerjava z obsegom izvornega korpusa kaže veliko izgubo materiala (okrog 40 %). Razlog je verjetno v načinu priprave izvornega gradiva, ki smo ga opisali v drugem poglavju.

Enota	Poravnanih enot v korpusu
poved	110.481
podnapis	112.293

Tabela 10: Vzporedni korpus SUMAT

5. Zaključek

V članku smo opisali postopke, ki smo jih izvedli pri pripravi vzporednega korpusa SUMAT. Korpus bomo uporabili za učenje sistema za strojno prevajanje podnapisov. Korpus omogoča primerjavo uspešnosti sistemov za strojno prevajanje na osnovi podnapisov in povedi. Ker je korpus po obsegu precej skromen in ne dosega želenih lastnosti za optimalno učenje strojnega prevajanja, bo potrebno opiranje na druge obstoječe slovensko-srbske vzporedne korpuse, kot je na primer korpus OpenSubtitles. Korpus SUMAT bo v prihodnje tudi avtomatsko oblikoslovno označen, da bomo lahko preizkusili vpliv oblikoslovnih oznak na kvaliteto statističnega strojnega prevajanja. Korpus žal ne bo dostopen za javnost, ampak samo v okviru projektnega konzorcija, saj podjetja, ki so tekstodajalci, izredno zaščitniško skrbijo za svoja gradiva in ne dovolijo, da bi v kakršni koli obliki prišla v javnost.

6. Literatura

Abdul-Rauf, S., Fishel, M., Lambert, P., Noubours, S., Sennrich, R. (2010). Evaluation of Sentence Alignment Systems. Available at: http://lium3.univ-lemans.fr/mtmarathon2010/ProjectFinalPresentation/SentenceAlignment/sentence_alignment.pdf.

Braune, F., Fraseer, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. V: COLING 2010, Peking, Kitajska.

European Commision (2010). *Audiovisual Media Services Directive* (AVMSD – 2010/13/EU). Official Journal of the European Union, 10 March 2010.

Gale, W.A., Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Comput. Linguist.* 19(1): 75-102.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation, *MT Summit*.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst. (2007). Moses: open source toolkit for statistical machine translation. In: ACL 2007: proceedings of demo and poster sessions, Prague, Czech Republic, pp. 177–180.

Kovačič., I. (1996). Subtitling strategies: a flexible hierarchy of priorities. In: *Traduzione multimediale per il cinema, la televisione e la scena / Multimediale Übersetzung für Film, Fernsehen und Bühne / Multimedia translation for film, television and the stage: atti del convegno internazionale*. Forlì, 26-28 October 1995 (pp. 297--305).

Media Consulting Group (2007). *Study on dubbing and subtitling needs and practices in the European audiovisual industry*. On behalf of the Information Society and Media Directorate General and the Culture Directorate of the European Commission, November 2007.

Moore, R. E. (2002). Fast and accurate sentence alignment of bilingual corpora. V: AMTA'02 proceedings, London, Velika Britanija (pp. 135—144).

Rojc, M., Kačič, Z. (2007). Time and space-efficient architecture for a corpus-based text-to-speech synthesis system. *Speech commun*. 49(3): 230-249.

Sennrich, R., Volk, M. (2010). MT-based sentence alignment for OCR-generated parallel texts. V: AMTA'10 proceedings, Denver, Kolorado.

de Sousa, S., Aziz, W., Specia, L. (2011). Assesing the Post-Editing Effort for Automatic and Semi-Automatic Translations of DVD Subtitles. *Proceesings of Recent Advances in Natural Language Processing Conference* (RANLP-2011). Hissar, Bulgaria.

Tiedemann, J. (2009). News from OPUS – A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In: N. Nicolov, K. Bontcheva, G. Angelova,, R. Mitkov (eds.): *Recent Advances in Natural Language Processing* (vol. V) (pp. 237--248). Amsterdam, Philadelphia: John Benjamins.

Varga, D., L. Németh, P. Halácsy, A. Kornai, V. Trón, V. Nagy (2005). Parallel corpora for medium density languages. In: *Proceedings of the RANLP 2005* (pp. 590--596).

Volk, M. (2008). The Automatic Translation of Film Subtitles: A Machine Translation Success Story? In: J. Nivre, M. Dahllof, B. Megyesy (eds.): *Resourfceful language Technology: Festschrift in Honor of Anna Sagvall Hein* (vol 7 of Studia Linguistica Upsaliensia, Uppsala, Sweden) (pp. 202—214).

Topic ontology construction from English and Slovene language technologies corpora

Jasmina Smailović[1], Senja Pollak[1,2]

[1]Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia
[2]Faculty of Arts, University of Ljubljana, Aškerčeva 2, Ljubljana, Slovenia
{jasmina.smailovic, senja.pollak}@ijs.si

Abstract

This paper presents the OntoGen topic ontology construction tool and the process of building topic ontologies from English and Slovene research papers in the domain of language technologies. We were interested in how cleaning the documents (e.g. removing the references section), manual concept moving and renaming, or using supervised active learning affect the ontologies.

Gradnja ontologij tematik iz angleškega in slovenskega korpusa jezikovnih tehnologij

V članku predstavljamo orodje OntoGen ter proces gradnje ontologij tematik iz angleških in slovenskih znanstvenih člankov s področja jezikovnih tehnologij. Zanimalo nas je, kako čiščenje člankov (npr. brisanje poglavja z viri), ročno preimenovanje in premeščanje konceptov ter uporaba metode aktivnega učenja vplivajo na ontologije tematik.

1. Introduction

Manual construction of taxonomies and ontologies represent a significant investment of human resources when used for modeling a new domain. Therefore, methods for (semi-)automatic extraction of domain knowledge from unstructured texts were developed. Automatic taxonomy construction was addressed in e.g. Navigli et al. (2011) and Kozareva and Hovy (2010).

While an *ontology* is a "formal, explicit specification of a shared conceptualization" (Gruber, 1993), represented as a set of domain concepts and the relationships between them, a *topic ontology* is a set of domain topics or concepts[1] – formed of related documents – represented by the most characteristic topic keywords and related by the *subconcept-of* relationship (Fortuna et al., 2005; 2007).

The task addressed in this paper is to semi-automatically construct a topic ontology from documents in the area of language technologies. This domain has already been modeled in previous research. The main domain publications were collected by Bird (2008). Joseph and Radev (2007) performed the citation analysis, the domain topic and trend analyses were done by Hall et al. (2008) and Paul and Girju (2009), using LDA (Blei et al. 2003).

In this work, our goal is to get an overview of the topics covered at the Slovene language technologies conference. To do so, we semi-automatically constructed two topic ontologies, one from papers written in Slovene, and the other from papers written in English. The constructed topic ontology is corpus-driven and represents only the concepts covered in the given corpus. For addressing this task, we used OntoGen[2] (Fortuna et al., 2005; 2007), a data-driven ontology editor, focusing on extracting and editing of topic ontologies. We investigated how the fact of removing information specific to scientific articles, like references or authors' names, affect the ontology. A very interesting part of this research was to compare the resulting topic ontologies with and without using supervised *active learning* (Cohn et al., 1994; Settles, 2009). We continue the research presented in (Smailović and Pollak, 2011), where the English articles were modeled into a topic ontology. In addition to the English topic ontology, in this paper we also model the Slovene part of the corpus as a separate topic ontology.

The paper is structured as follows. Section 2 describes the corpus, data preparation and the ontology editing tool. In Sections 3 and 4 the topic ontology construction process is presented. Section 5 provides the conclusions.

2. The corpus, data preparation and the OntoGen topic ontology construction tool

The articles for this case study were taken from the proceedings of the Slovene Language Technologies Conference (proceedings of seven conference editions are available online: http://www.sdjt.si/konference.html). As the papers (79 in English and 109 in Slovene) were available as PDF documents, we had to transform them into an appropriate textual format for the OntoGen tool, i.e. to the named-line document format. The first step was to transform the documents to a text-only document format. PDF to text conversion was performed, using the PDFBox[3] and Nitro PDF reader[4]. The text files were transformed to UTF-8 encoding. Next, we split the English and Slovene articles.

In this research, we present two settings, in the first one, the topic ontology is constructed without cleaning the documents and in the second one, semi-automatic data preprocessing is first performed. For the latter, using Perl scripts, we discarded parts of articles, such as authors' names, institutions, references, section numbers, tables, page numbers, etc. to get the "cleaned documents".

After presenting the documents in the named-line document format, OntoGen was used for building a topic

[1] In this paper words *concept* and *topic* are used as synonyms.

[2] http://ontogen.ijs.si/

[3] http://www.codeproject.com/KB/string/pdf2text.aspx

[4] http://www.nitropdf.com/

ontology. OntoGen is a semi-automatic and data-driven ontology editor. Semi-automatic means that the system is an interactive tool that aids the user during the topic ontology construction process. Data-driven means that most of the aid provided by the system is based on the underlying text data (document corpus) provided by the user. The system combines text mining techniques (*K*-means document clustering) with an efficient user interface to reduce both the time spent and complexity of manual ontology construction for the user.

3. Topic ontology on raw documents

We first examined how the topic ontology looks like if we do not perform any additional cleaning of the corpus. In this case, the text documents in the named-line format consist of an ID, followed by a title, names of the authors, main text of the article and references.

OntoGen uses *K*-means clustering, i.e. a method of cluster analysis which aims to partition *N* instances (documents, in our case) into *K* clusters in which each instance belongs to the cluster with the nearest mean. If we build a topic ontology automatically, by only suggesting to OntoGen the number *K* of concepts at each node of the concept hierarchy, the result for the English articles can be seen in Figure 1. For every concept, we tried different *K*-values and chose the one that splits the concept in the best way according to the user's understanding of the area. Neither the active learning functionality nor the renaming of the concepts was performed in this topic ontology construction process.

As one can see from the figure, names of the concepts/topics are not intuitive, and in some cases it is hard to understand what they represent. This happens since for concept naming OntoGen selects the first three most frequent words from the automatically constructed keywords list. For example, if the concept is described by the following keywords: *slovenian, translation, vowel, speakers, synthesis, speech, corpus, tagging etc.*, OntoGen will name this concept "*slovenian, translation, vowel*".

A better way of naming concepts is by involving the expert who can quickly find an appropriate concept name after observing all the topic keywords. Using this approach, the previous topic could be called *Speech technologies*. All the concepts in the English and Slovene topic ontologies were thus manually renamed based on the automatically extracted topic keywords.

Next, we observed that several topics/concepts were not present in the topic ontology. For the terms often occurring in the keyword lists of different concepts, but not being one of the three main topics keywords**,** we decided to use the supervised method for adding topics. It is based on the Support Vector Machine (SVM) active learning method of OntoGen. For the English corpus, we entered queries for *Speech recognition* and *Speech translation* concepts and answered some automatically proposed questions of a type: *Would you classify the document number 41 as an article on the topic of Speech recognition?* which enabled the system to label the instances. After the concept node was constructed, it was added to the ontology as a sub-concept of the selected concept, in our case, as a sub-concept of the *Speech technologies* concept. Similarly, we performed active learning also on the Slovene corpus. We entered queries for *Prevajanje govora* (*Speech translation*). In this way we tried to identify the most common and important words for the missing subconcept and put them in the query. Then, as for English articles, we answered several questions, and a new sub-concept was added in the Slovene topic ontology.

After manually renaming the concepts, using active learning for adding concepts, and manually moving some documents from one concept to another, we got an improved topic ontology. The resulting English topic ontology is shown in Figure 2. This ontology is more intuitive and understandable. One can see from the figure that language technologies consist of *Computational linguistics* and *Speech technologies* as its core concepts. This is also the general division of the field of language technology (e.g. in Wikipedia, language technology is defined as follows: *Language technology is often called human language technology (HLT) or natural language processing (NLP) and consists of computational linguistics (or CL) and speech technology as its core but includes also many application oriented aspects of them.*).

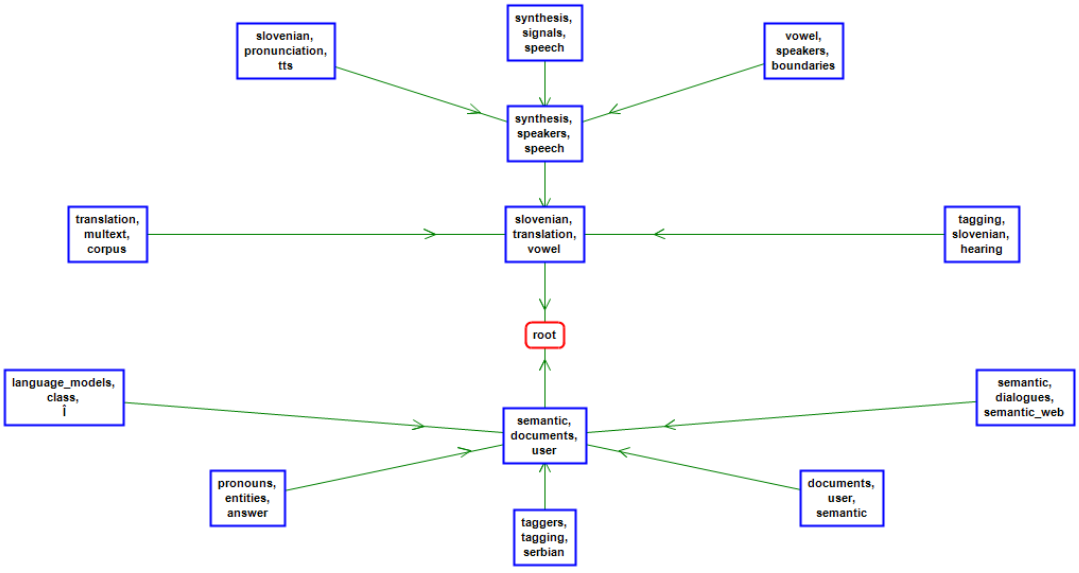


Fig. 1: English topic ontology without cleaning text document and without concepts renaming.

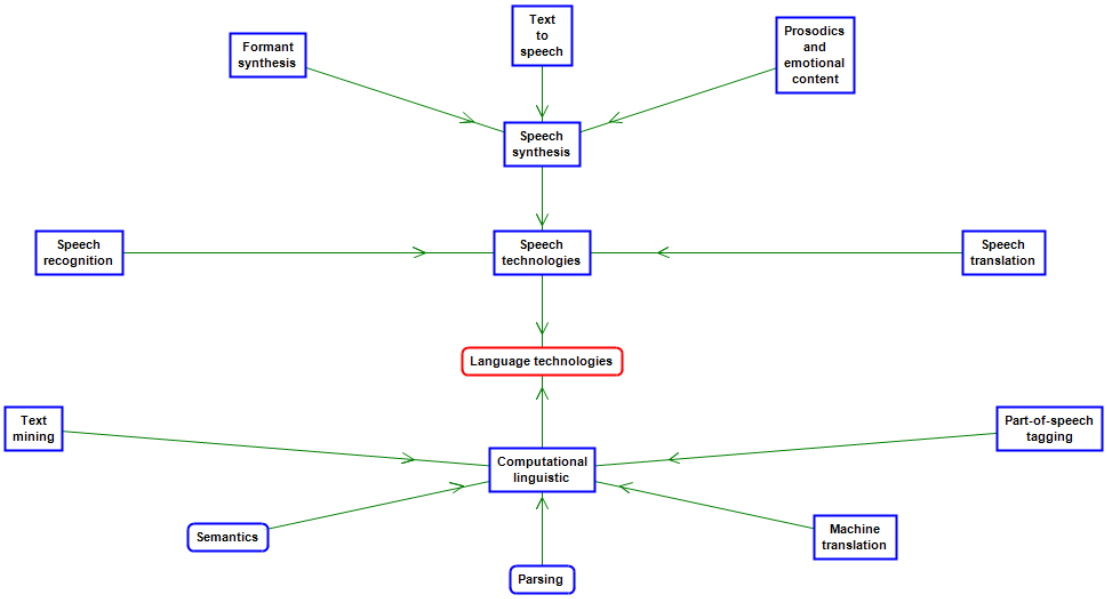


Fig. 2: English topic ontology after manually renaming the concepts, using active learning for adding concepts and manually moving some documents from one concept to another.

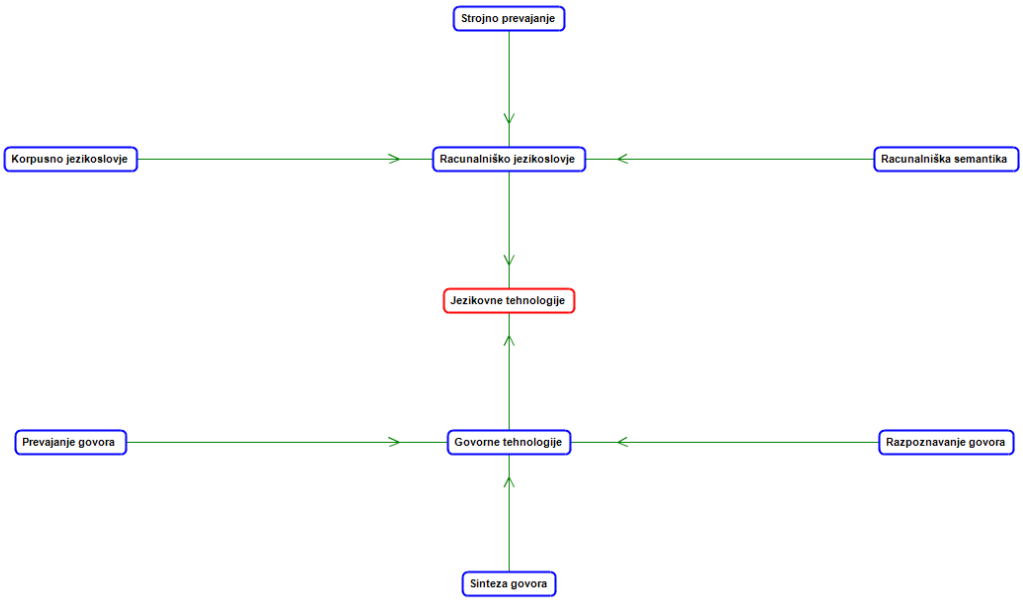


Fig. 3: Slovene topic ontology after manually renaming the concepts, using active learning for adding concepts and manually moving some documents from one concept to another.

Precise evaluation of the ontology coverage is a very hard task since we do not have a golden standard ontology for this specific corpus. We were therefore only able to approximately evaluate the coverage of the research area of language technologies separately for individual topics, as illustrated on the following subtopic. Concept of *Speech technologies* is in Wikipedia divided into 6 subfields (*Speech synthesis, Speech recognition, Speaker recognition, Speaker verification, Speech compression, Multimodal interaction*). The division in our ontology covers 2 out of these 6 concepts (and adds one more). However, as all the missing concepts occur very rarely in the corpus, the evaluation shows that OntoGen performs well, as the constructed ontology indeed adequately reflects the nature of the corpus.

Thus, OntoGen did the splitting very well for the root concept, we just had to change the sub-concepts' names. More manual work - supervised learning and manually moving some documents from one concept to another, needed to be done in further concepts splitting.

Slovene topic ontology, on the other hand, (after renaming the concepts, using active supervised learning and by manually moving some documents from one concept to another) is shown in Figure 3. Given that OntoGen does not have a stemmer for Slovene, we lemmatized the input documents in data preprocessing.

One can see from the figure that the Slovene topic ontology is simpler than the English one. For the Slovene topic ontology we had to do much more manual work (moving some documents from one concept to another). Interestingly, one third of the Slovene articles belong to

topic *Korpusno jezikoslovje* (*Corpus linguistics*). Note that the *Corpus linguistics* category in our ontology comprises also design of corpora and tools for corpus processing, such as PoS taggers and is not used in its strict sense, but in the sense of "Corpus construction and use".

Concept visualization for English articles can be seen in Figure 4. Two main concepts are marked with green and orange dashed lines. In the upper left corner of concept visualization one can notice some non-standard characters. These show OntoGen encoding problems probably due to the Slovene characters which are present in authors' names and references or some special characters from the tables.

Concept visualization for Slovene articles can be seen in Figure 5. The visualization is similar to the visualization for English articles, i.e. the articles are divided into two major topics (*Computational linguistics* and *Speech technologies*) and the *Computational linguistics* topic contains much more articles than the Speech technologies topic.

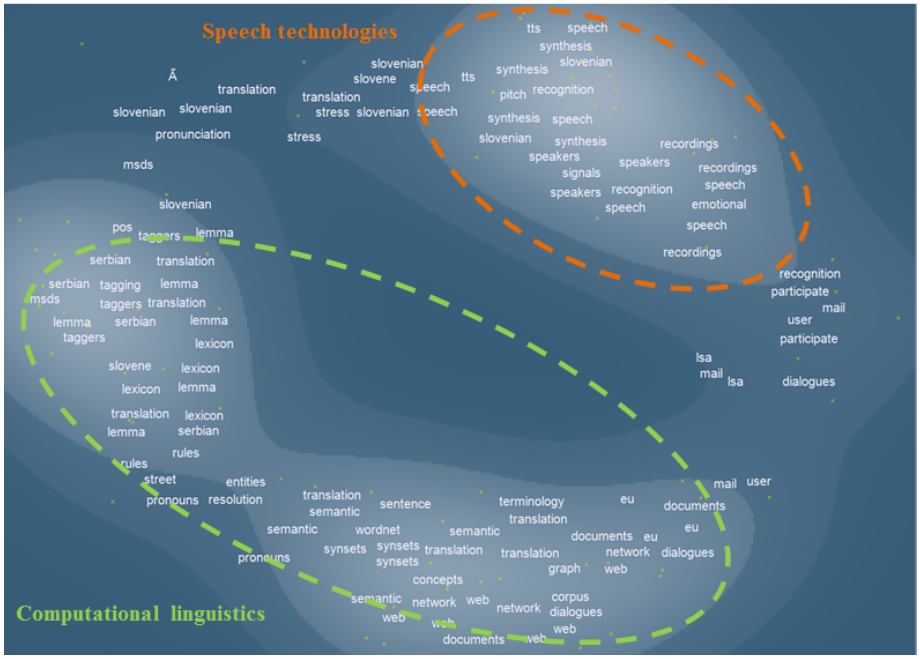


Fig. 4: Concept visualization of English text documents without cleaning (the automated splitting into two main topics, which we label *Computational linguistics* and *Speech technologies*, can be observed).

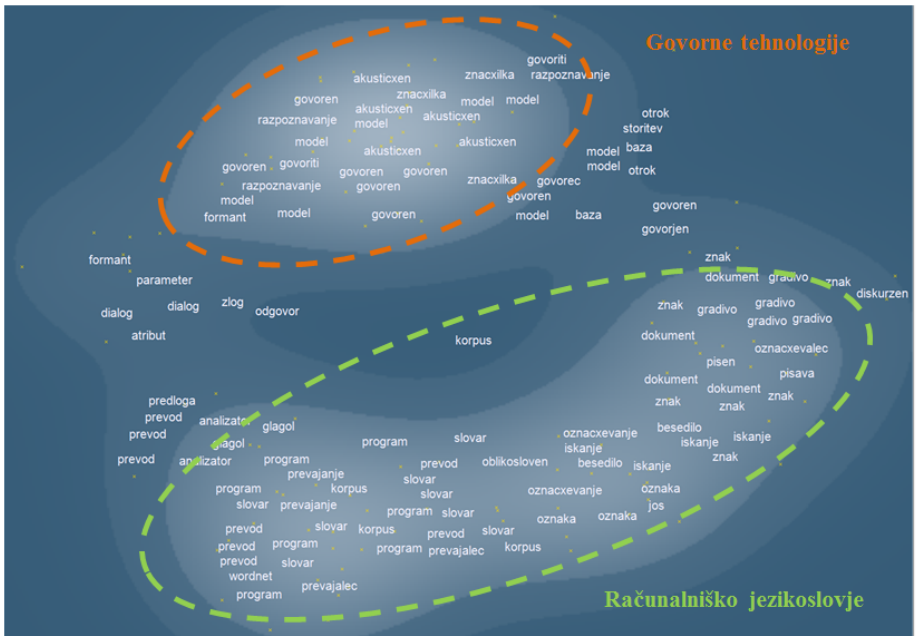


Fig. 5: Concept visualization from Slovene text documents without cleaning (the automated splitting into two main topics, that we label *Računalniško jezikoslovje* and *Govorne tehnologije* can be observed).

4. Topic ontologies constructed from "cleaned" corpora

We consider a text document to be "cleaned" once the names of the authors, references, page numbers, etc. were removed from the article.

In general, English topic ontology after cleaning the text documents has a similar structure as the topic ontology for text documents without cleaning (see Figure 6). Additional supervised learning and manually moving documents from one concept to another were also needed. One of the main differences is that the topic ontology for cleaned documents does not include the *Parsing* concept. We expected this type of differences, since the articles listed in the references may have an impact on the ontology, in our case for example, a new concept was created.

While building the Slovene topic ontology after cleaning the text documents (Figure 7), we noticed that splitting of the topic makes much more sense. Even the suggested topics` names were very similar to the actual names of the topics, even for leaf nodes of the hierarchy. For this ontology we did not perform any active learning or other manual work (except renaming), this is why the ontology is simpler than the topic ontology constructed from raw documents.

Concept visualization of English articles has visible differences, as shown in Figure 8. One can notice that once the text documents cleaned, certain groups of documents appear more distant. It is obvious that they were closer before because of the names of authors and names of papers and authors in the references. One can notice a non-standard character "Î" in the concept visualization. This character is also present after adding it to the stopword list, due to a mismatch with the encoding in OntoGen. After carefully reading the articles, we noticed that some of them contained Cyrillic characters which could not be properly encoded after PDF to text conversion. Concept visualization of Slovene articles is very similar to the one on uncleaned documents, but slight differences can be observed. In the *Računalniško jezikoslovje* (*Computational linguistics*) topic, one can see that documents which belong to concepts *Korpusno jezikoslovje* (*Corpus linguistics*) and *Strojno prevajanje* (*Machine translation*) are now distant (cf. Figure 9). Again, they were probably closer before because of the authors' names and the titles of papers and authors in the references.

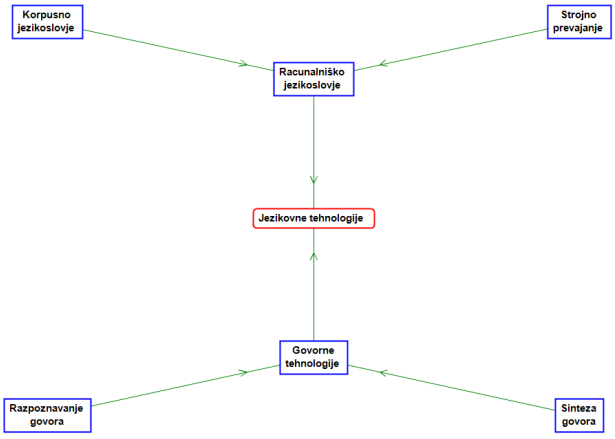


Fig. 7: Slovene topic ontology on cleaned text documents.

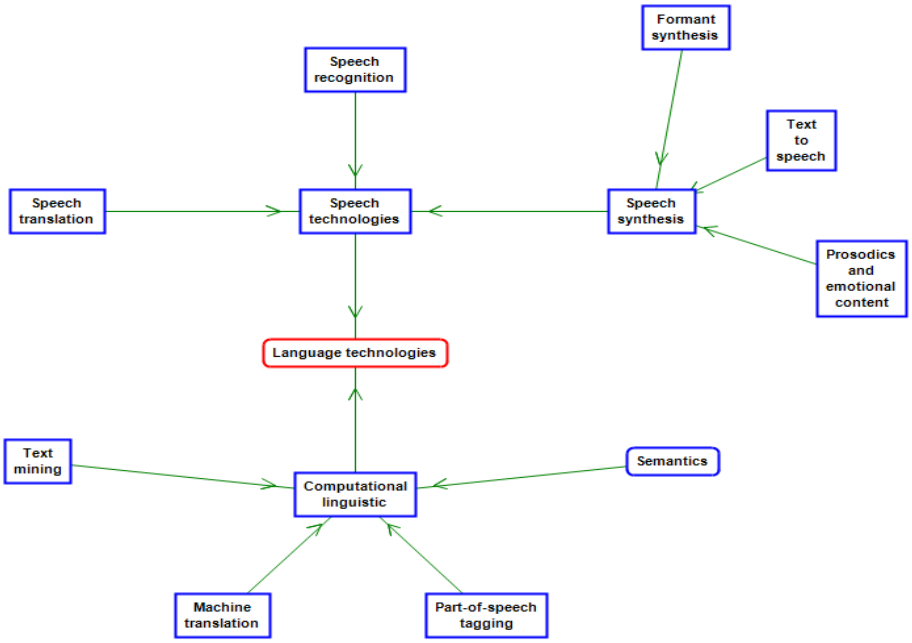


Fig. 6: English topic ontology on cleaned text documents after manually renaming the concepts, using active learning for adding concepts and manually moving some documents from one concept to another.

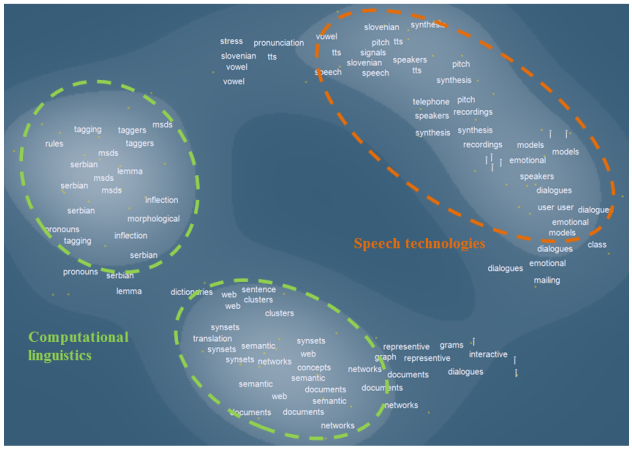


Fig. 8: Concept visualization of cleaned English articles marked with two main concepts.

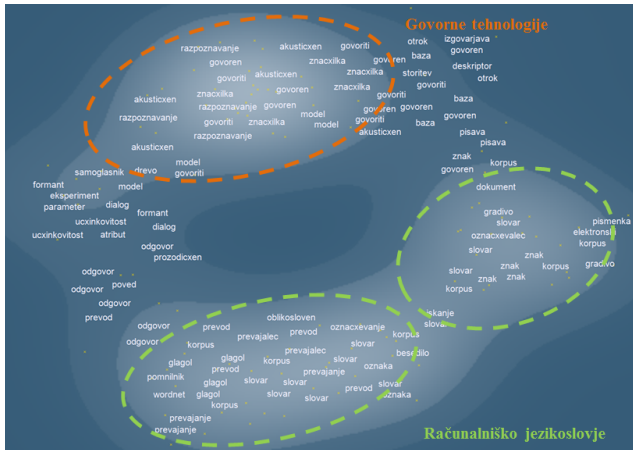


Fig. 9: Concept visualization of cleaned Slovene articles marked with two main concepts.

In other experiments, we also investigated how the topic ontology looks like if we use only the title and the abstract of the articles. However, we noticed that in the ontology building the splitting was done very well for the root concept, but further concepts splitting did not give satisfying results. Therefore, due to the space limitation, we do not present these topic ontology visualizations.

5. Conclusion

In this paper we have shown that OntoGen is a useful tool for constructing and editing topic ontologies, but as expected, to get perfect results additional manual work is necessary, using the OntoGen's semi-automatic editor.

We were interested in a specific domain of language technologies, and limited ourselves to the Slovene scientific conference in this field. Its small dataset (79 English and 109 Slovene articles) enabled us to evaluate the constructed ontologies, showing that adequate and interesting topic ontologies can be constructed. We extracted meaningful concepts from the language technologies domain and observed that the subtopics division (e.g. computational linguistics vs. speech technologies) make a lot of sense. This indicates that OntoGen could be used also as a tool for conference organization support, helping to assign articles to different tracks of the conference. Besides extracting the sub-concepts, we also identified hierarchical relations between them, which were not discussed in other studies. We can derive also some conclusions about data preprocessing. If one first cleans the documents by excluding authors' names and references from the document the cluster visualization reflects better the similarity between the articles contents.

6. Acknowledgements

We would like to thank T. Erjavec, N. Lavrač and B. Fortuna for their useful comments in the preparation of this paper.

7. References

Bird, S. (2008). Association for Computational Linguistics Anthology. http://www.aclweb.org/anthology-index/.

Blei, D., Ng, A. and Jordan, M. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022.

Cohn, D., Atlas, L. and Ladner, R. (1994). Improving generalization with active learning. Machine Learning, 15(2): 201–221.

Fortuna, B., Grobelnik, M. and Mladenić, D. (2005). Semi-automatic construction of topic ontology. Semantics, Web and Mining, Joint International Workshop, EWMF 2005 and KDO 2005, Porto, Portugal, Revised Selected Papers.

Fortuna, B., Grobelnik, M. and Mladenić, D. (2007). OntoGen: Semi-automatic ontology editor. In Smith, M. J. and Salvendy, G. (Eds.). In Proceedings of Human Interface, Part II, HCI International 2007, LNCS 4558, Springer, 309–318.

Gruber, T. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition. 5 (2):199–220.

Hall, D., Jurafsky, D. and Manning, C. (2008). Studying the history of ideas using topic models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08), 363–371.

Joseph, M.T. and Radev, D.R. (2007). Citation analysis, centrality, and the ACL anthology. Technical Report CSE-TR-535-07, University of Michigan.

Kozareva, Z. and Hovy, E. (2010). A semi-supervised method to learn and construct taxonomies using the web. In Proceedings of EMNLP 2010, 1110–1118.

Navigli, R., Velardi, P. and Faralli, S. (2011). A graph-based algorithm for inducing lexical taxonomies from scratch. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011), 1872-1877.

Paul, M. and Girju, R. (2009). Topic modeling of research fields: An interdisciplinary perspective. In Proceedings of the International Conference RANLP-2009. Association for Computational Linguistics, 337–342.

Settles, B. (2009). Active learning literature survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Smailović, J. and Pollak, S. (2011). Semi-automated construction of a topic ontology from research papers in the domain of language technologies. In Proceedings of the Human language technologies as a challenge for computer science and linguistics (LTC'11), 121–125.

Translating news to CycL using the XLE parser

Janez Starc, Blaž Fortuna

Artificial Intelligence Laboratory
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana
{janez.starc, blaz.fortuna}@ijs.si

Abstract

We built a pipeline for translating text into logical representation, which falls in the category of Machine Reading (MR). The essential component of the pipeline and our main contribution is the non-probabilistic rule-based framework. Other components are: syntatic parser, XLE, to extract gramamtical features from text, Enrycher, for additional natural language processing, and Cyc for semantic resources, reasoning, and its language CycL to respresent the translated knowledge. We defined and implemented several rules on the framework. We evaluated them on business news. In the discussion we identified several challenges in MR.

Prevajanje novic v jezik CycL s pomočjo razčlenjevalnika XLE

Zgradili smo cevovod za prevajanje besdila v logično predstavitev. Naše delo spada v področje strojnega branja. Glavna komponenta cevovoda in naš glavni prispevek je neprobabilistično programsko ogrodje, ki temelji na pravilih. Ostale komponente so: XLE – sintaktični razčlenjevalnik, Enrycher – storitev za procesiranje naravenega jezika in Cyc – semantični vir ter avtomatski pojasnjevalnik. Za predstavitev prevedenega znanja smo uporabili jezik CycL. Na programskem ogrodju smo definirali in implementirali nekaj pravil. Ocenili smo, kako prevajajo besedila iz poslovnih novic. V zaključku smo odkrili številne izzive v strojnem branju.

1. Introduction

Vast amounts of text are published online every minute. People all over the world are communicating through Facebook updates, Tweets, blogs, etc., or more formal discourses like news articles, academic papers, books, etc. The tendency is to extract as much information as possible from these sources and express this information in a logical representation. This data can be used to populate a particular knowledge base. Consequently, new knowledge can be inferred from the knowledge base.

We built a non-probabilistic rule-based system, which does several tasks sequentially. In the beginning, it obtains English textual data from a news article stream, IJS Newsfeed. Then it processes the data with natural language processing tools XLE parser (Maxwell and Kaplan, 1996) and Enrycher (Štajner et al., 2010). In the next step, the system translates the processed text into a logic language of Cyc system (Lenat, 1995), CycL. These logical statements are then asserted into the knowledge base of Cyc, CycKB. The main contribution of our work is a framework for implementing rules, which translate the output of the XLE parser, f-structures, to CycL. We named this component the Translator. We also defined, implemented and evaluated a few rules. Our system exploits external resources to semantically enrich its output.

Our paper falls into category of Machine Reading (MR). A study of requirements for MR has been done by (Clark and Thompson, 2009). Similar work to ours was done in (Witbrock et al., 2004) using different parsers. Also similar to our work is the approach taken by researchers at Parc (Crouch, 2005) (Crouch and King, 2006) (Bobrow et al., 2007). Many others took the semi-supervised approach, for e.g. (Etzioni, 2007) (Carlson et al., 2010). In (Ghosh et al., 2011) Markov Logic Networks (MLNs) were used for MR.

The rest of this paper is organized in the following way. We describe all third-party components, especially parts that we use in our system in Section 2. We present the workflow of our system in Section 3. In Section 4., we evaluate our system. We finish with conclusions and proposals for future work in the last section.

2. Description of components

2.1. IJS Newsfeed

To obtain online news we have used software IJS newsfeed[1]. It periodically crawls a list of RRS feeds and a subset of Google News to obtain links to news articles. Then it downloads and parses the articles to extract cleartext version of the article body, which can be further processed by Enrycher. This data is then available on two streams, cleartext and Enrycher processed. We captured news from the stream that is processed by Enrycher, which we present in the next subsection.

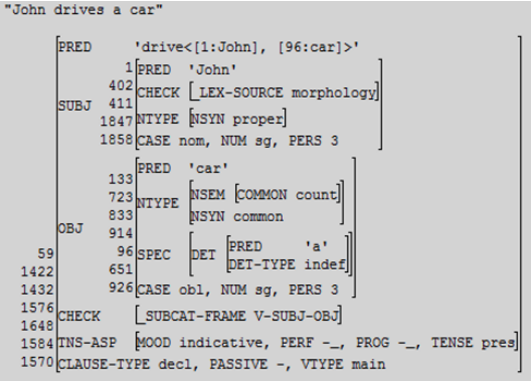
2.2. Enrycher

Enrycher[2] (Štajner et al., 2010) is a service that provides shallow and deep text processing at the document level. Main features of the system include topic and keyword detection, named entity extraction, word sense disambiguation, triplet extraction. We have used sentence splitting, topic detection, named entity resolution, co-reference and anaphora resolution in our system.

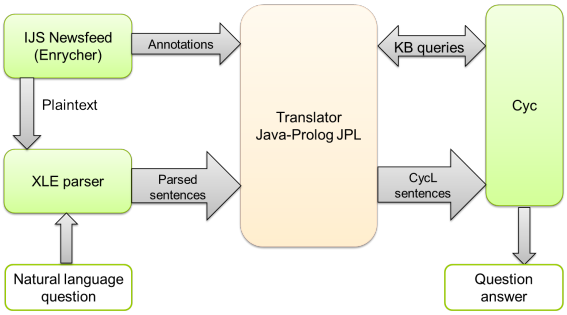
2.3. XLE parser

This software[3] (Maxwell and Kaplan, 1996) parses text with Lexical Functional Grammars (LFGs). It is a part of larger platform, XLE, which also provides rich graphical user interface for writing and debugging LFGs and facility

[1] `newsfeed.ijs.si`
[2] `http://enrycher.ijs.si`
[3] `www.parc.com`



Slika 1: F-structure



Slika 2: The Translator Pipeline

for generating text from LFGs. The main part of the platform is ordered rewrite rule system, which was used to produce semantic representations from the syntactic output of the parser (Crouch and King, 2006). This work has similarities to ours. We both use XLE parser, but different system to generate semantic representations.

The XLE parser produces two mutually constraining types of output, c-structures and f-structure. C-structures or structures of syntactic constituents are constituency trees. The f-structure analysis, on the other hand, treats the sentence as being composed of attributes, which include features such as number and tense, or functional units such as subject, predicate, or object. For example, f-structure of the sentence *"John drives a car."* is depicted on Figure 1. This sentence has only one f-structure. Since the system does not use any semantics there can be more than one solution to one sentence, and the parser produces a packed representation of all possible solutions. Using a form of Optimality Theory (Prince and Smolensky, 2004) solutions are ranked, based on ordered violable constraints. We use the highest ranked solution, which is written in Prolog format, for further processing.

2.4. Cyc

The last third-party component that we use is Cyc[4] (Lenat, 1995). This system includes CycKB, which is an ontology and knowledge base of every day common sense knowledge. The knowledge base contains nearly 500.000 terms, about 15.000 types of relations, and about 5 million assertions. The knowledge is represented by a formal language CycL. The CycL representation of the sentence *"John drives a car."* is presented on Listing 1.

Listing 1: CycL sentence

```
(#$thereExists ?ACTION
  (#$thereExists ?CAR
    (#$and
      (#$isa ?ACTION
        #$TransportInvolvingADriver)
      (#$isa ?CAR #$Automobile)
      (#$vehicle ?ACTION ?CAR)
      (#$driverActor ?ACTION #$John)
)))
```

[4] www.cyc.com

Cyc concepts start with *#$* and Cyc variables start with *?*. CycL sentences are split into microtheories. Each microtheory must not contain any contradictions. Furthermore, Cyc offers an inference engine, which derives answers on the queries using the knowledge base. It is also used to reject assertions that would be inconsistent with the knowledge base. We use Cyc to assert the translated sentence into its knowledge base. We use its inference engine to answer the CycL queries of the corresponding interrogative sentences. We also utilize data about subcategorization frames, which are stored in CycKB. There are two versions of Cyc technology available: the open source OpenCyc and the more complete version, ResearchCyc, which we use in our work.

3. The Workflow

In this section, we will present how our system works. The workflow is shown on Figure 2.

3.1. From news stream to f-structures

In the first step, we listen to IJS newsfeed stream for some time to download a sample of articles together with additional information produced by Enrycher. Each article is tagged with several topics keywords from SIOC ontology. We built a filter, which passes through articles that have been tagged with the selected topic. Using Enrycher's sentence and token splitting feature, another filter was built to filter out either too long or too short sentences. These types of sentences can be ungrammatical or the possibility of correct parse is very low. The parameters for this filter are minimal and maximal number of tokens in a sentence. All the filtered sentences are concatenated, and gathered in the plaintext file. This file is then sent as input to the XLE parser. The parser produces one Prolog file for each sentence. Each file includes several attributes including the actual sentence, metadata, the most probable f-structure, and c-structure.

3.2. Semantic resources from Enrycher

Enrycher annotates every named instance with its type. The following types are resolved: person, location, organization, date, percentage, amount of money. For every named instance we generated one corresponding Prolog compound term, which is available to Translator. One term has four arguments: the sentence, in which the instance appears, the mention of the instance, the name of the instance,

and the Cyc concept, which corresponds to the type of the instance. Here is an example of a compound term:

```
ne('She died in 1957 at age 90.','She','
    Mary',p_Person).
```

We can notice that there was a person named Mary mentioned in a sentence preceding the observed sentence. The author of the discourse replaced the word *Mary* with the pronoun *she* in the observed sentence. In this example, we exploited two Enrychers utilities: co-reference resolution and named entity resolution.

3.3. Semantic resources from Cyc

In the previous subsections, we described the first part of input to Translator: the actual sentence, its f-structure and information about its named entities. Now, we will present two types of semantic resources from Cyc that Translator uses. First, Translator can obtain all Cyc concepts denoted by a particular word. For instance, we would like to get all Cyc concepts, which correspond to word *plays* in the sentence *"John plays."*. The XLE parser provides shallow level linguistic information: this word is a verb and its lemma is play. For the pair *(play, verb)* Cyc returns the following list of concepts:

- `#$Playing`
- `#$PlayingAMusicalInstrument`
- `#$PlayingAGame`
- `#$PlayingAnAudioRecordedObject`

In addition, we also queried phrases for dentotaions. In this case, we did not add any linguistic information about the phrase.

The other type of semantic resources we utilize are *verbSemTrans* relations. This is the definition of the relation taken from CycKB:

> *(verbSemTrans VERB NUM FRAME TRANS)* means that the translation of word sense number NUM of VERB, appearing with subcategorization frame FRAME, is TRANS.

Example of such relation is presented on Listing 2. Subcategorization frame is type of sentence according to number and type of syntactic arguments that co-occur with the verb. For example, *Transitive noun phrase frame* corresponds to sentences that have subject and object. These two arguments must be noun phrases. Other examples of subcategorization frames are *Intransitive verb frame*, *Ditransitive noun phrase frame*, *Middle voice vrame*, etc. Last argument, the translation, is a CycL sentence that may include free variables like *:ACTION*, *:SUBJECT*, *:OBJECT*, *:INDIRECT-OBJECT*. These are later bound to corresponding CycL constructs.

Listing 2: Example of verbSemTrans relation

```
(#$verbSemTrans #$Drive-TheWord 0
  #$TransitiveNPFrame
    (#$and
      (#$isa :ACTION #
          $TransportInvolvingADriver)
      (#$vehicle :ACTION :OBJECT)
      (#$driverActor :ACTION :SUBJECT)))
```

The verbSemTrans relation from Listing 2 has been used to translate sentence *"John drives a car."* into CycL on Listing 1. Predicates *vehicle* and *driverActor* introduce semantics. Predicate vehicle requires *:OBJECT* to be instance of Cyc concept *#$TransportationDevice-Vehicle*. Analogously, predicate driverActor requires *:SUBJECT* to be instance of *#$Person*. Two interesting things can happen when trying to assert CycL sentence that is induced by this relation into CycKB. If one argument cannot fulfil the semantic requirements, e.g. *:SUBJECT* is not a person, then the whole sentence cannot be asserted into the knowledge base. Otherwise, if Cyc cannot prove that one of the arguments meets the requirements, e.g. cannot prove that *:SUBJECT* is a *#$Person*, than the whole sentence is asserted and *:SUBJECT* becomes a *#$Person*. Therefore, with the help of inference engine Cyc can also learn knowledge that was not explicit in the natural language sentence.

3.4. The Translator

In this subsection, we describe the main part of the Translator. We implemented this part of the system in Java and Prolog. We have chosen Prolog because one of the possible output format of the XLE parser is Prolog. The second reason is that the target language CycL is very similar to Prolog. To translate from Prolog to CycL and vice versa we developed a simple regex-based procedure. Because we could not make external calls to Cyc out of Prolog, we also had to use Java. To connect Java and Prolog we used JPL[5]. This is a bidirectional interface, which enables Prolog applications to exploit any Java classes, methods, instances, etc. Analogously, it allows any Java application to manipulate any Standard Prolog libraries, predicates, etc.

The main job of the Translator is to recursively execute the ordered rewrite rules. Rewrite rules replace the antecedent expression with the consequent expression if the antecedent expression complies with the rule. In our case, f-structures are replaced with CycL sentences. Rules are ordered in a sequence. The first rule that satisfies the conditions is executed and no other rules are tested or executed. One rule can have multiple outputs. Therefore, final output can consist of multiple CycL sentences. Translator executes the rules recursively. It starts with the whole sentence, then it recursively translates smaller parts of sentences, e.g. noun phrases, verb phrases. The smallest translated constituent is a word.

One of the main contributions of this paper is the implementation of the sequence of ordered rewrite rules, which is presented below.

1. ***Rule for verbSemTrans relations from Cyc.*** From the obtained f-structure of a sentence we can determine the subcategorization frame of the sentence. For example, the value of the feature *_SUBCAT-FRAME* in the f-structure on Figure 1 is *V-SUBJ-OBJ* (verb, subject, object), which corresponds to Transitive noun phrase frame. This frame together with the verb *drive* corresponds to verbSemTrans relation on Listing 2. The last argument of this relation is the incomplete

[5] `http://www.swi-prolog.org/packages/jpl/`

CycL of the underlying translated sentence. It is incomplete because terms like *:SUBJECT* still need to be instantiated. We implemented the following subcategorization frames:

- Transitive noun phrase frame (subject, verb, object), e.g., *"John drives a car."*
- Intrasitive verb frame (subject, verb), e.g., *"John swims."*
- Ditransitive prepositional phrase frame (subject, verb, object, preposition, oblique-object), e.g., *"John passed the ball to Tom."*
- Copula frame (subject, complement), e.g., *"John is sad."*

2. ***Rule for semantically poorer verbSemTrans relations.*** Frame and verb combinations that are not covered by the first rule, may be covered by this rule. This rule also uses verbSemTrans relations. But these are not stored in Cyc, instead, they are automatically generated. This is an example of such relation:

```
(#$verbSemTrans #$Drive-TheWord 1
  #$TransitiveNPFrame
  (#$and
    (#$isa :ACTION
       TransportInvolvingADriver)
    (#$actor :ACTION :OBJECT)
    (#$actor :ACTION :SUBJECT)
))
```

This relation is similar, but the CycL in the last argument is semantically poorer, than the one on Listing 2. This is because predicate *actors* means just that the last argument of the relation is somehow involved in the action.

3. ***Oblique-object rule.*** This rule is executed only if ditransitive prepositional phrase frame rule was applied before. It is used to correctly extract oblique objects from the prepositional phrase.

4. ***Free variable rule.*** This rule is executed on interrogative pronouns, e.g., who, what, in the interrogative sentences, which are recognized by *CLAUSE-TYPE - INT* attribute-value pair in the f-structure. Interrogative pronouns are recognized by *PRON-TYPE - INT* pair and are translated into CycL variables. Consequently, the final construct of the translation is not a CycL sentence, but a CycL query.

5. ***Cyc denotation rule.*** This rule finds a Cyc concept for a part of the sentence (see Subsection 3.3.)

6. ***Enrycher annotation rule.*** Details of resolving the annotation are presented in Subsection 3.2.. Names and types of entities are connected by this rule. Here is an example of the result of the rule:

```
(#$thereExists ?OBJECT
    (#$and
      (#$isa ?OBJECT #$Person)
      (#$nameString ?OBJECT
        "Tiger Woods")
)).
```

7. ***Noun phrase splitting rule.*** This rule splits the noun phrase into the main noun and the possible modifiers. Each modifier is connected to the main noun with the predicate *featureOf*, which is semantically very poor. For example, *"white rabbit"* is translated into `(#$featureOf #$Rabbit #$WhiteColor)`. Example of the semantically richer relation would be, `(#$mainColorOf #$Rabbit #$WhiteColor)`.

8. ***String instance rule.*** This is the last rule. It always succeeds. It is applied on a part or even whole sentences. It uses Cyc function InstanceFn. The result of the function is a Cyc concept, which is named after the functions only argument. Here is an example of the result of the rule:

```
(#$InstanceFn "quasicrystal")
```

3.5. Asserting sentences to Cyc KB

The result of the translation is one or, due to homonomy, multiple CycL sentences. For each natural language sentence, we create a microtheory, for example,

```
(#$MtWithFocalContentSentenceFn "Francisco
    Liriano will start for the Twins.")
```

We did not use a single microtheory for all senteces, because contradictions may appear due to news writing bias. We try to assert all possible translations of a particular sentence into its microtheory one by one. If a particular translated sentence is contradictory, it is rejected.

3.6. Question answering

The question answering feature was also added to our system. It is possible to make a natural language question. This question is translated to the CycL query. Cyc will answer the query by providing all the answers to it. If we asserted the translation of the sentence: *Clint Eastwood eats a steak.* We can ask a question like: *What does the actor eat?* This is the query translated from the sentence:

```
(#$thereExists ?SUBJECT
  (#$and
    (#$isa ?SUBJECT #$Actor)
    (#$thereExists ?ACTION
      (#$and
        (#$isa ?ACTION #$EatingEvent)
        (#$performedBy ?ACTION ?SUBJECT)
        (#$consumedObject ?ACTION ?OBJECT)
))))
```

The answer to the query, in which *?OBJECT* is the variable, is *(SKF-1534975054)*. This concept is a Skolem term that represents the particular steak that Clint Eastwood eats.

4. Evaluation

In this section, we will first present the empirical evaluation, and then an experiment that we conducted on our system. Since the start of this project, we have taken into account that translating text to knowledge representation is a hard problem, if not impossible. A system like ours would need a huge number of rules to get a good coverage. Since the basic unit of our input is a sentence, we filtered out sentences that are not going to produce good translations

without parsing. One type of such sentences are the ones that have a big probability of being incorrectly parsed by the syntactic parser. From our experience, these are either longer sentences, which consist of multiple clauses, or the ones that contain non-alphabetic characters, e.g., parenthesis, dashes, etc. Of course, there are also sentences that are grammatically incorrect, but these are hard to recognize without parsing. During evaluation, we also noticed that there is huge spread of quality of the evaluated sentences. Some of them are correctly translated, but they lack semantic richness. On the other hand, some translations are semantically very rich, but not all phrases are correctly translated.

In this paragraph, we will present an experiment that we conducted. We selected a controlled set of sentences from the IJS newsfeed stream, processed them, translated them to CycL, asserted them to CycKB and manually evaluated a fraction of translations. We only took articles from the business domain, because we expected text from this domain less complicated. To exclude other articles we used the SIOC tags from Enrycher. After obtaining the eligible articles, we split them into sentences. We retained the sentences that have 7 - 15 words, start with the capital letter, end with the period, and do not have any other characters than alphabetic characters, periods or currency signs. We got 19443 sentences from a total of 40624 articles. From these sentences we extracted 12245 mentions of named entities. We randomly selected 1000 sentences. These were then translated to CycL, and the translations were asserted to Cyc. Out of this set, 326 sentences had at least one valid Cyc translation. A valid translation is a CycL sentence, which is not necessary without contradictions. 204 sentences had at least one assertion. Out of these, 101 sentences had only one assertion; the others had ambiguous translations. One sentence had a maximum of 210 asserted translations. The average number of asserted translations was 3.0.

Of the sentences that had assertible translations, we randomly selected 50 sentences for human evaluation. One human evaluator observed three things on each sentence: the quality of the XLE f-structures (see Table 1), qualities of phrase denotations (see Table 2), the quality of the structure and semantic relations (see Table 3). Because word sense disambiguation was not applied in our system, there are multiple possible assertible translations. The evaluator made word sense disambiguation himself and has chosen the correct translation, and evaluated it. If some less important parts of the sentence, like adverbs in the beginning of the sentence, were not translated, the translation was not penalized.

No. of points	Description of quality class
3	The parse is completely correct.
2	The parse is almost correct. One part of the parse is not correct. However, it is good enough to be further translated.
1	The parse is wrong and it is not used for further processing.

Tabela 1: Scoring of f-structure quality

Annotation	Description of the denotation class
G (good)	The phrase is correctly denoted by a Cyc concept.
M (missing)	The phrase is encapsulated by InstanceFn function. Cyc should have a concept denoting this phrase.
P (poor)	The phrase is encapsulated by InstanceFn function. This phrase should be further split into smaller denotable units and Cyc should not have a concept denoting this phrase.
W (wrong)	The phrase is incorrectly denoted by a Cyc concept.
E (Enrycher)	The phrase denoted with Enrycher named entity resolution.

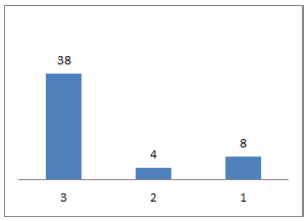
Tabela 2: Phrase denotation quality scoring

No. of points	Description of quality class
4	The structure is good and semantically rich.
3	The structure is good. However, some predicates have no sematic meaning.
2	Something in the structure is wrong.
1	The structure is completely wrong.

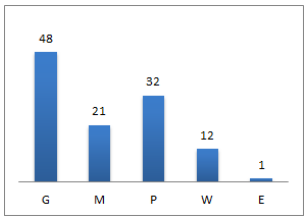
Tabela 3: Scoring of the quality of the structure and sematic relations

The results of the human evaluation are presented on Figure 3. The evaluation showed that 78% of the sentences were correctly parsed (Figure 3a). However, we should not judge the overall precision of the parser based on this number, because in this sample there are only sentences that have valid translation. Phrase denotation quality shows that about 42% of phrases have a corresponding Cyc concept (Figure 3b). There is only one phrase denoted with the help of Enrycher. This number is very low because Cyc denotations have priority over Enrycher denotations. The semantic and structure qualities are quite evenly distributed (Figure 3c). Sentences that had the lowest parsing score were automatically given the lowest structure score.

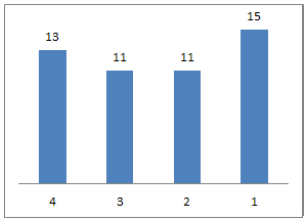
To analyse execution time to translate one sentence, we have to divide the processed into three parts: pre-processing part, XLE parsing and the translation. The whole dataset is processed in each step. These steps are not parallelized. The preprocessing part does not include the time that IJS newsfeed spends to download the articles and annotate them with Enrycher. About 30 articles are preprocessed in one second. XLE parsing of one sentence takes about a quarter of a second. However, we made this measurement on short sentences, which we used in the experiment. It took substantially more time, if we parsed sentences that were longer than the ones in the experiment. The translation and assertion time heavily depend on the number of ambiguous translations. The more translations that one sentence has the more time it takes to assert them to Cyc.



(a) F-structure quality



(b) Phrase denotation quality



(c) Structure and semantic quality

Slika 3: Quality distributions

However, in average two sentences can be translated and asserted in one second.

5. Conclusion

We have made many constraints in the process of translating and evaluation to get the precision that reflects on Figure 3. Therefore, our system is not complete enough to translate large amounts of news articles and then reason on the translated data. It turned out that our system is very useful to identify the problems that arise in text to logic translation. In contrary to many other systems, our system is non-probabilistic, recursive and rule based. Therefore, Prolog and JPL were very suitable for the job. Because our system is sequential, the drawback is that error propagates through the workflow. Evidence of this is also seen on Figure 3.

The XLE parser and its f-structures proved to be very useful in this kind of translation. Although, the diverse nature of the news language is not suitable for the parser. Its accuracy, especially for the longer sentences, is not acceptable. In addition, many sentences were grammatically questionable.

On the other hand, Cyc has a large ontology covering most of the phrases. However, ontology should be supplemented with additional concepts to improve to accuracy and semantic richness of the translation systems. The translation patterns stored in CycKB were manually created. We used the ones for verbs. It would be interesting to implement patterns for nouns, adverbs, etc. in our system. Nevertheless, there are not enough patterns to cover a language like news media language. This raises the question, whether is it possible to automatically create such patterns.

Our system also needs the mechanism for word sense disambiguation. Many systems learn from corpora that are annotated with word senses from a particular database. We believe that corpora for Cyc concepts do not exist yet. Therefore, a different kind of word sense disambiguation solution must be implemented.

Acknowledgements

This work was supported by Slovenian Research Agency and the ICT Programme of the EC under XLike (ICT-STREP-288342).

We would like to thank Tomaž Erjavec for useful comments and for the review.

6. References

D.G. Bobrow, B. Cheslow, C. Condoravdi, L. Karttunen, T.H. King, R. Nairn, V. Paiva, C. Price, , and A. Zaenen. 2007. Parcs bridge and question answering system. In *Proceedings of the GEAF 2007 Workshop*.

A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 101–110, New York, NY, USA. ACM.

P. Clark and J. Thompson. 2009. A study of machine reading from multiple texts. In *Proceedings of AAAI Spring Symposium on Learning by Reading and Learning to Read*.

D. Crouch and T.H. King. 2006. Semantics via f-structure rewriting. *Proceedings of LFG06*, pages 145–165.

R. Crouch. 2005. Packed rewriting for mapping semantics to kr. In *Proceedings of the 6th International Workshop on Computational Semantics*, pages 103–14. Citeseer.

O. Etzioni. 2007. Machine reading of web text. In *Proceedings of the 4th international conference on Knowledge capture*, K-CAP '07, pages 1–4, New York, NY, USA. ACM.

S. Ghosh, N. Shankar, and S. Owre. 2011. Machine reading using markov logic networks for collective probabilistic inference. In *Appearing in the Proceedings of the European Conference on Machine Learning (ECML) Workshop on Collective Learning and Inference from structured data (CoLISD)*.

D.B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

J. Maxwell and R. Kaplan. 1996. An efficient parser for lfg. In *Proceedings of LFG*, volume 96, page 131.

A. Prince and P. Smolensky. 2004. *Optimality Theory: Constraint interaction in generative grammar*. Wiley Online Library.

T. Štajner, D. Rusu, L. Dali, B. Fortuna, D. Mladenić, and M. Grobelnik. 2010. A service oriented framework for natural language text enrichment. *Informatica (Ljublj*, 34(3):307–313.

M. Witbrock, K. Panton, S.L. Reed, D. Schneider, B. Aldag, M. Reimers, and S. Bertolo. 2004. Automated owl annotation assisted by a large knowledge base. In *Workshop Notes of the 2004 Workshop on Knowledge Markup and Semantic Annotation at the 3rd International Semantic Web Conference ISWC2004*, pages 71–80.

Guessing the Correct Inflectional Paradigm of Unknown Croatian Words

Jan Šnajder

University of Zagreb
Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
jan.snajder@fer.hr

Abstract

A real-life morphological analyzer must be able to handle properly the out-of-vocabulary words. We address the task of guessing the correct inflectional paradigm of unknown Croatian words. We frame this as a supervised machine learning problem: we train a model for deciding whether a candidate lemma-paradigm pair is correct based on a number of string- and corpus-based features. Our aim is to examine the machine learning aspect of the problem: we analyze the features and evaluate the classification accuracy using different feature subsets. We show that satisfactory level of accuracy (92%) can be achieved with SVM using a combination of string- and corpus-based features. We discuss a number of possible directions for future research.

Ugibanje pravilne pregibne paradigme za neznane hrvaške besede

Uporaben morfološki analizator mora znati pravilno obravnavati tudi besede, ki jih nima v leksikonu. Prispevek je posvečen ugibanju pravilne pregibne paradigme za neznane hrvaške besede z uporabo nadzorovanega strojnega učenja. Model se odloči, ali je kandidat oz. par lema-paradigma, pravilen glede na večje število lastnosti, ki temeljijo na nizih in korpusu. Namen prispevka je, da preuči razne vidike strojnega učenja tega problema: analiziramo uporabljene lastnosti in ovrednotimo natančnost klasifikacije glede na različne podmnožice lastnosti. Pokažemo, da lahko zadovoljivo raven natančnosti (92%) dosežemo s SVM in z uporabo kombinacije lastnosti nizov in korpusa. Obravnavamo tudi več smernic za nadaljnje delo.

1. Introduction

Morphological analysis plays a vital role in many natural language processing applications, especially for morphologically rich languages such as Croatian. Morphological analysis typically relies on some form of a morphological lexicon, which lists the stems (or lemmas) and their associated word-forms. In a word-and-paradigm setting (Hockett, 1954), the relation between the stem and its word-forms is defined by an inflectional paradigm (pattern). The unavoidable problem of lexicon-based morphological analysis is the limited lexicon coverage. A real-life morphological analyzer must be able to deal in a satisfactory manner with out-of-vocabulary words. In a word-and-paradigm setting, this means being able to guess the correct inflectional paradigm of an unknown word-form.

In this paper we address the task of guessing the correct inflectional paradigm of unknown Croatian words. We frame this as a supervised machine learning problem: we train a model that decides which paradigm is correct based on a number of string- and corpus-based features. To guess the paradigm of an unknown word, we first generate the candidate lemma-paradigm pairs using a morphology grammar, and then use the classifier to decide which pair is correct. This is in contrast to most earlier approaches, which use hand-crafted scoring functions to decide on the correct paradigm. The aim of this paper is to examine the machine learning aspect of the problem: what the relevant features are and how well can we do on this classification task. We carry out feature analysis and evaluate the classification accuracy using different feature subsets. We show that satisfactory level of accuracy can be achieved with a combination of string- and corpus-based features.

The rest of the paper is structured as follows. In the next section we give a brief overview of related work. In Section 3 we define the problem, while in Section 4 we describe the features used for building the models. In Section 5 we analyze the features and evaluate the classification accuracy. In Section 6 we discuss the results and outline directions for further research. Section 7 concludes the paper.

2. Related Work

Much work on paradigm guessing comes from research in part-of-speech (POS) tagging and the related task of POS guessing (Mikheev, 1997; Kupiec, 1992). The problem has also been addressed in the context of rule-based machine translation systems (Esplá-Gomis et al., 2011). However, most work seems to address paradigm guessing in relation to (semi-)automatic lexicon acquisition (Oliver, 2003; Tadić and Fulgosi, 2003; Oliver and Tadić, 2004; Clement et al., 2004; Sagot, 2005; Forsberg et al., 2006; Hana, 2008; Šnajder et al., 2008; Adolphs, 2008; Kaufmann and Pfister, 2010; Esplá-Gomis et al., 2011). The basic idea is to first use a lemmatizer to obtain the lemmas and paradigms for each word-form from corpus. Because of grammar ambiguity, this usually results in a number of possible candidates. Thus, the next step is to disambiguate the output of the morphology grammar by assessing the plausibility of each lemma-paradigm pair. This is most commonly done by generating the corresponding word-forms and analyzing their corpus frequencies. An incorrect lemma-paradigm pair is likely to produce linguistically invalid word-forms that will not be attested in the corpus, and a suitably designed corpus-based scoring function can be used to decide which paradigm is correct. Some approaches use the

web as additional source of information (Oliver and Tadić, 2004; Cholakov and Van Noord, 2009). Moreover, some approaches use word-form properties to decide on the correct paradigm: Forsberg et al. (2006) use hand-crafted constraints, while Segalovich (2003) guesses the stems and the paradigms based on morphological similarity. It is also possible to use context-based information when analyzing the word-forms from corpus (Kaufmann and Pfister, 2010). More recent approaches use machine learning to predict the stem and the morphosyntactic features (Kaufmann and Pfister, 2010). In many cases the problem of paradigm guessing is also addressed in an unsupervised setting, in which paradigms are induced by clustering the word-forms from corpus and an analysis of their endings (Nakov et al., 2004; Oliver, 2003; Esplá-Gomis et al., 2011) – an instance of the more general task of morphology induction (Goldsmith, 2001).

3. Problem Definition

The problem of guessing inflectional paradigms of (unknown) words can be formulated as follows: given a word-form w, determine its correct stem s and its correct inflectional paradigm p. The correct paradigm is the one which, when used with stem s, generates the valid word-forms, including word-form w. The stem and the paradigm are tied together: given w, the inflectional paradigm (possibly ambiguously) determines the stem of w. Moreover, the stem and the inflectional paradigm (possibly ambiguously) determine the lemma l. Thus, the problem actually amounts to determining, for a given word-form, its lemma and the associated inflectional paradigm. In what follows, we call a pair (l, p), consisting of lemma l and inflectional paradigm p, a *lemma-paradigm pair*, or an LPP for short. We call an LPP (l, p) *correct* if (1) the lemma l is valid (it is an existing word of the language and it is indeed a lemma) and (2) the paradigm p is the correct paradigm for l; otherwise we call the LPP *incorrect*. To difficulty in determining the correct inflectional paradigm arises from the fact that for most word-forms there are many candidate LPPs. This problem is typically approached in two steps: generation of LPP candidates and selection of LPP candidates. Selection can be accomplished using scoring or, as we do, using classification.

3.1. LPP generation

The candidate LPPs of a given word-form are generated using a morphology grammar (an inflectional morphology model). Te concrete implementation of the grammar does not concern us here. We assume that the grammar is generative (capable of generating word-forms given a lemma) and reductive (capable of lemmatizing a word-form). We can abstract this with two functions:

$$wfs(l, p) \mapsto \{(w_1, t_1), (w_2, t_2), \ldots, (w_n, t_n)\} \quad (1)$$

which, given a LPP, generates a set of word-forms $w_1, \ldots, w_n$ paired up with the corresponding morphological tags $t_1, \ldots, t_n$, and

$$lm(w) \mapsto \{(l_1, p_1), (l_2, p_2), \ldots, (l_m, p_m)\} \quad (2)$$

which lemmatizes a word-form to a set of candidate LPPs. In general, one lemma may be associated with more than one paradigm, and one paradigm may be associated with more than one lemma. We also assume that the grammar can reduce each lemma to its stem.

In this work we use the Croatian higher-order functional morphology (HOFM) grammar described by Šnajder and Dalbelo Bašić (2008) and refined by Šnajder (2010). The current version of the grammar uses 93 paradigms: 48 for nouns, 13 for adjectives, and 32 for verbs. The morphological tags are encoded as MULTEXT-East descriptors (Erjavec et al., 2003). Following are examples of word-form generation and lemmatization using the grammar:

```
> wfs "vojnik" N04
[("vojnik","N-msn"),("vojnika","N-msg"),
("vojnika","N-msa"),("vojnika","N-mpg"),
("vojniku","N-msl"),("vojniče","N-msv"),...]

> lm "vojnika"
[("vojnik",N01),("vojnikin",N03),
("vojnik",N04),("vojniak",N05),
("vojniak",N06),("vojniko",N17),...]
```

The second example illustrates the ambiguity of the grammar: many LPPs have been generated (22 in total), of which only the third one is correct. Despite the fact that HOFM defines applicability conditions for certain paradigms, the level of ambiguity is still quite large. On average, each word-form is lemmatized to 17 candidate LPPs, among which there are 7 distinct lemmas and 15 distinct paradigms.

3.2. LPP classification

Given candidate LPPs generated for an unknown word, we wish to decide which one is correct. In a supervised machine learning setting, the problem may in principle be cast as (1) multiclass classification (choosing one LPP among candidate LPPs), (2) multilabel classification (choosing a number of LPPs among candidate LPPs), or (3) binary classification (deciding for each LPP from candidate LPPs whether it is correct). The problem with (1) is that it does not account for homographs (the cases in which a single word-form has more than one correct LPP). The problem with (2) is that it is difficult to define the possible classes (they should encode both the stem transformation and the paradigm). Moreover, both (1) and (2) are difficult to combine with the output of a morphology grammar. Approach (3) is the most straightforward and we shall follow it here.

For classification, we use the SVM with an RBF kernel. The SVM algorithm tends to outperform other machine learning algorithms on a variety of learning problems. The RBF kernel implicitly defines an infinite-dimensional feature space, and is thus a good choice for problems for which the number of examples is much larger than the number of features, which will be the case here.

As source of training data, we use the semi-automatically acquired inflectional lexicon from (Šnajder et al., 2008). The lexicon contains 68,465 manually verified LPPs for Croatian nouns, adjectives, and verbs. We will use a fraction of this data for training and testing. It should be

noted that the distribution of LPPs in the lexicon with respect to the paradigms is very uneven; the ten least frequent paradigms appear only 40 times in the lexicon, whereas the ten most frequent paradigms appear over 50,000 times.

4. Features

Given an LPP, we compute a set of features based on which the LPP can be classified as either correct or incorrect. We distinguish between two groups of features: string-based and corpus-based.

4.1. String-based features

The string-based features are based on the orthographic properties of the lemma or the stem. The intuition behind this is that incorrect LPPs tend to generate ill-formed (or somewhat odd-formed) stems and lemmas. For example, there is no adjective in Croatian language that ends in *-kč*; an LPP that would generate such a stem could be discarded immediately. In fact, many paradigms defined in traditional grammar books are conditioned on the stem ending, requiring that it belongs to a certain group of phonemes or that it forms a consonant group. Similarly, there are paradigms that are applicable only to one-syllable stems. We use the following string-based features:

1. *EndsIn* – the ending character of the stem;
2. *EndsInCgr* – a binary feature indicating whether the word-forms ends in a consonant group (two consecutive consonants);
3. *EndsInCons* – a binary feature indicating whether the word-form ends in a consonant;
4. *EndsInNonPals* – a binary feature indicating whether the word-form ends in a non-palatal (*v*, *r*, *l*, *m*, *n*, *p*, *b*, *f*, *t*, *d*, *s*, *z*, *c*, *k*, *g*, or *h*);
5. *EndsInPals* – a binary feature indicating whether the word-form ends in a palatal (*lj*, *nj*, *ć*, *d*, *č*, *dž*, *š*, *ž*, or *j*);
6. *EndsInVelars* – a binary feature indicating whether the word-form ends in a velar (*k*, *g*, or *h*);
7. *LemmaSuffixProb* – the probability $P(s_l|p)$ of lemma l having a three-letter suffix s_l given inflectional paradigm p;
8. *StemSuffixProb* – the probability $P(s_s|p)$ of stem s having a three-letter suffix s_s given inflectional paradigm p;
9. *StemLength* – the number of characters in the stem;
10. *NumSyllables* – the number of syllables in the stem (determined heuristically);
11. *OneSyllable* – a binary feature indicating whether *NumSyllables* equals 1.

4.2. Corpus-based features

The corpus-based features are calculated based on the frequencies of word-forms attested in the corpus. The general idea is that a correct LPP should have more of its word-forms attested in the corpus than an incorrect LPP. Instead of only looking at total counts of attested word-forms, one can also look at the distributions of attested word-forms across the morphological tags. The intuition behind this is that every inflectional paradigm has its own distribution of morphological tags, and that a correct LPP will generate word-forms that obey such a distribution. For instance, in case of a noun paradigm, we can expect a genitive word-form to be far more frequent than a vocative word-form. Hence, an LPP that generates more vocative word-forms than genitive word-forms is unlikely to be correct.

In what follows, we use $\#(w, C)$ to denote the number of occurrences of word-form w in corpus C. Set $T(p)$ denotes the set of morphological tags of inflectional paradigm p. Let $P(t|p)$ denote the probability distribution of morphological tag t conditioned on the inflectional paradigm p, and let $P(t|l,p)$ denote the probability of morphological tag t generated by LPP (l,p). We obtain these distributions as maximum likelihood estimates using the LPPs from the inflectional lexicon L and word-form frequencies from corpus C:

$$P(t|p) = \frac{\sum_{(l,p')\in L;p'=p;(w,t')\in wfs(l,p);t'=t} \#(w,C)}{\sum_{(l,p')\in L;p'=p;w\in wfs'(l,p)} \#(w,C)}$$

$$P(t|l,p) = \frac{\sum_{(w,t')\in wfs(l,p);t'=t} \#(w,C)}{\sum_{w\in wfs'(l,p)} \#(w,C)}$$

where wfs' is a simpler version of the wfs function that only returns the word-forms. Notice that, because we do not perform POS tagging of the corpus, we count the ambiguous word-forms (inner and outer homographs) multiple times. We use the following corpus-based features:

1. *LemmaAttested* – a binary feature indicating whether the lemma is attested in the corpus, i.e., $\#(l, C) > 0$;
2. *Score0* – the number of corpus-attested word-form types generated by the LPP:
$$score_0(l,p) = |wfs'(l,p) \cap C|$$
3. *Score1* – the sum of corpus frequencies of word-forms generated by the LPP:
$$score_1(l,p) = \sum_{w\in wfs'(l,p)} \#(w,C)$$
4. *Score2* – the proportion of corpus-attested word-form types generated by the LPP:
$$score_2(l,p) = \frac{|wfs'(l,p) \cap C|}{|wfs'(l,p)|}$$
5. *Score3* – the product of paradigm-conditioned distribution of morphological tags and the distribution of tags generated by the LPP:
$$score_3(l,p) = \sum_{t\in T(p)} P(t|p) \times P(t|l,p)$$

6. *Score4* – the expected number of corpus-attested word-form types generated by the LPP:

$$score_4(l,p) = \sum_{t \in T(p)} P(t|p) \times \min\left(1, \#(w,C)\right)$$

7. *Score5* – the Kullback-Leibler divergence between the paradigm-conditioned distribution of morphological tags, $p_1(t) = P(t|p)$, and the distribution of tags generated by the LPP, $p_2(t) = P(t|l,p)$:

$$score_5(l,p) \quad = \quad \mathrm{KL}(p_1||p_2)$$

8. *Score6* – the Jensen-Shannon divergence between the aforementioned distributions:

$$score_6(l,p) = \mathrm{KL}(p_1||p_2) + \mathrm{KL}(p_2||p_1)$$

9. *Score7* – the cosine similarity between the aforementioned distributions:

$$score_7(l,p) = \frac{\sum_{t \in T(p)} p_1(t) \times p_2(t)}{\sqrt{\sum_{t \in T(p)} p_1(t)^2 \times \sum_{t \in T(p)} p_2(t)^2}}$$

We computed the above features on the *Vjesnik* newspaper corpus totaling 23 million word-form tokens and 330,298 word-form types (the same corpus was that used for lexicon acquisition in (Šnajder et al., 2008)).

4.3. Other features

Besides the string- and corpus-based features, we also use the following two features:

1. *ParadigmId* – a nominal feature denoting the LPP's inflectional paradigm;

2. *POS* – the part-of-speech of the LPP's inflectional paradigm (noun, adjective, or verb).

5. Evaluation

The purpose of evaluation is twofold: apart from determining how accurately we can guess the inflectional paradigms, we also wish to analyze what features are most useful for this task.

5.1. Data set

We compiled the data set for training and testing from the aforementioned inflectional lexicon (Šnajder et al., 2008). We sampled from the lexicon 5,000 LPPs for training and 5,000 LPPs for testing. Because the distribution of paradigms is very uneven, we used stratified sampling with respect to the inflectional paradigms. Moreover, we ensured that there is no LPP that appears in the test set, but does not appear in the training set (otherwise the probability distributions would be undefined). To generate the negative training and testing examples, we proceeded as follows. For each LPP, we generate all word-forms using the function wfs. Then, for all corpus-attested obtained word-forms, we generate the candidate LPPs using the function lm, and filter out those LPPs that exist in the lexicon. This generates a large number of incorrect LPPs, from which we again sample 5,000 for training and 5,000 for testing. Thus we end up with 10,000 LPPs (5,000 correct and 5,000 incorrect) in each the training and the test set. Given the number of classes and features (a total of 146 binary-encoded features), the amount of training data ought to be sufficient; a larger training set would unnecessary increase the time required for training. Notice that the training set contains correct and some incorrect LPPs for each selected word-form, while the test set contains LPPs obtained from word-forms that did not appear in the training set.

5.2. Feature analysis

Some of the features we defined are redundant or perhaps irrelevant for LPP classification. Because in absolute terms the number of features is not large, we need not perform feature analysis in order to reduce this number. Instead, the purpose of our feature analysis is to gain insight into what features are useful for paradigm guessing.

For feature analysis we used the open source tool Weka (Hall et al., 2009). Table 1 summarizes the results. We used three univariate filtering methods: information gain (IG), gain ratio (GR), and RELIEF method (Kononenko, 1994). We lists feature rankings obtained on the training set, with first five ranks shown in bold. The first two methods produced similar rankings: among string-based features, suffix probabilities are ranked the highest, while among corpus-based features, feature *Score5* is often ranked high, while ranks of other features vary. There are a number of features that are low-ranked (rank > 10) by each of the three methods: the five *EndsIn** features, *NumSyllables*, *OneSyllable*, *StemLength*, *Score1*, *Score3*, and *POS*.

The univariate methods do not measure the dependencies between the features, thus they cannot detect feature redundancy. We therefore also analyzed the features using two multivariate feature subset selection (FSS) methods: correlation-based feature selection (CFS) (Hall, 1998) and consistency subset selection (CSS) (Liu and R., 1996), both with greedy forward search as the optimization method. Table 1 shows the optimal subset selection obtained with each of these methods. Notice that both selected subsets contain both string- and corpus-based features.

5.3. Classification accuracy

For training and testing of models, we used the LIBSVM implementation of the SVM algorithm (Chang and Lin, 2011). We trained eight models using different feature subsets. We optimized the parameters of each model separately using 5-fold cross-validation on the training set. Classification accuracy on the test set is shown in Table 2. The reliability of probability estimates used for some of the corpus-based features depends on the frequencies of word-forms in the corpus. In a realistic setting, the unknown words tend to be less frequent in corpus. The last two columns of Table 2 show the classification accuracy for LPPs for which the frequency of word-forms in the corpus is less than or equal to 100 (rare words, accounting for 66% of the test set) and less than or equal to 10 (very rare words, accounting for 22% of the test set). The performance baseline is the majority class in each test set.

Table 1: Feature selection analysis

	Ranking			FSS	
Feature	IG	GR	RELIEF	CFS	CSS
String-based features:					
EndsIn	12	13	**2**		×
EndsInCgr	21	21	11		×
EndsInCons	17	15	20		
EndsInNonpals	22	22	19		
EndsInPals	19	18	21		
EndsInVelars	20	19	18		
LemmaSuffixProb	**2**	**2**	**3**		×
NumSyllables	14	14	12		×
OneSyllable	16	17	17		×
StemLength	15	16	15		×
StemSuffixProb	**1**	**1**	6	×	×
Corpus-based features:					
LemmaAttested	11	**3**	8	×	
Score0	8	**4**	16	×	
Score1	13	12	22		×
Score2	6	8	**5**		×
Score3	10	11	13		×
Score4	9	10	14		
Score5	**4**	**5**	**4**		
Score6	**3**	6	9		×
Score7	**5**	7	7		×
Other features:					
ParadigmId	7	9	**1**		×
POS	18	20	10		

Table 2: Classification accuracy (%)

	Word-forms attested		
Features (count)	≥ 1	≤ 100	≤ 10
All (22)	**91.97**	**91.94**	**90.65**
String-based (13)	87.01	87.69	87.98
Corpus-based (11)	87.78	86.59	82.04
IG (5)	81.14	79.05	76.46
GR (5)	59.76	80.90	77.29
RELIEF (5)	90.62	90.60	89.27
CFS (3)	81.69	79.51	78.67
CSS (13)	27.41	91.56	90.37
Baseline	50.00	56.51	69.92

As expected, the maximum accuracy of about 92% was achieved when using all features. Interestingly, in this case the classification accuracy does not decrease much on rare or very rare word-forms. Using only string- or corpus-based features gives worse performance than when using both kinds of features. Moreover, as expected, using only corpus-based features decreases the performance on rare words. As regards the models with feature selected subsets, all perform above the baseline except the one obtained with CSS. The RELIEF method seems to have selected a very good subset of features; a model with only five features (*ParadigmID*, *EndsIn*, *LemmaSuffixProb*, *Score5*, and *Score2*) performs just slightly worse than the model using the full set of 22 features.

6. Discussion

As the work described in this paper is preliminary, there are a number of issues that should be pointed out, especially as regards the evaluation.

Considering that on average there are 17 candidate LPPs per word-form, accuracy of 92% means that for each unknown word we would on average wrongly classify at least one candidate LPP. However, the problem with the above evaluation is that the test set is balanced in the number of positive and negative examples. In reality, there are more negative examples (incorrect LPPs) than positive examples, of which many can probably be classified as such with high confidence. For future work, we need to evaluate the classifier in terms of precision and recall on a per word basis.

In this work we ignored the classifier confidence scores, which may be used to produce rankings. Paradigm guessing is often addressed as a ranking task, and it would make sense to evaluate it as such. It would also be possible to build a metaclassifier that uses the confidence scores assigned to candidate LPPs to decide which LPP to choose. Moreover, ranking-based classification enables the interactive use of a paradigm guesser, which is very convenient for semi-automatic lexicon enlargement.

Another issue that we did not address is the size and diversity of the training set. Often a large morphological lexicon is not available, and one wishes to use paradigm guessing to acquire such a lexicon. Related to this is the question of how many examples per paradigm we need to learn a good classifier. The active learning framework provides a way to minimize the number of training examples and hence reduce the manual labeling efforts. Active learning may also be combined with ranking-based classification to speed up the annotation process.

Furthermore, there are three additional evaluation scenarios that may be considered. First is the evaluation in the context of rule-based tagging (e.g., constraint grammar based tagging, as described by Peradin and Šnajder (2012)), in which the goal is to disambiguate ambiguous morphosyntactic tags, rather than ambiguous paradigms (the former is probably an easier task in most cases). Related to this is a setting in which corpus-based information is not available (e.g., on-the-fly tagging), and one must choose the correct paradigm using only string-based and possibly context-based features. Yet another interesting evaluation scenario is the acquisition of inflectional lexicons from a list of lemmas, which is obviously an easier task than the one we addressed here because the level of grammar ambiguity is lower.

7. Conclusion

We have addressed the problem of paradigm guessing for unknown Croatian words as a binary classification task over the output of a morphology grammar. We defined a

number of string- and corpus-based features and trained different models on selected subsets of these features. The highest accuracy (about 92%) was achieved using the complete set of 22 features. Just slightly worse performance can be obtained with a subset of only five features (a combination of string- and corpus-based features). Degradation in classification performance on rare words is minimal.

We have outlined several directions for further research. We plan to evaluate paradigm guessing as a ranking task on a per word basis, in the context of semi-automatic lexicon acquisition. We also intend to apply paradigm guessing for rule-based POS tagging of Croatian. From a machine learning perspective, we intend to experiment with additional features (including context-based features).

8. Acknowledgments

This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under Grant 036-1300646-1986.

9. References

P. Adolphs. 2008. Acquiring a poor man's inflectional lexicon for German. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco*.

C.-C. Chang and C.-J. Lin. 2011. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pages 2:27:1–27:27.

K. Cholakov and G. Van Noord. 2009. Combining finite state and corpus-based techniques for unknown word prediction. In *Proceedings of the 7th Recent Advances in Natural Language Processing (RANLP) conference*.

L. Clement, B. Sagot, and B. Lang. 2004. Morphology based automatic acquisition of large-coverage lexica. In *Proceedings of LREC'04*, pages 1841–1844, May.

T. Erjavec, C. Krstev, V. Petkevič, K. Simov, M. Tadić, and D. Vitas. 2003. The MULTEXT-East morphosyntactic specifications for Slavic languages. In *Proceedings of the EACL2003 Workshop on Morphological Processing of Slavic Languages*, pages 25–32.

M. Esplá-Gomis, V.M. Sánchez-Cartagena, and J.A. Pérez-Ortiz. 2011. Enlarging monolingual dictionaries for machine translation with active learning and non-expert users. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2011)*, pages 411–415.

M. Forsberg, H. Hammarström, and A. Ranta. 2006. Morphological lexicon extraction from raw text data. In *FinTAL*, pages 488–499.

J. Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The Weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

M. A. Hall. 1998. Correlation-based feature subset selection for machine learning. Technical report.

J. Hana. 2008. Knowledge- and labor-light morphological analysis. *Ohio State University Working Papers in Linguistics*, 58:52–84.

C. F. Hockett. 1954. Two models of grammatical description. *Word*, 10:210–234.

T. Kaufmann and B. Pfister. 2010. Semi-automatic extension of morphological lexica. In *Computer Science and Information Technology (IMCSIT), Proc. of the 2010 International Multiconference on*, pages 403–409. IEEE.

I. Kononenko. 1994. Estimating attributes: Analysis and extensions of relief. In *European Conference on Machine Learning*, pages 171–182.

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6(3):225–242.

H. Liu and Setiono R. 1996. A probabilistic approach to feature selection - a filter solution. In *13th International Conference on Machine Learning*, pages 319–327.

A. Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

P. Nakov, Y. Bonev, G. Angelova, E. Cius, and W. Von Hahn. 2004. Guessing morphological classes of unknown German nouns. *Recent Advances in Natural Language Processing III (RANLP'03), Nicolov, Nicolas, Kalina Bontcheva, Galia Angelova and Ruslan Mitkov (eds.)*, pages 347–356.

A. Oliver and M. Tadić. 2004. Enlarging the Croatian morphological lexicon by automatic lexical acquisition from raw corpora. In *Proceedings of LREC'04*, pages 1259–1262.

A. Oliver. 2003. Use of internet for augmenting coverage in a lexical acquisition system from raw corpora. In *Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL 2003), RANLP*.

H. Peradin and J. Šnajder. 2012. Towards a constraint grammar based morphological tagger for Croatian. In *Text, Speech and Dialogue*, pages 174–182. Springer.

B. Sagot. 2005. Automatic acquisition of a Slovak lexicon from a raw corpus. *Lecture Notes in Computer Science*, 3658:156–163.

I. Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. *Proceedings of MLMTA*.

J. Šnajder and B. Dalbelo Bašić. 2008. Higher-order functional representation of Croatian inflectional morphology. In *Proceedings of the 6th International Conference on Formal Approaches to South Slavic and Balkan Languages, FASSBL6*, pages 121–130, Dubrovnik, Croatia. Croatian Language Technologies Society.

J. Šnajder, B. Dalbelo Bašić, and Tadić M. 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing and Management*, 44(5):1720–1731.

J. Šnajder. 2010. *Morfološka normalizacija tekstova na hrvatskome jeziku za dubinsku analizu i pretraživanje informacija*. Ph.D. thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb.

M. Tadić and S. Fulgosi. 2003. Building the Croatian morphological lexicon. In *Proceedings of EACL'2003*, pages 41–46.

Razpoznavanje imenskih entitet v slovenskem besedilu

Tadej Štajner[*†], **Tomaž Erjavec**[*†], **Simon Krek**[*]

[*]Institut "Jožef Stefan",
Jamova cesta 39 1000 Ljubljana
{tadej.stajner, tomaz.erjavec}@ijs.si, simon.krek@guest.arnes.si

[†]Mednarodna podiplomska šola Jožefa Stefana
Jamova cesta 39, 1000 Ljubljana

Povzetek

Članek predstavi algoritem in implementacijo programa za razpoznavanje imenskih entitet v slovenskem jeziku s pomočjo strojnega učenja. Nadzorovan pristop na osnovi pogojnih naključnih polj je naučen na označenem korpusu ssj500k. V korpusu, ki je prosto dostopen pod licenco Creative Commons, so pri besednih pojavnicah poleg oblikoskladenjskih oznak oz. lastnosti in lem označena tudi osebna, zemljepisna ter stvarna imena. Članek predstavi vpliv na natančnost razpoznavanja ob uporabi oblikoskladenjskih oznak, leksikonov ter konjunkcij sosednjih značilk. Pomembna ugotovitev raziskave je, da oblikoskladenjske oznake koristijo pri razpoznavanju entitet. V kombinaciji z vsemi ostalimi značilkami doseže sistem na testni množici 77% natančnost in 75% priklic, pri čemer so lastna in zemljepisna imena razpoznana bistveno bolje kot stvarna, saj je razred stvarnih imen zelo raznolik in zato težaven za učenje. Programska oprema, razvita in uporabljena v teh poskusih, je prosto dostopna pod licenco Apache 2.0 na naslovu `http://ailab.ijs.si/~tadej/slner.zip`.

Named Entity Recognition in Slovene text

This paper presents an approach and an implementation of a named entity extractor for Slovene language, based on a machine learning approach. It is designed as a supervised algorithm, based on Conditional Random Fields and is using the ssj500k annotated Slovene corpus for training data. The corpus, which is available under a Creative Commons licence, is annotated with morphosyntactic tags, as well as named entities of people, locations and real names of other entities. The paper discusses the influence of morphosyntactic tags, lexicons and offset conjunctions of features of neighboring words. An important contribution of this investigation is that morphosyntactic tags benefit named entity extraction. In concert with all other features, it reaches a precision of 77% and a recall of 76%, having stronger performance on personal and geographical named entities than on other entities, since the class of other entities is very diverse and consequently difficult to predict. The software, developed in this research is freely available under the Apache 2.0 licence at `http://ailab.ijs.si/~tadej/slner.zip`.

1. Uvod

Članek opisuje sistem, namenjen razpoznavanju imenskih entitet v slovenskih besedilih. Razpoznavanje pojavnih oblik entitet (v angleščini *entity extraction, named entity recognition, entity identification*) je pomembna naloga pri izločanju informacij iz besedil, saj besede ali besedne zveze, ki predstavljajo imenske entitete, na primer lastno ime osebe, kraja ali organizacije, k vsebini besedila skupaj prispevajo več informacij, kot bi bilo moč razbrati zgolj iz števila posameznih besed.Razpoznavanje entitet obravnava besedilo na drugem nivoju abstrakcije, ker ne govorimo več o posameznih besedah, temveč (največkrat) o dvo- ali večbesednih entitetah. Pri iskanju informacij so lastna imena torej predstavljena kot entiteta, kar nam omogoča, da na besedilni korpus ali podatkovno bazo pogledamo na drugačen način - skozi indeksacijo entitet, ki se v tem korpusu pojavljajo. Na primer, *prikaži mi vse članke o Institutu Jožef Stefan*. V časopisni industriji in založništvu je pogosta praksa, da entitete in ključne besede, ki se pojavijo v člankih, indeksirajo ročno. Nekatere časopisne hiše to počnejo že od 19. stoletja, New York Times denimo od leta 1851 (Sandhaus, 2008). Razpoznavanje imen oseb, krajev in stvarnih imen se lahko uporablja tudi za namen povezovanja zgodb v časopisnih člankih (Štajner and Grobelnik, 2009), kjer uporaba entitet (poleg samega besedila) prispeva k natančnejšemu povezovanju različnih člankov v smiselne verige. V angleško govorečem delu znanstvene skupnosti je tehnologija razpoznavanja entitet doživela hiter razvoj v veliki meri kot rezultat serije konferenc *Message Understanding Conference* (Grishman and Sundheim, 1996), ki se je odvijala v devetdesetih letih in *TREC* (Balog et al., 2010), ki se v okviru sistemov za priklic informacij odvija še dandanes. V okviru obeh konferenc so bila organizirana odprta tekmovanja v raznih nalogah iz procesiranja naravnega jezika, pri čemer je bilo veliko nalog osredotočenih na razpoznavanje entitet. Najzmogljivejši sistemi trenutno uporabljajo predvsem postopke strojnega učenja, natančneje modele na probablističnih grafih, kot so npr. skriti Markovski modeli (*Hidden Markov Models*) (Rabiner and Juang, 1986) ali pogojna naključna polja (*Conditional Random Fields*) (Lafferty et al., 2001), npr. Mallet (McCallum, 2002) ali Stanford NER (Finkel et al., 2005). V praksi so ti sistemi implementirani z nadzorovanim učenjem na besedilu, kjer so entitete že označene. V procesu učenja se za vsako besedo generirajo posamezne lastnosti, kot na primer oblikoskladenjske oznake, velike začetnice, prisotnost pomišljaja in podobno, v procesu označevanja pa sistem uporabi model, zgrajen na osnovi teh lastnosti. Nekateri sistemi uporabljajo tudi eksplicitno predznanje, njihova slabost pa je ta, da ne zaznajo neznanih entitet, če jih nimajo v obstoječem leksikonu. Zato se jih pogosto kombinira s sistemom, osnovanem na stroj-

nem učenju, tako da tvorita hibridni sistem (Cohen and Sarawagi, 2004). Nekateri sistemi uporabljajo tudi nenadzorovano izločanje entitet, saj ta pristop ne zahteva vnaprejšnjega učenja (Etzioni et al., 2005).

V nadaljevanju ima članek naslednjo strukturo: v razdelku 2 predstavimo korpus, na katerem je bil sistem naučen in testiran, v razdelku 3 opišemo razviti razpoznavalnik, v razdelku 4 poskuse, ki smo jih izvedli, razdelek 5 pa vsebuje zaključke.

2. Učni korpus ssj500k

Za nadzorovano učenje je potreben korpus, kjer so pojavitve lastnih imen ustrezno označene. Za slovenski jezik do sedaj še nismo imeli tako označenega korpusa, vendar je bil pred kratkim izdelan ročni označeni korpus ssj500k, ki je bil nastal v okviru projekta Sporazumevanje v slovenskem jeziku (SSJ) in temelji na učnih korpusih jos100k in jos1M, izdelanih v okviru projekta JOS (Erjavec et al., 2010b; Erjavec et al., 2010a). Korpus ssj500k sestavljata dva dela: celotni korpus jos100k ter dodatnih 400.000 besed iz enomilijonskega korpusa jos1M. Vsi jezikoslovni metapodatki (oznake, leme, tokenizacija) so bili v korpusu ssj500k še enkrat ročno pregledani, skladenjsko razčlenjeni del pa je bil povečan na 11.411 stavkov. V celoti je bila ročno pregledana in popravljena tudi stavčna segmentacija in tokenizacija, kar omogoča preverjanje uspešnosti označevalnikov in razčlenjevalnikov tudi pri teh dveh postopkih. Učni korpus ssj500k je prosto dostopen pod licenco Creative Commons Priznanje avtorstva-Nekomercialno 3.0"na spletnih straneh projekta SSJ `http://www.slovenscina.eu/` oz. `http://nl.ijs.si/ssj/`. V delu, ki vsebuje podatke iz korpusa jos100k, so bile dodane tudi informacije o lastnih imenih za potrebe strojnih razpoznavalnikov imenskih entitet. Ta del zajema petino celotnega korpusa ssj500k, podatki zgolj za ta podkorpus (ssj100k) so podani v Tabeli 1.

elementov	n
besedil	248
odstavkov	1.599
stavkov oz. povedi	5.808
besed	100.135
ločil in simbolov	18.499
skladenjsko označenih stavkov	5.808
skladenjskih povezav	118.635
stavkov z imenskimi entitetami	2.177
imenskih entitet	4.397

Tabela 1: Število elementov v podkorpusu ssj500k, označenem s podatki o imenskih entitetah oz. lastnih imenih

Lastna imena v korpusu so segmentirana v tri kategorije: osebna (1.921), zemljepisna (1.284) in stvarna (1.192). Lastna imena vsebuje 2.177 oz. 37,48 % vseh stavkov, pri čemer je distribucija lastnih imen po teh stavkih razmeroma neenakomerna. Več kot polovico jih vsebuje eno lastno ime, četrtina dve, desetina tri, temu sledi potem dolgi rep do stavka s 47 kar lastnimi imeni.

3. Implementacija

V skladu s trenutno prakso obstoječih sistemov za druge jezike implementacija sistema, predstavljenega v tem članku, uporablja nadzorovano učenje s pogojnimi naključnimi polji (*Conditional Random Fields*, ali krajše CRF), ki temelji na sistemu Mallet (McCallum, 2002).

3.1. Model

Pogost pristop pri modeliranju zaznavanja imenskih entitet je verižni model, kjer besede označujemo zaporedno, pri vsaki odločitvi pa med drugim upoštevamo tudi odločitev klasifikacije na prejšnjem koraku. V takšnem modelu, kot je na primer sekvenčni CRF, so stanja določena z željenimi oznakami, ki predstavljajo tipe entitet. Množica stanj modela je torej $osebno, zemljepisno, stvarno, brez$. Ko predstavimo stavek kot zaporedje besed, v postopku označevanja vsaki besedi priredimo oznako najverjetnejšega stanje glede na oznako prejšnje besede ter glede na značilke trenutne besede. V splošnem lahko v modelu CRF predstavimo tudi odvisnosti višjih redov ali odvisnosti v poljubnem acikličnem grafu, vendar je za označevanje besedil najprimernejši verižni model prvega reda. Z drugimi besedami, model, ki ima lastnost, da je trenutno stanje odvisno le od lokalnih značilk ter od predhodnega razreda besede.

Naj bo $G = (V, E)$ graf, kjer je $Y = (Y_v)_{v \in V}$, tako da posamezne dimenzije Y predstavljajo vozlišča G. Iz stališča uporabe je X množica primerov v obliki vektorjev značilk, Y pa naključna spremenljivka, kjer posamezno stanja Y_v predstavljajo tudi ciljne razrede - tipe imenskih entitet. (X, Y) je pogojno naključno polje, če imajo naključne spremenljivke Y_v Markovsko lastnost glede na sosednost, kar pomeni, da je novo stanje odvisno le od prejšnjega stanja: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim w)$, kjer $w \sim v$ pomeni, da sta w in v soseda. Konkretno, to pomeni, da je oznaka trenutne besede odvisna od značilk trenutne besede ter oznake prejšnje besede.

Pogojna verjetnost med X in Y je tako opisana z množico funkcij značilk oblike $f_k(y, y', x_t)_{k=1}^K \in Re^K$. Na primer, $f_{upper-person}$ je lahko tovrstna funkcija, ki vrne 1 v primeru, ko se trenutna beseda začne z veliko začetnico in da je predhodna beseda označena kot osebno ime, sicer pa 0. Linearno verižno pogojno naključno polje (*Linear chain CRF*) je v tem primeru porazdelitev $p(y|x)$, ki jo opišemo z množico parametrov $\Lambda = \lambda_k \in \Re^K$:

$$p(y|x) = \frac{1}{Z(x)} \exp\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\} \quad (1)$$

kjer je $Z(x)$ normalizacijska funkcija. Za uporabo modela je nato potrebno oceniti vrednosti parametrov Λ, ki nam povedo v kolikšni meri je določena značilka povezana z določenim ciljnim razredom. V ta namen se tipično uporablja maksimizacija regulariziranega pogojnega log-verjetja (*conditional log-likelihood*) glede na množico učnih primerov, kar lahko predstavimo s sledečo enačbo:

$$l(\Lambda) = \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_t^{(i)}, y_{t-1}^{(i)}, x_t^{(i)}) -$$

$$-\sum_{i=1}^{N} \log Z(x^{(i)}) \\ -\sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2} \quad (2)$$

Zadnji del enačbe predstavlja regularizacijo, ki se uporablja za preprečevanje prekomernega prileganja modela podatkom. Izraz $\frac{1}{2\sigma^2}$ predstavlja moč regularizacije, ki nam pove, kolikšno kazen dobijo previsoke uteži λ_k. Ker pa $l(\Lambda)$ ni moč maksimizirati v zaprti obliki, se v ta namen uporablja numerična optimizacija s pomočjo delnih odvodov. V ta namen za rešitev optimizacije ocenjevanja parametrov uporabimo optimizacijski algoritem L-BFGS (Byrd et al., 1994). Ko se naučimo parametre modela, jih lahko uporabimo za označevanje neoznačenega besedila. Za to uporabimo inferenčni algoritem *loopy belief propagation* (Sutton and McCallum, 2004).

3.2. Značilke

Pri implementaciji pristopa za razpoznavanje entitet je ključno, da si lahko pomagamo s čim bolj raznolikimi tipi informacij. V ta namen uvajamo štiri kategorije značilk, kjer vsaka kategorija prinaša dodatno informacijo, kar demonstriramo s poskusi v razdelku 4.

3.2.1. Značilke črkovnih vzorcev

Pri obstoječih pristopih za zaznavanje entitet so najbolj tipične značilke grafemskih vzorcev, kjer lahko vsako posamezno besedo opišemo z binarno značilko vzorca. S pomočjo regularnih izrazov smo določili naslednje značilke, ki se že uporabljajo pri zaznavanju imenskih entitet. Značilka dobi vrednost 1 le če beseda ustreza regularnemu izrazu. To množico značilk vzamemo kot osnovo, kateri nato dodajamo ostale razrede.

Uporabljene značilke črkovnih vzorcev so:

- Velika začetnica le na začetku besede (npr. Ljubljana)
- Le velike črke v celotni besedi (npr. IJS)
- Mešane velike črke znotraj besede (npr. iPod)
- Številke v besedi (npr. ZVCP-1)
- Le številke v besedi (npr. 2012)
- Numerični izraz (npr. +3.14)
- Alfanumerični izraz, ki vsebuje le številke in črke (npr. E3)
- Rimska številka (npr. XVIII)
- Vsebuje vezaj ali pomišljaj (npr. Šmarje-Sap)
- Kratica, sestavljena le iz velikih črk, lahko ločenih s piko (npr. I.M.V.)
- Inicialka, posamezna velika črka, ki ji sledi pika. (npr. John *F.* Kennedy)
- Posamezna črka, ne glede na velike ali male črke (npr. odgovor *a*)
- Pozamezna velika črka (npr. plan *B*)
- Ločilo (npr. !)
- Narekovaj (npr. “)
- Le male črke (npr. *pisarna*)

3.2.2. Značilke zunanjih virov znanja

Poleg črkovnih vzorcev lahko uporabljamo tudi zunanje znanje v obliki leksikonov, ki vsebujejo že znana imena entitet. S tem pristopom v model vključimo znanje, ki bi ga bilo le z nadzorovanim učenjem težko nadomestiti. V ta namen definiramo leksikonsko značilko, ki dobi vrednost 1.0 le, če je lema besede vsebovana v določenemu leksikonu, kjer imamo za vsak leksikon po eno binarno značilko. Uporabimo prisotnost *leme*, saj bi bilo pripadnost posameznemu leksikonu pri besedni obliki zaradi bogate slovenske morfologije težko preverjati. Vsak leksikon se prevede v eno značilko. Večino leksikonov smo vzeli iz slovenske različice Wikipedije, ki je prosto dostopni vir in obsega dovolj široko paleto tematskih domen za splošno razpoznavanje entitet. Uporabljajo se sledeči leksikoni:

- kraji v Sloveniji iz slovenske Wikipedije (Wikipedia, 2012f)
- države iz slovenske Wikipedije (Wikipedia, 2012g)
- kraji v tujini iz slovenske Wikipedije (Wikipedia, 2012b)
- občine v Sloveniji iz slovenske Wikipedije (Wikipedia, 2012d)
- tipične besede v lokacijah (npr. vas, mesto, trg, gora)
- tipične besede v organizacijah (npr. institut, ministrstvo)
- tipične predpone in pripone osebnih imen (npr. dr, mag, ml)
- seznam pogostih in redkih imen iz Statističnega urada Slovenije (Statistični Urad Slovenije, 2012)
- seznam moških imen iz slovenske Wikipedije (Wikipedia, 2012c)
- seznam ženskih imen iz slovenske Wikipedije (Wikipedia, 2012a)
- seznam priimkov iz slovenske Wikipedije (Wikipedia, 2012e)
- imena dni v tednu
- imena mesecev

V primeru, da besedilo ni lematizirano, lahko uporabimo tudi samodejni lematizator (**?**).

3.2.3. Značilke oblikoskladenjskih lastnosti

Tretji potencialni vir informacij za razpoznavanje entitet so v korpusu že prisotne oblikoskladenjske oznake besed, ki jih prevedemo v značilke po specifikacijah tabele oznak (Erjavec et al., 2010c). Na primer, beseda *narediti* z oznako *Ggdn* dobi značilke *Category=verb, VerbType=main, Aspect=perfective, VForm=infinitive*, beseda *predsednik* z oblikoskladenjsko oznako *Sometd* pa značilke *Category=noun, NounType=common, Gender=masculine, Number=singular, Case=accusative, Animate=yes*. Če besedilo ni označeno, lahko uporabimo ustrezni oblikoskladenjski označevalnik (Rupnik et al., 2008). Uporaba oblikoskladenjskih oznak temelji na predpostavki, da iz vzorcev oznak lahko razbremo prisotnost entitet. Uporaba mestnika v kombinaciji z veliko začetnico lahko denimo nakazuje prisotnost zemljepisnega imena.

3.2.4. Strukturne značilke

Poleg regularnih izrazov, leksikonov in oblikoskladenjskih oznak lahko uporabljamo tudi različne strukturne značilke, ki izvirajo iz same zgradbe stavka kot zaporedja besed. Prva množica strukturnih značilk izvira iz **dolžine besede**, ki jo razbijemo v razrede dolžin 1, 2, 3 ali 4, od 5 do 9, ali več kot 10 znakov, sama značilka pa je odvisna od pripadnosti tem razredom (npr. *Length=5*).

Druga množica strukturnih značilk, **konjunkcija sosednjih značilk**, je definirana kot preslikava nad obstoječimi značilkami. To je metoda za generiranje dodatnih značilk, ki za vsako besedo sestavi nove značilke kot kombinacije značilk njenih sosedov znotraj določenega okna. Uporablja se predvsem v tistih verižnih klasifikatorjih, kjer soodvisnosti med značilkami in razredi niso odvisne le od prejšnjega in trenutnega stanja, ampak tudi od širše okolice, kar je lahko še posebej poudarjeno pri jezikih s prostim besednim redom. Ker je eksplicitno modeliranje soodvisnosti višjega reda računsko zelo zahtevno, konjunkcije sosednjih značilk uporabimo kot približek. Na primer, če se trenutna beseda nahaja dve mesti za besedo z veliko začetnico, dobi značilko $kapitalizacija_{-2}$. V nadaljnih poskusih obravnavamo tri možne razpone vzorcev: le predhodna in naslednja $((-1),(1))$, vse možne kombinacije parov predhodnika, trenutnega in naslednika $((-1,0),(-1,1),(0,1))$, tretji razpon pa predstavlja vse možne kombinacije parov značilk v razponu dve mesti naprej ter nazaj, npr. kombinacija $-2,1$ predstavlja konjunkcijo značik besede dve mesti pred trenutno z značilkami naslednje besede. Tovrstno generiranje značilk lahko izredno poveča število možnih značilk in s tem upočasni učenje ter povečuje nevarnost prekomernega prilagajanja.

4. Poskusi

S poskusi smo želili odgovoriti na vprašanja glede smiselnosti uporabe različnih razredov značilk glede na meritve:

- Ali oblikoskladenjske oznake izboljšajo model?
- Ali uporaba leksikonov izboljša model?
- Ali kombinacije parov značilk v soseščini izboljšajo model?

Poskuse smo izvedli z desetkratnim navzkrižnim preverjanjem, kjer naključnih devetdeset odstotkov podatkov uporabimo za učenje, preostalo pa za testiranje. Kakovost rezultata merimo z več metrikami: natančnostjo, ki nam pove, koliko od dobljenih entitet je pravilnih, priklicem, ki nam pove, koliko znanih entitet smo identificirali ter F_1, ki je geometrijsko povprečje natančnosti in priklica. Zaradi preglednosti obravnavamo vsako hipotezo posebej. Vsak nadaljni poskus kot osnovo uporablja različico predhodnega poskusa, ki je v tistem krogu imela najboljši izid.

Rezultati v tabeli 2 potrjujejo, da so oblikoskladenjske oznake pri razpoznavanju entitet izjemno koristne , saj sta tako priklic kot tudi natančnost pri vseh meritvah statistično značilno višja kot brez uporabe oznak. Poskusi tudi kažejo, da je sistem razmeroma uspešen pri razpoznavanju osebnih imen, nekaj slabši pri zemljepisnih imenih in neuspešen pri prepoznavanju stvarnih imen. Zemljepisna imena je lažje

Tip entitete	Natančnost	Priklic	F_1
Brez oblikoskladenjskih oznak			
Osebna	0.6207	0.6533	0.6342
Zemljepisna	0.4426	0.4868	0.4595
Stvarna	0.3171	0.1970	0.2412
Skupno	0.4932	0.4464	**0.4681**
Z oblikoskladenjskimi oznakami			
Osebna	0.7632	0.8526	0.8046
Zemljepisna	0.7303	0.6770	0.7016
Stvarna	0.5756	0.4283	0.4881
Skupno	0.7011	0.6585	**0.6788**

Tabela 2: Rezultati poskusov glede na uporabljene oblikoskladenjske oznake

razpoznati, ker gre za samostalnike z veliko začetnico, ki so tipično v mestniku, stvarna imena pa pogosto sestavljejo daljše zveze s pridevniki (*Evropska komisija*) ali predlogi (*Ministrstvo za obrambo*) in različnimi skloni znotraj besedne zveze, zaradi česar je možnih variacij preveč, da bi jih lahko zajeli v obstoječih učnih podatkih. V nadaljnih poskusih privzemamo, da so oblikoskladenjske oznake vedno prisotne. Z odebeljeno je označena glavna metrika - povprečje F_1 čez vse razrede.

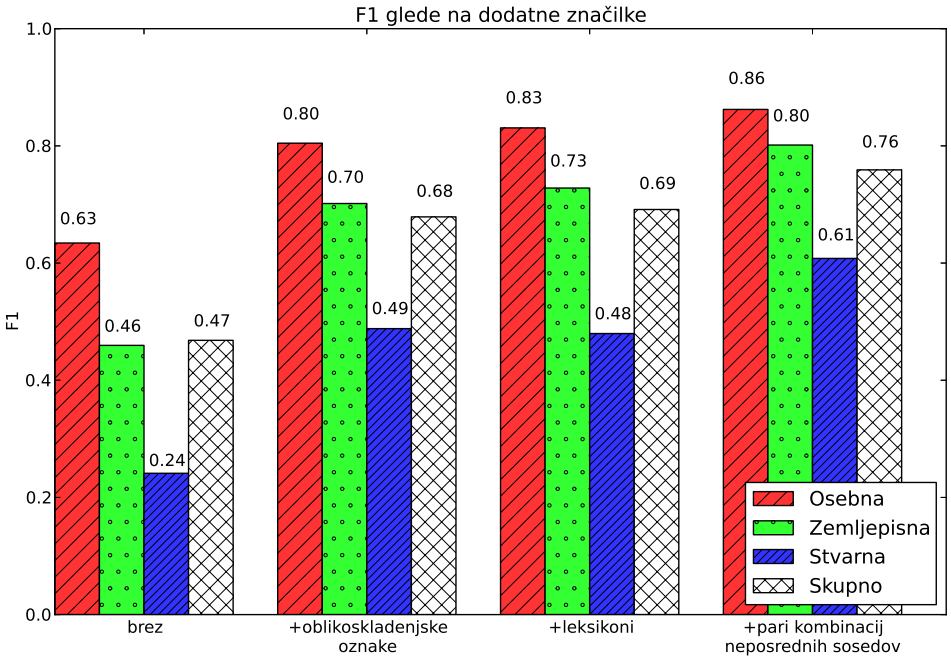
Tip entitete	Natančnost	Priklic	F_1
Z uporabo leksikonov			
Osebna	0.7851	0.8832	0.8307
Zemljepisna	0.7727	0.6892	0.7280
Stvarna	0.5524	0.4261	0.4797
Skupno	0.7126	0.6718	**0.6914**

Tabela 3: Rezultati poskusov glede na uporabljene leksikone

Tabela 3 kaže, da uporaba leksikonov opazno dvigne priklic in natančnost pri osebnih ter zemljepisnih imenih, medtem ko stvarna imena nimajo statistično značilne spremembe v primerjavi z uporabo le oblikoskladenjskih oznak brez leksikonov v drugem delu tabele 2. Skupna F_1 je statistično značilno višja od F_1, ko so uporabljene le oblikoskladenjske oznake. Ker so leksikoni uporabni le v primeru, da je besedilo lematizirano, je uspešna identifikacija imenskih entitet odvisna tudi od obstoja lematizatorja. V nadaljnih hipotezah in poskusih privzemamo uporabo značilk leksikonov in oblikoskladenjskih oznak kot osnovno verzijo.

Iz tabele 4 je razvidno, da je najboljše delovanje modela doseženo takrat, ko uporabimo kombinacije parov značilk neposrednih sosedov, in da je razširjanje soseščine škodljivo, saj se dimenzionalnost prostora značilk s tem močno poveča, kar otežuje proces učenja, saj je število primerov bistveno manjše ne le od števila vseh možnih značilk, temveč tudi ne-ničelnih značilk. Rezultati so podobni pri vseh tipih entitet, kar nakazuje, da je optimalno uporabiti le kombinacije značilk neposrednih sosedov. V primerjavi s tabelo 3 katerakoli uporaba kombinacij parov izboljša F_1, saj se iz 0.69 povzpne od 0.73 do 0.76, odvisno od števila kombinacij soseščine.

Slika 1 kaže rast F_1 metrike glede na dodajanje novih značilk. Meritve skrajno levo uporabljajo le značilke regularnih izrazov ter dolžino besede, naslednje meritve pa



Slika 1: F_1 glede na najboljše kumulativno dodane značilke

Tip entitete	**Natančnost**	**Priklic**	F_1
Značilke neposrednih sosedov			
Osebna	0.8255	0.8788	0.8507
Zemljepisna	0.8093	0.7492	0.7762
Stvarna	0.5851	0.5167	0.5469
Skupno	0.7405	0.7157	**0.7277**
Kombinacije parov značilk neposrednih sosedov			
Osebna	0.8463	0.8816	0.8622
Zemljepisna	0.8279	0.7786	0.8014
Stvarna	0.6472	0.5774	0.6079
Skupno	0.7729	0.7458	**0.7590**
Kombinacije parov značilk soseščine [-2,+2]			
Osebna	0.8310	0.8789	0.8538
Zemljepisna	0.8076	0.7616	0.7819
Stvarna	0.6494	0.5568	0.5966
Skupno	0.7656	0.7334	**0.7489**

Tabela 4: Rezultati poskusov glede na različne kombinacije parov značilk sosedov

kažejo razlike pri dodajanju novih značilk. Tu lahko vidimo, da oblikoskladenjske oznake statistično značilno izboljšajo kakovost na vseh tipih entitet, medtem ko leksikoni izboljšajo le osebna in zemljepisna imena, saj za stvarna imena še nismo uporabili primernega leksikona. Kljub temu pa so ravno stvarna imena imela najvišji napredek pri dodajanju parov kombinacij sosednjih značilk, kar lahko pojasnimo s tem, da so stvarna imena pogostokrat daljša in bolj odvisna od širšega konteksta. Podrobnejša analiza napak pokaže, da stvarna imena zajemajo mnogo različnih tipov entitet, kar učnemu modelu otežuje posploševanje. V literaturi (Grishman and Sundheim, 1996) se uporablja ožje definirane tipe, kot na primer *organizacija*, *geopolitična entiteta*, *izdelek* ter *dogodek*, saj je pri ožjih tipih lažje doseči višjo natančnost izločanja.

5. Zaključek

Članek je opisal implementacijo razpoznavanja entitet v slovenskem besedilu s pomočjo nadzorovanega učenja pogojnih naključnih polj z oblikoskladenjskimi in besednimi lastnostmi. Rezultati kažejo na visoko zanesljivost zaznavanja lastnih in zemljepisnih imen in nekoliko manj zanesljivo zaznavanje stvarnih imen, kar je glede na majhen učni korpus zadovoljiv rezultat. Rezultati tudi potrjujejo, da v slovenskem jeziku oblikoskladenjske oznake koristijo pri razpoznavanju entitet, prav tako pa se da kakovost izboljšati z uporabo leksikonov ter kombinacij značilk sosednjih besed. Pri stvarnih imenih bi bilo moč doseči boljši rezultat, če bi jih natančneje delili na organizacije, dogodke, izdelke in ostala stvarna imena, saj je trenutno razred stvarnih imen zelo raznolik in s tem težaven za učenje. Obstoj sistema za razpoznavanje entitet predstavlja tudi pomemben korak za razvoj sistema za razločevanje entitet (Štajner and Mladenić, 2009), ki razpoznavanje nadgradi še z določanjem točne identitete entitete. Razločevanje entitet nam omogoča povezovanje nestrukturiranih besedil s strukturiranimi podatkovnimi bazami, nove metode pa nam omogočajo tudi razločevanje enititet iz slovenskega besedila in povezovanje s podatkovnimi bazami, izraženimi v drugem jeziku (Štajner and Mladenić, 2012). Da bi bil sistem uporaben tudi za razpoznavanje entitet v besedilih brez oblikoskladenjskih oznak, je bila narejena tudi integracija z oblikoskladenjskim označevalnikom (Rupnik et al., 2008), ki je na voljo v slovenski različici spletne storitve Enrycher (Štajner et al., 2010). Programska oprema, razvita in uporabljena v teh poskusih, je prosto dostopna pod licenco Apache 2.0 na naslovu `http://ailab.ijs.si/~tadej/slner.zip`

Zahvale

To delo je podprla Javna agencija za raziskovalno dejavnost Republike Slovenije, 7. okvirni program Evropske Komisije s projektom XLike (ICT-288342-STREP).

6. Literatura

K. Balog, P. Serdyukov, in A.P. de Vries. 2010. Overview of the TREC 2010 Entity Track. *NIST Special Publication: TREC*.

R.H. Byrd, J. Nocedal, in R.B. Schnabel. 1994. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1):129–156.

W.W. Cohen in S. Sarawagi. 2004. Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods. V: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, str. 89–98. ACM.

T. Erjavec, D. Fišer, S. Krek, in N. Ledinek. 2010a. Jezikovni viri projekta JOS. *Zbornik Sedme konference Jezikovne tehnologije, 14. do 15. oktober 2010 : zbornik 13. mednarodne multikonference Informacijska družba - IS 2010, zvezek C.*, C:42–46.

Tomaž Erjavec, Darja Fišer, Simon Krek, in Nina Ledinek. 2010b. The JOS linguistically tagged corpus of Slovene. V: *Seventh International Conference on Language Resources and Evaluation, LREC'10*, Paris. ELRA.

Tomaž Erjavec, Simon Krek, Špela Arhar, Darja Fišer, Nina Ledinek, Amanda Saksida, Breda Sivec, in Blaž Trebar. 2010c. Oblikoskladenjske specifikacije JOS. `http://nl.ijs.si/jos/msd/`.

O. Etzioni, M. Cafarella, D. Downey, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, in A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

J.R. Finkel, T. Grenager, in C. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *Ann Arbor*, 100.

R. Grishman in B. Sundheim. 1996. Message understanding conference-6: A brief history. V: *Proceedings of the 16th conference on Computational linguistics-Volume 1*, str. 466–471. Association for Computational Linguistics Morristown, NJ, USA.

J. Lafferty, A. McCallum, in F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. V: *Machine Learning International Workshop*, str. 282–289. Citeseer.

A.K. McCallum. 2002. Mallet: A machine learning for language toolkit.

L. Rabiner in B. Juang. 1986. An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(1):4–16.

J. Rupnik, M. Grčar, in T. Erjavec. 2008. Improving morphosyntactic tagging of Slovene language through metatagging. *Informatica Special Issue: Intelligent Systems Guest Editors: Costin Badica*, str. 437–444.

E. Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia*.

T. Štajner in M. Grobelnik. 2009. Story link detection with entity resolution. V: *Proceedings of Semantic Search Workshop at WWW2009, Madrid, Spain*.

T. Štajner in D. Mladenić. 2009. Entity Resolution in Texts Using Statistical Learning and Ontologies. V: *3rd Asian Semantic Web Conference, Shanghai, China*, str. 91–104. Springer.

T. Štajner in D. Mladenić. 2012. Cross-lingual named entity extraction and disambiguation. *Proceedings of 4th Jožef Stefan International Postgraduate School Students Conference*, str. 176–181.

Statistični Urad Slovenije. 2012. Seznam pogostih in redkih imen. `http://www.stat.si/imena_top_imena_spol.asp?r=True.`, 5.

C. Sutton in A. McCallum. 2004. Collective segmentation and labeling of distant entities in information extraction. Tehnično poročilo, DTIC Document.

T. Štajner, D. Rusu, L. Dali, B. Fortuna, D. Mladenić, in M. Grobelnik. 2010. A Service Oriented Framework for Natural Language Text Enrichment. *Informatica*, str. 307–313.

Wikipedia. 2012a. Ženska osebna imena. `http://sl.wikipedia.org/wiki/Kategorija:%C5%BDenska_osebna_imena`, 5.

Wikipedia. 2012b. Glavna mesta. `http://sl.wikipedia.org/wiki/Kategorija:Glavna_mesta`, 5.

Wikipedia. 2012c. Moška osebna imena. `http://sl.wikipedia.org/wiki/Kategorija:Mo%C5%A1ka_osebna_imena`, 5.

Wikipedia. 2012d. Občine Slovenije. `http://sl.wikipedia.org/wiki/Kategorija:Ob%C4%8Dine_Slovenije`, 5.

Wikipedia. 2012e. Priimki. `http://sl.wikipedia.org/wiki/Kategorija:Priimki`, 5.

Wikipedia. 2012f. Seznam naselij v Sloveniji. `http://sl.wikipedia.org/wiki/Seznam_naselij_v_Sloveniji`, 5.

Wikipedia. 2012g. Seznam suverenih držav. `http://sl.wikipedia.org/wiki/Seznam_suverenih_dr%C5%BEav`, 5.

sloWCrowd: orodje za popravljanje wordneta z izkoriščanjem moči množic

Aleš Tavčar,* Darja Fišer,† Tomaž Erjavec‡

* Odsek za inteligentne sisteme, Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
ales.tavcar@ijs.si
† Oddelek za prevajalstvo, Filozofska fakulteta, Univerza v Ljubljani
Aškerčeva 2, 1000 Ljubljana
darja.fiser@ff.uni-lj.si
‡ Odsek za tehnologije znanja, Institut Jožef Stefan
Jamova cesta 39, 1000 Ljubljana
tomaz.erjavec@ijs.si

Povzetek

V prispevku predstavimo orodje sloWCrowd, ki smo ga razvili za lažje odpravljanje napak v avtomatsko generiranih semantičnih leksikonih tipa wordnet in je zasnovano tako, da nam odgovore za problematične literale omogoča zbirati iz široke množice uporabnikov. Orodje je prosto dostopno in temelji na razširjenih tehnologijah, kot sta PHP in MySQL. Sestavljata ga administratorski in uporabniški vmesnik. V administratorskem vmesniku izdelamo projekt, sledimo poteku projekta in izvažamo rezultate, v uporabniškem vmesniku pa reševalci glasujejo o (ne)pravilnosti naključno izbranih literalov. Rezultati prvega eksperimenta, ki smo ga izvedli z orodjem sloWCrowd, so spodbudni, saj so bili uporabniki orodja z njim zadovoljni, odločitev o dokončnem izbrisu nekega literala na podlagi ujemanja njihovih odgovorov pa enostavna, hitra in zanesljiva. Dodatna prednost razvitega orodja je, da ga je mogoče prilagoditi za najrazličnejše naloge, pri katerih je koristno sodelovanje večjega števila reševalcev.

sloWCrowd: a Crowdsourcing Tool for Cleaning Wordnets

The paper presents a tool called sloWCrowd, developed to facilitate error correction in automatically generated wordnets by crowdsourcing. The developed tool is open-source and based on popular technologies, such as PHP and MySQL. It consists of an administrator and a user interface. The administrator interface enables the creation of crowdsourcing projects, management of on-going projects and export of the results, while the user interface allows users to vote on the (in)correctness of the randomly displayed literals. The results of the first experiment that was performed to test the sloWCrowd tool are encouraging because the users were satisfied with the tool and the final decision on the deletion of a problematic literal based on the users' inter-annotator agreement was simple, fast and reliable. Another advantage of the tool is that it can be adapted to a broad range of other crowdsourcing tasks.

1. Uvod

Z razmahom avtomatskih pristopov za izdelavo jezikovnih virov, ki glede na zahtevnost problema dajejo rezultate različne kakovosti, so se v jezikovnotehnološki skupnosti povečale tudi potrebe po validaciji oz. čiščenju avtomatsko generiranih vsebin. Ker je tovrstno delo zamudno in drago, so raziskovalci kmalu začeli razmišljati o možnostih, ki bi postopek pospešile in pocenile, pri čemer se kvaliteta zbranih oznak ne bi bistveno znižala. Številni poskusi so pokazali, da je nalogo mogoče razdeliti na obvladljive in razumljive dele ter jo ponuditi v reševanje široki množici uporabnikov svetovnega spleta, ki niso nujno strokovnjaki z obravnavanega področja. Kvaliteto je mogoče zagotoviti s preverjanjem zanesljivosti uporabnikov skozi ponavljanje vprašanj različnim uporabnikom in filtriranjem njihovih odgovorov (Adda idr. 2011).

Ena najbolj razširjenih platform za uporabo moči množic (ang. *crowdsourcing*) za pridobivanje večje količine ročno potrjenih podatkov je Mechanical Turk[1] ameriškega podjetja Amazon, ki raziskovalcem ponuja administrativno podporo pri izvajanju eksperimentov, po želji pa tudi rekrutacijo reševalcev. S tovrstnimi platformami so zelo zadovoljni predvsem raziskovalci, ki zbirajo večje količine nejezikovnih podatkov (npr. označevanje slik), ter raziskovalci, ki se ukvarjajo z angleščino, saj imajo ti na spletu na voljo največ kompetentnih reševalcev. Z ustrezno zasnovanimi in izvedenimi nalogami pa so rezultati zelo uporabni tudi za zbiranje anotacij za zahtevnejše jezikoslovne in semantične probleme, kot so določanje afekta, presojanje semantične podobnosti besed, prepoznavanje besedilne vsebovanosti, časovno razvrščanje dogodkov, razdvoumljanje ipd. (Snow idr. 2008).

Za motivacijo povprečnih uporabnikov svetovnega spleta, da se pridružijo eksperimentu in prispevajo čim več odgovorov, so raziskovalci z različnih področij razvili t.i. igre z razlogom (ang. *games with a purpose*), ki od uporabnika na zabaven, a strukturiran način pridobivajo željene podatke. Ena prvih takšnih iger je bila ESP Game[2], v kateri sta uporabnika, ki se med seboj nista poznala, morala opisovati slike in zbirala točke vsakič, ko sta pri tem uporabila iste besede (von Ahn 2006). Med projekti, s katerimi so zbirali oznake za jezikovne podatke, pa je najbolj znana igra Word Detectives[3], s pomočjo katere označujejo anafore v besedilih (Chamberlain idr. 2008).

Povod za ta prispevek je bila potreba po odpravljanju napak iz ročno zgrajenega semantičnega leksikona za slovenščino sloWNet (Fišer 2009). Leksikon je zasnovan na sorodnem angleškem viru Princeton WordNet (Fellbaum 1998) in je bil grajen v več korakih s pomočjo

[1] https://www.mturk.com/mturk/welcome

[2] Igro je pred nekaj leti kupilo podjetje Google in jo vključilo v svoje produkte, zato je na spletu v prvotni obliki ni več mogoče igrati.

[3] http://anawiki.essex.ac.uk/phrasedetectives/

različnih tipov razpoložljivih dvo- in večjezičnih jezikovnih virov, kot so dvojezični slovarji, vzporedni korpusi in Wikipedija. Analiza vsebine je pokazala, da tako izdelan sloWNet vsebuje precej šuma, ki znižuje uporabno vrednost vira in ga je zato treba čim prej odpraviti. Najbolj problematične lekseme (literale), ki skoraj zagotovo ne ubesedujejo pojma (sinseta), ki so mu pripisane, smo identificirali avtomatsko (Sagot in Fišer, 2012). Za to smo uporabili referenčni korpus FidaPLUS[4], iz katerega smo izluščili kontekstualne informacije za literale iz sloWNeta. V skladu z načeli distribucijske semantike smo nato kontekstualne informacije, pridobljene iz korpusa, primerjali z neposredno okolico literala v sloWNetovi semantični mreži. Kandidate z najslabšim rezultatom smo označili kot potencialne napake, ki jih želimo s pomočjo orodja, ki ga predstavljamo v tem prispevku, ponuditi v glasovanje večjemu številu slovenskih uporabnikov interneta in nato po potrebi izbrisati.

V nadaljevanju prispevka predstavimo orodje sloWCrowd, ki smo ga razvili za zbiranje jezikovnih podatkov iz široke množice uporabnikov. Razdelek 3 predstavi eksperiment, v katerem smo orodje prvič preizkusili. Prispevek zaključimo s sklepnimi mislimi in načrti za prihodnje delo.

2. Orodje sloWCrowd

Z razvojem orodja sloWCrowd smo želeli pridobiti preprosto in prilagodljivo orodje, ki bi omogočalo uporabo množic za verifikacijo avtomatsko generiranih sinsetov in bi bilo uporabno tudi za najrazličnejše naloge na področju gradnje jezikovnih virov in razvoja orodij, pri katerih je potrebno zbrati večje število človeških odgovorov.

Za vire, ki jih gradimo z avtomatskimi pristopi, je značilno, da vsebujejo precej napak, vendar so tudi zelo obsežni, zato bi bilo strokovno in celovito ročno odpravljanje napak preveč zamudno in predrago. S prenosom bremena verifikacije na širšo množico se zmanjša čas verifikacije, dobljeni rezultati pa so lahko celo bolj zanesljivi, saj se o posameznem literalu odloča večje število uporabnikov. Pri tovrstnem načinu zbiranja zanesljivih podatkov je orodje, ki na preprost in zanimiv način izbere ter ponudi naloge uporabnikom, ključno.

Orodje sloWCrowd je sestavljeno iz dveh delov, administratorskega in uporabniškega. V administratorskem delu ustvarimo projekt in vodimo zbiranje odgovorov. Uporabniški del omogoča izbiro projekta, pri katerem uporabnik želi sodelovati, predstavitev projekta z navodili za reševanje, reševanje nalog in lestvico najboljših ocenjevalcev glede na število in pravilnost rešenih nalog.

Orodje je prosto dostopno in temelji na popularnem odprtokodnem programskem jeziku za strežniško rabo in razvoj dinamičnih spletnih vsebin PHP in podatkovni bazi MySQL, kar omogoča prenosljivost in enostavno namestitev. sloWCrowd deluje na večini spletnih brskalnikov, saj uporablja splošno razširjene tehnologije.

2.1. Zasnova in implementacija

sloWCrowd je spletna aplikacija, napisana v skriptnem jeziku PHP, kar nam omogoča izdelavo dinamičnih spletnih strani, ki so osnova za široko paleto storitev, ki jih danes najdemo na spletu. Prikaz vsebine omogočajo predloge HTML in CSS, zaradi česar je posamezne komponente orodja enostavno prilagajati trenutnim potrebam.

Podatki spletne aplikacije so shranjeni v odprtokodni podatkovni bazi MySQL. Struktura baze je definirana tako, da omogoča enostavno dodajanje novih projektov. V glavni tabeli so informacije o posameznih projektih, vsak projekt posebej pa ima dodeljeni še dve specifični tabeli. Prva vsebuje uporabniške podatke, druga pa uporabnikove odgovore na naloge.

Zaradi zagotavljanja kvalitete zbranih odgovorov smo sloWCrowd zasnovali tako, da se uporabniki pred reševanjem nalog najprej prijavijo v sistem. Prijava nam omogoča beleženje odgovorov uporabnikov, računanje zanesljivosti odgovorov in uporabnikov ter upravljanje z uporabniki. Registracija je zelo preprosta in poteka preko Googlovega računa. Za dostop do Googlove identifikacijske aplikacije smo uporabili odprtokodno PHP knjižnico HybridAuth[5], ki omogoča dostop do večine današnjih socialnih omrežij. Uporabnik s potrditvijo dostopa do prijavne aplikacije sistemu dovoli identifikacijo s podatki iz Googlovega računa, preko katerega orodje dobi uporabnikovo ime in elektronski naslov. Tak način registracije uporabniku prihrani vpisovanje osebnih podatkov in olajša prijavo v sistem, saj mu ni potrebno vsakič vpisovati uporabniškega imena in gesla.

sloWCrowd ima implementiran mehanizem za ugotavljanje zanesljivosti uporabnikov. Slednje je ključno pri obliki reševanja problemov, kjer sodeluje veliko različnih uporabnikov (od ekspertov do običajnih uporabnikov). Pri vsakem projektu je datoteki z nalogami, ki jih rešujejo uporabniki, mogoče dodati še datoteko, ki vsebuje referenčne naloge, t.j. naloge s predhodno označenimi pravilnimi odgovori. Uporabnik med reševanjem dobiva naloge iz obeh datotek, pri čemer referenčna datoteka služi za ugotavljanje zanesljivosti uporabnika:

$$\text{zanesljivost} = \frac{\text{število pravilnih odgovorov}}{\text{število odgovorov}}$$

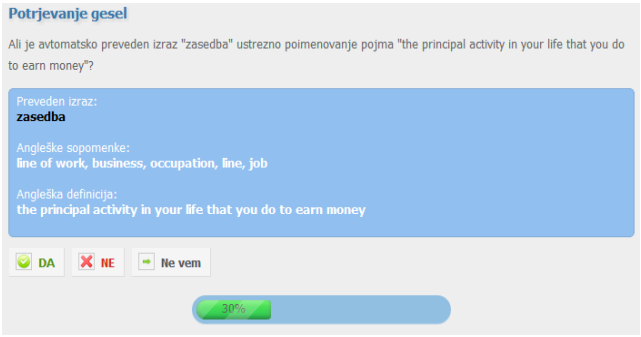
Pravilen odgovor je tisti, ki je enak odgovoru v referenčni datoteki, število odgovorov pa vključuje vse odgovore na naloge iz referenčne datoteke. Mero zanesljivosti uporabljamo za določanje razmerja med nerešenimi in referenčnimi nalogami, ki jih orodje ponudi uporabniku, omogoča pa tudi naknadno izločanje odgovorov zelo nezanesljivih uporabnikov.

[4] http://fidaplus.net/ [15.5.2012]

[5] http://hybridauth.sourceforge.net/index.html [15.5.2012]

2.2. Uporabniški vmesnik

Osnovna funkcionalnost orodja sloWCrowd je potrjevanje in zavračanje literalov. Po prijavi uporabnika se prikaže glavno okno, kjer uporabnik rešuje naloge. Na Sliki 1 je prikazan primer take naloge. Na vrhu zaslona so navodila za reševanje naloge, v konkretnem projektu preverjanje ustreznosti avtomatsko prevedenega slovenskega izraza za izbran koncept. Nato so za lažje odločanje navedene še dodatne informacije, kot na primer angleške sopomenke in angleška defiinicija za isti koncept. Na dnu zaslona so trije gumbi, med katerimi izbira uporabnik, in sicer DA, NE in NE VEM, s katerim vprašanje preskoči.



Slika 1. Potrjevanje in zavračanje literalov

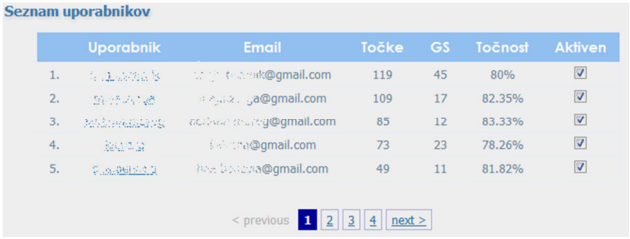
V vsakem sklopu se uporabniku prikaže 10 naključno izbranih vprašanj. Delež rešenih vprašanj v posameznem sklopu je prikazan na dnu zaslona. Orodje uporabniku ponudi določen delež novih in že rešenih nalog, s katerimi se ugotavlja zanesljivost uporabnika. Glede na pravilnost uporabnikovih odgovorov se uporabniku prikaže večji ali manjši delež nalog iz referenčne datoteke. Lestvica je progresivna, saj se z višanjem deleža pravilnih odgovorov iz referenčne množice viša delež novih, še neoznačenih vprašanj.

Poleg ugotavljanja zanesljivosti uporabnikov glede na referenčne naloge orodje beleži tudi stopnjo ujemanja z drugimi uporabniki. Za motivacijo uporabnikov pri reševanju nalog se njihovi odgovori točkujejo, najboljših pet uporabnikov pa je nato prikazanih na lestvici najboljših ocenjevalcev. Uporabnik točke dobi za vsak pravilen odgovor glede na referenčno datoteko in odgovore ostalih uporabnikov v bazi.

2.3. Administratorski vmesnik

V orodje je vključen tudi administratorski vmesnik, ki je namenjen skrbnikom orodja. V njem lahko urejajo aktivne projekte in definirajo nove. Skrbnik doda nov projekt tako, da vnese ime, opis projekta, v sistem naloži referenčno datoteko odgovorov, datoteko še nerešenih odgovorov in izbere eno od besedilnih datotek, v kateri je definirana celotna tekstovna vsebina projekta. V primeru, da skrbnik želi definirati projekt v drugem jeziku ali vzpostaviti projekt z drugimi funkcionalnostmi, le izbere ustrezno besedilno datoteko. Ob potrditvi se v bazi avtomatsko kreirajo tabele, ki jih projekt potrebuje in uporabniki lahko začnejo z reševanjem nalog projekta. Preostale funkcionalnosti so skupne vsem definiranim projektom in služijo pregledu poteka reševanja.

Prva funkcionalnost vmesnika je pregled vseh registriranih uporabnikov, razvrščenih po številu točk, ki so jih dosegli, njihova zanesljivost, na zahtevo pa tudi prikaz posameznih odgovorov. Uporabnike, ki ne dosežejo zadovoljive zanesljivosti, je mogoče onemogočiti z izklopom kljukice v polju Aktiven, s čimer se njegovi odgovori pri prikazu ne upoštevajo. Primer seznama uporabnikov je prikazan na Sliki 2.



Slika 2. Pregled vseh uporabnikov (imena uporabnikov so prekrita zaradi varovanja osebnih podatkov)

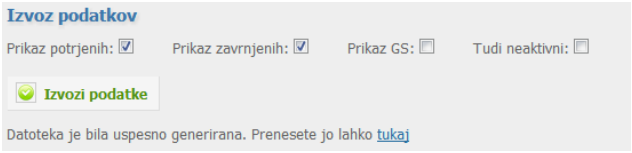
Naslednja funkcionalnost je prikaz seznama vseh rešenih nalog, število odgovorov na posamezno nalogo ter število potrditev in zavrnitev s strani uporabnikov. Na Sliki 3 je prikazan del odgovorov uporabnikov, ki smo jih zbrali med validacijo avtomatsko generiranih prevodov v sloWNetu. Literali, ki imajo v stolpcu GS (gold standard) zeleno kljukico, so iz datoteke z referenčnimi odgovori. Iz števila potrditev in zavrnitev lahko vidimo, da že z razmeroma majhnim številom zbranih odgovorov na posamezno vprašanje skupni rezultat hitro konvergira k pravilni rešitvi. Opazimo lahko namreč, da so v večini primerov uporabniki izbrali enak odgovor. Naj omenimo, da je v izvedenem eksperimentu množica referenčnih odgovorov precej večja od množice neoznačenih nalog, zato je število zbranih odgovorov za posamezni literal iz referenčne množice precej manjše od števila odgovorov na nove literale. Na Sliki 3 tako lahko vidimo, da so iz referenčne datoteke samo trije primeri: en odgovor za literal »simbol«, dva odgovora za literal »slovo« in eden za literal »soba«.

Slika 3. Seznam odgovorov

Orodje omogoča tudi enostavno filtriranje odgovorov, saj lahko skrbnik izbere, ali naj se literali iz referenčne datoteke prikažejo med rezultati ali ne. Pri izbiri posameznega literala se prikaže seznam, ki vsebuje vse odgovore uporabnikov, ki so se odločali o tem literalu. Za vsakega uporabnika se prikaže datum odgovora, ali je literal iz referenčne datoteke in pravilna rešitev. Vsakemu reševalcu je pripisana tudi njegova zanesljivost (glej Slika 4). Poleg tega lahko skrbnik izbira med prikazom uporabnikov, ki so literal zavrnili in uporabnikov, ki so literal potrdili.

Slika 4. Seznam odgovorov.

Zadnja funkcionalnost je izvoz vseh odgovorov v besedilno datoteko (Slika 5). Uporabnik lahko izbira, kateri podatki bodo vključeni v izvoz: samo potrditve ali zavrnitve literalov, ali naj bodo vključeni tudi odgovori na literale iz referenčne datoteke in odgovori uporabnikov, ki zaradi nezanesljivosti ne bodo upoštevani.



Slika 5. Izvoz podatkov

Izvoženi podatki so grupirani po literalih, kar pomeni, da so odgovori vseh uporabnikov za isti literal izpisani skupaj. Izvoženim podatkom je poleg vseh informacij o literalu dodano še uporabniško ime ocenjevalca, datum odgovora in njegova odločitev.

3. Preizkus orodja sloWCrowd

3.1. Opis eksperimenta

Razvito orodje smo preizkusili v manjšem eksperimentu, v katerem smo 10 uporabnikov, večinoma študentov in diplomantov prevajalstva ter tujih jezikov, prosili, da se prijavijo v orodje in rešijo 5 sklopov nalog, od katerih vsak vsebuje po 10 vprašanj. Za reševanje smo jim dali 10 dni časa. Pri tem smo jim dali naslednja navodila za reševanje:

»Naloge rešuješ tako, da prebereš slovensko besedo, angleško definicijo in angleške sopomenke ter se odločiš, ali je slovenska beseda ustrezen prevod za to angleško definicijo in sopomenke. Če se s tem strinjaš, klikneš na gumb DA, če se ne strinjaš, klikneš NE, če pa besede ne razumeš ali nisi prepričan, ali je pravilna ali ne, pa klikneš NE VEM.«

Vprašanja so bila sestavljena iz 100 samostalniških literalov, ki so na podlagi prejšnje raziskave (Sagot in Fišer 2012) najbolj vprašljivi in najverjetneje napačni. Enak delež vprašanj, ki so se uporabnikom naključno prikazala, pa je bil iz referenčne datoteke, ki je vsebovala vnaprej rešene naloge in nam služi za izračun stopnje zanesljivosti posameznih uporabnikov.

3.2. Predstavitev in analiza rezultatov

Pregled in analizo rezultatov začnemo s predstavitvijo sodelujočih pri eksperimentu. V Tabeli 1 so prikazani uporabniki, ki so sodelovali pri eksperimentu. Drugi stolpec v tabeli vsebuje število vseh literalov, o katerih so se uporabniki odločali. Od teh je v tretjem stolpcu prikazano število neoznačenih in v četrtem število označenih literalov. V četrtem stolpcu je prikazana zanesljivost uporabnikov, izračunana na podlagi odgovorov za literale iz referenčne datoteke, v zadnjem stolpcu pa je prikazana točnost odgovorov uporabnika za še neoznačene literale. Slednja je izračunana na podlagi odgovorov vseh uporabnikov (ang. *inter-annotator agreement*), pri čemer privzamemo, da je večinsko mnenje pravilno, odstopanja od njega pa nepravilna (čeprav bi v teoriji lahko tudi večina prispevala napačne odgovore, tovrstnih težav pri tej nalogi ne pričakujemo v omembe vrednem številu primerov). Iz primerjave stolpcev Zanesljivost in Točnost lahko ugotovimo, da so vrednosti obeh podobne in relativno visoke.

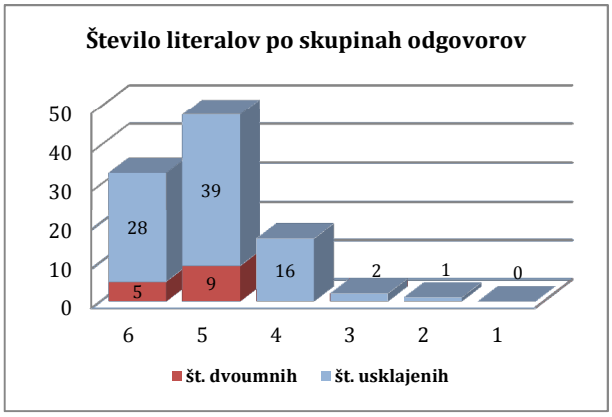
Upor.	Lit.	Neozn.	Ozn.	Zanesljivost	Točnost
U1	122	77	45	80,0 %	90,1 %
U2	110	94	16	82,3 %	91,5 %
U3	90	79	11	83,3 %	89,9 %
U4	78	56	22	78,3 %	87,5 %
U5	48	38	10	81,8 %	97,4 %
U6	49	35	14	92,8 %	88,6 %
U7	48	36	12	91,7 %	86,1 %
U8	48	36	12	91,7 %	91,7 %
U9	42	30	12	92,3 %	96,7 %
U10	40	29	11	81,8 %	75,9 %

Tabela 1. Pregled rezultatov, zbranih v eksperimentu

Iz tabele lahko opazimo, da so štirje uporabniki rešili veliko več nalog, kot je bilo od njih zahtevano, ostali pa nekaj nalog manj od zahtevanih 50. Ti uporabniki so verjetno nekajkrat izbrali možnost »Ne vem«, teh odgovorov pa v sistemu zaenkrat ne beležimo. Analiza odgovorov kaže, da so uporabniki so na splošno dosegli visoko zanesljivost v primerjavi z referenčno datoteko. Devet uporabnikov je doseglo zanesljivost, večjo od 80 %, štirje uporabniki pa celo zanesljivost nad 90 %. Povprečna dosežena zanesljivost pa znaša 85,6 %, kar je za naloge s področja leksikalne semantike zelo visok rezultat. Opazimo lahko tudi, da so uporabniki dosegli visoko stopnjo medsebojnega ujemanja. Tudi v tem primeru je le en uporabnik dosegel točnost, manjšo od 80 %, medtem ko povprečna točnost znaša 89,5 %. Na splošno pa so uporabniki dosegli večjo stopnjo točnosti kot zanesljivosti, kar pomeni, da so se bolj strinjali med seboj kot z referenčno množico. Opazimo pa trend, da se pri večini

uporabnikov z večanjem stopnje zanesljivosti veča tudi stopnja točnosti in obratno.

Pri analizi odgovorov nas je zanimalo tudi, kolikokrat so se literali med eksperimentom ponavljali. Na Sliki 6 je prikazana razporeditev literalov po pogostosti pojavitve. Opazimo lahko, da se je največ literalov (49) pojavilo petkrat, 33 literalov se je ponovilo šestkrat, 16 literalov se je ponovilo štirikrat, dva literala trikrat in en literal dvakrat. Na istem grafu je prikazano tudi število literalov, pri katerih so se uporabniki večinsko strinjali o pravilnosti pomena (zgornji del) in število dvoumnih literalov, za katere smo zbrali enako število potrditev in zavrnitev (spodnji del). V primeru petih odgovorov se prišteje literal med dvoumne tudi, če so trije uporabniki glasovali na en način, dva pa na drug.

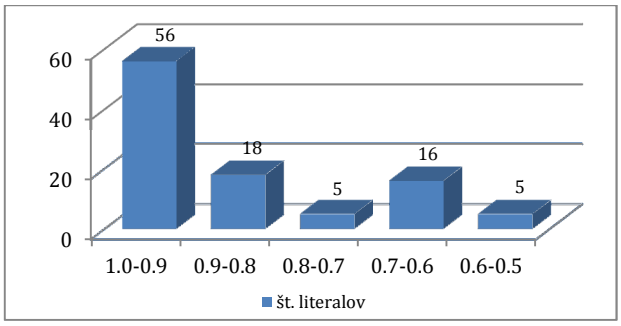


Slika 6. Število literalov po pogostosti pojavitve

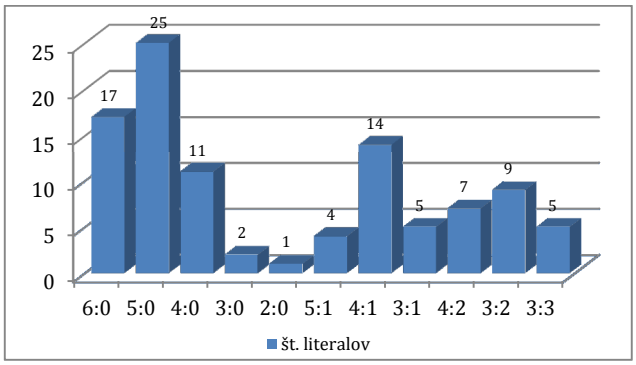
Opazimo lahko, da so se uporabniki v večini primerov strinjali o (ne)pravilnosti prevoda literala. Iz dobljenih rezultatov lahko sklepamo, da že z manjšim številom odgovorov dobimo zanesljivo oceno o nekem literalu, saj so že pri štirih odgovorih uporabniki dosegli konsenz glede pomena literala. Tega sicer zgolj na podlagi izvedenega eksperimenta ne moremo z gotovostjo potrditi, saj je število uporabljenih literalov premajhno, zato nameravamo v prihodnje minimalno število potrebnih odgovorov preveriti v večjem eksperimentu.

Pri nadaljnji analizi smo opazovali število literalov, ki so jih uporabniki potrdili ali zavrnili z določeno gotovostjo. Na Sliki 7 vidimo razporeditev literalov v skupine verjetnosti. V prvi so vsi literali, ki so bili s strani uporabnikov soglasno potrjeni ali zavrnjeni. V ostalih stolpcih pa so razporejeni literali s padajočim razmerjem med potrditvijo in zavrnitvijo. Opazimo lahko, da so bili uporabniki v večini primerov soglasni pri odločitvah o literalih, iz česar lahko sklepamo, da lahko z uporabo orodja sloWCrowd v večini primerov konvergiramo k splošno sprejemljivi pravilni rešitvi.

Na Sliki 8 je podrobneje prikazana razporeditev literalov po razmerjih odgovorov. Vsebinsko je podobna prejšnji sliki, le da natančneje prikaže razporeditev literalov po skupinah razmerij. Razmerje *x:y* predstavlja število potrditev (zavrnitev) *x* in število zavrnitev (potrditev) *y* za nek literal. Zadnja dva stolpca v grafu vsebujeta dvoumne literale, kjer je število potrditev in zavrnitev približno enako.



Slika 7. Število literalov po združenih deležih odgovorov



Slika 8. Število literalov po razmerju odgovorov

V nadaljevanju smo pregledali vse dvoumne literale. V Tabeli 2 so izpisani vsi literali, kjer je enako število potrditev in zavrnitev (3:3). V prvem stolpcu je prevod, nato sopomenke v angleščini in še angleška definicija.

Literal	Sopomenke	Definicija
hip	piece, while, spell, patch	a period of indeterminate length (usually short) marked by some action or condition
položaj	place, position	the particular portion of space occupied by something
prostor	room	opportunity for
razglas	rescript, fiat, order, edict, decree	a legally binding command or decision entered on the court record (as if issued by a court or judge)
vprašanje	topic, matter, subject, issue	some situation or event that is thought about

Tabela 2. Literali, za katere se uporabniki niso strinjali, ali so pravilni ali napačni.

Največ problemov predstavljajo literali, za katere je angleška razlaga precej ohlapna in zato nejasna. Naslednjo skupino problematičnih literalov predstavljajo angleške ustaljene fraze, ki se v slovenščini uporabljajo v drugem kontekstu ali z drugimi besedami. Med odgovori smo naleteli tudi na napačne odločitve. Dva primera, ki sta v resnici napačna, uporabniki pa so izglasovali, da sta pravilna, sta literala »del« (ang. *member*) v pomenu »*anything that belongs to a set or class*« in »člen« (ang. *division, part, section*) v pomenu »*one of the portions into which something is regarded as divided and which together constitutes a whole*«. Vendar je teh primerov zanemarljivo malo, pa še te glasovi drugih uporabnikov slej ko prej preglasujejo in s tem izničijo negativni vpliv napake na končne rezultate.

3.3. Diskusija in možnosti za izboljšave

Z eksperimentom, v katerem smo preizkusili razvito orodje sloWCrowd, smo ugotovili, da je njegova glavna prednost, da uporabnikom na preprost in zanimiv način ponudi v reševanje različne naloge. Kot je znano za večino iger z razlogom, se je tudi tu pokazalo, da vpeljava točkovanja uporabnike pritegne k pravilnemu reševanju večjega števila nalog, saj je kar nekaj tekmovalcev rešilo precej več nalog, kot je bilo od njih zahtevano. Čeprav na podlagi opravljenega eksperimenta težko zanesljivo določimo spodnjo mejo zahtevanih odgovorov, ki so potrebni za zanesljivo oceno literala, menimo, da 10 uporabnikov ob nizki obremenitvi in v krajšem času lahko validira 200–300 literalov, kar pomeni, da bi za ocenjevanje celotnega sloWNeta potrebovali 300–400 uporabnikov.

Da bi lahko orodje v prihodnje še izboljšali in nadgradili, smo prostovoljce, ki so v eksperimentu reševali naloge, dodatno prosili še, da nam posredujejo odgovore na naslednja vprašanja (možni odgovori NITI NAJMANJ, NE PREVEČ, PRECEJ, ZELO):

- Ali se ti je zdelo delo z orodjem sloWCrowd enostavno?
- Ali je zdelo delo z orodjem sloWCrowd zanimivo?
- Ali so se ti zdele naloge razumljive?
- Ali je bila dolžina posamezne naloge primerna?
- Ali imaš v zvezi z orodjem oz. eksperimentom kakšno pripombo, ki nam bo pomagala orodje še izboljšati?

Večina uporabnikov je menila, da je delo z orodjem sloWcrowd PRECEJ enostavno in zanimivo, nekaj uporabnikov je celo ocenilo enostavnost in zanimivost orodja z ZELO. Podobne ocene so uporabniki dodelili razumljivosti podanih nalog, s tem, da je en uporabnik podal oceno NE PREVEČ, z obrazložitvijo, da je pri veliko nalogah odgovor odvisen od konteksta, v katerem se literal uporablja, predvsem v primeru nejasnih definicij. Po njegovem mnenju zgolj prikaz angleških sopomenk in definicij v teh primerih ne zadošča, zato predlaga, da omogočimo še prikaz drugih semantičnih relacij, ki izhajajo iz ocenjevanega pojma, kar bomo v naslednji različici orodja tudi upoštevali. Podobne pomisleke so imeli tudi nekateri drugi uporabniki, saj je na primer eden od njih predlagal, da uvedemo še dodaten gumb »odvisno od konteksta«. Taka rešitev bi verjetno zmanjšala število zaželenih odločitev (DA, NE), saj bi se uporabniki v primeru dvoma odločali zanjo. Vendar bi s tem po našem mnenju po nepotrebnem dodatno zapletli nalogo in s tem zmanjšali uporabnost rezultatov. Menimo, da bomo tovrstne težave ustrezno odpravili že z zgornjim ukrepom. Bi pa bilo možno, da bi literale, za katere večje število uporabnikov izbere odgovor »NE VEM«, pregledal ekspert. Naslednja želja, ki jo je izrazil eden od uporabnikov, je povratna informacija, ali je bil izbrani odgovor pravilen. To bi v naslednji različici orodja lahko upoštevali tako, da bi uporabniku ob vsakem odgovoru sporočili, ali si je z odgovorom prislužil dodatno točko ali ne ter prikazali trenutno razmerje med zbranimi odgovori. Zadnja pripomba se nanaša na možnost predlaganja boljših prevodov, kar je pravzaprav funkcionalnost že razvitega orodja sloWTool (Fišer in Novak 2011), ki je namenjeno celovitemu urejanju wordneta. Glavni cilj orodja sloWCrowd je predvsem minimalistično in enostavno okolje za čim hitrejše pridobivanje odgovorov na ozko specializirana vprašanja.

4. Zaključek

V prispevku smo predstavili orodje sloWCrowd, ki je namenjeno ročni validaciji avtomatsko pridobljenih jezikovnih podatkov. Administrativni del vmesnika omogoča preprosto izdelavo projekta in uvoz podatkov, uporabniški vmesnik pa je zasnovan tako, da lahko uporabnik naloge rešuje čim hitreje in enostavneje. V eksperimentu, s katerim smo orodje testirali, smo uporabnike prosili, da pregledajo 100 najbolj vprašljivih avtomatsko generiranih literalov iz sloWNeta in označijo, ali so pravilni ali napačni. Po zaključenem preizkusu in analizi njihovih odgovorov ugotavljamo, da je orodje tako z administrativnega kot z uporabniškega vidika enostavno za uporabo, zbrani odgovori pa zanesljivi, saj je stopnja ujemanja med uporabniki visoka in število dvoumnih rešitev majhno. S tem tako nismo dobili samo orodja, s katerim je mogoče popravljati wordnet, temveč tudi platformo za evalvacijo uspešnosti različnih pristopov in avtomatskih metod za avtomatizirano luščenje leksikalno-semantičnih informacij iz strukturiranih in nestrukturiranih jezikovnih virov.

Glede na komentarje prostovoljcev, ki so pri preizkusu sodelovali, bomo v prihodnje še nekoliko izboljšali točkovanje uporabnikov, na večjem eksperimentu pa skušali ugotoviti optimalno število potrebnih odgovorov na vsako zastavljeno vprašanje za zagotavljanje zanesljivih rezultatov po eni stran in čim večje količine validiranih primerov po drugi. Projekti, ki trenutno tečejo v orodju sloWCrowd, so dostopni na naslovu http://nl.ijs.si/slowcrowd/select_project.php. Ker je orodje prosto dostopno pod licenco Creative Commons, pa ga je mogoče tudi prenesti, namestiti na lastni strežnik in prilagoditi svojim potrebam. Za namestitev sta potrebna le PHP in MySQL. Namestitvene datoteke so na: http://nl.ijs.si/slowcrowd/sloWCrowd.rar.

Literatura

G. Adda, B. Sagot, K. Fort, J. Mariani. 2011. Crowdsourcing for Language Resource Development: Critical Analysis of Amazon Mechanical Turk Overpowering Uses. *Zbornik konference LTC*, Poznań, Poljska.

L. von Ahn. Games with a Purpose. 2006. *Computer*, 39/6, str. 92-94.

J. Chamberlain, M. Poesio, U. Kruschwitz. 2008. Phrase Detectives: A Web-based collaborative annotation game. *Zbornik konference iSemantics*, Gradec, Avstrija.

D. Fišer, J. Novak. 2011. Visualizing sloWNet. *Zbornik konference eLEX*. Bled, Slovenija.

D. Fišer. 2009. Pristopi za avtomatizirano gradnjo semantičnih zbirk. *Terminologija in sodobna terminografija*. Ljubljana: Založba ZRC, ZRC SAZU, str. 357-370.

B. Sagot, D. Fišer. 2012. Cleaning noisy wordnets. *Zbornik konference LREC*, Istanbul, Turčija.

R. Snow, B. O'Connor, D. Jurafsky, A. Y. Ng. 2008. Cheap and Fast - But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. *Zbornik konference EMNLP*, str. 254-263.

Korpus slovenskega znakovnega jezika

Špela Vintar, Boštjan Jerko, Marjetka Kulovec

Univerza v Ljubljani, Oddelek za prevajalstvo
Aškerčeva 2, SI - 1000 Ljubljana
E-mail: {spela.vintar, bostjan.jerko, marjetka.kulovec}@ff.uni-lj.si

Povzetek

Prispevek opisuje projekt gradnje prvega korpusa slovenskega znakovnega jezika (SZJ). Predstavimo postopke zbiranja gradiva, predvsem vzpostavljanja stika z gluho skupnostjo ter snemanja gluhih oseb, ter postavljanja metodološkega okvirja za transkripcijo in označevanje. V nadaljevanju predstavimo posebnosti slovenske situacije in spregovorimo tudi o težavah, na katere smo naleteli med zbiranjem in označevanjem gradiva. V času pisanja prispevka je zbiranje gradiva končano, posnetih je prek 70 uporabnikov SZJ, v teku pa je transkribiranje s programom iLex. Pomemben cilj projekta je jezikoslovni opis SZJ, predvsem nabora kretenj z njihovimi variacijami ter slovnične strukture SZJ.

The Slovene Sign Language Corpus

The paper presents the project of compiling the first corpus of the Slovene Sign Language (SSL). We describe the procedures of data collection, the decisions regarding informant selection and the guidelines for transcription and annotation. We outline the particularities of the Slovene situation, especially the high variability of the language, issues concerning language competence and the attitutes of the deaf community towards such data collection. At the time of writing, the data collection stage is finished with over 70 recorded persons, and trancriptions with iLex are underway. The aim of the project is to use the corpus for explorations into the grammatical properties of SSL.

Keywords: Slovene Sign Languge, corpus compilation, sign language transcription

1. Uvod

Slovenski znakovni jezik (SZJ) je sredstvo za vizuelno sporazumevanje gluhih in je eden od uradnih jezikov Republike Slovenije. Vse dosedanje jezikoslovne raziskave SZJ - ki jih ni bilo veliko - so temeljile na podatkih, pridobljenih na zelo omejenem vzorcu oseb, jezikovni priročniki za SZJ pa se močno naslanjajo na slovenski knjižni jezik. Namen projekta SIGNOR (http://lojze.lugos.si/signor) je ustvariti reprezentativen korpus SZJ, ki bo omogočal prvi pravi vpogled v njegove značilnosti in bo služil kot podlaga za jezikoslovne študije na leksikalni, skladenjski in drugih ravneh.

Sorodne korpuse poznajo marsikje na svetu, med bolj znanimi so korpus nemškega znakovnega jezika DGS (Konrad et al. 2003; Konig et al. 2008), korpus avstralskega znakovnega jezika (Johnston et al. 2006), zanimive korpusno podprte raziskave pa se pričenjajo tudi v sosednjih državah, denimo v Avstriji (Krammer et al. 2001, Dotter 2011).

Izzivi, s katerimi se soočamo pri računalniški obdelavi znakovnega jezika, so na eni strani povezani z njegovo vizuelno naravo, saj ga ni mogoče niti slišati niti zapisati z običajno pisavo, na drugi pa s številnimi vprašanji v zvezi z interpretacijo in kategorizacijo posameznih kretenj in pomenov. O nekaterih od teh vprašanj spregovorimo tudi v nadaljevanju tega prispevka.

2. Slovenski znakovni jezik nekoč in danes

Gluha skupnost v Sloveniji šteje med 700 in 1600 oseb. Natančno število gluhih je težko določiti, saj nekateri ne uporabljajo nacionalnega sistema vaučerjev, s katerimi je gluhim sicer zagotovljena pravica do tolmačenja, drugi so morda oglušeli kasneje v življenju in jih uradne statistike ne beležijo, tretji pa se morda ne želijo identificirati z gluho skupnostjo in torej raje ostajajo "nevidni".

Podobno kot drugod po svetu tudi pri nas velja, da biti gluh ne pomeni nujno biti uporabnik znakovnega jezika, niti obratno. Mnogi gluhi se sporazumevajo z govorom in s pomočjo branja ustnic, mnogi slišeči pa uporabljajo znakovni jezik za sporazumevanje z gluhimi svojci ali iz drugih razlogov.

Sistematični razvoj SZJ sega v sedemdeseta leta prejšnjega stoletja, ko je bila Slovenija še del Jugoslavije in se prvič pojavijo tečaji in seminarji predvsem pod okriljem Zveze društev gluhih in naglušnih Slovenije (ZDGNS), na katerih se poučuje SZJ. Sredi osemdesetih se pojavijo tudi prve oblike izobraževanja za tolmače, leta 1986 je bil prvič organiziran tolmaški preizkus, ki ga je uspešno opravilo 16 tolmačev.

Javna zavest o kretnji kot jeziku gluhih se je pričela krepiti po letu 1980, ko se na TV Koper pojavi prva oddaja za gluhe. Nekaj let zatem je tudi RTV Slovenija začela opremljati posamezne oddaje s prevodom v SZJ ali v slovenščino v kretnjah. Vseeno pa se odnos do znakovnega jezika v izobraževanju še dolgo ni spremenil. Tradicionalno prepričanje, da je za gluhe otroke najboljša "inkluzija", se pravi vključitev v običajne š ole ter spodbujanje govora in odčitavanja, namreč marsikje š e vedno prevladuje.

Pomemben mejnik v razvoju SZJ je bilo sprejetje Zakona o uporabi slovenskega znakovnega jezika leta 2002, s katerim se je med drugim uzakonila pravica do uporabe SZJ v vseh javnih in zasebnih življenjskih situacijah, udejanjanje te pravice pa naj bi omogočal sistem vaučerjev za ure tolmačenja. A kljub temu pomembnemu mejniku je stanje vse prej kot zavidljivo, še posebej kar se tiče enakih možnosti za izobraževanje.

Znakovni jezik se poučuje le na eni šoli v Sloveniji, in sicer na ljubljanskem Zavodu za gluhe, pri odločanju za študij pa so gluhi postavljeni pred neprijetno dejstvo, da sto ur brezplačnega tolmačenja letno nikakor ne more zadostiti običajnim študijskim zahtevam. Kot posledica je gluha populacija v povprečju precej nižje izobražena, gluhih študentov pa je v Sloveniji trenutno le okrog dvajset.

3. Zbiranje gradiva

Cilj projekta je zgraditi reprezentativen korpus SZJ, ki zajema jezikovne vzorce okrog 10 odstotkov uporabnikov SZJ ter uravnoteženo predstavlja slovensko skupnost gluhih glede na regionalno zastopanost, spol in starost. V projekt smo tako zajeli vseh 13 lokalnih društev gluhih in naglušnih.

Še pred začetkom snemanja je bilo treba sprejeti nekaj pomembnih odločitev, predvsem v zvezi s strukturo posamezne snemalne seanse ter načini elicitacije komunikacije. Ker želimo, da vzorci predstavljajo čim bolj spontano jezikovno rabo, sta snemanja na terenu izvajali gluhi študentki, snemanje gluhih dijakov v Zavodu za gluhe pa je izvedla Marjetka Kulovec, gluha profesorica znakovnega jezika ter sodelavka pričujočega projekta.

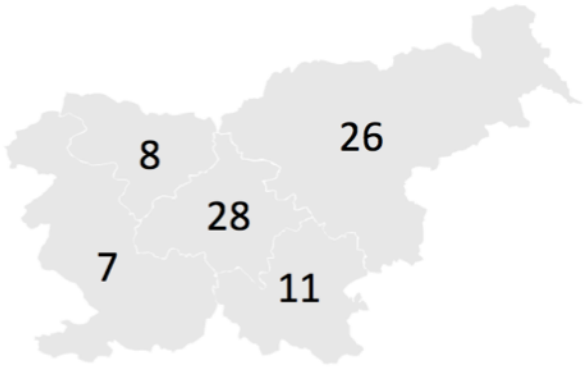
Stik z gluho skupnostjo je omogočila Zveza Društev gluhih in naglušnih Slovenije, ki je projekt že v samem začetku podprla ter nas povabila na eno od rednih srečanj predsednikov društev. Tam smo predstavili raziskovalna izhodišča in cilje projekta ter društva prosili za sodelovanje pri zbiranju gradiva, o projektu pa smo posneli tudi kratek televizijski prispevek, ki je bil objavljen na spletni televiziji ZDGNS. Pri predstavitvi projekta smo poudarili, da namen raziskave ni predpisovanje kretenj ali načina rabe znakovnega jezika, temveč želimo na podlagi zbranega gradiva opisati besedišče in skladnjo SZJ. Kljub temu so bili začetni odzivi gluhe skupnosti zelo previdni, še posebej ker je raziskovalna pobuda prišla z akademske institucije "z druge strani".

Po predstavitvenem dogodku smo društvom po e-pošti poslali še informativni letak, nato pa je dogovarjanje o snemalnih terminih ter snemanih osebah potekalo izključno med našo mobilno snemalno enoto[1] ter društvi. Večinoma so snemanja potekala v prostorih lokalnih društev, v nekaj primerih pa sta se snemalki podali tudi k intervjuvancu na dom. Snemanje dijakov Zavoda za gluhe je potekalo v šolskih prostorih, pri čemer je bilo v ta namen pridobljeno dovoljenje ravnateljice ter pisna dovoljenja vseh staršev dijakov.

Vsaka snemalna seansa s posameznim informantom je bila sestavljena iz treh delov:

- spontano kretanje **o sebi**; spraševalka prosi informanta, da se predstavi, spregovori o sebi, družini in poklicu, o svoji gluhosti, šolanju, itd.,
- kretanje po video predlogi s **splošno** vsebino; spraševalka informantu pokaže kratek video posnetek, ki prikazuje situacijo oz. dogajanje iz vsakdanjega življenja, nato prosi za povzetek in/ali komentar dogajanja v SZJ;
- kretanje po video predlogi s **specializirano** vsebino
  ali
  prosto kretanje na specializirano temo po izboru informanta.

Pri izboru elicitacijskih video posnetkov smo se naslonili na področja, ki jih definira slovar SZJ, ter skušali z video posnetki zajeti vsa področja človekovega življenja in delovanja. Ti posnetki vključujejo izseke iz filmov, nanizank in risank, pri čemer smo izbirali prizore, ki vključujejo čim manj govora. Da bi poleg splošnega besedišča zajeli tudi čimveč specializiranih pojmov, smo se v tretjem delu snemalne seanse usmerili v strokovna področja, ki jih prikazujejo denimo dokumentarni filmi. Če je informant že v prvem delu omenil, da se ukvarja s posebnim konjičkom ali strokovnim področjem, v tretjem delu nismo uporabili elicitacijskega posnetka, ampak je spraševalka informanta pozvala, naj pripoveduje o svojem priljubljenem področju.



Slika 1: Porazdelitev informantov po regijah

V času pisanja imamo na voljo posnetke 80 oseb, od katerih morda ne bodo prav vsi uporabni, ker se nekatere snemane osebe niso uspele dovolj sprostiti, da bi komunicirale spontano, ali pa njihova jezikovna kompetenca v SZJ ni dovolj visoka. Kar se tiče slednje, v okviru projekta namenoma nismo želeli postavljati strogih definicij v zvezi s stopnjo gluhosti ali načinom usvajanja SZJ. V Sloveniji se znakovni jezik poučuje šele zadnji dve desetletji, in še to zgolj na eni šoli v Sloveniji. Starejša generacija tako ni bila deležna nikakršnega spodbujanja pri usvajanju znakovnega jezika, celo nasprotno - dolga leta so v šolah vztrajali pri branju z ustnic in učenju govora. Tako smo se odločili za pragmatično usmeritev: primeren informant je vsak, ki SZJ redno uporablja za komuniciranje.

Posnetki so pridobljeni s pisnim soglasjem informanta, da dovoljuje uporabo gradiva v raziskovalne namene. Ob posnetkih smo o vsakem informantu zbrali še naslednje metapodatke:

- način nastanka, stopnja in trajanje gluhosti,
- starost in spol,
- primarna roka,
- izobrazba,
- kraj in regija rojstva,
- kraj in regija šolanja.

[1] Snemanja sta izvajali Maja Kuzma in Mateja Kanalec, ki sta se v vlogah spraševalke in snemalke odlično dopolnjevali.

Posnetki se v anonimizirani obliki hranijo na projektnem strežniku, kjer je poskrbljeno tudi za varnostne kopije.

4. Transkripcija in označevanje

Za transkripcijo in označevanje posnetkov smo izbrali program iLex (Hanke in Stolz 2008), zmogljivo in prilagodljivo orodje za večslojno označevanje video posnetkov. Prednost orodja iLex je predvsem, da se pri lematizaciji uporablja baza pomenskih oznak, zato se ne more zgoditi, da bi se oznake kretenj po nepotrebnem podvajale ali variirale. Ker z označevanjem znakovnega jezika pri nas še nihče nima posebnih izkušenj, se je bilo potrebno pred začetkom označevanja odločiti, katere informacije potrebujemo za jezikoslovno analizo in v katerih ravneh se bodo vnašale v korpus, ne da bi se ob tem že vsiljevala določena interpretacija podatkov.

4.1. Označevalne ravni

Pri zasnovi označevalne sheme smo se opirali na izkušnje projekta DGS (Rainer in dr. 2011), saj je večino opažanj, ki jih je hamburška skupina zabeležila za nemški znakovni jezik, moč prenesti na analizo slovenskega znakovnega jezika.

Označevalna shema SIGNOR tako zajema naslednje ravni označevanja:

- **Segmentacija** oziroma **tokenizacija**. Tu se tok pripovedi v kretnjah razdeli na posamezne kretnje, ki se v programu iLex zabeležijo kot časovni intervali v določenem posnetku. Pri tokenizaciji se srečujemo z dvema dilemama, in sicer prvič z vprašanjem, kako segmentirati sestavljene kretnje, in drugič z obravnavanjem prehodov med kretnjami. V našem primeru smo se odločili za razčlenjeno transkripcijo sestavljenih kretenj ter za izločanje oziroma rezanje prehodov kot delov kretnje.
- **Pripisovanje pomenskih oznak** (glos) oziroma **lematizacija**. Vsaki kretnji v SZJ je dodeljena edinstvena pomenska oznaka (npr. MAMA1), ki ima točno določeno obliko in ki se navezuje na točno določen pojem. Seznama pomenskih oznak (glos) ter pomenov (pojmov) sta v iLexu urejena kot medsebojno povezani bazi podatkov. Da ne bi prihajalo do zamenjav z besedami slovenskega jezika, se pomenske oznake zapisujejo s samimi veliki črkami.
- **Oblika ust**. Pogosto je razumevanje določene kretnje tesno povezano z obliko ust, ki lahko posnema določeno besedo ali njen začetek ali na drug način usmeri sogovornika proti pravemu pomenu.
- **Transkripcija** kretnje v zapis **HamNoSys** (Schmaling and Hanke 2001). HamNoSys je poseben način zapisovanja kretenj, pri katerem s posebnimi znaki zapišemo obliko, položaj in gibanje rok. Načeloma je zapis HamNoSys prirejen vsaki posamezni lemi oziroma glosu in je v tem smislu edinstven.
- **Pomen**. Vsaka kretnja kot določeni leksem ima enega ali več ustaljenih (slovarskih) pomenov, ki so vnešeni v pomensko bazo. Bazo pomenov smo prevzeli iz slovenskega semantičnega leksikona sloWNet. Ker je pomen v določenem kontekstu nedvoumen, se kretnji pripiše nameravani pomen.
- **Sestavljeni pomen**. Na vseh prejšnjih ravneh se sestavljene kretnje označujejo po posameznih delih. Šele na tej ravni označimo pomen, ki nastane iz kombinacije več kretenj; npr. DELATI1 + ŽENSKA1 = delavka
- **Besednovrstno označevanje**. Čeprav je ta faza v načrtu predvsem zaradi lažjih raziskav skladenjskih značilnosti SZJ, je kategorizacija kretenj v tradicionalne besedne vrste izjemno problematična. Ker bi bilo zatorej pripisovanje besednih vrst pomenskim oznakam preveč kompleksno, bodo besedne vrste pripisane pomenom.
- P**revod v slovenščino**. Zadnja raven označevanja bo prevod v slovenščino, ki bo pripisan na ravni posameznih izjav (angl. *utterance*), podobno kot bi tolmačili govorjeni jezik.

4.2. Dileme pri označevanju

Podobno kot v drugih znakovnih jezikih je tudi v slovenskem mnogo variacij, polisemnih kretenj, sinonimnih kretenj in drugih leksikalnih pojavov, ki jih je v bazi pomenskih oznak in v bazi pomenov treba ustrezno predstaviti. Če se pri tem znova opremo na hamburške kolege (Langer in dr. 2009), govorimo o varianti kretnje takrat, kadar sta si dve kretnji z istim pomenom po obliki zelo podobni, kot denimo X1a in X1b. O polisemiji govorimo, kadar se ista kretnja z isto obliko in pomensko oznako uporablja za označevanje dveh ločenih pomenov, npr. NAGLUŠEN1 - naglušen in NAGLUŠEN1 - ponedeljek. O sinonimiji pa govorimo takrat, kadar za izražanje določenega pomena obstaja več kretenj, ki so morda vezane na določeni regionalni dialekt SZJ ali na drugo podskupino uporabnikov SZJ, npr. ZDRAVNIK1, ZDRAVNIK2 in ZDRAVNIK3.

Kadar se oblikovno ista ali zelo sorodna kretnja uporablja na različne načine v odvisnosti od vloge, ki jo igra v stavku, govorimo o modifikaciji. Tako je denimo kretnjo UČITI1 mogoče uporabiti v oblikah UČITI1a - učiti nekoga, UČITI1b - učiti se, UČITI1c - biti poučevan od, kar ustreza različnim oblikam pregibanja v govorjenih jezikih.

Posebno področje v znakovnem jeziku predstavljata uporaba prostora na eni strani in ikonicitete na drugi strani. Z obema sredstvoma si znakovni jezik močno razširi izrazno polje in obenem na ta način poskrbi za izjemno gospodarnost - mnogih propozicij v govorjenem jeziku na tako jedrnat način sploh ni moč izraziti. Po drugi strani znakovnemu jeziku kronično primanjkuje specializiranega besedišča, iz tega razloga morajo biti uporabniki SZJ pri mnogih temah kreativni pri uporabi obstoječih kretenj z novimi in prenešenimi pomeni.

Čeprav smo doslej označili šele majhen del posnetega gradiva, se že izpostavljajo številne konkretne težave:

- Težave pri ločevanju dveh kretenj, ko sta kretnji prikazani na hitro in hkrati, nista dovolj jasno prikazani in ostro ločeni. V teh primerih enemu časovnemu segmentu priredimo dve lemi, na ravni pomena pa zapišemo nameravani pomen, če ga je označevalec uspel razbrati.
- Težave pri zapisu kretenj, ki se nanašajo na konkretne osebe. Gluhi številnim osebam v svoji skupnosti priredijo edinstveno kretnjo, ki označuje točno določeno osebo (npr. Marjetka Kulovec). Edina

pravilna lema za to kretnjo je torej MARJETKA_KULOVEC1, po drugi strani pa bi želeli na ravni lematizacije ohraniti informacijo o lastnoimenski specifiki teh kretenj.

- Težave pri transkribiranju dvoumnih kretenj, kjer do dvoumnosti pride tudi zaradi krajšanja sestavljenih kretenj; npr. ŠTUDIRATI1 – prikazana je ena kretnja z artikulacijo študirati, za pomen fakulteta pa se sicer uporablja sestavljena kretnja ŠTUDIRATI1 + STAVBA1, vendar je prikazana le kretnja ŠTUDIRATI1 skupaj z artikulacijo fakulteta.
- Težave pri razumevanju znakovnih terminov. Na poklicnih področjih se pojavljajo specifične kretnje, ki niso splošno znane in je za njihovo označevanje potrebno znova kontaktirati informanta; tak primer se je pojavil denimo pri mizarskih orodjih, materialih in postopkih obdelave.

4.3. Slovenski znakovni jezik in slovenščina v kretnji

Razmerje med slovenskim znakovnim jezikom kot naravnim sporazumevalnim jezikom gluhe skupnosti in slovenščino v kretnji, ki nastane kot neposredni prenos slovenske stavčne strukture v kretnje, je že v osnovi problematično. Večina gluhih se strinja, da je slovenščina v kretnji umetno ustvarjen jezik, ki se v spontani komunikaciji skoraj nikoli ne uporablja, po drugi strani pa je njen vpliv nesorazmerno velik, saj so ji gluhi precej izpostavljeni prek televizijskih oddaj na nacionalki. Tako se pod vplivom "kretane slovenščine" v znakovnem jeziku pogosteje pojavljajo določeni elementi, ki sicer za znakovni jezik niso značilni; npr. kretnja za "in", pomožni glagoli, zaimki. Posebej zanimivo vprašanje je tudi, kako način tolmačenja, se pravi bodisi tolmačenje iz slovenščine v pravi SZJ bodisi tolmačenje v slovenščino v kretnji, vpliva na razumevanje besedila.

S pomočjo našega korpusa bo sicer mogoče raziskati nekatere vidike odnosa med SZJ in slovenščino v kretnji, ne pa tudi tolmačenja in morebitnih tolmaških norm, ki skozi svojo ideologijo lahko prav tako vplivajo na razvoj znakovnega jezika.

Z večino omenjenih težav so se spoprijemali že raziskovalci drugih znakovnih jezikov, zato se pri njihovem reševanju opiramo na izkušnje tistih jezikov, ki so geografsko blizu - avstrijski (Krammer in dr. 2001, Dotter 2011), italijanski (Prinetto in dr. 2011), hrvaški (Tarczay 2010); kot tudi tistih, ki imajo posebej močan vpliv v evropskem prostoru - nemški (Konrad et al. 2003; Konig et al. 2008) - in v svetovnem merilu, denimo avstralski znakovni jezik (Johnston et al. 2006). Vsekakor si želimo tudi, da bi z rezultati projekta posredno vplivali na status znakovnega jezika in odnosa do gluhote v slovenski družbi.

5. Sklep

Ker je projekt še v teku in smo z označevanjem korpusa pričeli šele v zadnjih mesecih, zaenkrat ne moremo poročati o številnih znanstvenih izsledkih, zato pa toliko več o načrtih za prihodnost. Tako nameravamo najprej zaključiti zbiranje gradiva in zagotoviti reprezentativno pokritost vseh društev gluhih, vzporedno s tem pa že teče označevanje korpusa in dodajanje novih ravni označevanja. Kot kažejo tudi izkušnje iz tujine, se predvsem skozi samo označevanje porajajo temeljna teoretična vprašanja o jezikoslovni analizi znakovnega jezika. V označevanje korpusa z iLexom nameravamo v prihodnje vključiti še gluhe študente.

Ko bo označen dovolj velik del korpusa, bodo prvi rezultat pogostostni podatki o rabi temeljnih kretenj SZJ, ti pa bodo uporabni tudi pri gradnji obstoječega slovarja SZJ in pri prenovi obstoječih učbenikov. Naš naslednji cilj je pilotni slovnični opis izbranih prvin SZJ, še posebej skladenjske strukture in uporabe prostorskosti. V načrtu so tudi poskusi samodejne sinteze SZJ na podlagi zapisa HamNoSys.

S povsem drugega vidika pa bi želeli poiskati tudi odgovore na nekatera sociolingvistična vprašanja, povezana s SZJ, ter raziskati dejavnike, ki vplivajo na njegov razvoj. Eno od tovrstnih vprašanj je povezano z vlogo šolanja na splošno in konkretne izobraževalne ustanove, saj se SZJ trenutno poučuje na le eni šoli. Zanimivo bi bilo tudi raziskati znakovni sleng med mladimi ter vpliv drugih kultur, prav tako pa tudi variantnost SZJ glede na regijo, družbeno okolje in starost, vse to v razmerju do izražene potrebe gluhe skupnosti po standardizaciji SZJ.

6. Viri

Dotter, F. (2011). Sign Languages and Their Communities Now and in the Future. "Multilingualism in Europe. Prospects and Practices in East-Central Europe", Budimpešta, 25.-26. marec 2011.

Hanke, T. in Stolz, J. (2008). iLex – A Database Tool For Integrating Sign Language Corpus Linguistics and Sign Language Lexicography, LREC 2008, May 28 – May 30 2008, poster

Johnston, T. in A. Schembri (2006). “Issues in the creation of a digital archive of a signed language”. V L. Barwick and N. Thiesberger (ur.) *Sustainable data from digital fieldwork*. Sydney: Sydney University Press, 7-16.

König, S., R. Konrad in G. Langer. (2008). “What’s in a sign? Theoretical lessons from practical sign language lexicography”. V J. Quer (ed.) *Signs of the time. Selected papers from TISLR 2004*. Hamburg: Signum, 379-404.

Konrad, R., A. Schwarz, S. König, G. Langer, T. Hanke in S. Prillwitz (2003). *Fachgebärdenlexikon Sozialarbeit/Sozialpädagogik*. Hamburg: Signum. Dostopno na: http://www.sign-lang.uni-hamburg.de/slex/.

Krammer, K., Bergmeister, E., Dotter, F., Hilzensauer, M., Okorn, I., Orter, R. in Skant, A. (2001). The Klagenfurt database for sign language lexicons. Sign Language and Linguistics, vol. 4, issue 1, pp. 191-201

Prinetto, P. in dr. (2011). The Italian Sign Language Sign Bank: Using WordNet for Sign Language corpus creation, ICCIT, 2011 Internation Conference , 29-31 March 2011, 134-137

Schmaling, C. in T. Hanke (2001). *HamNoSys 4.0*. Dostopno na: http://www.sign-lang.uni-hamburg.de/Projekte/HamNoSys/HNS4.0/englisch/HNS4.pdf.

Tarczay, S. (2010.). Pretpostavke profesionalizacije prevoditelja znakovnoga jezika za gluhe i gluhoslijepe osobe. Magistarski rad. Zagreb: Edukacijsko-rehabilitacijski fakultet Sveučilišta u Zagrebu.

ZEN: zasnova glasovnih e-storitev v zdravstvu

Jerneja Žganec Gros[1], Tanja Majcen[2], Marko Ivančič[2], Žiga Golob[1], Aleš Miheljč[1], Boštjan Vesnicer[1], Boris Kern[3], Andrej Perdih[3], Primož Jakopin[3], Petar Brajak[4]

[1]Alpineon d.o.o.
Ljubljana, Slovenija
jerneja.gros@alpineon.si
[2]SRC d.o.o.
[3]ZRC-SAZU
[4]Medius d.o.o.

Povzetek

V projektu ZEN smo s pomočjo sinteze slovenskega govora razvili prototip e-storitve za podporo pri jemanju zdravil s fokusom na ostarelih uporabnikih ter uporabnikih s posebnimi potrebami, predvsem na slepih in slabovidnih. Storitev uporabnikom preko set-top box-a ali mobilnega telefona omogoča vpogled v seznam zdravil, ki so jim predpisana, v podatke o posameznih predpisih, kot tudi v navodila zdravil. Prav tako storitev s pomočjo glasovnih in besedilnih sporočil opominja uporabnika, da mora vzeti predpisani odmerek zdravil.

ZEN: Advanced Voice-Enabled e-Health Services

We present how Slovenian text-to-speech synthesis technologies have been used to develop a prototype solution of a novel e-Health service ZEN, which will focus on the elderly user group, along with blind and visually impaired users. The service features two communication channels for delivering information to the users: via telephone and via a TV screen, connected to a set-top-box. The user can browse and listen to descriptions of prescribed medicines and therapies. Further, he can receive textual and/or visual reminders related to his therapy. Validation results of the ZEN service are presented.

1. Uvod

V sodobni družbi je zagotavljanje sistema zdravstvenega varstva kritičnega pomena, saj med drugim predstavlja merilo za demokratično razvitost družbe. Za njegovo uspešno delovanje je potrebno vzpostaviti učinkovit način sodelovanja in nadzora med vsemi vpletenimi v celotni verigi zdravstvenega varstva.

Pomemben del interakcije človeka s strojem je *uporabniška izkušnja*, ki v večini primerov predstavlja bistven kriterij za odločitev o posvojitvi in redni uporabi nove naprave ali e-storitve s strani končnih uporabnikov. Izhodiščna predpostavka projekta je bila, da uvedba glasovnega uporabniškega vmesnika lahko pomembno doprinese k izboljšanju uporabniške izkušnje v zdravstvenih e-storitvah. Da bi to preverili, smo razvili celotno verigo tehnoloških komponent pri vzpostavitvi zdravstvene e-storitve za izbrano zdravstveno situacijo ter ugotavljali spremembe v uporabniški izkušnji ob dodatku naprednega glasovnega uporabniškega vmesnika.

Namen projekta *ZEN: zdravstvene e-storitve z naprednimi glasovnimi uporabniškimi vmesniki* je bila zasnova nove e-storitve na področju e-vključenosti in e-zdravja ter interdisciplinarni predkonkurenčni razvoj informacijsko-komunikacijskega sistema, podprtega z naprednimi glasovnimi uporabniškimi vmesniki. Poglavitno pozornost smo posvetili razvoju in validaciji novih tehnoloških rešitev v zdravstvenih e-storitvah, ki povečujejo uporabniško izkušnjo pri uporabi tovrstnih storitev.

Pri izdelavi demonstracijskega prototipa ZEN smo uporabili dolgoletne izkušnje vseh projektnih partnerjev s področij razvoja govornih tehnologij (Alpineon) in jezikovnih tehnologij (Inštitut za slovenski jezik v okviru ZRC-SAZU), informacijskih tehnologij v zdravstvu (SRC d.o.o.) ter odprtokodnih komunikacijskih tehnologij (Medius d.o.o.). V okviru projekta smo razvili nove jezikovne vire (slovar izgovarjav), številne nove tehnološke rešitve (glasovni strežnik, podatkovno-komunikacijski strežnik, glasovno-podprto aplikacijo za set-top box) ter novo odprtokodno tehnološko rešitev (odprtokodni TK strežnik za konvergentno povezovanje spletnih zdravstvenih e-storitev in govornih tehnologij).

V fazi integracije smo razvite tehnološke rešitve integrirali v enovit demonstracijski prototip, ki ga je možno prilagoditi za številne primere uporabe. Rezultate projekta je preveril zunanji recenzor. Med prvimi testnimi uporabniki prototipa je bila skupina predvidenih končnih uporabnikov rezultatov projekta – ostareli ter slepe in slabovidne osebe, ki so preverjali tako primernost uporabe glasovnih uporabniških vmesnikov v zdravstvenih e-storitvah, kot tudi primernost izvedbe celotne rešitve na demonstriranem primeru uporabe v okviru Festivala za tretje življenjsko obdobje 2012.

2. Glasovne tehnologije

Govor predstavlja najnaravnejši način komunikacije med človekom in strojem (Lazzari, 2006). Govorno podprti uporabniški vmesniki omogočajo uporabniško prijazno komunikacijo, še posebej v okolju mobilnih komunikacij. Ponujajo tudi možnost enakopravnega vključevanja skupin oseb s posebnimi potrebami, predvsem ostarelih, slepih in slabovidnih v sodobno informacijsko družbo. Sistemi, ki vključujejo govorne tehnologije, omogočajo hitre odzivne čase, znižujejo stroške poslovanja in prispevajo k večji prepoznavnosti na trgu. Nudijo možnost avtomatizacije obstoječih storitev in cenenega razvoja množice novih storitev in naprav na številnih sektorjih uporabe.

Za uspešen razvoj in uporabo govorno podprtih rešitev je potrebno zagotoviti učinkovite in visoko kakovostne komponente sistema govornega dialoga, to je uspešnost avtomatskega razpoznavanja govora in kvalitetno, razumljivo in naravno zvenečo sintezo govora, ki omogoča samodejno pretvarjanje vhodnih besedil v glasovno obliko (CHIL).

Raziskave in razvoj na področju govornih tehnologij se danes hitro prenašajo v komercialne sisteme, ki postajajo vse bolj razširjeni. Za jezike s široko bazo govorcev se rešitve samodejne prepoznave govora (angl. automatic speech recognition ali ASR) in samodejne sinteze govora (angl. text-to-speech synthesis ali TTS) vgrajujejo v cenovno ugodne programske pakete, namenjene predvsem uporabi na osebnih računalnikih (Karpov in drugi, 2006; Burlieanu, 2004). Evropa danes predstavlja enega najnaprednejših trgov govornih tehnologij. Evropska unija si prizadeva, da so potrebna orodja in viri na razpolago za vse jezike Evropske unije kot tudi glavne svetovne komercialne jezike, s čimer utira pot prodorni več jezikovni informacijski evropski družbi. Z uvajanjem večjezičnih proizvodov in storitev poskuša Evropska komisija doseči svoj ambiciozni cilj – posplošitev dostopa do informacij za vse evropske državljane, ki je tudi ključni cilj pobude *i2010*.

Vendar se obseg sistematične raziskanosti jezikov, ki se govorijo v Evropi, od enega jezika do drugega zelo razlikuje, pri čemer je bila v sklopu posebnih projektov znotraj EU, pa tudi nacionalnih in komercialnih projektov, dobro raziskana le peščica jezikov (angleščina, španščina, francoščina in nemščina), nekateri pa so bili komajda obravnavani. Pogosto so bile prav nove države članice tiste, ki niso imele možnosti za razvoj jezikovnih tehnologij za svoje pisne in govorjene jezike. Za slovenski jezik je na voljo komercialno dostopen prepoznavalnik govora za omejeno področje uporabe ter več raziskovalnih prototipov.

Sinteza govora predstavlja postopek samodejnega pretvarjanja vhodnih besedil, zapisanih v elektronski obliki, v govor. Za slovenščino sicer obstaja več sintetizatorjev govora, ki so namenjeni uporabi na osebnih računalnikih (Mihelič in drugi, 2006). Ni pa na voljo robustne in razširljive strežniške rešitve, ki bi ponujala usluge sinteze govora v širokemu spektru e-storitev in aplikacij. Prav ta tehnološka rešitev je bila razvita v okviru projekta ZEN. Izboljšan je bil tudi del sintetizatorja govora, ki določa izgovarjavo novih, neznanih besed, kar je še zlasti pomembno pri vključevanju sinteze govora na novo področje uporabe. Razširjen je bil tudi slovar izgovarjav, ki sedaj pokriva vse iztočnice iz Slovenskega pravopisa.

3. Opis e-storitve ZEN

Pomemben del interakcije človeka s strojem je *uporabniška izkušnja*, ki v večini primerov predstavlja bistven kriterij za odločitev o posvojitvi in redni uporabi nove naprave ali e-storitve s strani končnih uporabnikov. V okviru projekta smo razvili celotno verigo tehnoloških komponent pri vzpostavitvi značilne zdravstvene e-storitve ter ugotavljali spremembe v uporabniški izkušnji ob dodatku naprednega glasovnega uporabniškega vmesnika.

Zdravstvena e-storitev z naprednimi uporabniškimi vmesniki – ZEN – je namenjena spremljanju pacientovega stanja in poteka zdravljenja na daljavo. Predstavlja pripomoček pri izvajanju poteka zdravljenja. Uporabniku lahko služi kot opomnik za redno izvajanje aktivnosti, predpisanih s strani zdravnika, ter kot informator in podatkovni posredovalec med njim ter zdravnikom oz. zdravstvenim osebjem.

V sistemu ZEN se podatki uporabniku lahko posredujejo preko dveh komunikacijskih kanalov.

Prvi je telefonski kanal, kjer uporabnik dostopa do informacij v obliki glasovnega dialoga, podprtega s sintetizatorjem govora, ki govor samodejno generira iz dinamičnega elektronskega besedila.

Drugi komunikacijski kanal predstavlja set-top box (STB), kjer v dialogu z uporabnikom poleg glasovnih komponent nastopajo še vizualno grafične, kot so npr. prikaz besedila prilagodljive velikosti, slike, ipd. Preko istih dveh komunikacijskih kanalov lahko tudi uporabnik sproži zahtevo za izvajanje e-storitve.

4. Opis tehnološke rešitve ZEN

Sistemska arhitektura ZEN modulov, ki smo jih razvili v okviru projekta, je predstavljena na sliki 1. Poglavitni moduli sistema so naslednji: podatkovno-komunikacijski strežnik, TK strežnik, glasovni strežnik, aplikacija za STB napravo.

4.1. Podatkovno-komunikacijski strežnik

Podatkovno-komunikacijski strežnik, glede na nastavitve in podatke, ki jih sistem ZEN želi posredovati uporabniku, določi tip dialoga in komunikacijski kanal. Skrbi za podatke, ki so potrebni za ustvarjanje dinamičnih dialogov z uporabnikom za komunikacijo preko STB naprave. Dialogi poleg glasovnih vsebujejo tudi vizualno grafične elemente. Način prikaza podatkov lahko pacient, glede na svoje potrebe, spreminja. Velikost na ekranu prikazanega besedila je nastavljiva, kar je primerno za starejše ter slabovidne paciente. Podatki se med sistemom in STB napravo prenašajo preko varnega šifriranega kanala.

Vsi podatki o poteku zdravljenja, predpisanih zdravilih ipd., se hranijo na podatkovno-komunikacijskem strežniku v šifrirani obliki. Do njih ima dostop samo pooblaščeno medicinsko osebje ter, v določeni obliki in obsegu, pacient. Podatke sistem pacientu posreduje preko dveh kanalov: preko STB naprave ter preko telefonskega kanala.

Aplikacija za zdravstveno osebje lečečemu zdravniku in ostalemu pooblaščenemu zdravstvenemu osebju omogoča spremljanje poteka pacientovega zdravljenja na daljavo, vpisovanje in spreminjanje podatkov o zdravljenju, predpisanih zdravilih, časovnih intervalih zaužitja zdravila ter drugih s pacientovim zdravljenjem povezanih podatkov. Vsi ti podatki se zapisujejo v podatkovno zbirko na podatkovno-komunikacijskem strežniku.

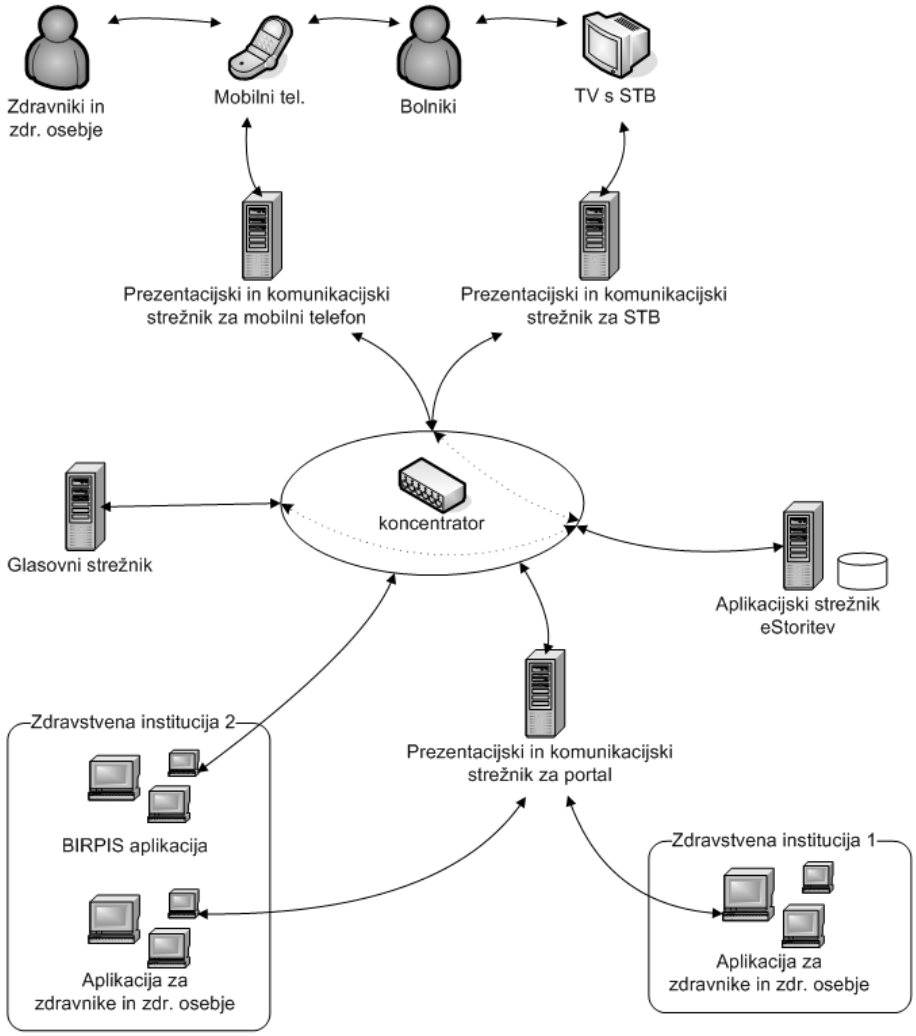
4.2. TK strežnik

Skrbi za ustvarjanje dialogov z uporabnikom za komunikacijo preko telefonskega kanala. Odvisen je od podatkovno-komunikacijskega strežnika, od katerega prejema zahteve po ustvarjanju dialoga z uporabnikom ter podatke, ki jih mora le temu posredovati. V domeni TK strežnika so govorni dialogi, ki potekajo preko telefonskega govornega kanala, za razliko od dialogov, ki so namenjeni komunikaciji preko STB naprave in vsebujejo tudi vizualno grafične elemente. V smeri od sistema proti pacientu poteka pretok informacij v obliki sintetiziranega govora, v obratni smeri pa preko detekcije pritiska tipk (DTMF) ali snemanja glasovnega sporočila.

Rešitev je bila izvedena na podlagi odprtokodne platforme SDP (Service Delivery Platform) in konceptov okolij IMS (IP Multimedia Subsystem) in omogoča integracijo spletnih storitev z govorno telefonijo (klasična stacionarna telefonija, mobilna telefonija, VoIP, SoftPhone).

Arhitektura je zgrajena na osnovi najboljših odprtokodnih orodij in tehnologij za SDP in na ta način zagotavlja dolgoročno odprtost rešitev, hiter razvojni cikel, prilagoditev novim produktom in standardom ter nizko ceno. Arhitektura je zasnovana na konceptih SOA (Service Oriented Architecture) in zajema celoten sklad komponent, od operacijskega sistema do vmesnikov, ki omogočajo:

- pošiljanje individualiziranih opomnikov,
- snemanje in pošiljanje glasovnih sporočil,
- objave posnetkov telefonskega govora,
- sintezo govora iz obvestil na oglasni deski s pomočjo glasovnega strežnika,
- integracijo storitev s koledarji.



Slika 1. Osnovna arhitektura sistema ZEN.

4.3. Glasovni strežnik

Glasovni strežnik skrbi za pretvorbo elektronskega besedila v govor. Vsi dialogi, ki predvidevajo komunikacijo z bolnikom oz. posredovanje podatkov preko govornega dialoga, uporabljajo glasovni strežnik za dinamično sprotno tvorjenje govora iz podatkov, ki so shranjeni v sistemu v obliki elektronskega besedila. Klientom dodeljuje kanale za sintezo ter druge potrebne vire.

Strežnik skrbi za identifikacijo pošiljatelja zahteve za sintezo govora iz elektronskega besedila. Iniciatorji zahtevka so lahko ostali strežniki sistema ZEN ali neposredno STB naprave. Identifikacijo lahko izvede glede na nameščeno serijsko številko, IP naslov ali nameščen certifikat za varno šifrirano komunikacijo. Glasovni strežnik skrbi tudi za upravljanje s profili uporabnikov in vodi statistiko po uporabnikih. Zaradi hitrejšega odziva je sposoben tudi hraniti v govor že pretvorjena sintetizirana besedila (posebno v primeru vnaprej definiranih dialogov, navodil ipd.) v obliki zvočnih datotek. Prav tako skrbi za pretvorbo vhodnega besedila (SSML – W3C Speech Synthesis Markup Language) v format, primeren za pretvorbo v govor.

V okviru projekta je bil tudi nadgrajen modul za grafemsko-fonetično pretvorbo vhodnih besedil ter

zgrajen nov slovar izgovarjav za preko 90.000 iztočnic iz Slovenskega pravopisa (Toporišič, 2007). Izgradnja slovarja je temeljila na formatu slovarja izgovarjav SI-PRON (Žganec Gros in drugi, 2006).

4.4. Podatki in komunikacija med moduli

Vsi podatki se shranjujejo na podatkovno-komunikacijskem strežniku v šifrirani obliki. Do njih lahko dostopa samo pooblaščeno zdravstveno osebje. Dostop do svojih podatkov, v vnaprej določenem obsegu in obliki, pa ima tudi pacient/uporabnik. Podatki se na pacientovo zahtevo pretvorijo v obliko, primerno za izbrani tip komunikacijskega kanala in dialoga ter se posredujejo pacientu na STB ali telefon.

Komunikacija med moduli v sistemu (med napravo STB ter podatkovno-komunikacijskim strežnikom, med aplikacijo za zdravstveno osebje ter podatkovno-komunikacijskim strežnikom) poteka preko varnih šifriranih kanalov. Podatki so med prenosom zaščiteni pred vpogledom tretjih oseb. Sistem je sposoben zagotoviti vzpostavitev varnega kanala ne glede na pot prenosa podatkov (preusmeritve, podomrežja, usmerjevalniki in druge namenske naprave). Identifikacija uporabnika temelji na osnovi nameščene serijske številke, uporabnikovega certifikata ter IP oz. MAC naslova.

4.5. Komunikacija z uporabnikom

V sistemu ZEN sta za komunikacijo sistema oz. zdravstvenega osebja s pacientom predvidena dva komunikacijska kanala: set-top box (STB) ter telefon.

Set-top box (STB) je naprava, namenjena priklopu na televizor oz. monitor. Pacient jo krmili s pomočjo daljinskega upravljalnika in preko nje dostopa do različnih podatkov v zvezi z zdravljenjem. Naprava podatke pacientu posreduje v slikovni in/ali zvočni obliki. Prav tako pacientu omogoča posredno oz. neposredno komunikacijo z zdravstvenim osebjem, deluje pa tudi kot pripomoček za spremljanje dnevnega izvajanja zdravljenja - tako količinsko kot časovno. Primer uporabniškega vmesnika ZEN je prikazan na sliki 2. Pacienta opozarja na termin bližajoče se aktivnosti zdravljenja, in od njega pričakuje tudi potrditev o izvedeni aktivnosti.

Slika 2. Primeri posnetkov zaslona uporabniškega vmesnika ZEN.

Telefon predstavlja alternativni komunikacijski kanal za posredovanje informacij v obe smeri. Komunikacija poteka preko govorjenega dialoga (v smeri k pacientu) oz. pritiska tipk na telefonskem aparatu (v smeri od pacienta) in za razliko od STB ne vsebuje vizualno grafičnih elementov.

5. Značilni primeri uporabe e-storitve ZEN

Uporabnik lahko preko e-storitve ZEN prejema sporočila od zdravnika, vezana na dnevno spremljanje poteka zdravljenja, prejema opozorila na bližajoči se ali spuščen termin za izvajanje določene aktivnosti, povezane z zdravljenjem (denimo redno razgibavanje ali jemanje zdravil), zdravnik lahko od svojega pacienta zahteva potrditev o izvedeni aktivnosti ipd. Pacient lahko preko e-storitve dostopa do rednih napotkov svojega zdravnika, do opisov predpisanih zdravil (navodila, stranski učinki, doze, trajanje, količine…), lahko posredno ali neposredno komunicira z lečečim zdravnikom oz. zdravstvenim osebjem, lahko zaprosi za dodatne informacije v zvezi z zdravljenjem ali preveri, kdaj je nazadnje izvedel predpisano aktivnost v zdravljenju.

Primer 1: Zdravnik ureja koledarje za svoje paciente (npr. datumi in ure za predpisano jemanje zdravil, opomniki in obvestila). Vsakemu dogodku zdravnik na portalu lahko pripne informacijo bodisi v obliki besedila ali govornega sporočila. Sistem ob uri, ki je določena s koledarjem, kontaktira pacienta, bodisi preko STB, bodisi preko telefonskega kanala, in ga obvesti o terminu

predpisane aktivnosti v poteku zdravljenja. Pacient lahko sistemu potrdi izvajanje dejavnosti preko istega komunikacijskega kanala.

Primer 2: Pacient iz Primera 1 se ne more spomniti, ali je ob predpisani uri izvedel razgibavanje ali ne. S pomočjo e-storitve ZEN se prepriča o dejanskem stanju.

Primer 3: Zdravnik na oglasni deski svojih pacientov pušča redna obvestila (navodila, informacije o vrsti in napredku zdravljenja, informacije o predpisanih zdravilih, termin pregleda pri zdravniku) v obliki besedila ali govornega sporočila. Pacient preverja obvestila preko e-storitve ZEN. Slepi in slabovidni uporabnik se lahko odloči za sprejem sporočila v glasovni obliki.

6. Evalvacija sistema

V fazi integracije smo razvite tehnološke rešitve integrirali v enovit demonstracijski prototip, ki ga je možno prilagoditi za številne primer uporabe.

Posebej smo ročno evalvirali nove jezikovne vire. Samodejna grafemsko-fonetična transkripcija iztočnic iz Slovenskega pravopisa namreč ni bila vedno uspešna (Jakopin, 2010).

K preskusu nove e-storitve ZEN smo povabili zunanjega recenzorja in skupino končnih uporabnikov rezultatov projekta – ostarele ter slepe in slabovidne osebe. V nadaljevanju podajamo rezultate obeh evalvacij.

6.1. Evalvacija s strani končnih uporabnikov

Med prvimi testnimi uporabniki prototipa je bila skupina predvidenih končnih uporabnikov rezultatov projekta – ostareli ter slepe in slabovidne osebe, ki so preverjali tako primernost uporabe glasovnih uporabniških vmesnikov v zdravstvenih e-storitvah, kot tudi primernost izvedbe celotne rešitve na demonstriranem primeru uporabe v okviru Festivala za tretje življenjsko obdobje 2010. Na istem Festivalu smo leto poprej zbirali uporabniške zahteve za e-storitev ZEN.

Sestavili smo anketo, ki je obsegala 6 vprašanj. Vprašanja so se nanašala na opremljenost bivalnih prostorov anketirancev z informacijsko komunikacijsko opremo, njihove potrebe po dostopu do zdravstvenih storitev ter na ocenjevanje bistvenih vidikov uporabnosti e-storitve ZEN.

Anketiranih je bilo 60 obiskovalcev festivala F3ŽO 2010. Vsem anketirancem je bila najprej predstavljena razvita aplikacija za dostop do e-zdravstvenih storitev, nato so izpolnjevali anketo.

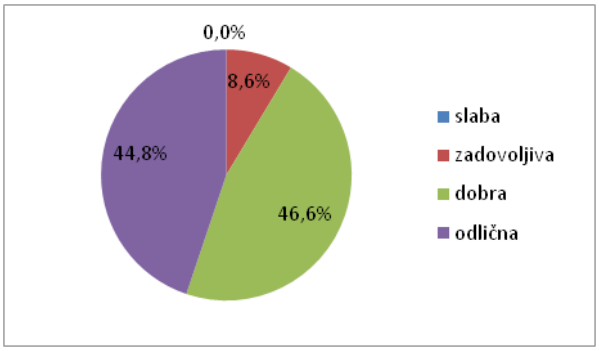
Večino anketirancev so predstavljali starejši. 52% anketirancev je bilo starejših od 64 let, 47% pa med 20 in 64 let. 75% anketirancev je bilo ženskega spola.

Vsi anketiranci so imeli dostop do tv sprejemnika ter telefona. Telefon uporablja pogosto ali zelo pogosto več kot 63% anketirancev, vsi izmed njih pa so že uporabljali telefon.

78% anketirancev se zdi koristno, da bi bili na bližajoč termin za obisk zdravnika opozorjeni preko telefona ali tv sprejemnika.

Enako vprašanje smo zastavili tudi pri zbiranju uporabniških zahtev na istem festivalu eno leto pred evalvacijo, ko smo dobili praktično enake rezultate.

Anketirance smo vprašali, ali se jim zdi koristno, da bi lahko nasvete zdravnika enostavno preverili ali v tekstovni ali v glasovni obliki preko tv sprejemnika oz. v glasovni obliki preko telefona. Enako vprašanje smo postavili tudi pri zbiranju uporabniških zahtev. Anketirancem se je v tej anketi zdelo takšno preverjanje precej bolj koristno. Kar 92% anketirancem se je zdelo primerno preverjanje informacij na tv sprejemniku v tekstovni obliki, kar je več kot 20% več v primerjavi s predhodno anketo.



Slika 3. Razporeditev odgovorov anketirancev na vprašanje o uporabnosti e-storitve ZEN. Na vprašanje je odgovorilo 97% anketirancev.

Kriterij	Ocena	Opis kriterija
Interoperabilnost in standardi	4	Zagotavljanje interoperabilnosti in uporaba odprtih standardov
Namestitev/priprava za uporabo	N	Preprostost/kompleksnost/samodejnost/ trajanje namestitvenega postopka, ...
Stopnja informacijske varnosti	4	Prenos, dostopnost in zaščita podatkov
Kvaliteta govornega up. vmesnika	4	Avtomatska sinteza govora
Berljivost in jasnost besedila	4	Uporaba barv, fontov, kontrastov, ...
Uporabnost	4	Preprostost/kompleksnost sistema za končnega uporabnika rešitve
Navigacija	4	Preprostost navigacije po menijih, informacija o trenutnem meniju

Tabela 1. Rezultati ocen tehniško uporabniških vidikov glasovno podprte e-storitve ZEN. Pri vrednotenju tehnično uporabniškega vidika je bila uporabljena štiristopenjska ocenjevalna lestvica. 1 pomeni najnižjo, 4 pa najvišjo stopnjo kvalitete. 'N' pomeni, da ocena glede na opisni kriterij ni možna. V pojasnilu so dodatno razloženi opisni kriteriji s tehnično-uporabniškega vidika.

Preverjanje v glasovni obliki preko tv sprejemnika se je zdelo koristno 84% anketirancev (v prejšnji anketi 62%), v glasovni obliki preko telefona pa 86% anketirancev (v prejšnji anketi 77%). Vzrok za porast deleža anketirancev, ki se jim zdi tako preverjanje informacij koristno, je najverjetneje v tem, da so si ljudje pred demonstracijo e-storitve ZEN težko predstavljali, na kakšen način bi lahko takšne informacije preverjali preko domačega tv sprejemnika.

Anketiranci so ocenjevali tudi različne vidike razvite aplikacije, kot so berljivost, razločnost, hitrost govora, možnost prekinitve, prijaznost navigacije ter uporabnost sistema.

Anketiranci so vse vidike aplikacije večinoma ocenili kot dobre ali odlične, kar je prikazano na sliki 3. Izrazili pa so tudi željo po možnosti prekinitve glasovnega predvajanja.

6.2. Evalvacija s strani zunanjega recenzorja

Rezultate projekta je preveril tudi zunanji recenzor z vidika doseganja ciljev projekta, kot tudi s tehniško-uporabniških vidikov projekta (Prijatelj, 2010). V tabeli 1 podajamo rezultate recenzorske ocene.

7. Zaključek

V okviru projekta ZEN smo izvedli raziskave s področja razvoja na storitvah temelječih rešitev za podporo sodelovanja ponudnikov in uporabnikov storitev zdravstvenega varstva z namenom povečanja dostopnosti, uporabe, prijaznosti do uporabnika in preglednosti storitev z zdravstvenega področja. Pokazali smo primernost uporabe govornih tehnologij v zdravstvenih e-storitvah.

Z vključitvijo govornega kanala in televizorja ter telefona kot dostavnega kanala sistem ZEN dosega ciljno populacijo v veliko večji meri kot druge informacijske rešitve. To namreč predstavlja poenostavitev uporabe informacijske tehnologije s stališča ciljne populacije – praviloma starejšega, gibalno in informacijsko podrejenega segmenta državljanov.

Takšna zasnova omogoča izboljšanje penetracije informacijskih rešitev v ciljni populaciji in zagotavlja platformo, na kateri bo mogoče v prihodnosti razviti in ponuditi širok nabor informacijsko temelječih storitev, ki bodo dopolnjevale realne zdravstvene e-storitve.

8. Zahvala

Opisano razvojno-raziskovalno delo je nastalo v okviru projekta *ZEN: zdravstvene e-storitve z naprednimi glasovnimi uporabniškimi vmesniki*, ki ga je delno financirala Evropska unija iz sredstev Evropskega sklada za regionalni razvoj v okviru pogodbe št. 3211-09-000523.

9. Literatura

Burileanu D., Fecioru A., Ion D., Stoica M., Ilas C., 2004. An optimized TTS system implementation using a Motorola StarCore SC140-based processor. Proceedings of the ICASPP, 5 : 17-21.

CHIL, Computers in the Human Interaction Loop. EU FP6 project, http://chil.server.de/

Karpov E., Kiss I., Leppänen J., Olsen J., Oria D., Sivadas S., Tian J., 2006. Short Message Dictation on Symbian Series 60 Mobile Phones. Workshop on Speech in Mobile and Pervasive Environments (SiMPE) in Conjunction with MobileHCI. Helsinki, Finland.

Jakopin P., 2010. Računalnikov izgovor : besede, besede, besede. *Delo*. Ljubljana, 31. jul. 2010, 8 52 (175): 10.

Lazzari G., 2006. Human Language Technologies for Europe. http://www.tc.star.org/ pubblicazioni/ D17_ HLT_ENG.pdf.

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*, 36(1).

Prijatelj V., 2010. Recenzija prototipne rešitve na projektu 'ZEN' – Zdravstvene e-storitve z naprednimi glasovnimi uporabniškimi vmesniki'.

Toporišič, J. (ur) , 2007. Slovenski pravopis. Slovar, 3.natis. Založba ZRC Ljubljana, ZRC SAZU.

Žganec Gros J., Cvetko-Orešnik V., Jakopin P., Mihelič A., 2006. SI-PRON pronunciation lexicon : a new language resource for Slovenian. *Informatica 2006*, 30(4): 447-452.

Indeks avtorjev / Author index

Uredila / Edited by

Tomaž Erjavec, Jerneja Žganec Gros