Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA - IS 2017

Zvezek C

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY - IS 2017

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Uredila / Edited by
Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

9.–13. oktober 2017 / 9–13 October 2017
Ljubljana, Slovenia

Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2017

Zvezek C

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY – IS 2017

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Uredila / Edited by

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

9. - 13. oktober 2017 / 9th – 13th October 2017
Ljubljana, Slovenia

Urednika:

Dunja Mladenić
Laboratorij za umetno inteligenco
Institut »Jožef Stefan«, Ljubljana

Marko Grobelnik
Laboratorij za umetno inteligenco
Institut »Jožef Stefan«, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2017

Kataložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=292473088
ISBN 978-961-264-114-6 (pdf)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2017

Multikonferenca Informacijska družba (http://is.ijs.si) je z **dvajseto** zaporedno prireditvijo osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Letošnja prireditev je ponovno na več lokacijah, osrednji dogodki pa so na Institutu »Jožef Stefan«.

Informacijska družba, znanje in umetna inteligenca so spet na razpotju tako same zase kot glede vpliva na človeški razvoj. Se bo eksponentna rast elektronike po Moorovem zakonu nadaljevala ali stagnirala? Bo umetna inteligenca nadaljevala svoj neverjetni razvoj in premagovala ljudi na čedalje več področjih in s tem omogočila razcvet civilizacije, ali pa bo eksponentna rast prebivalstva zlasti v Afriki povzročila zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so planetarni konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali dvanajst odličnih neodvisnih konferenc. Predstavljenih bo okoli 200 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic. Prireditev bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica, ki se ponaša s **40-letno** tradicijo odlične znanstvene revije. Odlične obletnice!

Multikonferenco Informacijska družba 2017 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Soočanje z demografskimi izzivi
- Kognitivna znanost
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Izkopavanje znanja in podatkovna skladišča
- Vzgoja in izobraževanje v informacijski družbi
- Četrta študentska računalniška konferenca
- Delavnica »EM-zdravje«
- Peta mednarodna konferenca kognitonike
- Mednarodna konferenca za prenos tehnologij - ITTC
- Delavnica »AS-IT-IC«
- Robotika

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija, SLAIS, DKZ in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in inštitucijam, še posebej pa 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 2017 bomo petič podelili nagrado za življenjske dosežke v čast Donalda Michija in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe bo prejel prof. dr. Marjan Krisper. Priznanje za dosežek leta bo pripadlo prof. dr. Andreju Brodniku. Že šestič podeljujemo nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobilo padanje slovenskih sredstev za akademsko znanost, tako da smo sedaj tretji najslabši po tem kriteriju v Evropi, jagodo pa »e-recept«. Čestitke nagrajencem!

Bojan Orel, predsednik programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2017

In its **20th year**, the Information Society Multiconference (http://is.ijs.si) remains one of the leading conferences in Central Europe devoted to information society, computer science and informatics. In 2017 it is organized at various locations, with the main events at the Jožef Stefan Institute.

The pace of progress of information society, knowledge and artificial intelligence is speeding up, and it seems we are again at a turning point. Will the progress of electronics continue according to the Moore's law or will it start stagnating? Will AI continue to outperform humans at more and more activities and in this way enable the predicted unseen human progress, or will the growth of human population in particular in Africa cause global decline? Both extremes seem more and more likely – fantastic human progress and planetary decline caused by humans destroying our environment and each other.

The Multiconference is running in parallel sessions with 200 presentations of scientific papers at twelve conferences, round tables, workshops and award ceremonies. Selected papers will be published in the Informatica journal, which has **40 years** of tradition of excellent research publication. These are remarkable achievements.

The Information Society 2017 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Facing Demographic Challenges
- Cognitive Science
- Collaboration, Software and Services in Information Society
- Data Mining and Data Warehouses
- Education in Information Society
- 4th Student Computer Science Research Conference
- Workshop Electronic and Mobile Health
- 5th International Conference on Cognitonics
- International Conference of Transfer of Technologies - ITTC
- Workshop »AC-IT-IC«
- Robotics

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, SLAIS, DKZ and the second national engineering academy, the Slovenian Engineering Academy. In the name of the conference organizers we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

For the fifth year, the award for life-long outstanding contributions will be delivered in memory of Donald Michie and Alan Turing. The Michie-Turing award will be given to Prof. Marjan Krisper for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, an award for current achievements will be given to Prof. Andrej Brodnik. The information lemon goes to national funding of the academic science, which degrades Slovenia to the third worst position in Europe. The information strawberry is awarded for the medical e-recipe project. Congratulations!

Bojan Orel, 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, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
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, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Robert Blatnik
Aleš Tavčar
Blaž Mahnič
Jure Šorn
Mario Konecki

Programme Committee

Bojan Orel, 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
Mitja Luštrek
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

Invited lecture

AN UPDATE FROM THE AI & MUSIC FRONT

Gerhard Widmer
Institute for Computational Perception
Johannes Kepler University Linz (JKU), and
Austrian Research Institute for Artificial Intelligence (OFAI), Vienna

Abstract
Much of current research in Artificial Intelligence and Music, and particularly in the field of Music Information Retrieval (MIR), focuses on algorithms that interpret musical signals and recognize musically relevant objects and patterns at various levels -- from notes to beats and rhythm, to melodic and harmonic patterns and higher-level segment structure --, with the goal of supporting novel applications in the digital music world. This presentation will give the audience a glimpse of what musically "intelligent" systems can currently do with music, and what this is good for. However, we will also find that while some of these capabilities are quite impressive, they are still far from (and do not require) a deeper "understanding" of music. An ongoing project will be presented that aims to take AI & music research a bit closer to the "essence" of music, going beyond surface features and focusing on the expressive aspects of music, and how these are communicated in music. This raises a number of new research challenges for the field of AI and Music (discussed in much more detail in [Widmer, 2016]). As a first step, we will look at recent work on computational models of expressive music performance, and will show some examples of the state of the art (including the result of a recent musical 'Turing test').

References
Widmer, G. (2016).
Getting Closer to the Essence of Music: The Con Espressione Manifesto.
ACM Transactions on Intelligent Systems and Technology 8(2), Article 19.

KAZALO / TABLE OF CONTENTS

Zbornik 20. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2017

Zvezek C

Proceedings of the 20th International Multiconference

INFORMATION SOCIETY – IS 2017

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Uredila / Edited by

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

9. oktober 2017 / 9th October 2017
Ljubljana, Slovenia

PREDGOVOR

Tehnologije, ki se ukvarjajo s podatki so v devetdesetih letih močno napredovale. Iz prve faze, kjer je šlo predvsem za shranjevanje podatkov in kako do njih učinkovito dostopati, se je razvila industrija za izdelavo orodij za delo s podatkovnimi bazami, prišlo je do standardizacije procesov, povpraševalnih jezikov itd. Ko shranjevanje podatkov ni bil več poseben problem, se je pojavila potreba po bolj urejenih podatkovnih bazah, ki bi služile ne le transakcijskem procesiranju ampak tudi analitskim vpogledom v podatke – pojavilo se je t.i. skladiščenje podatkov (data warehousing), ki je postalo standarden del informacijskih sistemov v podjetjih. Paradigma OLAP (On-Line-Analytical-Processing) zahteva od uporabnika, da še vedno sam postavlja sistemu vprašanja in dobiva nanje odgovore in na vizualen način preverja in išče izstopajoče situacije. Ker seveda to ni vedno mogoče, se je pojavila potreba po avtomatski analizi podatkov oz. z drugimi besedami to, da sistem sam pove, kaj bi utegnilo biti zanimivo za uporabnika – to prinašajo tehnike odkrivanja znanja v podatkih (data mining), ki iz obstoječih podatkov skušajo pridobiti novo znanje in tako uporabniku nudijo novo razumevanje dogajanj zajetih v podatkih. Slovenska KDD konferenca pokriva vsebine, ki se ukvarjajo z analizo podatkov in odkrivanjem znanja v podatkih: pristope, orodja, probleme in rešitve.

Dunja Mladenić, Marko Grobelnik

FOREWORD

Data driven technologies have significantly progressed after mid 90's. The first phases were mainly focused on storing and efficiently accessing the data, resulted in the development of industry tools for managing large databases, related standards, supporting querying languages, etc. After the initial period, when the data storage was not a primary problem anymore, the development progressed towards analytical functionalities on how to extract added value from the data; i.e., databases started supporting not only transactions but also analytical processing of the data. At this point, data warehousing with On-Line-Analytical-Processing entered as a usual part of a company's information system portfolio, requiring from the user to set well defined questions about the aggregated views to the data. Data Mining is a technology developed after year 2000, offering automatic data analysis trying to obtain new discoveries from the existing data and enabling a user new insights in the data. In this respect, the Slovenian KDD conference (SiKDD) covers a broad area including Statistical Data Analysis, Data, Text and Multimedia Mining, Semantic Technologies, Link Detection and Link Analysis, Social Network Analysis, Data Warehouses.

Dunja Mladenić, Marko Grobelnik

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Dunja Mladenić

Marko Grobelnik

ANOTATING DOCUMENTS WITH RELEVANT WIKIPEDIA CONCEPTS

Janez Brank, Gregor Leban, Marko Grobelnik
Artificial Intelligence Laboratory
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773778; fax: +386 1 4251038
e-mail: {janez.brank,gregor.leban,marko.grobelnik}@ijs.si

ABSTRACT
We describe an efficient approach for annotating a document with relevant concepts from the Wikipedia. A pagerank-based method is used to identify a coherent set of relevant concepts considering the input document as a whole. The proposed approach is suitable for parallel processing and can support any language for which a sufficiently large Wikipedia is available.

1 INTRODUCTION

Recent years have seen a growth in the use of semantic technologies. However, in many contexts we still deal with largely unstructured textual documents that lack explicit semantic information such as might be required for further processing with semantic technologies. This leads to the problem of semantic annotation or semantic enrichment as an important preparatory step before further processing of a document. Given a document and an ontology covering the domain of interest, the challenge is to identify concepts from that ontology that are relevant to the document or that are referred to by it, as well as to identify specific passages in the document where the concepts in question are mentioned.

A specific type of semantic annotation, known as *wikification*, involves using the Wikipedia as a source of possible semantic annotations [1][2]. In this setting, the Wikipedia is treated as a large and fairly general-purpose ontology: each page is thought of as representing a concept, while the relations between concepts are represented by internal hyperlinks between different Wikipedia pages, as well as by Wikipedia's category memberships and and cross-language links.

The advantage of this approach is that the Wikipedia is a freely available source of information, it covers a wide range of topics, has a rich internal structure, and each concept is associated with a semi-structured textual document (i.e. the contents of the corresponding Wikipedia article) which can be used to aid in the process of semantic annotation. Furthermore, the Wikipedia is available in a number of languages, with cross-language links being available to identify pages that refer to the same concept in different languages, thus making it easier to support multilingual and cross-lingual annotation.

The remainder of this paper is structured as follows. In Section 2, we present the pagerank-based approach to wikification used in our wikifier. In Section 3, we describe our implementation and present some experimental evaluaton. Section 4 contains conclusions and a discussion of possible future work.

2 PAGERANK-BASED WIKIFICATION

The task of wikifying an input document can be broken down into several closely interrelated subtasks: (1) identify phrases (or words) in the input document that refer to a Wikipedia concept; (2) determine which concept exactly a phrase refers to; (3) determine which concepts are relevant enough to the document as a whole that they should be included in the output of the system (i.e. presented to the user).

We follow the approach described by Zhang and Rettinger [1]. This approach makes use of the rich internal structure of hyperlinks between Wikipedia pages. A hyperlink can be thought of as consisting of a source page, a target page, and the *link text* (also known as the *anchor text*). If a source page contains a link with the anchor text *a* and the target page *t*, this is an indication that the phrase *a* might be a reference to (or representation of) the concept that corresponds to page *t*. Thus, if the input document that we're trying to wikify contains the phrase *a*, it might be the case that this occurrence of *a* in the input document also constitutes a *mention* of the concept *t*, and the concept *t* is a *candidate annotation* for this particular phrase.

2.1 Disambiguation

In the Wikipedia, there may be many different links with the same anchor text *a*, and they might not all be pointing to the same target page. For example, in the English-language Wikipedia, there are links with *a* = "Tesla" that variously point to pages about the inventor, the car manufacturer, the unit in physics, a band, a film, and several other concepts.

Thus, when such a phrase *a* occurs in an input document, there are several concepts that can be regarded as candidate annotations for that particular mention, and we have to determine which of them is actually relevant. This is the problem of *disambiguation*, similar to that of word sense disambiguation in natural language processing.

There are broadly two approaches to disambiguation, local and global. In the local approach, each mention is disambiguated independently of the others, while the global approach aims to treat the document as a whole and disambiguate all the mentions in it as a group. The intuition behind the global approach is that the document that we're annotating is about some topic, and the concepts that we use as annotation should be about that topic as well. If the document contains many mentions that include, as some of their candidate annotations, some car-related concepts, this makes it more likely that we should treat the mention of "Tesla" as a reference to Tesla the car manufacturer as opposed to e.g. a reference to Nikola Tesla or to Tesla the

rock band.

2.2 The mention-concept graph

To implement the global disambiguation approach, our Wikifier begins by constructing a *mention-concept graph* for the input document. (Some authors, e.g. [2], refer to this as a *mention-entity* graph, but we prefer to use the term "mention-concept graph" as some of the Wikipedia pages do not necessarily correspond to concepts that we usually think of as entities, and our wikifier does not by default try to exclude them.) This can be thought of as a bipartite graph in which the left set of vertices corresponds to mentions and the right set of vertices corresponds to concepts. A directed edge $a \to c$ exists if and only if the concept c is one of the candidate annotations for the mention a (i.e. if there exists in the Wikipedia a hyperlink with the anchor text a and the target c). A transition probability is also assigned to each such edge, $P(a \to c)$, defined as the ratio [number of hyperlinks, in the Wikipedia, having the anchor text a and the target c] / [number of hyperlinks, in the Wikipedia, having the anchor text a].

This graph is then augmented by edges between concepts, the idea being that an edge $c \to c'$ should be used to indicate that the concepts c and c' are "semantically related", in the sense that if one of them is relevant to a given input document, the other one is also more likely to be relevant to that document. Following [1], the internal link structure of the Wikipedia is used to calculate a measure of semantic relatedness. Informally, the idea is that if c and c' are closely related, then other Wikipedia pages that point to c are likely to also point to c' and vice versa. Let L_c be the set of Wikipedia pages that contain a hyperlink to c, and let N be the total number of concepts in the Wikipedia; then the semantic relatedness of c and c' can be defined as

$$SR(c, c') = 1 - [\log(\max\{|L_c|, |L_{c'}|\}) - \log|L_c \cap L_{c'}|] / [\log N - \log(\min\{|L_c|, |L_{c'}|\})].$$

In the graph, we add an edge of the form $c \to c'$ wherever the semantic relatedness SR(c, c') is > 0. The transition probability of this edge is defined as proportional to the semantic relatedness: $P(c \to c') = SR(c, c') / \Sigma_{c''} SR(c, c'')$.

This graph is then used as the basis of calculating a vector of pagerank scores, one for each vertex. This is done using the usual iterative approach where in each iteration, each vertex distributes its pagerank score to its immediate successors in the graph, in proportion to the transition probabilities on its outgoing edges:

$$PR_{new}(u) = \tau\, PR_0(u) + (1 - \tau)\, \Sigma_v\, PR_{old}(v)\, P(v \to u).$$

The baseline distribution of pagerank, PR_0, is used both to help the process converge and also to counterbalance the fact that in our graph there are no edges pointing into the mention vertices. In our case, $PR_0(u)$ is defined as 0 if u is a concept vertex; if u is a mention vertex, we use $PR_0(u) = z \cdot$ [number of Wikipedia pages containing the phrase u as the anchor-text of a hyperlink] / [number of Wikipedia pages containing the phrase u], where z is a normalization constant to ensure that $\Sigma_u\, PR_0(u) = 1$. We used $\tau = 0.1$ as the stabilization parameter.

The intuition behind this approach is that in each iteration of the pagerank calculation process, the pagerank flows into a concept vertex c from mentions that are closely associated with the concept c and from other concepts that are semantically related to c. Thus after a few iterations, pagerank should tend to accumulate in a set of concepts that are closely semantically related to each other and that are strongly associated with words and phrases that appear in the input document, which is exactly what we want in the context of global disambiguation.

2.3 Using pagerank for disambiguation

Once the pagerank values of all the vertices in the graph have been calculated, we use the pagerank values of concepts to disambiguate the mentions. If there are edges from a mention a to several concepts c, we choose the concept with the highest pagerank as the one that is relevant to this particular mention a. We say that this concept is *supported* by the mention a. At the end of this process, concepts that are not supported by any mention are discarded as not being relevant to the input document.

The remaining concepts are then sorted in decreasing order of their pagerank. Let the i'th concept in this order be c_i and let its pagerank be PR_i, for $i = 1, \ldots, n$. Concepts with a very low pagerank value are less likely to be relevant, so it makes sense to apply a further filtering step at this point and discard concepts whose pagerank is below a user-specified threshold. However, where exactly this threshold should be depends on whether the user wants to prioritize precision or recall. Furthermore, the absolute values of pagerank can vary a lot from one document to another, e.g. depending on the length of the document, the number of mentions and candidate concepts, etc. Thus we apply the user-specified threshold in the following manner: given the user-specified threshold value $\theta \in [0, 1]$, we output the concepts $c_1, \ldots, c_m$, where m is the least integer such that $\Sigma_{i=1..m}\, PR_i^2 \geq \theta\ \ \Sigma_{i=1..n}\, PR_i^2$. In other words, we report as many top-ranking concepts as are needed to cover θ of the total sum of squared pageranks of all the concepts. We use $\theta = 0.8$ as a broadly reasonable default value, though the user can require a different threshold depending on their requirements.

For each reported concept, we also output a list of the mentions that support it.

2.4 Treatment of highly ambiguous mentions

Our wikifier supports various minor heuristics and refinements in an effort to improve the performance of the baseline approach described in the preceding sections.

As described above, anchor text of hyperlinks in the Wikipedia is used to identify mentions in an input document (i.e. words or phrases that may support an annotation). One downside of this approach is that some words or phrases occur as the anchor text of a very large number of hyperlinks in the Wikipedia and these links point to a large number of different Wikipedia pages. In other words, such a phrase is highly ambiguous; it is not only unlikely to be disambiguated correctly, but also introduces noise into the mention-concept graph by introducing a large number of concept vertices, the vast majority of which will be completely irrelevant to the input document. This also slows down the annotation process

by increasing the time to calculate the semantic relatedness between all pairs of candidate concepts.

We use several heuristics to deal with this problem. Suppose that a given mention a occurs, in the Wikipedia, as the anchor text of n hyperlinks pointing to k different target pages, and suppose that n_i of these links point to page c_i (for $i = 1, \ldots, k$). We can now define the entropy of the mention a as the amount of uncertainty regarding the link target given the fact that its anchor text is a: $H(a) = -\Sigma_{i=1..k} (n_i/n) \log(n_i/n)$. If this entropy is above a user-specified threshold (e.g. 3 bits), we completely ignore the mention as being too ambiguous to be of any use. For mentions that pass this heuristic, we sort the target pages in decreasing order of n_i and use only the top few of them (e.g. top 20) as candidates in our mention-concept graph. A third heuristic is to ignore candidates for which n_i itself is below a certain threshold (e.g. $n_i < 2$), the idea being that if such a phrase occurs only once as the anchor text of a link pointing to that candidate, this may well turn out to be noise and is best disregarded.

Optionally, the Wikifier can also be configured to ignore certain types of concepts based on their Wikidata class membership. This can be useful to exclude from consideration Wikipedia pages that do not really correspond to what is usually thought of as entities (e.g. "List of…" pages).

Another heuristic that we have found useful in reducing the noise in the output annotations is to ignore any mention that consists entirely of stopwords and/or very common words (top 200 most frequent words in the Wikipedia for that particular language). For this as well as for other purposes the text processing is done in a case-sensitive fashion, which e.g. allows us to ignore spurious links with the link text "the" while processing those that refer to the band "The The".

2.5. Miscellaneous heuristics

Semantic relatedness. As mentioned above, the definition of semantic relatedness of two concepts, $SR(c, c')$, is based on the overlap between the sets L_c, $L_{c'}$ of immediate predecessors of these two concepts in the Wikipedia link graph. Optionally, our Wikifier can compute semantic relatedness using immediate successors or immediate neighbours (i.e. both predecessors and successors) instead of immediate predecessors. However, our preliminary experiments indicated that these changes do not lead to improvements in performance, so they are disabled by default.

Extensions to disambiguation. Our Wikifier also supports some optional extensions of the disambiguation process. As described above, the default behavior when disambiguating a mention is to simply choose the candidate annotation with the highest pagerank value. Alternatively, after any heuristics from section 2.4 have been applied, the remaining candidate concepts can be re-ranked using a different scoring function that takes other criteria besides pagerank into account. This is an opportunity to combine the global disambiguation approach with some local techniques. In general, a scoring function of the following type is supported:

$$score(c|a) = w_1 f(P(c|a)) PR(c) + w_2 S(c, d) + w_3 LS(c, a)$$

Here, a is the mention that we're trying to disambiguate, and c is the candidate concept that we're evaluating. $P(c|a)$ is the probability that a hyperlink in the Wikipedia has c as its target conditioned on the fact that it has a as its anchor text. $f(x)$ can be either 1 (the default), x, or $\log(x)$. $PR(c)$ is the pagerank of c's vertex in the mention-concept graph. $S(c, d)$ is the cosine similarity between the text of the input document d and of the Wikipedia page for the concept c. $LS(c, a)$ is the cosine similarity between the context (e.g. previous and next 3 words) in which a appears in the input document d, and the contexts in which hyperlinks with the target c appear in the Wikipedia. Finally, w_1, w_2, w_3 are weight constants. However, our preliminary experiments haven't shown any improvements from the addition of these heuristics, so they are disabled by default ($f(x) = 1$, $w_2 = w_3 = 0$) to save computational time and memory (storing the link contexts needed for the efficient computation of LS has turned out to be particularly memory intensive).

3 IMPLEMENTATION AND EVALUATION

3.1. Implementation

Our implementation of the approach described in the preceding section is running as a web service and can be accessed at http://wikifier.org. The approach is suitable for parallel processing as annotating one document is independent of annotating other documents, and any shared data used by the annotation process (e.g. the Wikipedia link graph, and a trie-based data structure that indexes the anchor text of all the hyperlinks) need to be accessed only for reading and can thus easily be shared by an arbitrary number of worker threads. This allows for a highly efficient processing of a large number of documents.

Our implementation currently processes on average more than 500,000 requests per day (the total length of input documents averages about 1.2 GB per day), including all the documents from the JSI Newsfeed service [3]. The output is used among other things as a preprocessing step by the Event Registry system [4]. The wikifier currently supports all languages in which a Wikipedia with at least 1000 pages is available, amounting to a total of 134 languages. Admittedly, 1000 pages is much too small to achieve an adequate coverage; however, about 60 languages have a Wikipedia with at least 100,000 pages, which is already enough for many practical applications.

Annotations are returned in JSON format and can optionally include detailed information about support (which mentions support each annotation), alternative candidate annotations (concepts that were considered as candidates during the disambiguation process but were rejected in favour of some other more highly scored concept), and WikiData/DbPedia class membership of the proposed annotations. Thus, the caller can easily implement any desired class-based postprocessing.

3.2. Evaluation

One way to evaluate wikification is to compare the set of annotations with a manually annotated gold standard for the same document(s). Performance can then be measured using

metrics from information retrieval, such as precision, recall, and the F_1-measure, which is defined as the harmonic mean of precision and recall. We used a manually annotated set of 1393 news articles that was made available from the authors of the AIDA system and was originally used in their experiments [2]. This manually annotated dataset excludes, by design, any annotations that do not correspond to named entities. Since our wikifier does not by default distinguish between named entities and other Wikipedia concepts, we have explicitly excluded non-entity concepts (based on their class membership in the WikiData ontology) from the output of our Wikifier for the purposes of this experiment. In addition to our wikifier, we obtained annotations from the following systems: AIDA [2], Waikato Wikipedia Miner [6], Babelfy [7], Illinois [8], and DbPedia Spotlight [9].

	Gold	JSI	AIDA	Waikato	Babelfy	Illinois	Spotlight
Gold	1.000	0.593	0.723	0.372	0.323	0.476	0.279
JSI		1.000	0.625	0.527	0.431	0.489	0.363
AIDA			1.000	0.372	0.352	0.434	0.356
Waikato				1.000	0.481	0.564	0.474
Babelfy					1.000	0.434	0.356
Illinois						1.000	0.376
Spotlight							1.000

Table 1: F_1 measure of agreement between the various wikifiers and the gold standard.

Table 1 shows the agreement not only between each of the wikifiers and the gold standard, but also between each pair of wikifiers (the lower left triangle of the matrix is left empty as it would be just a copy of the upper right triangle, since the F_1-measure is symmetric). As this experiment indicates, our wikifier ("JSI" in the table) performs slightly worse than AIDA but significantly better than the other wikifiers. Furthermore, it turns out that there is relatively little agreement between the different wikifiers, which indicates that wikification itself is in some sense a vaguely defined task where different people can have very different ideas about whether a particular Wikipedia concept is relevant to a particular input document (and should therefore be included as an annotation) or not, which types of Wikipedia concepts can be considered as annotations (e.g. only named entities or all concepts), etc. Possibly the level of agreement could be improved by fine-tuning the settings of the various wikifiers; in the experiment described above, default settings were used.

4 CONCLUSIONS AND FUTURE WORK

We have presented a practical and efficient approach to Wikification that requires no external data except the Wikipedia itself, that can deal with documents in any language for which the Wikipedia is available, and that is suitable for a high-performance, parallelized implementation.

The approach presented here could be improved along several directions. One significant weakness of the current approach concerns the treatment of minority languages. When dealing with a document in a certain language, we need hyperlinks whose anchor text is in the same language if we are to identify mentions in this input document. Thus, if the document is in a language for which the Wikipedia is not available at all, it cannot be wikified using this approach; and similarly, if the Wikipedia is available in this language but is small, with a small amount of text, low number of pages, and generally poor coverage, the performance of wikification based on this will be low. One idea to alleviate this problem would be to optionally allow a second stage of processing, in which Wikipedias in languages other than the language of the input document would also be used to identify mentions and provide candidate annotations. This might improve coverage especially of concepts that are referred to by the same words or phrases across multiple languages, as is the case with some types of named entities. For the purposes of pagerank-based disambiguation in this second stage, a large common link-graph would have to be constructed by merging the link-graphs of the Wikipedias for different languages. This can be done by using the cross-language links which are available in the WikiData ontology, providing information about when different pages in different languages refer to the same concept.

Another interesting directon for further work would be to try incorporating local disambiguation techniques as a way to augment the current global disambiguation approach. When evaluating whether a mention *a* in the input document refers to a particular concept *c*, the local approach would focus on comparing the context of *a* to either the text of the Wikipedia page for *c*, or to the context in which hyperlinks to *c* occur within the Wikipedia. Preliminary steps taken in this direction in Sec. 2.5 did not lead to improvements in performance, but this subject is worth exploring further. Instead of the bag-of-words representation of contexts, other vector representations of words could be used, e.g. word2vec [5].

Acknowledgments

This work was supported by the Slovenian Research Agency as well as the euBusinessGraph (ICT-732003-IA) and EW-Shopp (ICT-732590-IA) projects.

References

[1] L. Zhang, A. Rettinger. *Final ontological word-sense-disambiguation prototype.* Deliverable D3.2.3, xLike Project, October 2014.

[2] J. Hoffart, M. A. Yosef, I. Bordino, *et al.* Robust disambiguation of named entities in text. *Proc. of the 2011 Conf. on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, 2011, pp. 782–792.

[3] M. Trampuš, B. Novak. Internals of an aggregated web news feed. *Proc. SiKDD 2012*.

[4] G. Leban, B. Fortuna, J. Brank, M. Grobelnik. Event registry: Learning about world events from news. *Proc. of the 23rd Int. Conf. on the World Wide Web* (WWW 2014), pp 107–110.

[5] T. Mikolov, K. Chen, G. Corrado, J. Dean. *Efficient estimation of word representations in vector space.* Arxiv.org, 2013.

[6] D. Milne, I. H. Witten. An open-source toolkit for mining Wikipedia. *Artificial Intelligence*, 194:222–239 (January 2013).

[7] A. Moro, A. Raganato, R. Navigli. Entity linking meets word sense disambiguation: A unified approach. *Trans. of the Assoc. for Comp. Linguistics*, 2:231–234 (2014).

[8] L. Ratinov, D. Roth, D. Downey, M. Anderson. Local and global algorithms for disambiguation to Wikipedia. *Proc. of the 49th Annual Meeting of the Assoc. for Comp Linguistics: Human Language Technologies* (2011), pp. 1375–84.

[9] J. Daiber, M. Jakob, C. Hokamp, P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. *Proc. of the 9th Int. Conf. on Semantic Systems*, 2013.

Impact of News Events on the Financial Markets

Miha Torkar
Jožef Stefan International Postgraduate School
and
Artificial Intelligence Laboratory,
Jožef Stefan Institute,
Jamova 39, 1000 Ljubljana,
Slovenia
miha.torkar@ijs.si

Dunja Mladenič
Jožef Stefan International Postgraduate School
and
Artificial Intelligence Laboratory,
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana,
Slovenia
dunja.mladenic@ijs.si

ABSTRACT

In this work we investigate how news events can be used to predict the financial markets. Namely we built a time series model that includes features obtained from the news and investigated whether the changes in volume of traded shares can be predicted more accurately with this information. The time series model that was built is of an ARMA-GARCH type, because we wanted to account for any clustering of the volatility that is normal for the financial markets. The models were evaluated with the Akaike and Bayesian Information Criterion, while also being compared to the baseline model that did not include any features from the news. Overall our results show that there is an improvement in the model when the information from the news is used and hence show a promising avenue for future research work.

1. INTRODUCTION

The predictability of future movements of financial markets is a well researched area in the literature and offers many interesting theories. Financial institutions and investors have been using various sources of information in order to increase the accuracy of their predictions and consequently outperform others. There are several approaches to building the best predictive model, but in general we can divide them into two categories: technical analysis and fundamental analysis. On one side we have models that are based on the historical market data and believe that the past movements will repeat themselves. This is the so called technical analysis approach to modelling markets and believes that an experienced observer can detect the repetition of a pattern from a graph of market data. The effectiveness of this approach was tested by [9].This approach however does not offer any reasons why market movements would repeat, and in order to incorporate more fundamental believes, a second approach named fundamental analysis was developed. These models use data that is available from multiple sources, which ranges from company's balance sheet data, financial market data like company's index, financial data about government activities to data about political or geographical circumstances presented in news.

From the variety of sources for fundamental data, we will focus on how news can be used in modelling financial markets. There have been various studies on this topic, which differ by the extent of the analysis of textual data describing news story. Initially research focused on the impact of frequency of news stories on the market movements (see [6] or [7]). In these papers authors found some correlation between increased number of published news stories and larger market movements. To extend this approach, researchers also analysed the content of the news stories, which led to determining the sentiment of the news articles and consequently determining the impact on the market on the basis of whether it is positive (upward trend predicted) or vice versa (see [3] or [4]). Our approach is similar to the second one, but instead of determining sentiment for each news event, we use the effect that the past similar events had on the market as a proxy for the impact of the current event.

We define an event as a collection of news articles from different sources about the same story in the news. From this collection of articles we will be able to extract the topic, date, location, social score (how trending was this event on social media) and all of the entities involved in the event, which will add to the complexity of our dataset. One possible source of such dataset is a system called Event Registry (see [5]), which automatically extracts events from news articles. Using this type of data source is novel for this research area and hence serves as an additional contribution from this work.

In order to see the impact that these events had on the financial market, we looked at how the volume of traded shares changes on the days of the event. We obtained several features from the news events and checked whether they would allow us to improve the time series model for the volume change.

2. DATA

For the historical market data of the company, we collected the following values (prices): Open, High, Low, Close, Volume. In addition to the market values of a company we also used the value of the market volatility index VIX (closing price). Secondly with the use of the Event Registry system we obtained all of the news events relating to the company. The general description above was intentional as it can be applied to any publicly listed company, but for our specific example we will use data about investment bank Goldman Sachs (GS). In Figure 1 we present the dataset we will be modelling and predicting, where it can be seen that there are large spikes at certain time periods. Moreover the volume change graph demonstrates that we have clusters of periods with high volatility.

2.1 Data Description

Our dataset spanned from 2.12.2013 to 30.12.2016, which offered us 777 trading days on which we were able to collect historical market data. On the other hand the number of news events that occurred in that time period was significantly higher. When we singled out events using 50 as the relevance threshold we obtained 4336 events (details of how Event Registry calculates the weights of concepts in

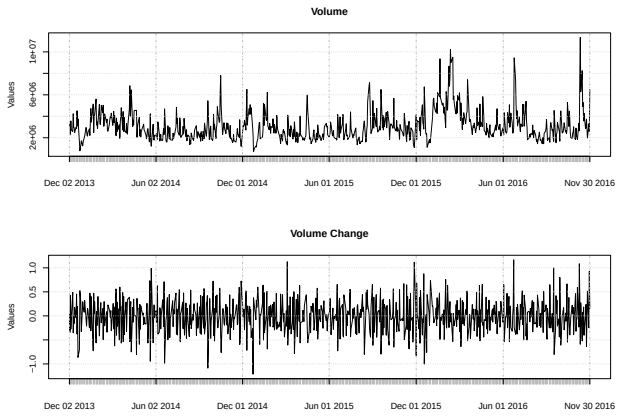

Figure 1: Volume Dataset

the event are presented in [5]). For testing purposes we split our dataset into two parts, where we allocated 757 observations for training and testing, while we used the remaining 21 observations for the out of sample prediction.

For each event we also obtained related events to it. These related events are obtained by computing the TF-IDF weights on the concepts present in the event and then by using cosine similarity measure other events with similar concept weights are found (see [5]). These past similar events formed a crucial dataset, because we could link them to the market movements and deduce their impact.

2.2 Data preprocessing

In order to measure the impact of the event we will be considering the change in volume that occurs between the closing value of today and the day after the event. Specifically we will be analysing and predicting the value of

$$(\text{Volume Change})_t \equiv VC_t = \frac{V_{t+1} - V_t}{V_t}, \tag{1}$$

where V_t is the value of volume at time t. In this formulation we used future values, so that we can observe the impact an event has on future volume. It should be noted that this value is calculated in the same manner as that of the returns of shares, which we will also use in our analysis. The formula is identical except that we replace values of volume with those of closing price P_t. Hence we write

$$(\text{Returns})_t \equiv r_t = \frac{P_{t+1} - P_t}{P_t}. \tag{2}$$

Changes in the volatility index VIX were calculated with the same formula. Additionally we also added rolling 5 and 10 day moving average of volume change to the feature set, so that we would have another measure of impact an event has on value. Hence the complete list of all of the features that were obtained from the stock market is:

- Returns
- Volume Change
- Open
- High
- Low
- Close
- Volume
- VIX Close
- VIX Change
- Rolling mean 5 days
- Rolling mean 10 days
- Rolling EMA 5 days
- Rolling EMA 10 days

In order to reduce noise in the dataset of events we selected only the most important ones. Namely we set a limit, which determined the lower boundary for the relevance of the events to the given company. Hence if an event was not relevant enough we discarded it. This naturally raises the issue of selecting the appropriate boundary for the relevance of the events and after testing several values we selected 89 out of 100. After this we were left with 424 events.

What should be noted here is that many events occurred on the days when the markets were closed (weekends, holidays), so they had to be linked with the next possible trading day that followed. Another issue was also that in many cases multiple events occurred on the same day and hence we were not able to isolate the effect of a single event, but rather looked at the average effect of all the events on a specific day. Therefore we are also relying on the fact that there are no duplicates in the our dataset.

3. METHODS

3.1 Predictions from Similar Events

In order to build our time series model, which uses data from news events, we examined the effects of past similar events on the market for each new event separately. This was done by first extracting the market information on the days of the similar events and combining them with the additional information about the event (social score, relevance to the company, correlation to current event, number of articles). On average each event had between 5-17 similar events from our dataset.

Illustration of this processes for an event E *"Goldman Profit Rises 74% as Bond Trading Beats Estimates"* with two similar events (SE) is represented in Figure 2. On the two days of the SE_1 and SE_2 we then collect all of the available stock market values for the company and index VIX. It should also be noted that the time difference between event and past similar events is not limited or predetermined. In this case we have events that are years apart.

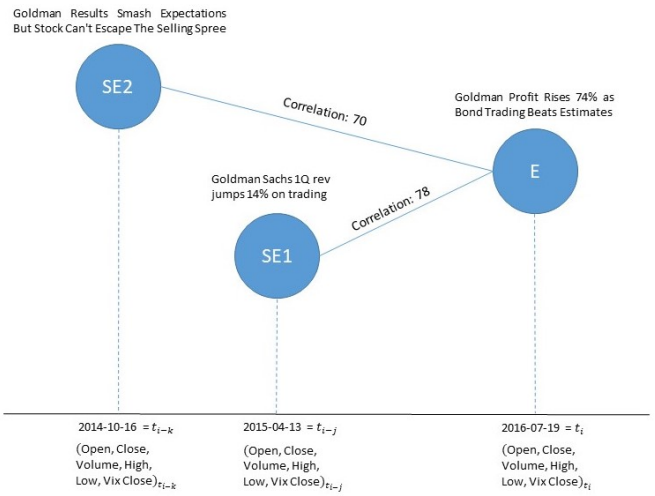

Figure 2: Similar Events

Once the dataset of market impacts from past similar events was obtained an Ordinary Least Squares (OLS) model was fit to the dataset. In the Figure 2 this would be all of the market variables available at times t_{i-k} and t_{i-j}. This

dataset then allowed us to use data from the market and current event to form a prediction of what the future market movement could be. This procedure was done for both volume change and returns. Several choices of external regressors that were used in OLS were tested, where the main issue arose when there were fewer similar events than features to be fit. Namely we fitted two models, one with all of the market information available and one where only Returns, VIX Change and concept weight were used to predict Volume Change. Similarly two models were fit to predict Returns, where Volume Change, VIX Change and concept weights were used as features in the second one. In addition we also calculated the average of volume changes and returns from dates of similar events and used it as a feature. In order to take into account the correlation to the similar events we created an extra feature that represented the value of average of volume changes and returns multiplied by the normalized correlation (correlation/100). All together the following features were added to the model (all of which are predictions from similar events):

- Number of Events that day
- Returns (all regressors)
- Returns (selected regressors)
- Volume Change (all regressors)
- Volume Change (selected regressors)
- Average Returns
- Average Returns $*$ Correlation
- Average Volume Change
- Average Volume Change $*$ Correlation

With the OLS model trained on the dataset of similar events we were then able to predict the impact of the original event on the volume change. It should also be noted that the OLS model was selected in order to avoid over fitting an ARMA type model. The predictions that were obtained in this way were then used in the next step when we were building our regression model.

3.2 ARMA-GARCH MODEL

The main model in our analysis will be of the ARMA-GARCH type, because this formulation allows us to capture the effects of values from previous periods and account for clustering in volatility.

The ARMA model is in its general form specified as:

$$\begin{aligned} X_t &= \mu_t + Z_t, \\ \mu_t &= \sum_{i=0}^{p} \alpha_i X_{t-i} + \sum_{j=1}^{q} \omega_j Z_{t-j} \\ Z_t &= \sigma\epsilon \Rightarrow Z \sim N(0, \sigma^2) \end{aligned} \tag{3}$$

where X_t is the target variable at time t, μ_t is the equation for the mean at time t,$\epsilon \sim N(0,1)$ is an iid normally distributed noise term and σ is the variance at time t. The first step in modelling is to determine the best values for (p, q) according to the evaluation criteria.

The above model however assumes that the variance (σ) is constant. This assumption is dropped due to the clustering in the volume change (periods of high changes are followed by lower ones). This type of models are called Generalised Autoregressive Conditional Heteroskedasticity (GARCH) models ([2]) and have been shown across literature to improve the models (see [8]). They allow us to model $\sigma = \sigma_t$ by an additional non linear model. Namely general formulation of the problem is the following:

$$\sigma_t^2 = \beta_0 + \sum_{i=1}^{r} \beta_i Z_{t-i}^2 + \sum_{j=1}^{s} \gamma_j \sigma_{t-j}^2 \tag{4}$$

So the current value of the variance also depends on the previous values of the variance. From this general formulation one has to determine the value of (r, s) that is most suitable for the model. Finally we can add external regressors (features) to the equations 3 as an additional sum term in the first line of the equation.

3.3 Evaluation Criteria

In order to asses which model was most suitable for the given dataset we have used a variety of different tests. In order to be able to assume that the time series is stationary we first ran Augmented Dicky Fuller test and KPSS test. In both cases the p-value was below and above the 5% threshold respectively. To differentiate between variety of different model we analysed the performance of each model by the Akaike information criterion (AIC) and Bayesian information criterion (BIC), which are the classical information criteria used across literature that also penalize model for high complexity ([1]). Finally in order to determine the significance of the features we will also build a model without the predictions from the similar news events. This will serve as our baseline model and the difference in performance will be then the main measure of how significant these features are.

4. RESULTS

Our first step in determining which features are significant for our analysis was running a t-test selection procedure for all regressors. From this analysis it was determined that only the following features were relevant for our model:

- VIX Close price
- Rolling mean 10 days
- Rolling mean 5 days
- Rolling EMA 5 days
- Rolling EMA 10 days
- Prediction of Returns

Hence our first finding was that the predictions that we obtained from the similar events were significant, but instead of using the predicted volume changes, the predicted returns turned out to be more significant for our model.

With this set of regressors the best ARMA model was of order (2,2), so two lagged terms in both variables were included. This model preformed according to the evaluation criteria shown in the table 1. As mentioned before in order to capture clustering in the dataset we modelled the variance term even further with a GARCH type model. Again the same evaluation criteria were used and a grid search was preformed to find the best coefficients (r, s) for order of the model. This resulted in a GARCH (5,1) model, with the evaluation criteria presented in the same table 1.

Comparison of results yields what improvements have been made by including the feature. This table also serves as a comparison to the baseline model. Since the AIC is merely a heuristic, differences between its values are important. So the predicted feature from similar events improves AIC value by 10.22, while the improvement of our final model is 32.69.

One way of interpreting this difference is also in terms of relative likelihood, which is defined as $exp((AIC_{min} - AIC_i)/2)$ for model i, where AIC_{min} represents the lowest AIC value from all models. Hence the baseline model is $7.97 * 10^{-8}$ times as probable as the best model to minimize the information loss, while the ARIMA(2,2) with feature is $1.06 * 10^{-5}$ times as probable. However we can see that due to additional complexity of our final model, BIC criterion is actually the highest for our chosen model. Hence an optimisation by the BIC criterion would result in a different model.

	AIC	BIC
Baseline	-925.57	-874.66
ARMA(2,2)	-935.35	-879.81
ARMA(2,2)-GARCH(5,1)	-958.26	-869.96

Table 1: Evaluation criteria train set

It should be noted that we have also tested the assumption about the distribution of the standardized residuals. So instead of using normal distribution we fitted the model with student t distribution with and without skew. Some minor improvements in AIC value were obtained, but not really significant. Hence we kept the distribution as normal.

Final test included out of sample predictions, where our best chosen model was ARMA(2,2)-GARCH(5,1). The time period for prediction was 1 month (December 2016) with 21 trading days. We also calculated errors when making these predictions, where our best model from above scored a value of 0.1344944 for Mean Absolute Error and 0.1565652 for Root Mean Squared Error. Figure 3 shows how the model predicted future values, where the middle line represent predictions for the mean value. Additionally the plot also included intervals of upper and lower 95 and 80 quantile range.

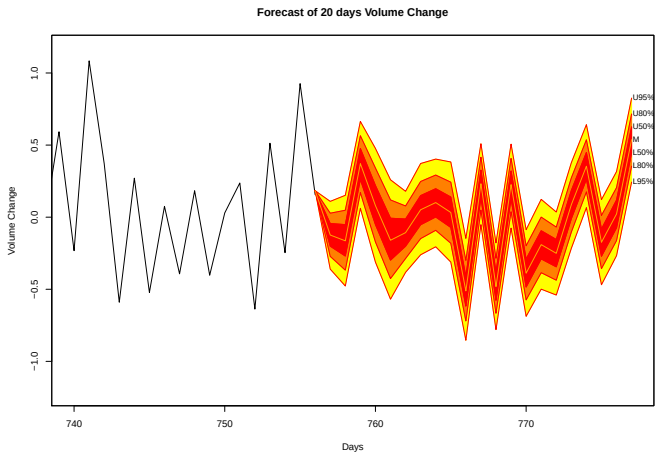

Figure 3: Out of Sample Forecast

5. CONCLUSION

In this paper we investigated one possible approach of determining the impact that news have on the financial markets. This is a growing research area since nowadays any model that wishes to capture market dynamics has to account for the effect of world events. In order to extend previous work in this area, we demonstrated how a more complex text data can be used to obtained relevant features for modelling changes in financial markets. Our data consisted of news events that were automatically extracted from the news articles. For each event we then collected past similar events and observed how the market reacted to those events. On the basis of these reactions we built various features vectors that helped us improve our model. Results show that when predicting change in volume, the predicted returns from similar events served as a useful feature. Additionally we tested the performance of our improved times series model on the out of sample dataset for the period of 1 month. Future work will be done in this direction, where we will look for further similarities between news events and possibly obtain new features that could be used for modelling financial markets.

6. ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675044.

7. REFERENCES

[1] H. Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1998.

[2] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307 – 327, 1986.

[3] X. Ding, Y. Zhang, T. Liu, and J. Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2327–2333. AAAI Press, 2015.

[4] R. Fehrer and S. Feuerriegel. Improving Decision Analytics with Deep Learning: The Case of Financial Disclosures. *ArXiv e-prints*, August 2015.

[5] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Event registry: Learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 107–110, New York, NY, USA, 2014. ACM.

[6] M. L. Mitchell and J. H. Mulherin. The impact of public information on the stock market. *The Journal of Finance*, 49(3):923–950, 1994.

[7] D. Shen, W. Zhang, X. Xiong, X. Li, and Y. Zhang. Trading and non-trading period internet information flow and intraday return volatility. *Physica A: Statistical Mechanics and its Applications*, 451:519 – 524, 2016.

[8] R. Tsay. *An Introduction to Anaysis of Financial Data with R*. John Wiley & Sons, New Jersey, 2013.

[9] H. Yu, G. V. Nartea, C. Gan, and L. J. Yao. Predictive ability and profitability of simple technical trading rules: Recent evidence from southeast asian stock markets. *International Review of Economics & Finance*, 25:356 – 371, 2013.

Challenges in media monitoring of worldwide news sources to support public health

Joao Pita Costa * **, Flavio Fuart *, Marko Grobelnik *, Gregor Leban * and Evgenia Belyaeva ***

***Abstract*—Real-time global media monitoring is nowadays an essential resource to public health. Multilingual capabilities can enrich this potential allowing a worldwide overview based on online news sources, blog posts or social media. In this paper we propose research topics related with the exploration text mining tools used to provide real-time global media monitoring in the context of health. We aim to understand how media can contribute to a better overview of health related and well being matters. With it we shall also identify open research questions that motivate further technological development to better fit the needs, interests and workflow of public health professionals.**

I. Introduction

Dealing with media (written online media in particular) has several issues, one of which is a lack of common publishing standards. Another issue is related to the global nature of the service: the mere fact of a system being universal, requires a system that can manage a variety of languages, possibly hundreds of them. This creates issues, as todays language technologies can deal only with words and sentences, highlighting the need to bridge the gap from simple textual representation towards semantic representation, where we would want to understand the semantic and conceptual aspect of the textual content and not just lexical (words and phrases) and syntactical (sentences).

Considerable research and commercial activity in the past period has led to a development of high performance online news monitoring systems and accompanying tools, methods and models. For example, a complete cross-lingual news processing pipeline consisting of the following components has achieved good results in both research and commercial usage scenarios: NewsFeed system [16] to monitor, gather an produce clear-texts of online (HTML) news articles; Sentiment Detection module [17]; Enrycher module for text annotation [7]; Wikifier for advanced text categorization [2]; Cross-lingual document linking [11]; Document clustering [1]; and Event Registry [9], an advanced news visualization and analysis tool. There is a relatively large offer of similar online news monitoring ("clipping") systems available, each of them with a distinctive set of features. Authors, however, are not aware, of a monitoring system that would implement a rich set of cross-lingual features as the online news processing pipeline mentioned above.

* Quintelligence, Ljubljana, Slovenia
** University of Rijeka, Croatia
*** Jozef Stefan Institute, Ljubljana, Slovenia

Category: H.4.0 Information Systems Applications: General reporting, statistical analysis, visualisation, networks
Keywords: Text mining, public health, news, media.

Fig. 1. The contribution of text mining efforts based on online multi-lingual news to public health raises several research questions. In the image above shows an ER visualisation module representing a real-time stream of news that permits us to explore the range of some of those questions.

According to the ECDC (European Centre for Disease Prevention and Control), the objective of epidemic intelligence is to produce timely, validated and actionable intelligence on events related to communicable diseases or of unknown origin that are of interest for public health and health authorities [4]. A similar definition is provided also by the WHO [18]. Part of this effort is to gather unofficial, unstructured and unverified information about those events, that are then later verified and analyzed by Public Health experts. Several ICT solutions to support these efforts have emerged. Most notable solutions used by authorities are GPHIN [3] (Global Public Health Intelligence Network), developed and operated by the Canadian Government, MedISys [15], developed and operated by the Joint Research Centre of the European Commission and Healthmap.org, a system developed by Boston Children's Hospital receiving external funding.

All those systems are multi-lingual to some extent, i.e. they monitor news in more than one language. However, it seems they do not leverage the usefulness of cross-lingual approaches to increase the quality of detected health events. Also, they seem no to use Wikipedia, which is nowadays the biggest freely available knowledge base, to extract meaningful information from news articles.

In this paper we aim to identify research questions, related to features highlighted above (cross-linguality, wikification, among others), that can contribute to the appropriate software development serving the needs of Public Health professionals. Part of the work presented in it was developed in the context of the European Union research project MIDAS, under the program Horizon 2020.

II. Collecting the data

Online news media sources represent a reliable and structured near-real time stream of a heterogeneous, multilingual text documents that describe real-word events. A range of services offers the aggregation of both social media and online news, following the uptake of news publishing through the latter channel. Several media news aggregators provide web crawlers for information extraction and media monitoring aiming for newsworthy stories. In order to extract meaningful information for a particular application domain, automatic event detection mechanisms exist allowing us to measure the media impact of public health awareness rising campaigns, health related news coverage and bias across different outlets. Those also include sentiment detection and cross-lingual linking of documents applied to news sources. In the public health domain, these approaches have been used to detect disease outbreaks and other public health threats (e.g. monitoring of international social and sports events, anti-vaccination campaigns, among others).

Among the available news aggregator and analysis services that provide some level of access to online news streams we highlight the NewsFeed system (available at newsfeed.ijs.si). It is a real-time aggregated stream of semantically enriched news articles tracking over RSS-enabled global media sources worldwide. In particular, the online media monitoring system NewsFeed currently monitors around 900.000 RSS news feeds (800.000 web sites) and collects between 350.000 and 600.000 articles per day, assuming an article archive available since May 2008. Using the current API it is possible to get annotated articles since June 2013. Currently, about 50% of all articles are in English. All languages present in Wikipedia are included, but are covered in respect of quality, volume and extent of analysis performed [8] to differing degrees. For research purposes, authors where granted access to the NewsFeed system, available at newsfeed.ijs.si. The data is accessed through a HTTP API and the result is provided in XML format. Additional metadata can be obtained through the API: named entities, concepts, categories, mentioned places and similar [14]. Furthermore, this technology could be used for additional data annotation and analysis tailored to public health applications (e.g. the annotation with *wikifier* as later discussed in Section III).

The news monitoring software EventRegistry (ER) (available at www.eventregistry.org) feeds on Newsfeed, tracking over 100,000 global media sources in near-real-time, operating across 100 languages, and aggregating global media content in a semantically meaningful way. It collects over 300,000 news articles on average per day and arranges them into events, events are further connected into story-lines which enable tracking of evolving topics [9]. Since ER covers most of the global media reporting on any topic, it can be used also to track topics like health and well-being on different levels of resolution from small local issues, up to the higher level country issues and global trends.

III. Automatic annotation

The complexity of identification of entities makes the automatic multilingual text analysis a difficult task. Wikifier

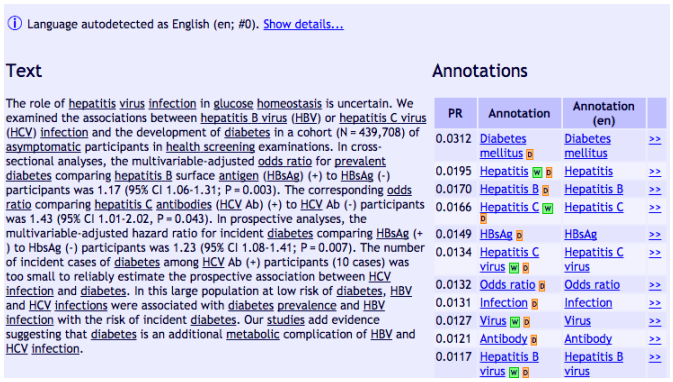

Fig. 2. The automatic annotation tool Wikifier used to enrich a WHO article on vaccination taken as an example. It is copied to the *Text* field and has several underlined words corresponding to the identified Wikipedia concepts. When hovering the emphD icon on disease name, we notice that this article is mostly about Yellow Fever, we see that yellow fever is a disease and we see that Kinshasa is a settlement. If we further explore the DBpedia entry the area, number of inhabitants, country and other information in available.

(available at wikifier.ijs.si) takes profit of Wikipedia, the biggest open, online and up-to-date knowledge repository on the internet, to annotate text and link it to relevant knowledge resources. Wikifier allows for annotating large quantities of free text in a very short time. The type of analysis provided permits us to identify trends (e.g. health related lifestyles) or ask questions like: *Provide me all texts (articles) that are about vaccination campaigns in Africa in cities with population of more than 2 million people*. Figure 2 shows the output of the automatic annotation of the abstract of a recent article on Cholera. The top ten concepts annotated include ”Vibrio cholerae”, ”Fresh water” or ”Bacteria”.

By being constructed over the knowledge-base of Wikipedia, it is essential that Wikipedia coverage on health topics is of high quality which is itself an open question. It is also another open question of whether an extension of Wikipedia can capture all of the concepts covered by MeSH.

IV. From global to local media monitoring

In the context of Public Health, a graphical dashboard can provide contribution to the real-time monitoring of the global health by allowing to a continuous observation of medical and well-being issues, limiting to the presentation of articles/items related to those topics (a possible health dashboard is available at [6]). Such a dashboard presents the incoming multi-lingual health related media content published somewhere in the world. That allows the health professional to have an overview of what is happening globally at any given moment in time (cf. [5]). This is a convenient way to observe global health monitoring by using the ER system introduced above in Section II. Its key feature is to be able to observe health issues across many languages and in temporal detail, over a variety of scales, which is what most other systems have difficulties [10]. The image in Figure 1 shows a snapshot of the dashboard with health related events on July 20th, 2017.

The event of the Zika outbreak is identified immediately after the collection of news articles that report about it. With

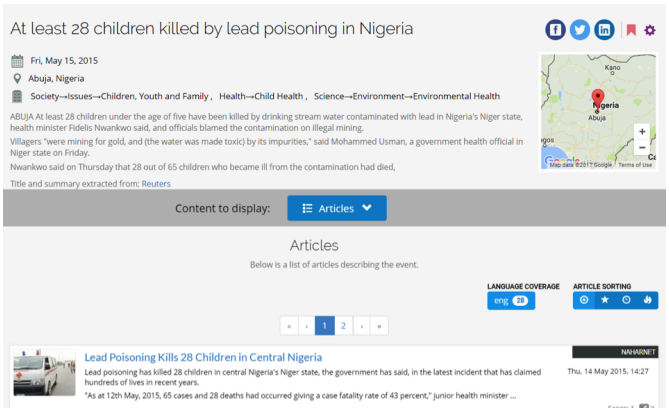

Fig. 3. ER screenshot showing the event of "lead poisoning" in Nigeria on May 12, 2015. It includes date, location, categories, number of languages covered and social media (Twitter) count.

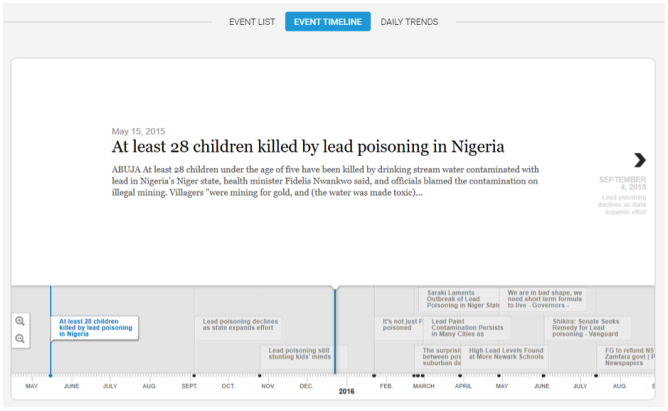

Fig. 4. ER screen-shot showing story-line developed after the event of "lead poisoning" in Nigeria on May 12, 2015. It describes below the event a timeline of related news.

it, the health professional can explore the evolution of the news publishers awareness of the epidemics in time by looking at the related news articles represented in a world map, as they were identified or updated during a selected period of time. ER can find articles and events related to a particular entity, topic, date, location or category, as well as measure their impact on social media (in particular, Twitter). Moreover, its cross-lingual capabilities allow to consider events where the news appears only in e.g. Chinese or some unknown language where the news never came into English speaking space. This is of particular interest when considering the monitoring of rare diseases worldwide.

Another perspective is the micro view of a particular health related event happening somewhere in the world. The event of "lead poisoning" in Nigeria on May 12th, 2015, which went mostly unnoticed at a global level, is an example that can be explored through ER. With a simple query to the system, the health professional can extract all the reports related to that event, and check the event in the context of other related events. To illustrate this, the screen-shots in Figures 3 and 4 show the event itself and the story-line developed after the event, respectively.

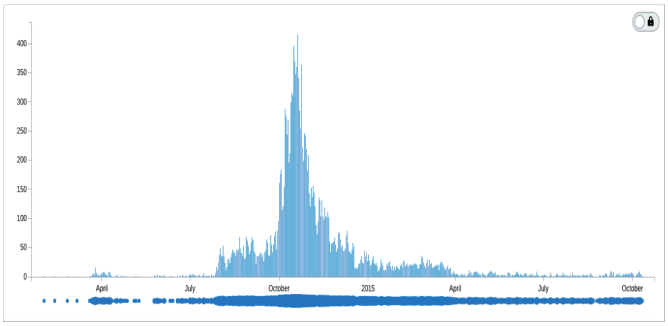

Fig. 5. A temporal intensity visualisation in ER for the query 'Ebola virus disease' showing that the system could notice the outbreak of the epidemic before it was made public.

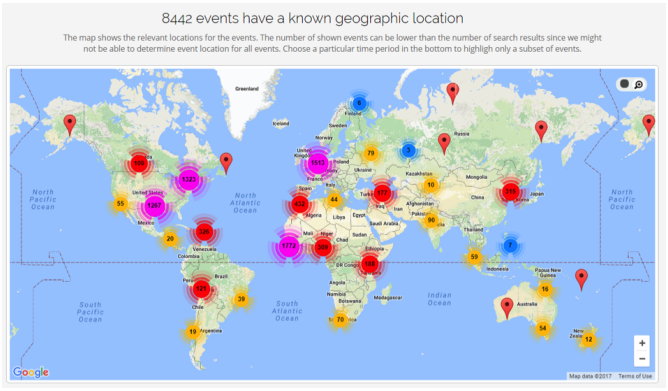

Fig. 6. ER screenshot showing a geographical spread visualisation module for the query 'Ebola virus disease'. Each mark shows the number of related news per location in the map.

V. Historical Perspective

Another view relevant to any health related issue is a historical perspective based on an aggregation of a particular topic. The evolution of Ebola virus and reporting after 2014 until 2017 can be an example to be explored through ER. Querying ER for *Ebola virus disease* provides over 20,000 events related to Ebola appearing after 2014. The content (over 200,000 news articles) could be analysed through several visual modules: temporal intensity, geographical spread, topical spread, among others. The Figures 5 and 6 illustrate temporal (with the peak in October 2014) and geographical spread (West Africa and the US) of Ebola related events. We highlight that ER was able to notice the outbreak of the epidemics before it was made public. Though, the question was not asked then, i.e., the query wasn't done because it was not yet an identified issue and there was a lack of continuous attention. Though a complete attention of all such epidemic related topics could be a heavy burden to the system. This rises the research problem of predicting and alerting of high density events like this.

In ER the Zika outbreak event is identified immediately after the collection of the news articles that report about it. One can explore the evolution of the news publishers awareness of the epidemics in time by looking at the related news articles represented in a world map, as they were identified or updated during a selected period of time.

ER can find articles and events related to a particular entity, topic, date, location or category, as well as measure their impact on social media (specifically, Twitter). This social media monitoring is complemented by TwitterObservatory (cf. [12]), leveraging in-house technology that uses data observation, enrichment and storage techniques for social media data presentation, search and analytics. Moreover, there have been several successful tests done to extract sentiment from news based on the sentiment of tweets associated with news [13]. The sentiment directly from news is still an open problem that shall be tackled.

VI. Conclusions and Further Work

In this paper we discussed the potential of several text mining tools dedicated to explore worlwide multilingual news, focusing matters of interest to Public Health. Further research (exploring the potential of Newsfeed, Wikifier and ER) includes: (i) the correlation of high level concepts with low level features; (ii) the showcase of hierarchies (e.g. in some disease) and how they can be drilled down to variety of sub topics (e.g. different aspects of such a disease; (iii) the analysis of the impact of health related issues on society (e.g. Ebola news impact in adherence to insurance); (iv) the presence of PubMed/Medline in global news; and (v) the prediction of consequences of a health event. Other research directions consider the dynamics of Public Health/Healthcare institutions where activities happen affecting: (1) the decision makers that make choices based on the legislation and on the available in-house monitoring systems operated by their own data scientists, (2) those data scientists that explore the data and extract relevant information that contributes to evaluation of Public Health scenarios, and (3) the technical team that deploys and maintains the data infrastructure where data scientists are active. It could be very useful to have easy to handle data visualisation modules (like the ones offered by Kibana and shown in Figure 7) allowing decision-makers to choose with a few clicks the representation of data that makes sense to the problems they focus on. The data scientists could then manipulate the current workflows, maximising critical outputs, presenting data in a meaningful way whilst minimising resource required to drive data interrogation and presentation.

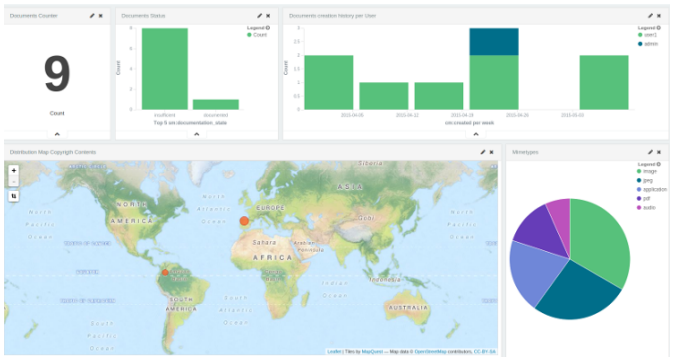

Fig. 7. Kibana screenshot showing the different visualisation modules based on queries to elasticSearch composing an interactive dashboard. This is a puppet example from venzia.es to show the practical potential of this tool.

Acknowledgment

The authors would like to thank the EC Horizon 2020 MIDAS Project and funded by the European Union under grant agreement no. 727721, under the call SC1-PM-18-2016 - Big Data supporting Public Health policies.

References

[1] J. Brank, G. Leban, and M. Grobelnik. A high-performance multi-threaded approach for clustering a stream of documents. *In Proceedings of the 17th International Multiconference Information Society*, 2014.

[2] J. Brank, G. Leban, and M. Grobelnik. Annotating documents with relevant wikipedia concepts. *Proceedings of SiKDD 2017, forthcoming*, 2017.

[3] M. A. Dion M, AbdelMalik P. Big data and the global public health intelligence network (gphin). *CCDR: Volume 41-9, September 3, 2015: Big Data*, 2015.

[4] ECDC. Epidemic intelligence. ecdc.europa.eu/en/threats-and-outbreaks/epidemic-intelligence. Accessed: 2017-09-05.

[5] M. Grobelnik. Observing global health and well-being. http://www.midasproject.eu/2017/07/24/observing-global-health-and-well-being/.Accessed6/8/2017, 2017. MIDAS Project Blog, 2017.

[6] M. Grobelnik and G. Leban. Eventregistry's health pannel. https://tinyurl.com/wits2017qnt, 2017. Accessed: 2017-07-25.

[7] M. Grobelnik and D. Mladenić. Simple classification into large topic ontology of web documents. *CIT*, 13.4:279–285, 2005.

[8] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Cross-lingual detection of world events from news articles. *Proceedings of the 2014 International Conference on Posters and Demonstrations Track, CEUR-WS. org*, 1272:21–24, 2014.

[9] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik. Event registry: learning about world events from news. *Proceedings of the 23rd International Conference on World Wide Web, ACM*, 2014.

[10] J. P. Linge, J. Belyaeva, R. Steinberger, M. Gemo, F. Fuart, D. Al-Khudhairy, S. Bucci, R. Yangarber, and E. van der Goot. Medisys: medical information system. *In Advanced ICTs for disaster management and threat detection: collaborative and distributed frameworks*, pages 131–142, 2010.

[11] A. Muhič, J. Rupnik, and P. Škraba. Cross-lingual document similarity. *Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces (ITI), IEEE*, pages 387–392, 2012.

[12] I. Novalija, M. Papler, and D. Mladenić. Towards social media mining: Twitterobservatory. *Proceedings of the Slovenian Data Mining and Data Warehouses 2014*, 2014.

[13] L. Rei, M. Grobelnik, and D. Mladenić. Event detection in twitter with an event knowledge base. *Proceedings of SiKDD 2015*, 2015.

[14] J. Rupnik, A. Muhic, G. Leban, P. Škraba, B. Fortuna, and M. Grobelnik. News across languages-cross-lingual document similarity and event tracking. *Journal of Artificial Intelligence Research*, 55:283–316, 2016.

[15] R. Steinberger, F. Fuart, B. Pouliquen, and E. van der Goot. Medisys: A multilingual media monitoring tool for medical intelligence and early warning. *Proceedings of the International Disaster and Risk Conference IDRC Davos 2008 - Short and Extended Abstracts p. 612-614*, 2008.

[16] M. Trampus and B. Novak. The internals of an aggregated web news feed. *Proceedings of 15th Multiconference on Information Society IS-2012*, 2012.

[17] T. ˇ Stajner, I. Novalija, and D. Mladenić. Informal multilingual multi-domain sentiment analysis. *Informatica*, 37.4:373–380, 2013.

[18] WHO. Epidemic intelligence. www.who.int/csr/alertresponse/epidemicintelligence/en/. Accessed: 2017-09-05.

Ontology-based translation memory maintenance

Andraž Repar
Iolar d.o.o.
Parmova 51 and
Jozef Stefan International Postgraduate School
Jamova 39
1000 Ljubljana, Slovenia
repar.andraz@gmail.com

Senja Pollak
Jozef Stefan Institute
Jamova 39
1000 Ljubljana, Slovenia
senja.pollak@ijs.si

ABSTRACT

In this paper, we explore the use of text mining techniques for translation memory maintenance. Language service providers often have large databases of translations, called translation memories, which have been in use for a long time - leading to a slow population of the translation memory with other domains (i.e. adding financial content to a medical translation memory). To our best knowledge, no tools exist that would effectively separate the content of a translation memory according to different domains. Having the ability to extract individual domains from low-quality translation memories could mean a significant benefit to language service providers looking to utilize modern translation methods, such as machine translation and automated terminology management. In the first stage, we used OntoGen, a semi-automatic ontology building tool which uses text mining techniques, to separate the segments in the translation memory according to domains. In the second stage, we wanted to test whether we could use the domains defined in the previous stage to build classification models - effectively using them as class labels in place of the costly and time-consuming manual annotation of segments.

Keywords

translation memory, language service provider, ontology, OntoGen, text classification

1. INTRODUCTION

In the translation industry, language service providers (LSP) often offer a guarantee to their customers that they will never have to pay twice for the translation of the same text. In order to do so, they have come up with a way of saving and re-using past translations to reduce costs and offer discounts. Starting in the 1970s and 1980s, translation companies began using translation memories which are essentially databases of bilingual segment pairs (source text – target text) along with some metadata. Whenever a new document is received for translation, it is leveraged against the translation memory for "exact" and "fuzzy" matches and the results are used to calculate the final price of the translation. This technology really took off in the 1990s and today virtually every language service provider on the market uses some kind of a translation memory to store translations.

In theory, the translation memory concept involves the use of metadata to clearly mark the segments belonging to different domains and/or customers. However, metadata are often not added due to time pressure or other issues and the information about the domain or customer is lost. Without this information it is difficult to reuse the translation memories for machine translation and/or terminology management. Finally, the quality of a translation memory can also degrade over the years because segments may get accidentally stored in the wrong translation memory (the domain of the segments is not the same as the domain of the translation memory).

In this paper, we analyze one such translation memory used by the translation company Iolar to see whether we could use text mining techniques to extract domains and clean low-quality translation memories. We used OntoGen [4] topic ontology editor to separate the dataset into distinct domains and then used these domains for text classification in Weka [5].

Ontology learning is a well-researched area with researchers using various techniques, such as natural language processing([10]), machine learning ([13]) and information retrieval ([3]). The same can be said of using machine learning for text classification (for example, [11], [9] and [7]). On the other hand, research into using data mining techniques for translation memory maintainance is scarce with most authors focusing on spotting low-quality individual segments. Barbu [1] uses several machine learning algorithms to spot false segment pairs in translation memories, Sabet et al. [6] describes a system for unsupervised cleaning of translation memories without labeled training data based on a configurable and extensible set of filters, and Nahata et al. [8] defines a set of rules for a rule-based classifier which is in turn used to find low-quality segment pairs. A more recent topic that serves a similar purpose is quality estimation of machine translated segments. For example, Specia et al. [12] describe a system that tries to predict the quality of machine translated segments using machine learning.

2. DATA DESCRIPTION

The translation memory analyzed for this article has been in use for almost 15 years and contains parallel translation segments in English and Slovene. Initially, it was meant to store Marketing, Legal and Financial translations, but over the years various other domains have been stored in this translation memory. In addition to the three domains mentioned above, this translation memory also contains a large chunk of IT-related segments, such as user interface strings, user assistance texts and technical documentation of various IT devices (printers, scanners, monitors etc.). Given the content of these documents, we expect to see some overlap between domains – for example, a printer user manual will typically contain some legal information as well as some marketing-like language.

3. EXPERIMENTS

The most obvious way to go about this task would be to manually annotate a dataset from this translation memory and then use it to train a classifier. However, manual annotation is time consuming and costly, so we first utilize OntoGen [4], a semi-automatic and data-driven ontology editor focusing on editing of topic ontologies, and then use the resulting ontology topics for building a text classifier that could be used for other translation memories and documents.

3.1 Preprocessing

The first step involved extracting the segment pairs and filtering them. The Slovene segment parts were discarded because only one language is needed for this task. English was chosen because it is the source language in this translation memory. The TMX file contained 247,103 English-Slovene segment pairs. To cut down on the noise and remove the segments most difficult to classify, we decided to remove all segments with less than 8 words leaving us with 121,593 segments.

3.2 Ontology creation

The selected segments were saved in a Named Line-Documents format suitable for OntoGen. Given the size of the file, the processing in OntoGen was slow-going. We tried various approaches in OntoGen and finally settled on using k-means clustering (with k=10) functionality to generate various sets of segments corresponding to different keywords and then manually group them into meaningful domains based on our translation experience with this translation memory.

After experimenting with various ontology building techniques in OntoGen, the following topic ontology was constructed (followed by the number of documents in parentheses): IT (51,247) (subdivided into ITGeneral and User Interface), Marketing (11,567), Financial (12,987), Legal (42,163) (subdivided into Contracts, Tenders and IT Legal[1]).

A graphical representation is shown in Figure 1.

3.3 Classification

In the final step we exported the domains from OntoGen, attached them to their corresponding segments and loaded the data into Weka machine learning toolkit. We tested various machine learning classification algorithms (Naïve Bayes Multinomial, SVM, J48) to find which one gives the best results. 10-fold cross-validation was used for all experiments. We applied Weka's StringToWordVector filter and used a stoplist (300 most frequent words from the BNC [2] corpus) to filter out the most common words.

[1]This group contains segments from privacy policies and license agreements of various software applications

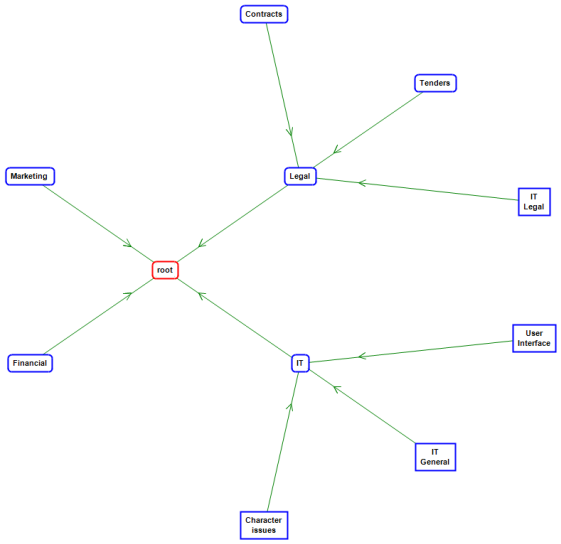

Figure 1: Ontology visualization: 4 main domains are extracted (Financial, Marketing, IT, Legal) with two of them having additional subdomains

In the first phase, we have both topics and subtopics – where a subtopic existed, we glued the topic and subtopic together to get a distinct class. This means we had 7 distinct classes: ITUserInterface, ITITGeneral, Financial, Marketing, LegalContracts, LegalTenders, LegalITLegal.

In the second phase, we used only the main topics – meaning that 4 classes were used: IT, Financial, Marketing, Legal.

Because the original dataset was fairly large (more than 100.000 segments), we had to significantly reduce it in order to be able to complete the calculations in Weka in reasonable time. However, we couldn't just take the first n segments, because the different topics were not uniformly distributed across the dataset. Therefore, we took every 10th segment, leaving us with a dataset of about 10,000 segments.

Tables 1 and 2 contain information about the performance of the three classifiers mentioned in section 3.3. For a detailed analysis see section 4.2.

4. EVALUATION AND INTERPRETATION OF RESULTS

When one evaluates the results of the hierarchical clustering by OntoGen and classification, one should bear in mind that in many cases no clear boundaries between domains exist. This was to be expected on the one hand due to the short length of the documents, and on the other due to the seg-

Table 1: Classifier performance with 7 labels (accuracy of the ZeroR classifier for the majority class = 0.349)

	J48	SMO	NB Multinomial
Accuracy	0.511	0.495	0.583
Precision	0.507	0.483	0.581
Recall	0.511	0.495	0.583
F-measure	0.472	0.483	0.580

Table 2: Classifier performance with 4 labels (accuracy of the ZeroR classifier for the majority class = 0.406)

	J48	SMO	NB Multinomial
Accuracy	0.597	0.619	0.671
Precision	0.615	0.608	0.678
Recall	0.597	0.619	0.671
F-measure	0.576	0.610	0.673

ments that are very difficult to assign to a single domain, for example:

- The system must support operation of the HSM system and the archiving of files even if the file system operates in the Windows cluster.
- The latest Windows operating systems have a firewall built in.

The first sentence comes from a tender document, while the second one comes from an IT user manual. Even for a human annotator, this would be a difficult task and we would most likely see low levels of inter-annotator agreement.

4.1 Ontology creation

To evaluate the results of ontology creation in OntoGen, we extracted 50 random segments for all 7 topics/subtopics, manually annotated them and compared the results.

Overall, a precision of 0.81 is quite good considering that we are working with sentences which are difficult to classify. It is also important to not lose sight of the fact that there can be some overlap between the topics and that certain sentences cannot be adequately classified into any of the available topics. The overlap between the various topics causes a certain degree of ambiguity, but we believe that the precision is high enough to use the topics extracted in OntoGen as class labels for building a classifier.

Table 3: Manual evaluation of the ontology results on 50 segments per domain

Topic	Precision
Financial	0.76
ITGeneral	0.80
ITUserInterface	0.86
LegalContracts	0.80
LegalITLegal	0.86
LegalTenders	0.78
Marketing	0.80
Average	**0.81**

4.2 Classification

The results of the classification with 7 labels are not promising. The performance of all classifiers does exceed the majority class classifier significantly, but the accuracy is not high enough for production use (close to 60% for the best performing classifier). Looking at the confusion matrix in Figure 2, we can observe that the ITGeneral topic overlaps with quite a few other topics and is the largest culprit for the low performance. A significant part of the false positives originate in the ITGeneral topic for all topics apart from Financial and LegalContracts (class c and e in Figure 2). These two topics also have the highest precision (0.688 and 0.668, respectively).

```
=== Confusion Matrix ===

   a    b    c    d    e    f    g   <-- classified as
 236  258   10   24   13   39   25 |    a = ITUserInterface
 197 2512   80  380  137  267  118 |    b = ITITGeneral
   7  131  817   73  231   41   18 |    c = Financial
   7  372   46  511   56  115    9 |    d = Marketing
   9  187  183   52 1321  121  108 |    e = LegalContracts
   9  401   43  150  152  447   38 |    f = LegalTenders
  12  164    8   20   68   34  325 |    g = LegalITLegal
```

Figure 2: Confusion matrix of the Naïve Bayes Multinomial classifier - 7 labels

When we focus only on the main 4 labels, the results are better. Naïve Bayes Multinomial is again the best performing classifier with its accuracy reaching a little over 67%. Looking at the confusion matrix in Figure 3, it is evident that the first 3 labels perform significantly better than the last one. Indeed, the precision of Legal, IT and Financial is around 0.7, while that of Marketing is just a little over 0.4 (for detailed results see Table 4).

```
=== Confusion Matrix ===

   a    b    c    d   <-- classified as
2482  838  261  271 |    a = Legal
 566 3206   80  447 |    b = IT
 268  126  852   72 |    c = Financial
 163  336   50  567 |    d = Marketing
```

Figure 3: Confusion matrix of the Naïve Bayes Multinomial classifier - 4 labels

	Precision	Recall	F-measure
Legal	0.713	0.644	0.677
IT	0.711	0.746	0.728
Financial	0.685	0.646	0.665
Legal	0.418	0.508	0.459

Table 4: Detailed performance of the Naïve Bayes Multinomial classifier

The largest issue that we have not been able to overcome in this analysis is that a huge chunk of the segments in this dataset are IT related – this is especially true of the Marketing and certain Legal segments (e.g. terms of use, privacy

statements or press releases or advertising material for IT devices) which means that it is often difficult to differentiate between a Legal/Marketing segment and a regular IT one. This issue is very clearly seen in the confusion matrix in Figure 3. In contrast, the Financial segments have no immediate relation to any IT content making them a much more distinct category.

5. CONCLUSION AND FUTURE WORK

This paper tries to determine whether text mining techniques can be used to facilitate translation memory maintenance in a language service provider environment. Given the fast-paced nature of work in the translation industry, it is only natural that the quality of translation memories reduce over time. Even if they are perfectly designed, noise will inevitably be introduced leading the reduced usefulness for other language applications.

At the outset, we had two questions: 1) whether OntoGen can be used to divide the content of a particular low-quality translation memory, and 2) whether the resulting topics can be used as labels to build a classifier for other translation memories and documents. The main reason was to find a shortcut for manual annotation which is costly and time-consuming.

We successfully managed to build an ontology, but the boundaries between some topics were relatively vague. One reason for this is that we had to deal with sentences – as opposed to larger chunks of text – which are difficult to classify. The second issue was the fact that many of these topics were in fact inter-related and some of the segments could have easily been classified in more than one domain. In particular, the Legal, IT and Marketing domains are closely related, because a lot of Legal and Marketing segments originated in IT-related translation jobs. One could argue that the IT and Marketing domains could be combined into one category, since there is so much overlap, however from a strictly translator's point of view it makes sense to have separate categories, because different translation strategies are normally used for marketing (i.e. press releases) and general IT (i.e. user manuals, help articles) translation jobs.

The results of the ontology creation were promising with manual evaluation (see Table 3) showing that around 4 in 5 strings were assigned a correct label. However, the picture was much less clear when it came to building a classifier. It turned out that the full ontology was too complex for the classification algorithms used in this paper (see Section 4.2). When we used only the four main topics as labels, the results started approaching acceptable with an accuracy of 67% (compared to 0.406 as majority class). We would still ideally like to see the accuracy breaking the 75% or 80% barrier.

In the current state, the classifier is not accurate enough to be used in production. However, when there are reasonably clear boundaries between topics in OntoGen, the resulting labels can be successfully used – as evident by the performance of the Financial label. This is in itself a useful achievement, because there is currently no way to export just the finance-related segments from the translation memory. An obvious route to better classification performance would be to use just those topics that are clearly separated from the other parts of the dataset.

In terms of future work, we will explore text classification on manually annotated high quality translation memories. Finally, an interesting route would be to utilize domain terminology to enhance highly domain-specific terms assigning higher weight to terminological features.

6. REFERENCES

[1] E. Barbu. Spotting false translation segments in translation memories, 2015.

[2] J. H. Clear. The digital word. chapter The British National Corpus, pages 163–187. MIT Press, Cambridge, MA, USA, 1993.

[3] H. Cunningham. Information extraction, automatic. *Encyclopedia of Language and Linguistics, 2nd Edition*, 5:665–677, 2006.

[4] B. Fortuna, M. Grobelnik, and D. Mladenic. *OntoGen: Semi-automatic Ontology Editor*, pages 309–318. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[6] M. Jalili Sabet, M. Negri, M. Turchi, J. G. C. de Souza, and M. Federico. Tmop: a tool for unsupervised translation memory cleaning. In *Proceedings of ACL-2016 System Demonstrations*, pages 49–54. Association for Computational Linguistics, 2016.

[7] T. Joachims. *Text categorization with Support Vector Machines: Learning with many relevant features*, pages 137–142. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[8] N. Nahata, T. Nayak, S. Pal, and S. Naskar. Rule based classifier for translation memory cleaning. 05 2016.

[9] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2):103–134, May 2000.

[10] S. P. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1440–1445. AAAI Press, 2007.

[11] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[12] L. Specia, G. Paetzold, and C. Scarton. Multi-level translation quality prediction with quest++. In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China, 2015.

[13] F. M. Suchanek, G. Ifrim, and G. Weikum. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 712–717, New York, NY, USA, 2006. ACM.

Audience Segmentation Based on Topic Profiles

Matic Kladnik, Luka Stopar, Blaz Fortuna, Dunja Mladenić
Jožef Stefan Institute
and
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
e-mail: matic.kladnik@ijs.si

ABSTRACT

Audience segmentation is often applied on Web portals to gain insights into the audience, support targeted marketing and in general provide on-line recommendations to the users. We propose an approach to audience segmentation that is based on using machine learning on topic profiles of the visited content. Our preliminary experiments on a small sample of log-data show that the proposed approach is promising and the proposed combination of features capturing short term and long-term user interest gives better results than using only the short-term interests of the user.

1. INTRODUCTION

Large number of retuning users regularly visiting the same Web portals offer an opportunity to apply audience modeling considering descriptions of the visited content, different characteristics of the user and the user behavior. Instead of the usual audience segmentation based on user profiles that is also commonly adopted in recommendation systems [1], we propose an audience segmentation approach that is based on the topic profiles of the visited content. As the users potentially have interests in different topics we allow the same user to occur in several segments.

In this paper, we described the proposed approach and on a small sample of real-world log-data test the hypothesis that the proposed combination of features capturing the content of the recently visited pages and the properties of all the pages visited by the user improves the quality of the segmentation.

The rest of this paper is structured as follows. Section 2 describes the problem setting and the dataset used in the experiments. The proposed approach is described in Section 3 together with the description background knowledge that we have used for mapping form the space of users into the space of topic profiles. Section 4 gives the results of the preliminary experiments, while the conclusions are presented in Section 5.

2. PROBLEM SETTING AND DATA

High-quality Web portals that offer regularly updated content, such as market data, news articles or financial data, attract many loyal users [2]. Today, vendors offer user data obtained by third-party cookies that cover a whole range of user properties including demographics, interest, geography. The problem that we are addressing is automatic audience segmentation where potentially vendor data on the users is available in addition to the usual Web log files and content of the visited pages. In addition we propose to use background knowledge in the form of pre-trained machine learning model that classifies Web pages into a predefined custom taxonomy. In this way, each Web page is based on its textual content assigned a ranked list of content topics of different granularity. For instance, the assigned topics may be Business, Business/Financial_Services/Medical_Billing, Business/News_and_Media, Society/Issues/Gun_Control.

The dataset that we have used to test the proposed approach was obtained from an international media company. Almost 3 000 Web pages were crawled from the company Web site. The anonymized user data was obtained for more than half a million users that have visited the Web site within one selected day. All the considered text is in English language.

We have pre-processed the data to remove references to Web pages that have limited textual content or un-standard formatting. The Web pages were processed in a standard way to extract the textual content, remove the standard English stop-words and represent each Web page as a bag-of-words (BoW) with TFIDF weight. In addition to the content, each page has metadata including a set of manually assigned content labels done by the editorial team. For instance, brexit, Europe, money, davos, jobs, London, markets. These content labels were historically used to annotate the users visiting the pages. Each user is thus described by a set of properties including demographics and the content labels of the pages visited over a longer period of time.

3. APPROACH DESCRIPTION

Audience segmentation is commonly based on grouping the users by their common interests and some other e.g., demographic properties and behavioral similarity. However, the same user may have several interests and exhibit different behavior depending on the current focus. This may result is grouping together the users that do not have much in common except that they share some (but not the same) interests with the third user.

Thus we propose an approach to audience segmentation based on the similarity of the topics that the users are interested in. The idea is to view the problem through topics of the visited Web pages and based on that obtain segments of the users.

Architecture of the proposed approach is shown in Figure 1. The whole pipeline consists of several steps:

1. From the log file of the user visits we obtain a list of visited pages (URLs).
2. By using background knowledge in the form of machine learning model for classifying documents into a predefined custom taxonomy, we assign a ranked list of topics to each URL.
3. For each URL we select one or a few topics with the highest rank and form a collection of URL – Topic pairs.
4. Representation of the topics is based on a list of URLs that were assigned the topic and the list of UserIds of the users that visited the URLs.
5. Topic profiles are processed by a clustering algorithm to obtain segments of the users.

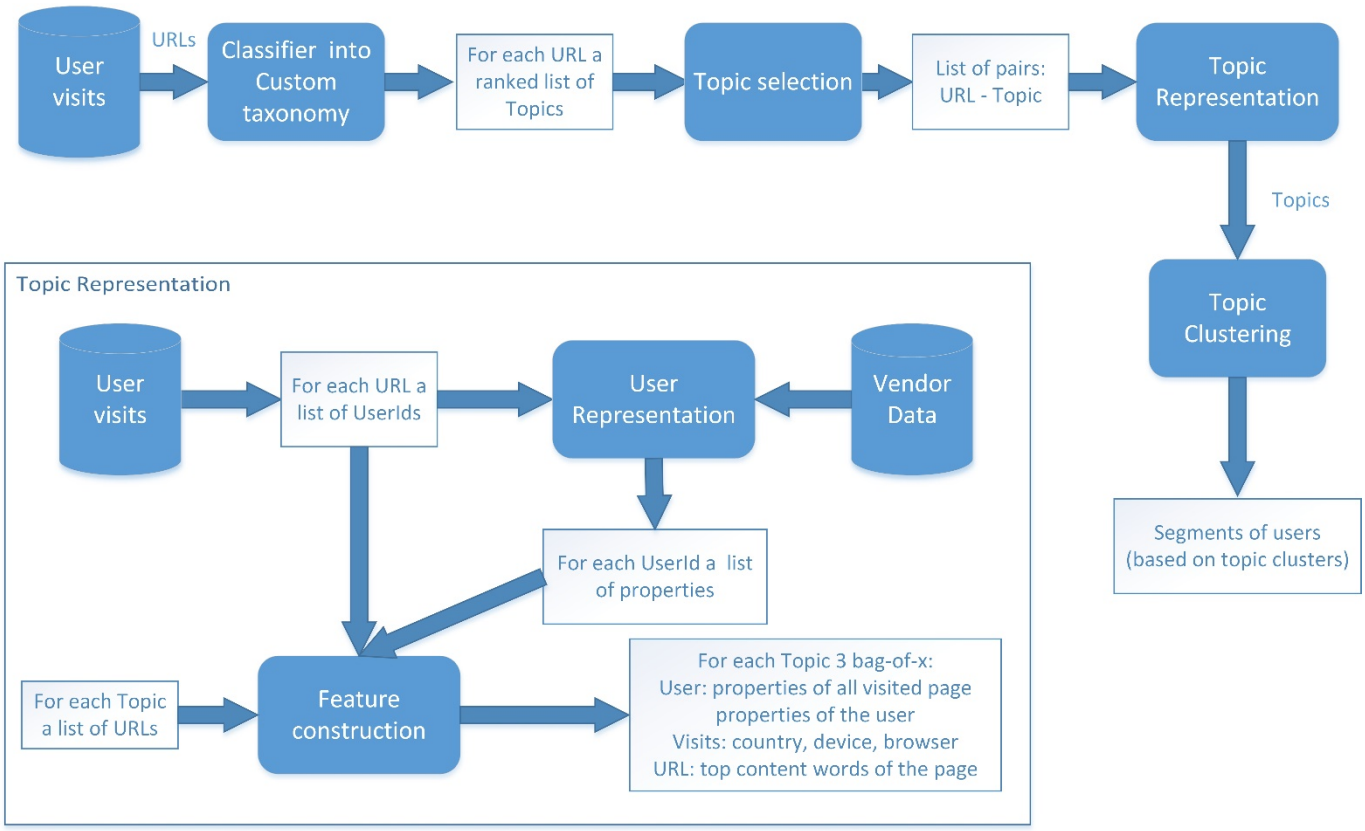

Figure 1. Architecture of the proposed approach to audience segmentation. The user visits of the Website and combined with the visited content and properties of the users to obtain segments of the users based on the topic clusters.

As topic profiles are built around the URLs that were assigned the topics using background knowledge, we separately keep a list of UserIds for each URL (obtained from the log file). When constructing features for each topic, we combine different data source:

- content of the Web pages visited by the users,
- properties of the visited pages,
- properties of the users, and
- information about the visits.

Background knowledge that we have used in the experiments is based on a subset of a large custom taxonomy [2]. The subset was defined by the domain expert from the company of the Web portal and consists of several hundred of topics.

We use DMOZ classifier with custom taxonomy to classify each Web page into a hierarchical content topic. Pages can be classified into topics on different levels of hierarchy, where lower levels give more specific classification. Upper levels also give context to the lower levels of classification. If we compare the following two topics that mention aerospace in their hierarchy:

- Science/Technology/Aerospace,
- Business/Aerospace_and_Defense/Aeronautical,

we can see that the first content topic is put into the context of Science and Technology, whereas the second is put into the context of Business and Aerospace and Defense. This approach gives us more information about the content topic.

In our experiments, content of the Web pages is obtained by crawling the Web portal. User data including properties of the visited pages is obtained from the Vendor data.

4. EVALUATION

In the experimental evaluation we combine two sources of data: URLs from the log file and the user data from the user's history. The two features sets used for data representation correspond to these two data sources: bag-of-words from the Web page corresponding to the URL; the user interest in the form of a collection of content labels of the Web pages visited by the user over a longer period of time (see Table 1).

Table 1. Features used for audience segmentation.

Source	Description	No. of values
Web page	BoW - Words from the Web pages	59929
User interest	Content Labels of the visited Web pages	1268

To compare the influence of different feature sources on the obtained segmentation we use a cluster dispersity measure. Specifically, we measure the weighted average distance between the examples and their centroid normalized by the average distance to the global mean (i.e. center of the data). The formula is given below:

$$D = \frac{\sum_{i=1}^{k} \frac{n_i}{n} \sum_{j=1}^{n_i} \frac{1}{n_i} d(\mu_i, x_j)}{\frac{1}{n} \sum_{j=1}^{n} d(\mu, x_j)} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_i} d(\mu_i, x_j)}{\sum_{j=1}^{n} d(\mu, x_j)}$$

D represents the dispersity, d is a distance measure (in our case cosine distance), n_i and μ_i are the size and centroid of the i-th cluster, x_j is the j-the example and μ is the global mean. Intuitively, examples in more compact clusters will lie closer to the centroid and so will contribute less to the dispersity than more disperse clusters.

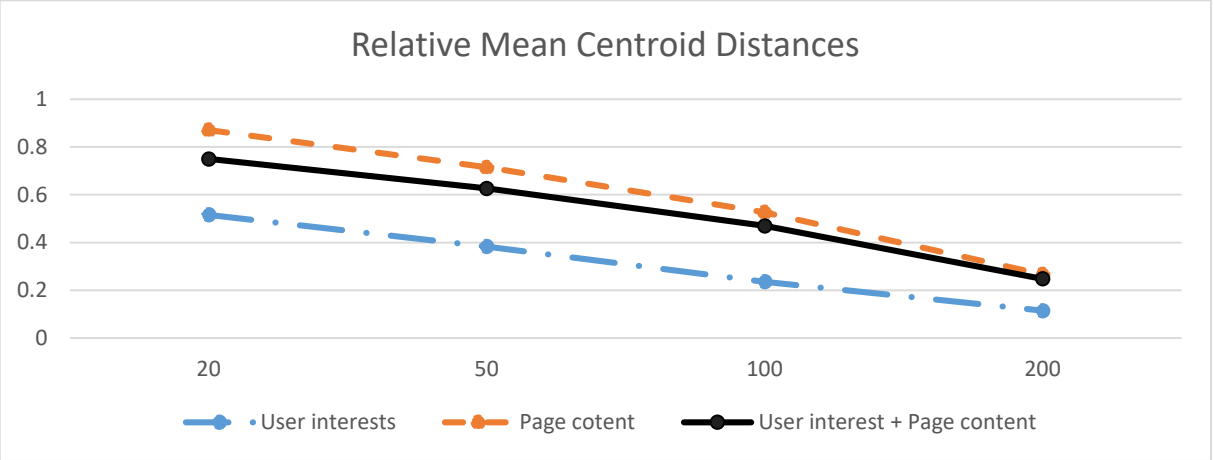

Figure 2 Experimental results comparing relative mean centroid distances for the three data representations over different number of clusters (*k* ranging from 20 to 200).

The experiments on clustering topic profiles were performed on three different data representations. One that considers only content of the visited pages, one that considers only user interest and a combination of the two feature sets. We have applied k-means clustering, varying the value of parameter *k* (the number of clusters) form 20 to 200. As the results of the applied clustering method depends on the random seed for choosing the initial clusters, we have repeating the process five time for each value of k. Figure 2 shows the results of the experiments averaged over five runs.

We can see that the smallest distance of topic clustering is obtained when the data is represented only by user interests, which represents a long-term interest of the user on an aggregated level. This can be particularly attributed to the fact that the number of different content labels is much smaller than the number of different words from the page content. Combining user interest (capturing history of the user) and page content (of pages visited in the considered log file) gives better results than using only page content.

Looking at the topic clusters that we have obtained, we can see that the similar topics are clustered together (see Table 2). For instance, the topics such as Health/Addiction, Business/Chemicals/Wholesale_and_Distribution, Recreation/Drugs are in the same cluster.

Table 2. Illustrative example of some clusters obtained when generating 50 clusters using both feature sets. For a few selected clusters we show the topics that belong to the cluster.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Business/Biotechnology_and_Pharmaceuticals	Recreation/Models	Home/Personal_Finance	Health/Addictions	Business/Financial_Services/Venture_Capital/Regional
Health/Child_Health	Science/Astronomy		Recreation/Drugs	Business/Transportation_and_Logistics/Bus
Health/Conditions_and_Diseases/Cancer			Business/Chemicals/Wholesale_and_Distribution	Business/Transportation_and_Logistics/Rail
Health/Conditions_and_Diseases/Immune_Disorders			Business/Food_and_Related_Products/Beverages	Government/Agencies
ealth/Conditions_and_Diseases/Infectious_Diseases			Science/Biology/Bioinformatics	Recreation/Autos/Makes_and_Models/Honda
Health/Pharmacy			Society/Issues/Gun_Control	Science/Environment
				Science/Environment/Carbon_Cycle, …

To obtain audience segments from the clustering of the topic profiles, we map the topic clusters onto a set of UserIds based on the user visits of the URLs that are classified to each of the topic in the cluster. In this way we obtain non mutually exclusive audience segments. The average number of users per segment is given in Table 3. From the table we can see

Table 3. Average size of the audience segments in relation to the granularity of the segmentation.

No. of segments	Average size	Median size
20	43592.35	589.5
50	17436.94	301
100	8718.47	229.5
200	4359.235	193

5. CONCLUSION

We have proposed an approach to audience segmentation based on topic profiles of the visited Web pages instead of the commonly used user profiles. The topic are obtained by classifying the visited Web pages into a custom taxonomy. The classification is performed automatically using a pre-trained machine learning model. The topic profiles are formed from properties of the users visiting the Web page that are classified into the topic, and the content of the Web page.

Preliminary experiments on a small sample of log-data show that the proposed approach is promising, grouping together similar topics and based on that segmenting the audience into reasonably populated segments. Namely, one of the issues with audience segmentation when the users have multiple interests is that many users that are not similar to each other are assigned to the same segment, due to similarity with the other users form the same segment.

Larger scale experiments are needed in the future work to confirm the results and provide additional insights into the other properties of the users form the same segment, such as demographics, geography, job.

6. ACKNOWLEDGMENTS

This work was partially supported by the Slovenian Research Agency.

References

[1] Adomavicius, G. and Tuzhilin, A., Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on on Knowledge and Data Engineering*, Vol. 17, No. 6, June 2005.

[2] Fortuna, B., Fortuna, C., Mladenic, D., Real-time news recommender system. In *Proceedings of Machine learning and knowledge discovery in databases : European conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010* (Lecture notes in computer science, ISSN 0302-9743, Lecture notes in artifical intelligence, vol. 6323). Berlin; Heidelberg; New York: Springer. 2010, vol. 6323, pp. 583-586.

[3] Grobelnik, M., Bank, J., Mladenic, D., Novak, B., Fortuna, B.. Using DMoz for constructing ontology from data stream. In Proceedings of the 28th International Conference on Information Technology Interfaces, June 19-22, 2006, Cavtat/Dubrovnik, Croatia, (IEEE Catalog, No. 06EX1244). Zagreb: University of Zagreb, SRCE University Computing Centre. cop. 2006, pp. 439-444.

Building Client's Risk Profile Based on Call Detail Records

Zala Herga[1, 3], Casey Doyle[2], Stephen Dipple[2], Caleb Nasman[2], Gyorgy Korniss[2], Boleslaw Szymanski[2], Janez Brank[1], Jan Rupnik[1], and Dunja Mladenić[1, 3]

[1]Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia

[2]Rensselaer Polytechnic Institute, 110 8th Street,Troy, NY 12180, USA

[3]Jožef Stefan Postgraduate School, Jamova 39, Ljubljana, Slovenia

{*zala.herga,janez.brank,jan.rupnik, dunja.mladenic*}*@ijs.si*

{*doylec3, korniss, nasmancc1, dippls, szymab*}*@rpi.edu*

ABSTRACT

Data collected from mobile phones can be used to uncover underlying social network dynamics and individual's behavioral patterns. Based on a Call Details Records dataset, we build a weighted, directed network and analyze it's properties. In addition to node-level network measures we extract an extensive consumption and mobility-based feature set. We show that extracted network and consumption features can be used to model individual's risk profile.

Keywords

Mobile Phone Network, CDR, Supervised learning

1. INTRODUCTION

The Call Detail Records (CDR) dataset is a relatively standard dataset obtained by mobile phone operators. One record in the CDR dataset corresponds to a communication event between two mobile phone users and includes time stamp, type of event (call, text), direction (in- or outgoing) etc. This data type reveals behavioral patterns that can be used to identify user's personality [2], spending habits [11] or socioeconomic level [5]. Here, we are interested in using the data to build each client's risk profile; in particular, we attempt to use this data to predict user defaults. To this end, we focus our analysis around whether the clients phone number was blocked at the end of month (indicating issues potentially related to the defaulting behavior), using this data to label clients as good or defaulted. The dataset used is completely anonymised.

Structure of the rest of the paper is as follows: Section 2 presents characteristics of the network built from the CDR, Section 3 describes feature extraction, Section 4 presents probability of default models and their evaluation and Section 5 concludes the paper.

2. NETWORK PROPERTIES

As the first step in analyzing the dataset and gaining an understanding of how the users operate we define the structure of the network. Here, we treat the mobile data set as a social network where each node is an individual and each edge represents a connection between them and another individual. Wherever possible, we use weighted, directed edges to preserve the strength of the connection between individuals [4]. Weights are assigned based on the frequency of outgoing communications between the source node and the target node. Where applicable, we use this metric to define the distance between two nodes as $w_{avg}/w_{i\to j}$ where w_{avg} is the average weight of all connections in the network and $w_{i\to j}$ is the weight of the connection between the source (i) and the target (j) [6]. Wherever it is not feasible to use a weighted edge scheme, we create an unweighted graph using a cutoff to define how many outgoing communications from one node to another constitutes a connection (ie we use the frequency to define whether a connection exists at all, and all connections are still directed but have equal weight) [5]. Using a low cutoff introduces a lot of noise into the system and is less representative of a true social network as many of the edges are too weak to accurately indicate a social connection between two individuals, but choosing too high of a cutoff restricts the network and discards potentially valuable data connecting nodes and communities together.

Using these methods, our data set translates to a network with a giant component comprising 99.14% of the it. Of course, the number of edges and size of the giant component decreases quickly when the unweighted cutoff scheme is used (Fig. 1). The size of the giant component decreases linearly with increases in the cutoff, while the decrease in the number of edges levels off as a power law with $\gamma \approx 0.75$. The degree distribution also changes slightly with the cutoff than without; in both cases the distribution has a fat tail that is well approximated by a power law, but the exponent increases with a larger cutoff. For the general case of the weighted edges with no cutoff, the power law tail has an exponent of $\gamma \approx -4.3$ while with a high cutoff such as thirty the power law tail exhibits an exponent of $\gamma \approx -6$ (Fig. 2), both in general agreement with prior work on mobile network data [5, 7]. Similarly, the distribution of node strengths (defined as the sum of the weights of its adjacent edges) also exhibits a heavy tailed decay as expected [7].

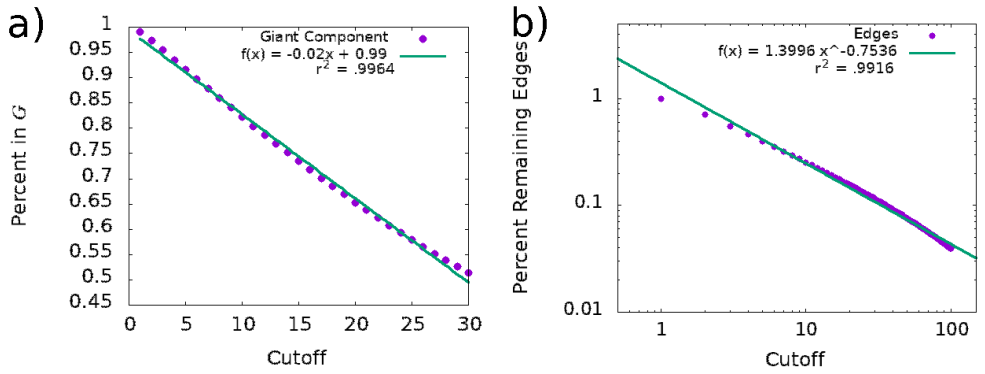

Figure 1: (a) The size of the giant component G as it decreases linearly with higher cutoff criteria to form an edge between two nodes. (b) The total number of edges in the system decreases as a power law with $\gamma \approx 0.75$.

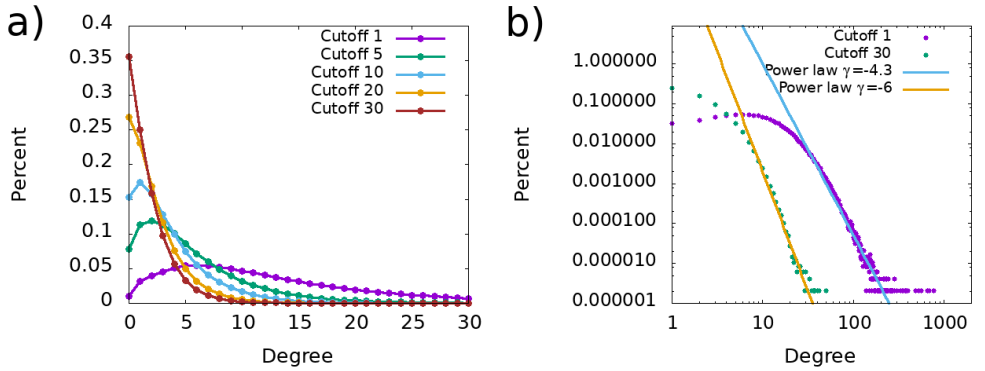

Figure 2: (a) The degree distribution for various cutoff criteria to create an edge between two nodes. As the cutoff increases, the peak of the distribution shifts left until it peaks at zero. (b) The same distributions on a log-log scale to highlight the power law tail of the distributions. Shown here are the two extreme cutoffs tested in order to highlight the increase in the gamma value for higher cutoffs.

Additionally, we study the distribution of some higher level node based measures such as reciprocity[14], which is surprisingly low. Without a cutoff only 38.65% of all links are reciprocated, while higher cutoffs increase the fraction of reciprocated links up to a maximum of only 41.57% when the cutoff is fifteen. We further measure the reciprocity using the weighted network scheme by defining the weighted reciprocity[12, 13] as $R_{ij} = |w_{ij} - w_{ji}|/(w_{ij} + w_{ji})$. Using this metric, the network shows an average weighted reciprocity of only .3235, further indicating the low reciprocity of the network.

Finally, we use node centrality to measure how the nodes position themselves in within the communication paths across the network (nodes with high centrality are most likely to connect communities and therefore are very important to the study of how risk patterns propagate across the network). For this purpose, we utilize the closeness centrality[1, 10, 3], a node level measurement that utilizes the shortest paths across the network to identify where nodes lie in the network structure. Specifically, it is a ranking of the distance from the node in question to every other node, defined as $C_C(i) = (N-1)(\sum_{i \neq j} d_{ij})^{-1}$ where C_C is the closeness centrality and d_{ij} is the length of the shortest path between nodes i and j (assuming a path exists). Unfortunately this measure only works on connected graphs, so to analyze the full unconnected graph, we also study the harmonic closeness centrality[9, 8], defined instead as $C_H(i) = \frac{1}{n-1} \sum_{j \neq i} \frac{1}{d_{ij}}$. As seen in Fig. 3, the closeness centrality has a tightly grouped, high density of relatively high values. This implies a very well connected graph such that the shortest path between any two nodes is low. This can further be seen via the harmonic centrality, which also has a relatively low density of low scoring individuals even with the inclusion of nodes not within the giant component. This implies that even the nodes that are not connected to the giant component tend to form small, tightly connected communities of their own.

3. FEATURE EXTRACTION

After understanding the network dynamics, our aim was to build individual's behavioral patterns. For that reason we extracted from the CDR three types of behavior-related features: individual's consumption, social network and mobility. Some of the features were extracted for different time window (e.g. per day, per week, hour of day), separately for incoming and outgoing events and/or separately for event type (call, text). We also added another more technical category which relates to individual's position in the underlying network. Together, more than 6000 features were extracted. Each category is described in more detail below.

1. Consumption features: These features are related to individual's usage of the mobile phone. We extracted for each individual the number of all calls, number of all texts, total duration of calls, average duration of calls, average time between consecutive events.

2. Social network features: This type of feature focuses

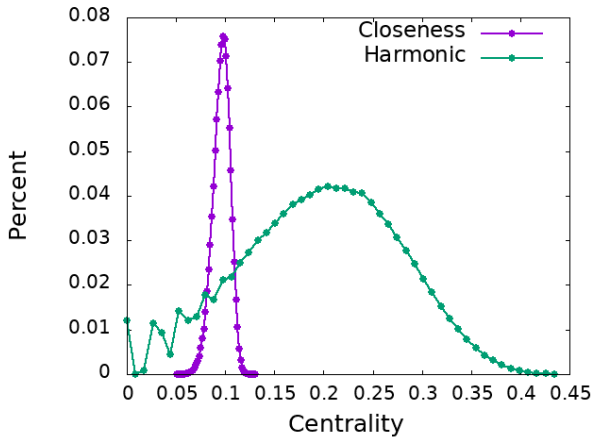

Figure 3: Distribution of closeness and harmonic centrality scores across the network where higher scores indicate a more central position in the network. The closeness centrality only considers nodes in the giant component (and the distribution is therefore only calculated for those nodes). The harmonic centrality includes all nodes in the network.

on the number of contacts and reciprocated events: the number of unique contacts, the number of contacts with which individual exchanges on average at least 5 texts per week / 2 calls per week, the number of reciprocated call events, the median time between reciprocated call events, and the median time to answer text.

3. Mobility features: These features that are based on used BTS tower location and include the average daily radius of gyration, the average distance traveled per day of week, the popular cell towers that sum up to 90% of records, and the average number of unique cell towers used per week.

4. Node level network measures: These features all rely on the individuals location within the social network built off of their usage statistics. The details of these metrics are discussed in Section 2, and represent each nodes level of importance to the overall social network as well as how deeply embedded the individual is.

3.1 Geographic analysis

Geographic analysis was performed to help us with the specific goal of building individual's risk profile. Analyzing geographic features requires a definition of their location that considers that most people connect to many different cell towers over the course of a three month period. For our purposes, we use each user's top two most used cell towers. We assign an individual to both their most used and second most used towers to account for the likelihood that a user will spend large amounts of time both at their residence and their workplace. From there we analyze the number of people that exhibit default behavior for each tower or district and identify high risk geographic regions. Based on that analysis we calculated empirical probability of default for *each cell tower*. These probabilities were used as two additional features, one for each of the two most commonly used towers of each user.

4. MODELING

Our aim was to model probability of default for each client based on extracted phone usage patterns and node-level network measures. We present the results of fitting several linear regression models with varying parameters. Features that are described in previous sections were used for modeling, and all features were normalized to standard score (z-score). We divided our dataset into train- and test set in 70:30 ratio.

We started with a linear model (labeled as *glm-6* in Figure 4) that was based only on six predictor variables: *frequency* (corresponds to the node's strength), *duration* (of user's call events; sum), *degree*, *harmonic centrality* and the two *geographic* (cell tower PD) variables. We chose with these features because we believed that they carry a lot stronger signals in contrast to the other 6048 features. The p-value is < 0.01 for all, except for *duration*, which has a p-value of 0.97. Overall this implies that network measures are good predictors for default behavior.

4.1 Principal Component Analysis

Further, when dealing with larger amount of features (6048) principal component analysis (PCA) was performed on the train set for feature space reduction. PCA is a method that decomposes the feature space into principal components (eigenvectors) and also provides information about how much variance in the data each component explains. Selection of a subset of PCA components reflects a trade-off between 1) model simplicity (we want to include a moderate number of features in our models) and 2) total variance explained by the component subset. All features were subjects to PCA, except for the 6 features that were used in *glm-6* model described above. Those were added to models in their original (but standardized) form.

We ordered the obtained PCA components decreasingly by explained variance of the data. The first component explains 20% of the variance, the second 7%, the first ten components together 37%, first thirty together 42%, and first five hundred sum up to 66%. We then created two linear models based on reduced feature subsets: first, using 30 PCA components and second, using 500 components (*pca-30* and *pca-500* in Figure 4). Because many variables in *pca-500* have large p-values, we fitted another model that didn't include those variables with p-value ≥ 0.5 (*pval-05*).

4.2 Oversampling

Only about 0.25% of users in the underlying dataset exhibited default behavior, which makes the dataset very unbalanced. For that reason, we implemented a simple oversampling method on train set: we multiplied defaulted users (and their features) by 20. The model using this method, *oversampled-20*, is also presented in Figure 4. Surprisingly, the oversampled dataset not only does not improve performance, but can be seen to provide slightly worse results than the originally unbalanced dataset.

4.3 Evaluation

Model comparison is presented in Table 1. We can see that at 95%, the level recall is high, up to 0.91 for both models based on 500 PCA components. Precision is low for all models due to the unbalanced dataset, but even with that drawback, our models still perform far better than random models.

Model	random	glm-6	pca-30	pca-500	pval-05	oversampled-60
Recall	0.05	0.13	0.79	0.90	0.91	0.86
Precision	0.003	0.007	0.042	0.049	0.049	0.046

Table 1: Recall and precision at 95% level for each of the models presented in Figure 4.

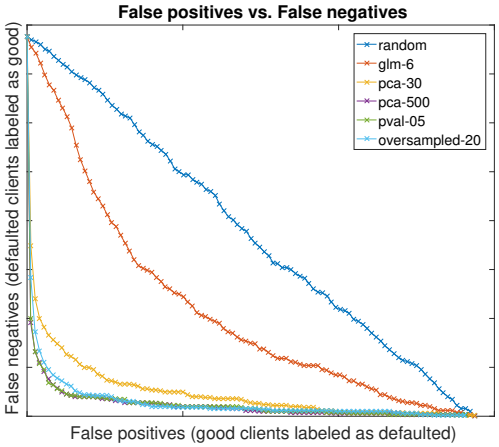

Figure 4: This graph presents prediction results on test set of the fitted models. y-axis corresponds to the false negatives (clients, that we're labeled as *good* but really defaulted), while x-axis corresponds to false positives. Results are shown for probability thresholds $0 - 1$ with step 0.01. *pca-500* (purple) is to great extent covered by *pval-05* (green) since both models provide very similar results.

5. CONCLUSION

This paper presents an analysis of a mobile phone data using a social network representation and various prediction models to understand default patterns. The analysis on the underlying network reveals a large giant component such that most nodes have at least some path to any other node in the network. Further, both the nodes within and without the giant component exhibit relatively high centrality scores; meaning that nodes are form tightly connected communities such that the path between nodes is generally quite short. Further, many nodes have a high degree and the degree distribution exhibits a heavy power law-like tail. Using many of these properties as features, we were able to make even more accurate predictive models of default.

Our model evaluation shows that there are many variables that carry weak signals about user behavioral patterns that have a strong predictive power when aggregated together. The unbalanced nature of the dataset makes the fitted models have a high recall but low precision, yet they strongly outperform the random model in both measures.

There is still a lot of space for improvement in the modeling including testing more complex oversampling methods, fitting additional models (SVM, LASSO, ANN), and including additional node-level network measures and community detection analysis.

6. ACKNOWLEDGEMENTS

This work was supported by RENOIR EU H2020 project under Marie Skodowska-Curie Grant Agreement No. 691152 as well as in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (the Network Science CTA), by the Office of Naval Research (ONR) Grant No. N00014-15-1-2640, and by NSF Grant No. DMR-1560266 under the REU Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies either expressed or implied of the Army Research Laboratory or the U.S. Government.

7. REFERENCES

[1] A. Bavelas. Communication Patterns in TaskOriented Groups. *J. of the Acoustical Society of America*, 22(6):725–730, nov 1950.

[2] Y.-A. de Montjoye, J. Quoidbach, F. Robic, and A. Pentland. Predicting personality using novel mobile phone-based metrics. In *SBP*, pages 48–55, 2013.

[3] L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, jan 1978.

[4] M. S. Granovetter. The Strength of Weak Ties. *American J. of Sociology*, 78(6):1360–1380, May 1973.

[5] S. Luo, F. Morone, C. Sarraute, M. Travizano, and H. A. Makse. Inferring personal economic status from social network location. *Nature communications*, 8:15227, may 2017.

[6] M. E. J. Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132, Jun 2001.

[7] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási. Structure and tie strengths in mobile communication networks. *PNAS*, 104(18):7332–6, May 2007.

[8] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, jul 2010.

[9] Y. Rochat. Closeness centrality extended to unconnected graphs: The harmonic centrality index. In *ASNA*, number EPFL-CONF-200525, 2009.

[10] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, dec 1966.

[11] V. K. Singh, L. Freeman, B. Lepri, and A. S. Pentland. Predicting spending behavior using socio-mobile features. In *SocialCom*, pages 174–179. IEEE, 2013.

[12] T. Squartini, F. Picciolo, F. Ruzzenenti, and D. Garlaschelli. Reciprocity of weighted networks. *Scientific Reports*, 3(1):2729, dec 2013.

[13] C. Wang, A. Strathman, O. Lizardo, D. Hachen, Z. Toroczkai, and N. V. Chawla. Weighted reciprocity in human communication networks. aug 2011.

[14] S. Wasserman and K. Faust. *Social network analysis : methods and applications*. Camb. Univ. Press, 1994.

Connecting Professional Skill Demand with Supply

Erik Novak
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova cesta 39
1000 Ljubljana
erik.novak@ijs.si

Inna Novalija
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana
inna.novalija@ijs.si

ABSTRACT
Todays job market demand from the job seekers to continuously learn new skills. When applying for a job position one must have the required skill set. If the applicant is missing a skill it can be learned by attending a course. Finding the appropriate courses can be tedious but necessary work to be up-to-date with the job market demand. In this paper, we present a dashboard which connects the job market skill demand with the courses that give the required skill knowledge. We developed a pipeline for continuous crawling of job postings and courses which feeds the dashboard with the appropriate data. The dashboard allows searching by keywords and returns relevant job postings, courses and basic statistics relevant to the given search query.

General Terms
Job Market, Skill Set, Courses, Lectures, Design

Keywords
Information Retrieval, Data Mining, Analysis, Wikifier, VideoLectures.NET

1. INTRODUCTION
In todays job market the required skills are constantly evolving. This can be seen in more technical fields such as web development and data science where new tools and libraries are developed and available to the public with an increasing rate. This is visible in both research and industry sectors where a job position might require a previously unseen skill and the applicant needs to learn it to be qualified. Finding the courses that would give the skill knowledge can be tedious and does not guarantee its sufficiency.

To this end, we developed a dashboard which would connect the job market skill demand with the courses that give the required skill knowledge. We focused on job positions that require data science skills and courses that are provided by acknowledged course providers.

Our contributions are a) creating a sizable data set of data science related job postings containing the job postings title, description, locations and other information, and b) developing a dashboard which for a given query shows relevant job postings as well as courses and lectures which give the appropriate skills. The dashboard is daily updated with new job postings showing the most recent changes. Basic statistics such as the most popular job locations and skills are also shown.

The remainder of the paper is structured as follows. In section 2 we present related work. Next, data acquisition is explained in section 3 followed by the presentation of the dashboard in section 4. Finally, we discuss and conclude our work in section 5.

2. RELATED WORK
There are multiple blogs that write about top skills needed for getting a job in data science. One such blog is [14] which lists both non-technical and technical skills a data scientist should have in the coming years. Another blog [15] lists the top data science skills and courses where they can be learned. A lot of these blogs are not up-to-date and not reflecting the current state.

A research report [11] writes about connecting supply and demand in Canada's youth labor market. They were interested in finding what skills young adults acquired during their education, how employers demand is conveyed to students and those who support them and how well are the acquired skills utilized on the job. They presented their results but did not develop an application that would help to narrow the gap between the skill demand and supply.

Another report [13] talks about the mismatch of the skills young adults get during their education and the skills the companies demand. They found that skills are a critical asset for individuals, businesses and societies and that many employers report difficulties in finding suitably skilled workers. Additionally, they find that a sizable qualification mismatch is one of the biggest problems.

The company *Year Up* [9] helps young adults get the appropriate skills and the needed work experience. They identify motivated individuals and companies that are prepared to help, send the individuals to learn new skills and afterwards apply the newfound skills at the companies, getting the critical work experience for their career. The work is done

manually which can be expensive and time consuming.

3. DATA ACQUISITION

Open job positions can be found using job search services. These services aggregate job postings by location, sector, applicant qualifications and skill set or type. One such service is Adzuna [6], a search engine for job ads which mostly covers English speaking countries. Another service is Trovit [7], a leading search engine for classified ads in Europe and Latin America. The service is available in 13 different languages and provides listings of jobs as well as cars, real estate and other products.

When applying for a job position the applicant requires to have a certain skill set. If the requirements are not fulfilled, he can enroll in courses to get the missing skills. Additionally, watching certain lectures can give a deeper understanding of a particular problem which can increase the probability of getting accepted for a job position. VideoLectures.NET [8] is an award-winning free and open access educational video lectures repository. It contains videos of individual lectures as well as lectures given at renown conferences.

Crawling. Since we needed a continuous flow of data, we developed a pipeline for acquiring job postings, courses and lectures. This will allow us to provide the dashboard, presented in section 4, with the most recent data. For job postings we targeted the portals like Adzuna with an emphasis on positions in Data Science and for courses we targeted different course providers, including Coursera [2], providing courses from top universities, and Hackr.io [4], a service which finds the best online programming courses & tutorials. We also targeted VideoLectures.NET to acquire video lectures containing the Data Science tag. The tags are given manually by the VideoLectures team.

For data acquisition and enrichment, we collected data either using dedicated APIs, including Adzuna API [1] as well as custom web crawlers. The data was formatted to JSON to aid further processing and enrichment.

Enriching. The next step of data preprocessing is *wikification* - identifying and linking textual components to the corresponding Wikipedia pages [16]. This is done using Wikifier [10] which also supports cross and multi-linguality enabling extraction and annotation of relevant information from job postings, courses and video lectures in different languages. Wikification will allow us to search for job postings, courses and lectures in multiple languages.

Next, we use the Skill and Recruitment Ontology (SARO) [17] to extract Data Science skills from job postings. For each job posting we match the Wikipedia concepts with the skills found in SARO ontology and declare the matched concepts as Data Science skills. These skills are then added to the job posting profile.

Finally, to allow searching by locations and countries the job postings were further enriched by using GeoNames ontology [5] to include the latitude and longitude and the corresponding GeoNames ID and the location name.

Data Set Statistics. The job postings data set contains almost 3.3M job postings acquired in the period of 18 months. Job postings were located for 144 different countries, the majority of them from Europe. Figure 1 shows the top fifteen countries with most found job postings. The UK dominates other countries with 906k job postings, followed by France with almost 539k.

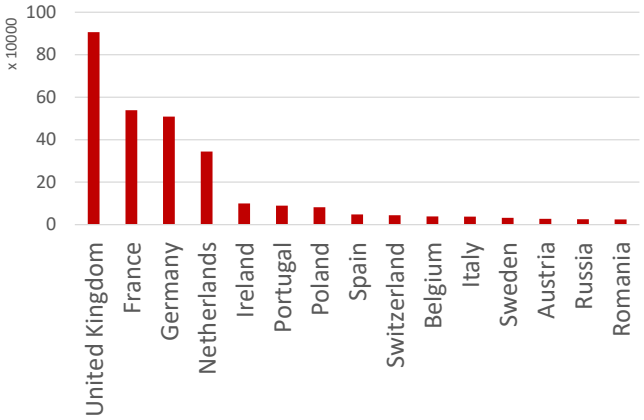

Figure 1: Top fifteen countries with most found job postings. The greatest number of job postings were found for UK, followed by France and Germany.

There were 650 unique Data Science skills extracted from the data set. These include soft skills, such as leadership and management, knowledge of a particular domain, such as machine learning and artificial intelligence, and programming languages. Figure 2 show the most demanded skills in the data set.

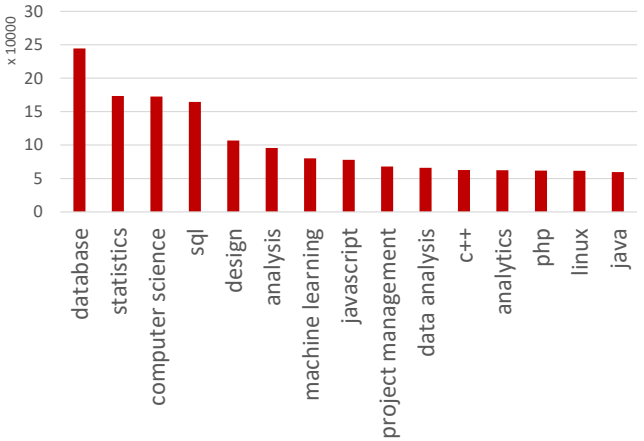

Figure 2: Top fifteen most demanded skills. They are mostly comprised of high-level skills, such as "database" and "computer science", and programming languages.

The course data set contains over 63k course information including their title, description and course providers. The data set is comprised of over 8k courses available online and 55k offline courses. Figure 3 shows the distribution of online courses by course providers. The most courses were acquired from Coursera with above 4k, followed by Hackr.io at 2k.

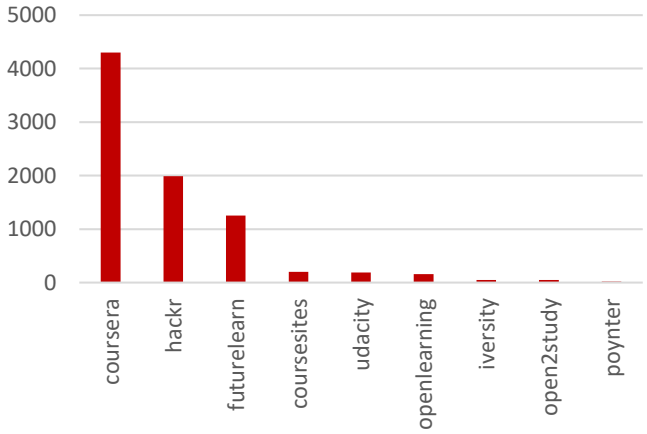

Figure 3: The distribution of online courses by course providers. The most courses were acquired from Coursera, followed by Hackr.io.

Finally, we acquired a data set of over 20k lectures published on VideoLectures.NET. It contains information about the lectures available on the video repository including title and description and link to the lecture.

4. DASHBOARD

Our objective is to automatically connect Data Science skill demand with the provided courses. To this end, we developed a dashboard [3] which enables its users to search for their desired job position, find out what is the required skill set and which are the appropriate learning materials and courses to acquire the missing skills. Additionally, the dashboard shows the most demanded skills and hiring location for the given results. In this section, we present the content retrieval methodology and describe the different components of the dashboard.

Methodology. Here we present the methodology used for retrieving the demand and supply content. The content is retrieved by inserting a query text in the search input. The user may add additional query conditions by selecting the Data Science skills, locations, countries and a time interval in which the job postings were published. Upon submitting, the query is used to fetch the content that matches the conditions. While all query values are used for retrieving job postings, only the input text and skills are used for retrieving the courses and video lectures content. Since courses and video lectures are available online the location and time interval are irrelevant for retrieving the supply content. To retrieve the content we first need to set an appropriate index. The job posting data set is indexed by Wikipedia concepts, Data Science skills, locations, countries and published date while the course and lecture data sets are indexed only by Wikipedia concepts. The query text is sent through wikification to acquire Wikipedia concepts which are used for retrieving the relevant content. Next, additional query conditions are used to filter out the content. The remaining content is used to calculate the most demanded skills and hiring locations. Finally, the query results are returned and used to update the dashboard components. This process is developed using QMiner [12], a data analytics platform for processing large-scale real-time streams containing structured and unstructured data.

Components. The dashboard is composed of different components. The largest component is a list of job postings. Each job posting is presented by its extracted information, including the Data Science skills extracted from the title and description. Figure 4 shows an example of a job posting in the list. Since Wikifier supports cross and multi-linguality the list consist of job postings written in different languages.

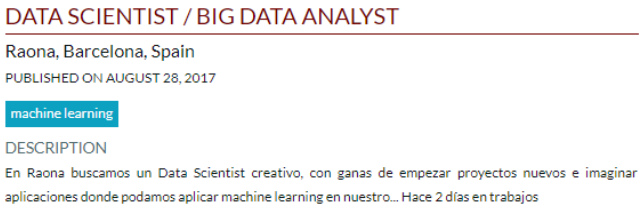

Figure 4: Example of a job posting returned by the query "machine learning". Even though the job posting is written in Spanish the methodology finds it relevant.

If the user does not have the required skill set it can be acquired by enrolling into courses shown in the course list. The list shows courses offered by different online course providers that are relevant to the users input query. Figure 5 shows the component containing the course list. Left and right arrows are used to navigate through the list where each course is presented by its name and a course provider.

Figure 5: A sample of recommended courses for the query "machine learning". Clicking on a course redirects the user to the course provider where he can enroll.

Additionally, the user can watch lectures to get a deeper understanding of a problem. Similar to courses the video lectures list show relevant content found on VideoLectures.NET. Clicking the lecture redirects the user to the video lecture homepage.

The dashboard also shows them most demanded skills and job posting timeline. The timeline shows how did the ratio between queried and all job postings change since the start

of the year 2016. Additionally, this shows a trend of the skill demand in the queried job posting subset. Figure 6 shows the visualizations used to show the skill demand and timeline.

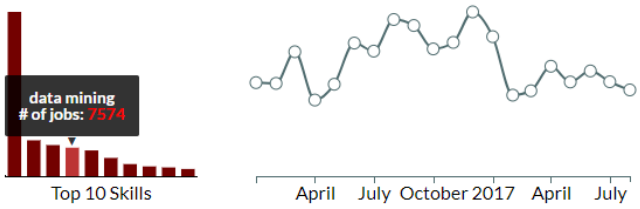

Figure 6: On the left the ten most demanded skills histogram, and on the right the number of job positions timeline, for the query "machine learning". Hovering over the histogram column shows the number of queried jobs demanding the skill.

Finally, a world map shows the most popular hiring locations extracted from the queried job postings. The locations are at first clustered where upon zooming the clusters divide and the individual locations are shown. Figure 7 show an example of clustered locations.

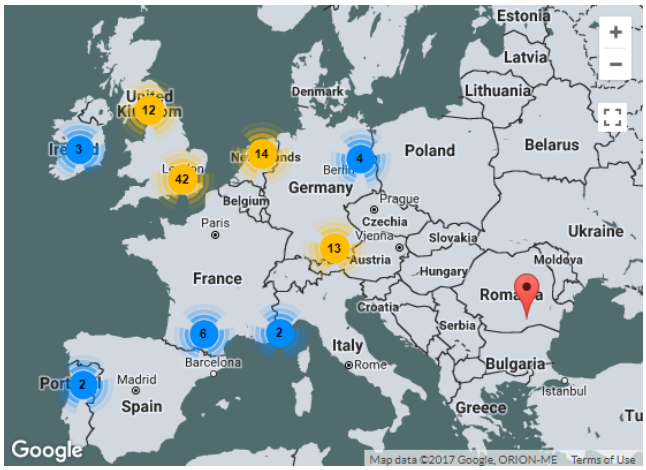

Figure 7: Top hundred hiring locations for the query "machine learning". The clusters show the number of locations it contains.

5. CONCLUSION AND FUTURE WORK

In this paper we present the methodology for automatically connecting skill demand and supply. We acquired a sizable job posting and course data set, developed a methodology for retrieving job postings, courses and lectures relevant to the user query and created a dashboard for showing the retrieved content.

In the future we wish to improve the data enriching process by handling skills that are not in the SARO ontology and add new features and improvements to the dashboard.

6. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency, EDSA EU H2020 project (Contract no: H2020-ICT-643937) and RENOIR EU H2020 project under Marie Skłodowska-Curie Grant Agreement No. 691152.

7. REFERENCES

[1] Adzuna api. https://developer.adzuna.com/. Accessed: 2016-09-07.

[2] Coursera | online courses from top universities. join for free. https://www.coursera.org/. Accessed: 2017-08-22.

[3] European data science academy dashboard. http://jobs.videolectures.net/. Accessed: 2017-08-23.

[4] Find the best online programming courses & tutorials - hackr.io. https://hackr.io/. Accessed: 2017-08-29.

[5] Geonames. http://www.geonames.org/. Accessed: 2017-08-23.

[6] Job search - find every job, everywhere with adzuna. https://www.adzuna.com/. Accessed: 2017-08-23.

[7] Trovit - a search engine for classified ads of real estate, jobs and cars. https://www.trovit.com/. Accessed: 2017-08-23.

[8] Videolectures.net - videolectures.net. http://videolectures.net/. Accessed: 2017-08-22.

[9] Year up - closing the opportunity divide. http://www.yearup.org/. Accessed: 2017-08-22.

[10] J. Brank. Wikifier. http://wikifier.org/. Accessed: 2017-08-23.

[11] R. Brisbois, L. Orton, and R. Saunders. *Connecting Supply and Demand in Canada's Youth Labour Market*. 2008.

[12] B. Fortuna, J. Rupnik, J. Brank, C. Fortuna, V. Jovanoski, M. Karlovcec, B. Kazic, K. Kenda, G. Leban, A. Muhic, et al. ■ qminer: Data analytics platform for processing streams of structured and unstructured data ■, software engineering for machine learning workshop. In *Neural Information Processing Systems*, 2014.

[13] W. E. Forum. Matching skills and labour market needs: Building social partnerships for better skills and better jobs. http://www3.weforum.org/docs/GAC/2014/WEF_GAC_Employment_MatchingSkillsLabourMarket_Report_2014.pdf, 2014.

[14] M. Mayo. KDnuggets analytics big data data mining and data science. www.kdnuggets.com/2016/05/10-must-have-skills-data-scientist.html. Accessed: 2017-08-22.

[15] E. Mcnulty. Top 10 data science skills, and how to learn them. http://dataconomy.com/2014/12/top-10-data-science-skills-and-how-to-learn-them/. Accessed: 2017-08-22.

[16] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011.

[17] E. Sibarani, S. Scerri, N. Mousavi, and S. Auer. Ontology-based skills demand and trend analysis, July 2016.

Analyzing raw log files to find execution anomalies

A novel approach to analyzing text-based log files to find changes in the execution profile of complex IT systems

Viktor Jovanoski
Carvic d.o.o.
Kotnikova 5
Ljubljana, Slovenia
viktor@carvic.si

Jan Rupnik
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
jan.rupnik@ijs.si

Mario Karlovčec
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
mario.karlovcec@ijs.si

Blaž Fortuna
Jozef Stefan Institute
Jamova 39
Ljubljana, Slovenia
blaz.fortuna@ijs.si

ABSTRACT

Anomaly detection (a.k.a. outlier detection) is the identification of events that do not conform to an expected pattern in a dataset. When applied to monitoring modern, complex IT systems, it keeps track of a plethora of incoming data streams. This paper provides an approach that uses the lowest and most unstructured source of data related to an IT system - the raw system log files. Several versions and parametrizations of basic building blocks will be presented to show how different types of anomalies can be extracted from the data. Several experiments on synthetic as well as real-world data show effectiveness of the algorithm. Special care is taken to keep the model and the resulting alerts interpret-able - since detecting an error without a meaningful explanation about its details is of limited use to end user (the results need to be actionable).

Keywords

Anomaly detection, Outlier detection, Infrastructure monitoring

1. INTRODUCTION

Modern IT systems are getting increasingly complex and distributed. On-line monitoring of their health is becoming critical for their normal operation. The data that is being collected about these systems comes in diverse time series (numerical, categorical, text), potentially huge in volume and arriving with different latencies. For instance, complex systems often expose metrics about their performance such as number of served requests or size of internal data structures. One can also monitor systems performance indirectly by inspecting CPU load, network load or database communication patterns. Quality assurance of input and output data can also be performed, such as the number and the size of the data records. Spotting unusual behavior of such systems is crucial and unhandled execution problems can lead to catastrophic results. The anomalies can be very different in nature, from abrupt changes that occur within a second to the gradual decay of performance that is only observable on a weekly or monthly scale.

Most of the research about anomaly detection has been concentrated on outlier detection in numerical time-series (e.g. financial time series like in [5], time series arising from infrastructure monitoring) and discrete-event sequences (e.g. fraud detection like in [4], cyber-security applications like in [3]). Data representation (the way of encoding the relevant information) is vital to the practical performance of an anomaly detection approach (does it capture relevant events or just insignificant variation).

However, all of these numeric data series have to be "prepared" in advance. A developer had to think ahead about the potential problems that can arise and expose appropriate measurements. When this data is available, it can clearly signal the problem to the operators. But what happens when a new type of outage occurs and no measure to detect it was put in place? In such case the infrastructure maintainers typically resort to inspection of *raw log files*. Writing a line of text to console or file output is often the only indication that something happened or has not happened when it should have. In our experience, all complex IT systems produce such files and they are still used to solve the hardest problems and errors.

The log files may be very unstructured in practice and may contain extremely diverse information. From errors, warnings, database calls and initialization steps, to casual counters and observations. The log lines themselves may be unstructured: their content might be unordered and they may contain text written in natural language (e.g. error messages). Even then, these bits of information may or may not

be written in a easy-to-parse form (e.g. JSON format). The only thing that can be expected of each line is a *timestamp* - a clear indication of time on the server, when this particular line was written to the log. Even if this data is missing from some lines, the sequential order of writing to file helps us narrow down the possible timestamp for each line. We can simply choose to re-use the last timestamp before that line.

Many algorithms and approaches to anomaly detection have been proposed in the literature. [1] provides an excellent overview of the field. New approaches are getting increasingly more sophisticated at dealing with multidimensional numeric data, discrete, sequential or even spatial data. The unstructured nature of raw log files makes the problem amendable to text-mining based approaches. This article presents work in that direction.

2. ANOMALY-DETECTION

End-users in charge of maintaining a large IT system will typically be concerned with two types of anomalies. Either they will want to avoid a sudden, critical degradation of performance, or they will want to know if the performance has been slowly degrading and attempt to prevent that.

Abrupt change - In this scenario, the system "falls of the cliff" - performance plummets and multiple parts of the system typically experience severe problems. Users will most often be notified of the problem via many channels. Hence, the raw log files are not the first place they will start looking at. However, if the cause of the problem cannot be determined or the exact timeline of events of the disaster cannot be established, the information from the log files will also be used as a part of the forensic analysis.

Gradual deterioration - IT systems that have been in production environment for a long time can experience gradual changes. These changes do not necessarily cause catastrophic failure overnight, but degrade the performance over a longer period of time. Such changes are very difficult to detect by the programmer as they are very subtle, e.g. they are only observable on the monthly scale. And often the only place where this can be detected is with a long term analysis based on the raw log files.

2.1 General pipeline

Algorithm 1 shows the general structure of a typical anomaly-detection system that operates on a stream.

First, a data record is extracted from the incoming data. As stated before, the timestamp is always defined. The record can also incorporate recent past data (previous lines of log file) and contextual data (e.g. server overall status, holidays indicator).

In the second step a *score* is calculated that should reflect a record's anomalousness (non-normality, novelty). The simpler the score, the easier the subsequent steps are. Additionally, we prefer models and scoring functions that can easily be explained to the end user, since he is supposed to act on them. Lastly, the score should be constructed in such manner that the anomalous examples fall on one edge of the spectrum (e.g. the higher the score, the bigger the anomaly).

Algorithm 1 General anomaly pipeline

```
while input data available do
  parse data and extract record
  calculate anomaly score
  if score above threshold then
    report anomaly
  end if
  add record to model
end while
```

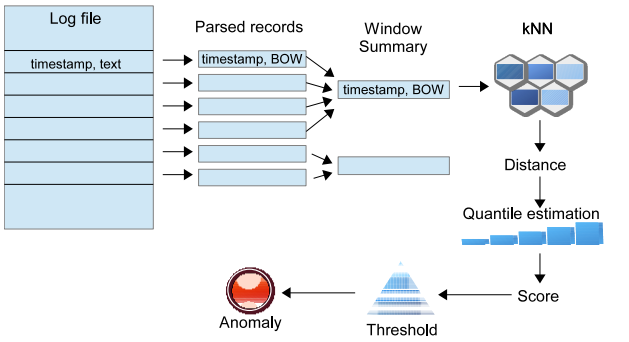

Figure 1: Text-processing anomaly pipeline

The score value is then used to decide if the incoming data record is an anomaly - either by comparing against some manually predefined static threshold or against a dynamic one, which uses historical score values to determine the threshold autonomously. Finally, in case of an anomaly, an alert is created that contains enough data to explain what was observed and why it was tagged as an anomaly.

Only when all of the above steps are done, we add the new data point to the model. We can store it internally in the model for some time, but we have no guarantee to be able to ever again access all the historical data (e.g. for retraining of the algorithm).

2.2 Defining normality

It is crucial for any anomaly-detection system to be able to tell what *normal* is, so that based on this notion it can say that something is *not normal*, i.e. anomalous. When available, domain knowledge and experience from human experts can be used to capture the appropriate aspect of normality in the data, which greatly improves the quality of the reported anomalies.

In general, however, the systems need the ability to autonomously define normality. This ability greatly depends on the scoring function and it is common practice to define it in a way that only the values on the border of observed values are indicators of an anomaly.

2.3 Detecting anomalies on raw log files

We will now present our approach to anomaly detection based on text processing techniques. The steps of the approach are given in Algorithm 2 and graphically presented in Figure 1.

The log parser processes a single line at the time and each

Algorithm 2 Log-file anomaly pipeline

```
while input data available do
  read data from log file
  parse data and extract BOW
  insert into time windows
  calculate distance in kNN
  if score quantile above threshold then
    create explanation using distance
    report anomaly
  end if
  add record to model
end while
```

time emits a record with several features. In the second step we extract standard *BOW* (bag-of-words) feature vectors. There are several possible ways of how to extract features and how to weight each feature dimension. We chose to use a simple representation where we extract tags, such as *server=x* and *process=mytask.tsk*. We collect all these tags from the record and assign them weight 1. This means that individual lines can produce vectors of varying lengths, as they are not normalized. We could also normalize them or re-weight them using the TFIDF weighting scheme ([6]) to down-weight frequent tags.

To capture the wider context of each record, we aggregate a set of records within a time-window of predefined length and emit a combined record (simple normalized sum of all vectors) that represents that window. In the fourth step, we use the *k-nearest-neighbor* algorithm (kNN) to find which k windows from the past are the closest to current window. This unsupervised algorithm was chosen because it can handle skewed distributions very well.

The average distance of the k neighbors is used as an *anomaly score*: the further away an instance is to its nearest neighbor set, the more anomalous it is. The value of the parameter k is usually small, between 1 and 5. The best value depends on the data domain and should be determined by experimenting.

The absolute scale of the score may vary from problem to problem and also depends on the feature representation. For that reason we used a quantile based approach: we compare the anomaly score of a new instance with scores of recently observed data points (the size of the of recently observed data point set is controlled with a parameter *learning window*). If the score is higher than a large (example 0.999) fraction of scores (controlled by a parameter *nn_rate*), then we classify the instance as an anomaly. The quantile (1 - *nn_rate*) directly controls the detection rate (0.001 corresponds to classifying 0.1% of instances as anomalies) under the assumption that the data distribution is stationary.

2.3.1 Tokenization

When tokenizing input tests we have several options. If we know that some common patterns exist on how the special entities are marked inside the text, we can extract them and create useful features for *BOW* step. For instance we can use message text directly to create BOW. Alternatively, we can just find identifiers of the origin of the message (e.g. process name, method name, class name, page name etc.) and use these instead of whole texts.

Table 1: Synthetic Data - window = 1 min

NN rate	Precision	Recall
0.003	0.06	1.00
0.001	0.15	0.66
0.0005	0.23	0.66

Table 2: Synthetic Data - window = 1 h

NN rate	Precision	Recall
0.05	0.07	0.66
0.03	0.14	0.66
0.01	0.33	0.33

2.3.2 Anomaly explanation

We construct the explanation for an individual alert by finding its nearest neighbor and subtracting one vector from the other and squaring each element of the resulting vector. Each dimension is thus attributed with an *"anomalousness"* score, and the highest scoring dimensions contributed the most to the distance to the nearest neighbor. The explanation to the user contains a sorted list of highest scoring dimensions (clipped to avoid information overload).

3. EXPERIMENTAL RESULTS

3.1 Evaluation

In most real-world anomaly-detection cases we receive a dataset that is largely unlabelled. The labels we have usually denote some special catastrophic situations that users experienced and want to avoid in the future. The rest of the data can be assumed to be mostly *"normal"*, but unknown anomalies may also remain in the dataset. The standard metrics to evaluate the performance of an anomaly detector are *precision* (#true anomalies / #predicted anomalies) and *recall* (#predicted true anomalies / #true anomalies). Low precision translates to a higher burden on the user that inspects the anomalies (each inspection has some cost) and low recall translates to missing many anomalies and increasing risk. If we suspect that the dataset is not labelled completely (unlabelled anomalies present) the precision might be estimated pessimistically and recall might be measured optimistically. In such cases manual inspection of false positives might lead to discovering new relevant types of anomalies present in the dataset.

3.2 Synthetic data

We generated log files by simulating parallel execution of several processes, each having a specific pattern of writing to log. We then manually inject 8 instances of anomalous entries, 2 per week, each pair occurring within one minute.

We use 10 days for the kNN learning window length, so there will be no anomalies in the first 10 days. For the length of the input-grouping window, we experimented with two settings: 1 minute and 1 hour. In the former setting we have 6 original anomalies, but in the later, we only have 3 since each pair of adjacent anomalies (they are 1 minute apart) gets collapsed into the same hour. We set the parameter k to 1 - thus making the explanation of the anomaly very simple.

Table 3: Web logs - an anomaly explanation example

File	Val	Near	Contr
/shuttle/missions/sts-68/mission-sts-68.html	0.707	0	0.354
/images/NASA-logosmall.gif	0	0.505	0.180
/htbin/cdt_main.pl	0	0.416	0.122

Table 1 shows the results for windows of length 1 minute. When parameter *nn_rate*, which controls the sensitivity to outliers, was set to 0.03, it correctly detected all 6 manually inserted anomalies. Table 2 shows the results for windows of length 1 hour. This granularity of data is too coarse and hides certain anomalies that remain undetected.

3.3 Web-server logs

We analyzed browsing pattern logs from a production web server where the logs contained information on web-page and file requests. The specific feature of this web site is that it is almost completely static - there are almost no new pages being added to it, so the browsing patterns should be relatively constant. The dataset spanned a period of one month. We set the summarization window length to 5 minutes and the kNN comparison window length to 10 days. The parameter k was again set to 1 and the anomaly rate was set to 0.001. The parameters were hand-tuned.

When analyzing such data the anomalies that we are interested should capture both system failure (malfunctioning software, infrastructure failure) as well as malevolent behavior (denial-of-service attack (DoS), a hacker-induced scanning for exploits). In these cases the system should ideally produce anomalies with **strong dimensional outliers**.

We manually inspected anomalies where the strongest dimension contributed more than 30% of the total anomaly score. An example is shown in Table 3, where columns Val means value in current record, $Near$ means value in the nearest record and $Contr$ means contribution to the total distance. It turns out that these anomalies corresponded to the rare events when new content was added to the web page. No DoS or hacker attacks were detected in the observed time period. So the vector dimension for the file $(mission-sts-68.htm)$ was $Val = 0.707$. This file was not present in the nearest record ($Near = 0$) and this dimension contributed 0.354 of the total distance between this record and the nearest one.

4. CONCLUSIONS AND FUTURE WORK

We presented a novel combination of known approaches to anomaly detection using techniques developed in the field of text mining. Using these we were able to extract valuable information from raw textual log files that are normally only used for manual inspection and forensic analysis.

Our algorithm is based on processing log files, but many other sources of information can be used to extract anomalies in modern IT systems. Our long-term goal is to design what we call *Full-spectrum anomaly detection system* (FSADS) that will be able to import many different types of data streams, covering a wide range of aspects of an IT system (inputs, outputs, internal performance, database performance, network communications etc.). After each single source of data is analyzed and anomalies are extracted, the next step in FSADS will correlate them, determine critical signals (which anomalies have a high impact on system?), indicate possible root causes (what might have caused a particular anomaly?) and give predictions (what may follow after detecting a particular type of anomaly?).

The popularity of deep learning techniques ([7]) is also felt in the anomaly-detection field and we plan to study their application to multivariate analysis. Most often, autoencoders are used to create compact descriptions of the data and may also be used to highlight the dimensions with high reconstruction error. Another interesting approach is to use generative adversarial networks ([2]), where two neural networks are contesting in a zero-sum game: one network generates "normal" candidates and the other one discriminates between generated examples and true examples. We currently see two major challenges for broader use of deep neural networks in anomaly detection systems. The first one is the ability to **explain the results** to the end-user in an actionable way. The second one is processing of **stream data** and updating the model on-the-fly. Currently, existing and simpler techniques (e.g. kNN, clustering, statistical tests) provide much better support for these requirements. However, the linearity in describing the feature space is an issue that we would like to address in the future and deep neural networks present a promising approach.

5. REFERENCES

[1] C. C. Aggarwal. *Outlier Analysis*. Springer New York, New York, New York, 2013.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[3] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 219–230, New York, NY, USA, 2004. ACM.

[4] X. Liu, P. Zhang, and D. Zeng. Sequence matching for suspicious activity detection in anti-money laundering. In C. C. Yang, H. Chen, M. Chau, K. Chang, S.-D. Lang, P. S. Chen, R. Hsieh, D. Zeng, F.-Y. Wang, K. M. Carley, W. Mao, and J. Zhan, editors, *ISI Workshops*, volume 5075 of *Lecture Notes in Computer Science*, pages 50–61. Springer, 2008.

[5] C. Phua, V. C. S. Lee, K. Smith-Miles, and R. W. Gayler. A comprehensive survey of data mining-based fraud detection research. *CoRR*, abs/1009.6119, 2010.

[6] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.

[7] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang. Deep Structured Energy Based Models for Anomaly Detection. *ArXiv e-prints*, May 2016.

Usage of SVM for a Triggering Mechanism for Higgs Boson Detection

Klemen Kenda
Jožef Stefan Institute, Aritificial Intelligence Laboratory
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
klemen.kenda@ijs.si

Dunja Mladenić
Jožef Stefan Institute, Aritificial Intelligence Laboratory
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT
Real-time classification of events in high energy physics is essential to deal with huge amounts of data, produced by proton-proton collisions in ATLAS detector at Large Hadron Collider in CERN. With this work we have implemented a triggering mechanism method for saving relevant data, based on machine learning. In comparison with the state of the art machine learning methods (gradient boosting and deep neural networks) shortcomings of Support Vector Machines (SVM) have been compensated with extensive feature engineering. Method has been evaluated with special metrics (average median significance) suggested by the domain experts. Our method achieves significantly higher precision and 8% lower average median significance than the current state of the art method used at ATLAS detector (XGBoost).

Categories and Subject Descriptors
H.2.8 [**Database Applications**]: Data mining, scientific databases

General Terms
Algorithms, Measurement, Experimentation.

Keywords
Support Vector Machine, Gradient Boosting, Classification, High Energy Physics, Higgs Boson

1. INTRODUCTION
ATLAS and CMS experiments have announced discovery of the Higgs boson in 2012 [1]. Experiments have been conducted on Large Hadron Collider (LHC) in CERN in Geneva. The discovery has been succeeded by a Nobel Prize in Physics, awarded to François Englert and Peter Higgs. The existence of the particle, which gives mass to other elementary particles, has been predicted around 50 years ago [6][7][8].

Higgs boson decays almost instantly and can be observed only through its decay products. Initially the particle has been observed through $H \to \gamma\gamma$, $H \to Z^0Z^0$ and $H \to W^+W^-$ decays. These decays leave a signature that is relatively easy to interpret. The next steps required analysis of Higgs boson decay into fermion pairs: τ leptons or b quarks.

In this paper we focus on a special topology of $H \to \tau^+\tau^-$ decay [9]. Due to similarities with other decays this particular decay is very difficult to classify. Distinguishing background (events that do not belong to the $H \to \tau^+\tau^-$ decay) from signal (events that belong to Higgs boson decay) requires the use of state of the art machine learning methods.

In the past the task has often been solved with simple cut-off techniques based on statistical analysis, performed by expert users. Today advanced classification methods based on machine learning are used regularly.

State of the art methods for this type of problems include deep neural networks and gradient boosting [10][11][12]. Experiments at CERN prefer the usage of gradient boosting classifiers as they are able to evaluate large amounts of data (more than 20×10^6 events/s) [4].

The success of both methods is based on their intrinsic property of introducing non-linearity into the system. In our work we want to compare basic linear methods and Support Vector Machines (SVM) with different kernels to the state of the art models. Additionally, we want to enrich the data by intensive feature engineering.

The results of feature engineering can be used for further physical interpretation of relevant physical phenomena.

2. DATA
Dataset has been made public by the ATLAS collaboration for the Higgs Boson Machine Learning Challenge on Kaggle in 2014 [3]. It contains data from the ATLAS detector simulator (real labelled data would be impossible to obtain). The winning method from the challenge is being used in the ATLAS experiment today [4].

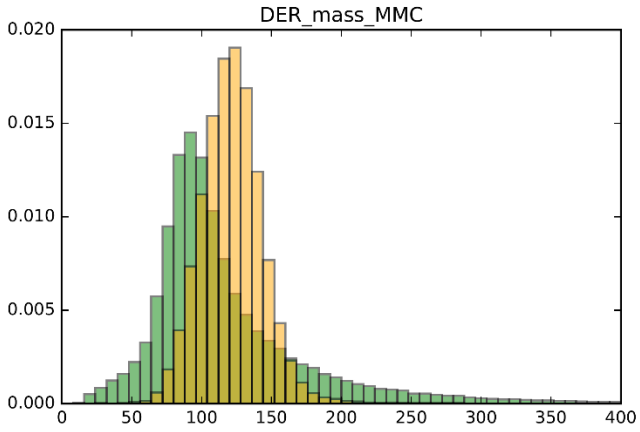

Figure 1. Distribution of signal (yellow) and background (green) according to most informative attribute DER_mass_MMC (mass of Higgs boson candidate) [5].

2.1 Data Description
Dataset consists of 250,000 instances. 85,667 represent signal, 164,333 represent background. Each instance consists of 32 attributes and 1 target variable. All the attributes are numerical (continuous), target variable is nominal (binary). 2 of the attributes

should not be used for classification purposes, as they represent id of the instance and probability of such an event happening in the experiment [4].

There are missing values in the data. 11 attributes could not always be measured due to characteristics of the detector. Distribution of the missing values is different for signal and for background.

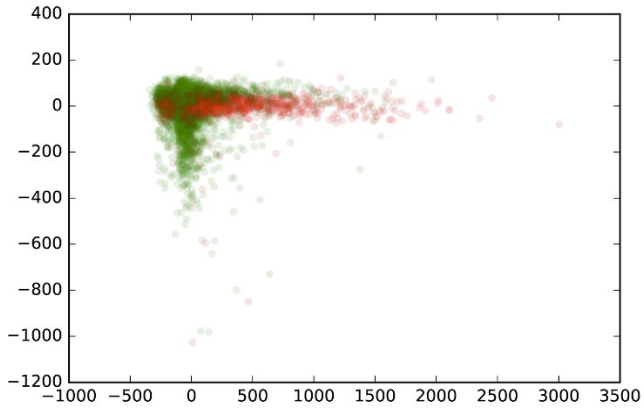

Figure 1. Plot of 1st PCA component against the 3rd. Red dots represent signal instances, green dots represent background instances [5].

The signal is limited to the events representing only one possibility for $\tau^+\tau^-$ pair decay [4].

2.2 Data Understanding

The main task of our method is to separate the signal from the background, based on the ATLAS detector measurements. As vast amounts of data (a few terabytes/day) are generated within the process it is crucial that only the relevant events are detected and stored [4].

Exploratory analysis has shown (see Figure 1) that this task can not be successfully accomplished with simple cut-off techniques based on a single attribute. Figure 2 depicting PCA components plot is a bit more promising as parts of phase space can clearly be assigned to one of the classes.

Attributes are divided into 3 groups. First group contains 18 primary attributes (measured in the detector), second group contains 12 derived attributes (relevant physical phenomena calculated from primary attributes) and 2 metadata values (weight and event id). Detailed exploratory data analysis can be found in [5].

2.3 Data Preprocessing

ATLAS detector enables good precision of all measurements, therefore expected noise in the data is very small and it can not be further filtered. Missing values have been dealt with in two different ways. Firstly – we used "replacement with average" strategy to fill in the missing data and secondly, we generated additional binary features, representing missing attribute values.

SVM expects input data to be normalized, therefore the features have been normalized with average and standard deviation values set to 1. Data transformation has been handled with Pandas library in Python.

2.4 Feature Engineering

The main task of our work has consisted of extensive feature engineering, where non-linear combinations of features were introduced to overcome the shortcomings of linear SVM in comparison with gradient boosting or deep neural networks.

We have built new features from original attributes by transforming them with some common functions like e^x, x^2, x^3, $\sqrt{x}$ and $log(x)$. Additionally we have used k-means clustering to generate an additional attribute (cluster id). All the generated feature sets are shown in Table 1.

Table 1. Attribute sets used for SVM.

Set	Description
1	Original feature set.
2	Added missing values.
3	Filtered missing values and all e^x derivatives.
4	Filtered missing values, e^x and all x^2 derivatives.
5	Filtered missing values, e^x, x^2 and all x^3 derivatives.
6	Filtered missing values, e^x, x^2, x^3 and all $\sqrt{x}$ derivatives.
7	Filtered missing values, e^x, x^2, x^3, $\sqrt{x}$ and all $log(x)$ derivatives.
8	Selection of most relevant transformations by one attribute.
9	Unfiltered set of transformations by one attribute.
10	Unfiltered set of $x_i x_j$.
11	Set of attributes by one of HiggsML winners (Tim Salimans, DNN).
12	Unfiltered set of $x_i^2 + x_j^2$.
13	Unfiltered set of $e^{x_i^2 + x_j^2}$.
14	Unfiltered set of $\sqrt{x_i^2 + x_j^2}$.
15	Unfiltered set of $(1 + x_i x_j)^2$.
16	Filtered set of transformations by 1 and 2 attributes.
17	(8) with k-means cluster id.

Filtering of the features has been done manually, with a simple cut-off technique based on feature importance as obtained from linear SVM model.

3. MACHINE LEARNING METHODS USED

Baseline experiments have been carried out with simple cut-off techniques and linear methods like logistical regression and Naïve Bayes classifier. As state of the art methods we included gradient boosting and gradient boosting adjusted for the approximate median significant metrics (see Section 3.2) [11].

We are proposing to use SVM method [12]. Linear SVM can be used for feature selection with large number of attributes. It can discover most relevant features in a large feature set.

3.1 Brief Description of SVM

In our setting we are solving a binary classification problem. Let us assume, that the classes are linearly separable in our space. In general, there are many different hyper planes that can separate the two classes. Support vector machine (SVM) method is also called maximum margin classifier. There exists only one hyper plane that maximizes margin between the two classes [12].

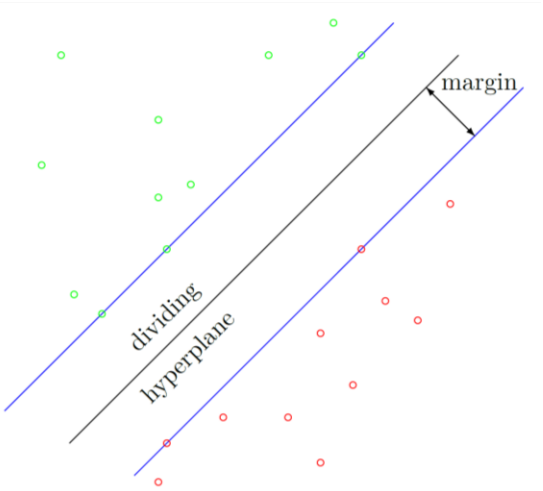

Figure 2. Maximum margin of dividing hyper plane in SVM [5].

SVM classifier is derived from maximization of the margin, which can be translated into minimization of $||w||^2$ [5][12]. As we are dealing with data sets, where classes are not separable, we need to consider a soft margin method that would take into account classification error. SVM is therefore solving minimization problem of $||w||^2 + C\sum_{i=1}^{n}\xi_i$, where ξ_i is a classification error metrics and C is a parameter that controls the influence of ξ.

3.2 Brief Description of the Evaluation Criteria

Evaluation of the results is to be done with measures derived from confusion matrix (accuracy, precision, recall, F_1). The evaluation metrics (*approximate median* significance) is defined as

$$AMS = \sqrt{2\left(s + b + b_{reg}\right)\ln\left(1 + \frac{s}{b + b_{reg}}\right) - 2s}$$

s represents sum of event probabilities of true positives (signal), b represents sum of event probabilities of true negatives (background), b_{reg} is set to 10 and represents a pre-set regularization parameter. The metrics favorizes recall before precision. In real setting this algorithm is used as a triggering mechanism for saving relevant data. Probability for a positive example in the real data is only around $p \approx 2 \times 10^{-5}$, therefore we do not want to lose many of them.

4. EVALUATION

Experiments have been carried out in Python. Data loading and cleaning has been accomplished with `Pandas` library, implementation of SVM, scaling and other methods have been taken from `scikit-learn` package. Default parameters for SVM have been used.

On our system SVM learning phase took ~1 hour. For time optimization purposes normal evaluation with training and test set has been performed. Training set consisted of 225,000 and test set of 25,000 instances.

Table 2. Evaluation of different attribute sets on SVM with linear kernel.

Attribute set	Prec.	Rec.	Acc.	F_1	AMS
1	0.665	0.548	0.749	0.600	1.999
3	**0.748**	0.655	0.805	0.698	2.526
4	**0.748**	0.654	0.805	0.698	2.528
5	0.740	0.657	0.802	0.696	2.478
6	0.743	0.683	0.809	0.712	2.547
7	0.734	0.690	0.807	0.711	2.516
8	0.732	0.670	0.802	0.700	2.482
10	0.744	0.705	0.815	0.724	**2.582**
11	0.694	0.584	0.768	0.634	2.201
12	0.744	0.705	0.815	0.724	**2.583**
13	0.744	0.709	**0.816**	**0.726**	**2.581**
14	0.744	0.705	0.815	0.724	**2.583**
15	0.744	**0.710**	**0.816**	**0.726**	2.578
16	0.740	0.684	0.809	0.711	2.553

Results from extensive feature engineering are shown in Table 2. Linar SVM performed similar to linear baseline methods (logistic regression, Naïve Bayes). AMS score was ~2.00. Best feature sets for linear SVM were (10), (12), (13) and (14). These feature sets include two-attribute transformations, e.g., $x_i x_j$. It is interesting to notice that filtered feature sets performed slightly worse. Extensive feature generation achieved almost 30% better AMS results (1.999 on basic feature set compared to 2.583).

Table 3. Evaluation of different methods and attribute sets compared to baseline and state-of-the-art methods.

Method and attribute set	Prec.	Rec.	Acc.	F_1	AMS
simple window	0.560	**0.824**	0.716	0.667	1.579
log. reg. (1)	0.668	0.535	0.749	0.594	2.015
SVM-LIN (13)	0.744	0.709	0.816	0.726	2.581
GBC (8)	0.787	0.703	0.832	0.742	2.856
SVM-r (8)	**0.791**	0.718	**0.837**	**0.752**	2.940
opt. SVM-r (8)	**0.907**	0.446	0.793	0.598	3.451
XGBoost (1)	0.665	**0.806**	0.793	0.729	**3.735**

Table 3 contains results of baseline, state-of-the-art and the proposed SVM. Best feature sets for selected methods were chosen. Baseline methods are simple window (based on cut-off technique on candidate particle mass) and logistic regression. As state of the art methods we included: gradient boosting (GBC) and current state of the art (XGBoost, gradient boosting optimized for AMS).

Proposed methods are linear SVM, SVM with RBF kernel (SVM-r) and optimized SVM with RBF kernel (opt. SVM-r).

Usage of kernels (RBF and polynomial kernels have been tested) improved AMS score for another ~15%. Because of the nature of SVM kernels in this setting 2-attribute transformations were less efficient than 1-attribute transformations. Selection of most relevant transformations by 1 attribute (set (8)) gave the best results. Method behaved better than gradient boosting classifier (GBC) on the same training set. However, methods were not optimized to maximize AMS score. The difference however suggests that the usage of SVM might be a promising way to proceed.

Finally we optimized SVM with RBF kernel for AMS score and compared it to XGBoost method, which implements gradient boosting, optimized for AMS. Optimization has been done based on threshold for SVM confidence score. Our method performs approximately ~8% worse than the state of the art. There is, however, a big difference with XGBoost. Our method yields higher precision than the other methods and still preserves very high AMS score. The proposed method also performs ~20% better than other SVM based methods reported in HiggsML Challenge [3].

5. CONCLUSION

In our work we have examined the potential of SVM for a triggering mechanism in high-energy physics domain. With extensive feature engineering we have also provided an interesting input for high energy physics experts, where most effective generated features could be analyzed through domain knowledge.

Our method achieves more than 200% better AMS score compared to cut-off techniques, based on statistical approach. Further, our methods achieves ~20% better AMS score than other SVM based methods reported by HiggsML Challenge competitors, but performs ~8% worse than current state of the art (XGBoost). There is however a significant difference between our method and state of the art. Although achieving comparable AMS score, our methods achieves much better precision. This might make SVM based methods valuable members of an ensemble method.

Beside adding SVM methods to ensembles and trying to improve state of the art, further work could be done with adapting the SVM optimization to AMS metrics. In our work features were selected based on weight-importance. Often different transformations of the same attributes have been selected. Features that could improve our models only by little have potentially been left out. This should be studied further. Optimization of SVM parameters should also be performed.

6. ACKNOWLEDGMENTS

This work was partially supported by the Slovenian Research Agency.

7. REFERENCES

[1] G. Aad et al. *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*. Physics Letters B, 716(1):1 – 29, 2012.

[2] S. Chatrchyan et al. *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*. Physics Letters B, 716(1):30 – 61, 2012.

[3] HiggsML challenge. https://www.kaggle.com/c/higgs-boson, 2014. [Online; access April 20, 2016].

[4] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl in D. Rousseau. *The Higgs boson machine learning challenge*. In Workshop on High-energy Physics and Machine Learning, HEPML 2014, held at NIPS 2014, Montreal, Quebec, Canada, December 8-13, 2014 [30], pages 19–55.

[5] Kenda, K., Podobnik, T., Gorišek, A. *Uporaba metod strojnega učenja pri analizi podatkov, zajetih z detektorjem ATLAS*. Diploma thesis. 2016. Faculty of Mathemathics and Physics, University of Ljubljana

[6] P. W. Higgs. *Broken symmetries, massless particlees and gauge fields*. Physics Letters, 12:132–133, September 1964.

[7] F. Englert, R. Brout. *Broken Symmetry and the Mass of Gauge Vector Mesons*. Physical Review Letters, 13:321–323, August 1964.

[8] P. W. Higgs. *Broken symmetries and the masses of gauge bosons*. Phys. Rev. Lett., 13:508–509, Oct 1964.

[9] G. Aad et al. *Evidence for the Higgs-boson Yukawa coupling to tau leptons with the ATLAS detector*. JHEP, 04:117, 2015.

[10] T. Chen, T. He. *Higgs boson discovery with boosted trees*. In HEPML 2014, held at NIPS 2014, pages 69–80.

[11] T. Chen, C. Guestrin. *XGBoost: A scalable tree boosting system*. CoRR, abs/1603.02754, 2016.

[12] Boser, B. E., Guyon, I. M., Vapnik, V. N. *A training algorithm for optimal margin classifiers*. Proceedings of the fifth annual workshop on Computational learning theory – COLT '92. p. 144.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot in E. Duchesnay. *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12:2825–2830, 2011.

A methodology to evaluate the evolution of networks using topological data analysis

Joao Pita Costa * ** and Tihana Galinac Grbac *

***Abstract*—Networks are important representations in computer science to communicate structural aspects of a given system of interacting components. The evolution of a network has several topological properties that can provide us information on the network itself. In this paper, we present a methodology to compare the the topological characteristics of the evolution of a network, encoded into a (persistence) diagram that tracks the lifetimes of those features. This will enable us to classify the evolution of networks based on the distance between the diagrams that represent such network evolution. In that, we also consider complex vectors that bring a complementary perspective to the distance-based classification that is closer to the computational methods, aims to enhance the computational efficiency of those comparisons, and that is by itself a source of open research questions.**

I. Introduction

A. Comparing the topology of the evolution of networks

Networks that change as a function of time - known as evolving networks - are a natural extensions of undirected graphs (i.e., standard (static) networks). Almost all real world networks evolve over time, either by adding or removing nodes or edges. The example of scientific collaboration analysis, such as in the example of Figure 1 shows such a network.

The analysis of the evolution of a network is a matter of interest transversal to many fields of knowledge, from social network analysis and scientific collaboration to computational biology. A standard example is the network dynamics of a social network such as Twitter should consider an evolution through time where new nodes come up as new members join, and new edges are created mirroring the new relationships between members that appear [1]. Often all of these processes occur simultaneously in social networks.

Collaborative networks are a prime example of evolving networks, where nodes represent authors and edges represent scientific collaborations. This is illustrated in Figure 1. It shows the plot of three phases of an instance in the scientific community in Slovenia [13] using ScienceAtlas, a web portal available at scienceatlas.ijs.si integrating data about 35272 researchers, 5716 projects, 82905 publications and 17190 video lectures. This too allows visualizing collaboration and competences of the researchers [14].

* University of Rijeka, Croatia,
** Quintelligence, Ljubljana, Slovenia

Category: G.2.2: Mathematics of Computing: Discrete Mathematics applications
Keywords: Network, undirected graph, persistent homology, computational topology, persistence diagram.

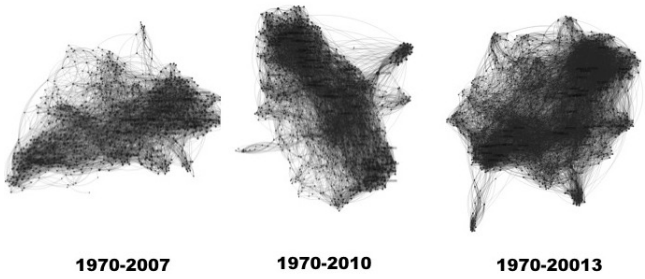

Fig. 1. The evolving network ScienceAtlas of the collaborations in scientific works by Slovenian researchers, evolving over a 9-year period with 3-year leaps. Each node represents an author and each edge represents a collaboration. The nodes with degrees smaller than 20 are filtered out so that the networks are not too large to be visualized.

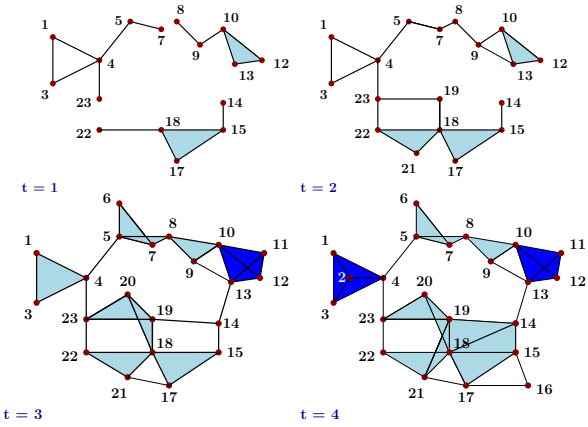

Fig. 2. An example of an evolving network based on an undirected graph that is growing over time by adding new edges and nodes at each step.

A biological network, on the other hand, is an approximate mathematical representation of connections found in ecological, evolutionary, and physiological research, among others. An example of a relevant application of such analysis of biological networks with respect to human diseases is network medicine. It considers networks in biological systems containing many components connected within complicated relationships but organized by simple principles [1].

In this paper, we focus on the comparison of the evolution of two (or more) given networks. Our approach considers topological data analysis (TDA), allowing us to encode the topological features of the corresponding evolving networks onto diagrams, and using standard methods to compute distances between them. In that, we can classify networks according to the distance between the topology of their evolution.

The TDA approach to the study of networks is not itself new. It had several widespread applications from collaboration networks [3] to functional brain networks [15]. There are several ways of considering a height function in a network including: (i) considering weights in the edges of the network - *weighted network* - and then having the function built by threshold those weights [17]; (ii) measuring the distance from each node to each other by counting the minimal number of edges between them and then building the height function based on that distance [11]; among others. This permits us to use persistent homology over such height function. Another possibility is to consider the maximal cliques as the simplicial complexes (named *clique complexes*) that feed the persistence algorithm and proceed with the computation directly over that [5]. We used the latter approach to compute the persistence of the networks generated for the purpose of this paper.

B. Basic notions in persistent homology

Topology is a field of study in mathematics concerned in the quality aspects of an object. It focus on the properties that are preserved through deformations, twistings, and stretchings of the given continuous objects (e.g. linear maps) in multidimensional scenarios. Computational topology takes advantage of simplification methods (e.g. the triangulation of a space) to permit the computation of topological invariants. One of those computations is homology which evaluates the connectedness of, e.g., a network at different dimensions separately. Thus, homology is a natural choice when it comes to the study of the topology of a network. Now, if we consider a monotone function describing the time variable in, e.g., an evolving network, we can track its homology changes. This notion is known as persistent homology and is rooted in TDA, allowing for retrieving the essential topological features of an object [2]. Formally, persistent homology computes the topological features of a growing sequence of spaces $\emptyset = X_0 \subseteq X_1 \ldots \subseteq X_n = X$, known as a *filtration* of the space X. $H_i(X)$ is the i-th homology group of X, with an associated i-th Betti number of X,β_i, corresponding to the measure of connectedness in the i-th dimension (cf. [5]). Using the inclusion maps $X_j \rightarrow X_{j+1}$ we can identify copies of Z_2 in the homology groups $H_i(X_j)$ and $H_i(X_{j+1})$ of a filtration and track where the homology changes. We do that by recording when a new copy appears (i.e. "is born"), and when an existing copy persists or merges to an existing one (i.e. "dies"). That persistence of the topological feature is tracked by a lifetime bar (as shown in Figure 3) that can be equivalently represented by an ordered pair (x, y), where x is the birth time and y is the death time. The multiset of all such points exists in the plane subset defined by $0 < x < y$ that encodes the topology of a space and is known as *persistence diagram*. Several topological features can have the same lifetimes and therefore some of the points in the persistence diagram are repeated in the multiset. We refer to their amount as *multiplicity*. We consider the infinite points in the diagonal as points of the persistence diagram with null lifetime. The standard method to compare two persistence diagrams - called *bottleneck distance* - measures the cost of finding a correspondence between their points. It identifies the closest matching elements of each persistence diagram and determines the global distance based on what is the biggest of those distances. The cost of taking a point $p = (p_1, p_2)$ to a point $q = (q_1, q_2)$ in R^2 is given by the L_∞ norm $\|p - q\|_\infty = max|p_1 - q_1|, |p_2 - q_2|$. Then, the bottleneck distance between persistence diagrams X and Y is computed by taking the infimum over all such matchings, i.e., $d_B(X, Y) = \inf_\eta \sup_{x \in X} \|x - \eta(x)\|_\infty$, where the infimum is taken over all bijections η from X to Y. Each point with multiplicity k in a multiset is interpreted as k individual points, and the bijection is interpreted between the resulting sets [4].

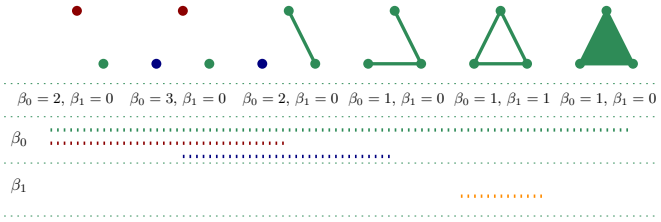

Fig. 3. The computation of persistent homology on a simplicial complex changing in time [3]. The colors correspond to the topological features to which the lifetime is tracked in the persistence barcode below. The Betti numbers indicate the number of connected components β_0 in dimension zero, the number of holes in the network β_1 in dimension one, and the number of tunels and voids β_2 in dimension two.

C. The motivation of EVOSOFT

Nowadays, software systems start to interconnect to provide new and innovative applications and services that drives new development opportunities in all domains. Therefore these software systems have gradually evolved into large scale complex systems and we lack models for their further management and evolution. One of the key aspects of such systems is the ability to model and predict their behaviour to achieve the required quality of operations to fulfill human expectations in all domains. In that, the project Evolving Software Systems: Analysis and Innovative Approaches for Smart Management (EVOSOFT) aims to understand how abstract software structures can be used to model global system properties (e.g. fault distributions). Understanding how to use software structure to model fault distributions can help us to improve system reliability. EVOSOFT observes software structure as networks with nodes representing various software functions that are interconnected to each other by function calls. In particular, a software graph structure considers nodes as program functions (e.g. classes in object oriented paradigm, functions or modules in functional programming) and edges as function calls or signals transferred in communication among these program functions. EVOSOFT aims to observe how large software systems evolve from version to version, and understand the relationship between the change in software structure during its evolution, and the change in software fault distributions across its structure. Previous empirical studies in [9], [10], [18] show that communication structures among the program functions significantly influence system fault distributions. This is what motivated us to further explore this relationship.

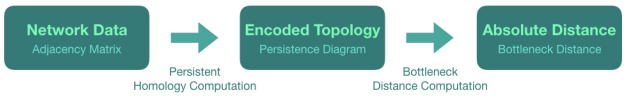

Fig. 4. The methodology diagram to encode and compare the topology of the evolution of two (or more) networks using TDA. It considers three phases: (i) the data, where we input the evolving network represented by an adjacency matrix; (ii) the topology, where a persistence diagram encodes the topological invariants of the network evolution; and (iii) the distance, held between persistence diagrams representing the topology of given evolving networks.

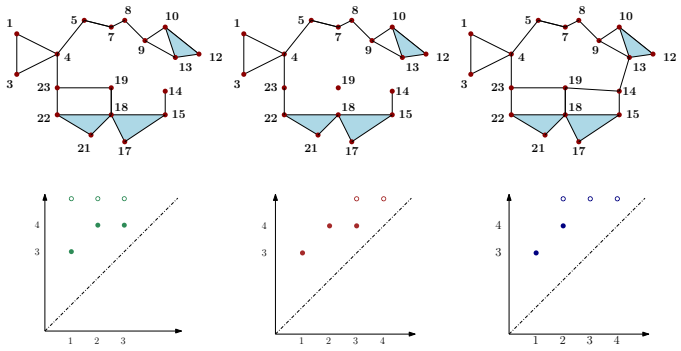

Fig. 5. The presented methodology applied to the comparison of the evolution of three networks sharing the same evolution as the network represented in Figure 2 but with differences in phase 2. Each evolving network is associated with one persistence diagram that encodes its topology. This permits us to visualize the relevant topological features of the evolution of the networks.

II. Use-case methodology to encode and compare evolving networks

The problem of tracking and comparing the evolution of networks can be very demanding and complex due to the combinatorial properties of networks. In the following section we shall describe the methodology diagram to encode a compare the topology of the evolution of networks (as illustrated in Figure 4). It considers persistent homology to encode the topological features of the evolution of a network using persistence diagrams. In that, we first provide the evolving network given by one Boolean adjacency matrix for each phase of network development. We then compute the persistent homology of the evolving network by feeding the concatenated matrices a suitable algorithm. It will encode the topology of each evolving network, representing it by one unique persistence diagram each. Finally, we measure the bottleneck distance between persistence diagrams to identify how close are the evolving networks to each other based on their topology.

To the purpose of this paper, we used the software library *Perseus* [16] to compute the homology of a the evolving network represented in Figure 2, given by the graph's Boolean adjacency matrix. The network is provided to *Perseus* as a list of cliques including the time of appearance. The output of that procedure is a persistence diagram that corresponds to the topological changes within the evolution of that network. The evolving network A on the left has four stages as illustrated in Figure 2. The evolving networks B and C are variations of the evolution of the end network in A with different phases at time $t = 2$, as represented in 5.

To compare the evolution of networks we consider the distance between the corresponding persistence diagrams, using the bottleneck distance. This permits a fast computation of the distance between the (persistence diagrams representing the) topology of two evolving networks. In the case of the persistence diagrams encoding the topology of evolving networks A, B and C represented in Figure 5, we get $d(A, B) = 0$ and $d(B, C) = d(A, C) = 1$. This discards the points with infinite persistence that are less relevant when considering dimension 1 diagrams. The computations were done using the TDA package available in R [7]. In this example we can explore the distance between several possible evolution of a network. In it, shows how TDA can contribute to better understand the behavior of a certain network.

III. The EVOSOFT Experiments

For the purpose of this research we will use the EVOSOFT motivation to generate networks that fit that scenario and allow us to compare the evolution of networks in that context. In these preliminary experiments we shall consider data representing the evolution of networks based on the empirical analysis of the evolution of complex software systems [8].

In these experiment we will generate networks with labeled nodes - not ordered pairs in $\mathbb{R}^2$ - and extract all maximal cliques from it. The maximal cliques serve us to construct clique complexes with which we are able to later on compute the topology of those networks. In these experiments we shall obtain the EVOSOFT evolving networks provided by their graph's Boolean adjacency matrix. Those matrices must be consistent with the evolution of the network in the sense that existing maximal cliques in phase i must maintain or enlarge in the phase $i + 1$ during the updates of a software version. The persistence diagrams computed by *Perseus* shall exhibit the encoded topology of evolving networks corresponding to different pieces of software.

The comparison between the topology of a pair of evolving networks given by the adjacency matrix is given by the bottleneck distance between the corresponding diagrams. That distance can be computed using the R library [7]. When considering other evolving networks we can calculate the pairwise distance between all of them and consider single linkage clustering based on this metric (as in earlier TDA applications to gene expression data as in [12]) to allow classification based on the topology of network evolution.

IV. Comparison through complex vectors

A possible algebraic representation of persistence diagrams is offered by complex polynomials. The method layed out in [6] can lead to avoid tedious and less meaningful computations of bottleneck distance, since far polynomials represent far persistence diagrams (the converse is known not to be true). A fast comparison of the coefficient vectors can reduce the size of the database to be classified by the bottleneck distance. We can then focus on close persistence diagrams for which we want to calculate precise measures. This should complement existing methods, rising the efficiency of computations for large evolving networks. Given a persistence diagram D described by

its points $p_1 = (u_1, v_1), \ldots, p_s = (u_s, v_s)$ with multiplicities $r_1, \ldots, r_s$, respectively, the method considers complex numbers $z_1 = u_1 + iv_1, \ldots, z_s = u_s + iv_s$. This allows us to associate to D the complex polynomial $fD(t) = \Pi_{j=1}^{s}(tz_j)^{r_j}$ where r_j is the multiplicity of the point p_j. It was shown in [6] that the first k coefficients are the ones carrying most of the relevant information and, therefore, the choice of a threshold k can reduce the computational complexity.

The unpublished 2-part algorithm by the authors of [6] permits us to input a persistence diagram in order to compute a complex vector out of it. Then the same algorithm compares two complex vectors corresponding to two persistence diagrams to output a float corresponding to the distance between those vectors. At the moment, this approach to convert persistence diagrams into complex vectors can be applied only when neglecting points with infinite persistence. In the running example we get the polynomial $p_A = (t - 1 - 3i)(t - 2 - 4i)(t - 3 - 4i) = p_B$ and $p_C = (t - 1 - 3i)(t - 2 - 4i)^2$, not considering points of infinite persistence. We then develop the polynomials to identify their coefficients into a complex vector. The distance between the three complex vectors corresponds to a basic classification of the given evolving networks. This is not a dense case where we would need additional tools like complex vectors. Though, real life examples of evolving networks are appropriate cases of such needs due to their inherent complexity.

V. Conclusions and further work

In this paper we have discussed the topological data analysis of evolving networks. In that we presented a method to encode the topology of the evolution of a given network through a persistence diagram, and its potential for a classification based on a chosen distance between diagrams. The inherent complexity of an evolving network demands for the data simplification methods to be available and appropriate to the nature of the considered object. In that, the TDA-based methodology in this paper can contribute to the analysis and interpretation of evolving networks and their behaviour. The experiments in real data are valuable to improve this method. In that, the collaborations with the earlier mentioned *Slovenian Science Atlas* would be welcome, allowing us to further explore the interpretation of the topology of the evolution of these collaborative networks and the distance between them. Further work includes the processing of EVOSOFT existing networks, as well as the interpretation of results in the context of that field of knowledge. It can provide new challenges specific to the available data and to its role and usage in the field. In particular, the interpretation of the persistent topological features captured in EVOSOFT experiments represents a relevant open problem that requires a deeper analysis based on the EVOSOFT expertise and the manipulation of the topological results. Lastly, the mathematical development of the complex vector method, that contributes to the study of evolving networks in general, is a rather computational method that is suitable to the application of compatible algorithms, allowing potential engineering applications. Moreover, it is itself a great source of open mathematical problems that we shall consider in further research (e.g. stability [5]).

Acknowledgment

The authors would like to thank to Barbara di Fabio for the usefull discussions on complex vectors and advice on further research, and to Primož Škraba for his comments and suggestions. The first and second authors would like to thank to the support of the Croatian Science Foundation's funding of the project EVOSOFT (UIP-2014-09-7945). The second author would also like to thank to the support by the University of Rijeka Research Grant 13.09.2.2.16 funding.

References

[1] A. L. Barabási. Network medicine - from obesity to the diseasome. *New England Journal of Medicine*, 357(4):404–407, 2007.

[2] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[3] C. J. Carstens and K. J. Horadam. Persistent homology of collaboration networks. *Mathematical problems in engineering*, 2013.

[4] H. Edelsbrunner and J. Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[5] H. Edelsbrunner and J. L. Harer. *Computational Topology*. American Mathematical Society, Providence, RI, 2010.

[6] B. D. Fabio and M. Ferri. Comparing persistence diagrams through complex vectors. *In International Conference on Image Analysis and Processing*, pages 294–305, 2015.

[7] B. T. Fasy, J. Kim, F. Lecci, and C. Maria. Introduction to the R package TDA. *arXiv:1411.1830*, 2014.

[8] T. Galinac and S. Golubić. Project overlapping and its influence on the product quality. In *Proceedings of the 8th International Conference on Telecommunications, 2005. ConTEL 2005.*, volume 2, pages 655–662, June 2005.

[9] T. G. Grbac and D. Huljenic. On the probability distribution of faults in complex software systems. *Information & Software Technology*, 58:250–258, 2015.

[10] T. G. Grbac, P. Runeson, and D. Huljenic. A second replicated quantitative analysis of fault distributions in complex software systems. *IEEE Trans. Software Eng.*, 39(4):462–476, 2013.

[11] D. Horak, S. Maletić, and M. Rajković. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(3):P03034, 2005.

[12] M. Juda. Topological structures in gene expression data. *unpuplished work presented at the Genetic Analysis Workshop 19*, 2014.

[13] M. Karlovčec, B. Lužar, and D. Mladenić. Core-periphery dynamics in collaboration networks: the case study of slovenia. *Scientometrics*, 109.3:1561–1578, 2016.

[14] M. Karlovčec, D. Mladenić, M. Grobelnik, and M. Jermol. Conceptualization of science using collaboration and competences. *The Electronic Library*, 34.1:2–23, 2016.

[15] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee. Discriminative persistent homology of brain networks. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 841–844, 2011.

[16] V. Nanda. Perseus, the persistent homology software. http://www.sas.upenn.edu/~vnanda/perseus. Accessed: 2017-09-05.

[17] G. Petri, M. Scolamiero, I. Donato, and F. Vaccarino. Topological strata of weighted complex networks. *PloS one*, 8(6):p.e66506, 2013.

[18] J. Petric and T. G. Grbac. Software structure evolution and relation to system defectiveness. In *18th International Conference on Evaluation and Assessment in Software Engineering13-14, 2014*, pages 34:1–34:10, 2014.

Improving mortality prediction for intensive care unit patients using text mining techniques

Primož Kocbek[1], Nino Fijačko[1], Milan Zorman[2], Simon Kocbek[1,3], Gregor Štiglic[1,2]

[1]Univerza v Mariboru Fakulteta za zdravstvene vede, Maribor, Slovenija, +386 2 300 47 00

[2]Univerza v Mariboru Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, Slovenija, +386 2 220 7000

[3]Kinghorn Centre for Clinical Genomics, Garvan institute of Medical Research, Sydney, Australia, +61 (02) 9295 8100

{primoz.kocbek, nino.fijacko, milan.zorman, gregor.stiglic}@um.si, skocbek@gmail.com

ABSTRACT

Numerous severity assessment scores for estimation of in-hospital mortality in Intensive Care Unit (ICU) have been developed over the last 40 years. In this study, we predicted 1-month mortality in chronic kidney disease (CKD) patients using the open Medical Information Mart for Intensive Care III (MIMIC III) database. Additionally, we observed the improvement in predictive performance and interpretability of the baseline model used in ICUs to a more complex model using simple features such as unigrams or bigrams, as well as advanced features extracted from textual nursing notes. For the latter, MetaMap extraction tool was used to extract medical concepts based on the Unified Medical Language System (UMLS) terminology. We used a logistic regression based classifier, built using Simplified Acute Physiology Score II (SAPS II), age and gender, as a baseline model. The baseline model was then compared to regularized logistic regression based classifier built using simple and more complex additional features. The Area Under the ROC Curve (AUC) results for the baseline predictive performance improved from of 0.761 to 0.782 when frequency of unigrams and bigrams were used to build the model. In a similar scenario, where unigram and bigram frequency was replaced with Term Frequency–Inverse Document Frequency (TF-IDF) based feature values, AUC further increased to 0.786.

This paper represents an opportunity in extracting new knowledge in the form of unigrams, bigrams or concepts extracted from textual notes accompanied by regression coefficient values that can be interpreted as relations between the features and the outcome. The combination of both can provide added value in decision support systems in ICU departments, where data is collected in electronic medical records (EMRs) in real-time.

Categories and subject descriptors

H.2.6 [Information Systems]: Database Machines

H.2.8 [Information Systems]: Database Applications

General Terms

Algorithms, Measurement, Documentation, Performance, Reliability, Experimentation.

Keywords

Text mining, ICU, database, machine learning, mortality prediction.

1. INTRODUCTION

Predicting the mortality of ICU patients is a complex and dynamic process. Critical illness severity assessment scores, such as Acute Physiology and Chronic Health Evaluation I-IV (APACHE), Sequential Organ Failure Assessment score (SOFA), Mortality Probability Model I-III (MPS), or Simplified Acute Physiology Score I-III (SAPS), help clinicians detect patient problems earlier, thus providing a better holistic treatment for patients and making patient care more cost-effective. The sheer number of different severity scores used is partly because of the quality of recorded data needed to calculate them. An example of such severity score is APACHE IV, which tends to have the best discriminative performance but the data needed to compute the score is complex and hospitals would need to develop a good enough high-quality database for analysis of risk stratification [1, 2].

The MIMIC III database [3], a free public-access intensive care unit repository, is widely used for predicting the mortality of ICU patients, where developers provided several severity scores for the database (e.g., OASIS, SAPS, SAPS II, SOFA), but also noted that for APACHE IV the coding of the diagnostic component is difficult and might lack accuracy [4]. Part of MIMIC III are free text nursing notes, which represent a good candidate source of information for mortality risk prediction, as they contain a detailed and regularly-updated record of the interventions performed, medications administered, vital signs, and physical examination findings, all of which carry highly specific information about the patient's dynamic physiological state and eventual outcome [5]. Because such data is unstructured, our purpose is knowledge discovery where we observe the improvements in predictive performance and interpretability of predictive models based on additional features extracted from nursing notes collected in EMRs. More precisely, we aim to predict one month mortality in CKD (ICD 9 code 585.x) patients and compare the improvement of the baseline model performance by including simple features such as unigrams or bigrams as well as more advanced features extracted from text in the form of medical concepts using MetaMap [6] to define mapping between textual notes and UMLS terminology.

2. METHODS

The data were obtained from the MIMIC III database, version 1.3, to select 58,976 hospitalizations for 46,520 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center in Boston between 2001 and 2012. The database included 26 linked tables, which can be merged mostly by patient or hospitalization identification numbers. Our focus were nursing notes (i.e., free text

notes from patients with CKD diagnosis), where we excluded hospitalization of patients that died within 24 hours of admission and nursing notes that were not fully updated, where duplication of data was likely. That left us with 10,867 nursing notes from 4,381 hospitalizations. The first nursing note was taken on average 7.8 hours after admission (85.2 % hospitalizations have at least two notes), second taken on average 14.7 hours after admission (38.1 % had at least 3 taken) and the third one was taken on average 17.5 hours after admission. A slight majority of hospitalized patients were male (59.4 %), with an average age of 65.6 (Standard Deviation (SD) 15.2)) and a 13.4 % mortality rate (death during or up to one month after the hospitalization was recorded).

Developers of the database also included source code for calculation of several severity scores (i.e., OASIS, SAPS, SAPS II and SOFA), and we selected SAPS II as the main feature in the baseline model, since it is used on daily bases in hospitals. The input features of our baseline model consisted of SAPS II score, age and gender.

Nursing notes were initially processed using traditional text extraction algorithms with stemming and removal of stop words, which produced 51,680 unique unigrams and 363,055 unique bigrams. Both frequency and TF-IDF tables were prepared. The text from the nursing notes was also processed using the MetaMap tool from the US National Library of Medicine. MetaMap identifies and normalizes biomedical terminology from the Unified Medical Language System (UMLS). Binary representations of Bags of Phrases (BOP) identified by MetaMap, and their UMLS (Concept Unique Identifiers) CUIs were used as features for the classifier. Space characters in phrases were replaced with underline character. Word sense disambiguation was used to distinguish similar words with same structure. In addition, phrases were marked with whether the associated concepts were found in a positive or negative context. To identify the polarity of phrases (negative or positive), the NegEx module [7] of MetaMap was enabled in order to identify the polarity of phrases (negative or positive). NegEx implements a simple algorithm that contains several regular expressions indicating negation, filters out sentences containing phrases that falsely appear to be negation phrases, and limits the scope of the negation phrases.

For better understanding, we provide a short example of MetaMap-annotated phrases from part of the sentence "URINE MICROSCOPY" (Table 1).

Table 1. Example of MetaMap-annotated phrases from part of the sentence "URINE MICROSCOPY"

Meta Candidates	
Score	**Matched concept**
1000	C0430397: Urine microscopy (Microscopic urinalysis) [Laboratory Procedure]
861	C0026018: Microscopy [Laboratory Procedure]
789	C0205288: Microscopic [Qualitative Concept]
694	C0042036: Urine [Body Substance]
694	C0042037: Urine (In Urine) [Functional Concept]
694	C2963137: Urine (Portion of urine) [Body Substance]
Meta Mapping	
Score	**Matched concept**
1000	C0430397: Urine microscopy (Microscopic urinalysis) [Laboratory Procedure]

The *meta candidates* are all discovered mappings that are ordered according to an evaluation metric described in [7], while *meta mappings* represent the selected phrases which finally represent features in our model. Please note that several meta mappings may be found in a sentence. Parentheses contain the concept's preferred name while square brackets contain the concept's semantic type.

One of our goals was interpretability and avoidance of over-fitting, therefore we restricted model building to regularized linear models, further we narrowed the selection to L1 regularization models or least absolute shrinkage and selection operator (lasso), which includes feature selection functionality which was needed due to high number of features in our datasets [8]. We expanded on the work of Marafino et al. [4], where they used the MIMIC II dataset and predicted mortality via stochastic gradient descent-based classifiers with TF-IDF on the extracted unigrams and bigrams for patients that died during the given ICU stay. All experiments were implemented in R language and environment for statistical computing [9] using glmnet package [10] to build and validate predictive models.

3. RESULTS

Results presented in this section were obtained from four scenarios where we compared different combinations of two types of extracted features (n-grams versus concepts) and two types of the extracted feature values (frequency vs. TF-IDF).

The baseline classifier with basic features (SAPS II score, age and gender) to predict mortality was used to evaluate the performance gain when more complex classifiers were built. Initially, we were interested in measuring the improvement of the baseline predictive performance by adding unigram and bigram features to SAPS II, age and gender of the patients. At the same time, we were observing the complexity of the predictive models by observing the number of features that were included in the models.

Second row in Table 2 presents the results when frequency of unigrams and bigrams were used to build the model. It can be seen that baseline predictive performance improved from the AUC of 0.761 to 0.782 when frequency information of unigrams and bigrams was used to build the model. With an improvement of more than 2% in AUC it is also important to note that in this scenario we obtained the simplest models in terms of interpretation with only 9 features on average. In a similar scenario where unigram and bigram frequency was replaced with TF-IDF based feature values resulted in further improvement with AUC of 0.786. In the next two experiments, we replaced unigrams and bigrams with UMLS concepts that were extracted from free text (nursing notes). Table 2 demonstrates further improvement of the classification performance as the AUC in case of TF-IDF increased to 0.789, while even further improvement with a mean AUC of 0.791 was measured in frequency based features experiment. Tables 3 and 4 provide more detailed overview of selected features along with the number of times a feature was included in a predictive model during 100 cross-validation runs. It can be seen that TF-IDF produced predictive models with a larger number of features and therefore represents a richer set of concepts that can be used to warn a medical expert of a potential threat to a patient. On the other hand, a complex model (in case of TF-IDF experiment, more than 35 features were used in a model on average) might represent a challenge for medical experts when interpretation of models is needed.

4. DISCUSSION AND CONCLUSIONS

In this paper, we observed the improvements in predictive performance and interpretability of predictive models based on new features extracted from nursing notes collected in EMRs. More precisely, we predicted one month mortality, at the end of 24-hours spent in the ICU, for CKD patients. The improvement of the

Table 2. Summary of predictive performance measures for different experiments using features extracted from nursing notes

	AUC	Sensitivity	Specificity	PPV	NPV	Selected features
Baseline (SAPS II)	0.761 [0.757-0.766]	0.712 [0.704-0.720]	0.687 [0.680-0.695]	0.283 [0.277-0.288]	0.933 [0.931-0.935]	1.0 [1.0-1.0]
Unigrams and bigrams (frequency)	0.782 [0.778-0.786]	0.727 [0.721-0.734]	0.714 [0.707-0.722]	0.306 [0.300-0.313]	0.939 [0.937-0.940]	9.1 [6.6-11.6]
UMLS concept mapping (frequency)	0.791 [0.787-0.795]	0.736 [0.728-0.745]	0.716 [0.708-0.724]	0.310 [0.304-0.316]	0.941 [0.939-0.943]	17.6 [14.8-20.5]
Unigrams and bigrams (TF-IDF)	0.786 [0.782-0.790]	0.733 [0.725-0.740]	0.712 [0.704-0.720]	0.306 [0.300-0.313]	0.940 [0.938-0.941]	25.1 [21.7-28.6]
UMLS concept mapping (TF-IDF)	0.789 [0.785-0.793]	0.747 [0.741-0.754]	0.700 [0.692-0.709]	0.302 [0.296-0.308]	0.942 [0.94-0.943]	35.5 [31.0-40.0]

baseline model (SAPS II, gender and age) in comparison to predictive models that included unigrams, bigrams as well as more advanced features extracted from text in the form of medical concepts using the MetaMap extraction tool was also observed.
The results show high level of predictive performance that can be compared to a similar study by Brabrand et al. [11] where it was shown that using clinical intuition of the admission staff produced comparable predictions in terms of AUC when identify patients at risk of dying. However, it has to be noted that Brabrand and colleagues did not focus on a specific group of patients.

Table 3. Frequency of specific features selected in the UMLS concept mapping (Frequency) experiment

Single_count_all_CKD_all_feat	N
DNR_(DNR_-_Do_not_resuscitate)_[Finding]	100
Map_(Functional_Map)_[Conceptual_Entity]	100
SAPSII	100
Meeting_(Meetings)_[Health_Care_Activity]	98
Coccyx_(Entire_coccyx)_[Body_Part_Organ_or_Organ_Component]	93
PICC_line_(Peripherally_inserted_central_catheter_(physical_object))_[Medical_Device]	86
Anuria_[Disease_or_Syndrome]	85
CMO_(Chronic_multifocal_osteomyelitis)_[Disease_or_Syndrome]	82
Family_[Family_Group]	70
vascular_(Blood_Vessel)_[Body_Part_Organ_or_Organ_Component]	51
Bilirubin_[Biologically_Active_SubstanceOrganic_Chemical]	45
Prognosis_(Forecast_of_outcome)_[Health_Care_Activity]	45
error_[Qualitative_Concept]	35
Necrotic_(Necrosis)_[Organ_or_Tissue_Function]	34
Pleural_effusion_(Pleural_effusion_fluid)_[Body_Substance]	28
Poor_prognosis_(Prognosis_bad)_[Finding]	28
Thick_[Qualitative_Concept]	28
Hypotensive_[Pathologic_Function]	27
dysfunction_(physiopathological)_[Functional_Concept]	25
Brain_[Body_Part_Organ_or_Organ_Component]	23

When providing the predictive models for healthcare experts to support their work in clinical practice, we should also pay attention to interpretability of such models. As already noted in [12] in case of similar predictive performance on training set, the simplest models often also perform the best on the test set. Therefore, we should also take the complexity of models with similar performance into account. In case of our study, the complexity of the four proposed models ranges from 9 up to approximately 35 selected features. In case of both unigram and bigram based models, it would perhaps make sense to use the simpler model as it does not significantly differ in predictive performance at a significantly lower complexity of the simpler, frequency based model.

Table 4. Frequency of specific features selected in the UMLS concept mapping (TF-IDF) experiment

Single_tfidf_all_CKD_all_feat	N
DNR_(DNR_-_Do_not_resuscitate)_[Finding]	100
Meeting_(Meetings)_[Health_Care_Activity]	100
SAPSII	100
CMO_(Chronic_multifocal_osteomyelitis)_[Disease_or_Syndrome]	98
PICC_line_(Peripherally_inserted_central_catheter_(physical_object))_[Medical_Device]	97
Family_[Family_Group]	96
Coccyx_(Entire_coccyx)_[Body_Part_Organ_or_Organ_Component]	88
Heels_(Heel)_[Body_Location_or_Region]	88
Pressors_[Pharmacologic_Substance]	83
Map_(Functional_Map)_[Conceptual_Entity]	74
Levophed_[Organic_ChemicalPharmacologic_Substance]	73
loosen_(Loosening)_[Functional_Concept]	70
neg_DNR_(DNR_-_Do_not_resuscitate)_[Finding]	68
Worsening_(Worse)_[Qualitative_Concept]	68
error_[Qualitative_Concept]	67
dysfunction_(physiopathological)_[Functional_Concept]	64
Bilirubin_[Biologically_Active_SubstanceOrganic_Chemical]	62
Anasarca_[Pathologic_Function]	59
Coccyx_(Bone_structure_of_coccyx)_[Body_Part_Organ_or_Organ_Component]	51
Poor_prognosis_(Prognosis_bad)_[Finding]	50

From the most frequently selected features (Table 3 and 4) we can observe some very general concepts, like “do not resuscitate

(DNR)" and high SAPS II score, indicating higher mortality. Also, family related concepts can be easily interpreted by a fact that physicians usually call family members to discuss the severity of the situation, especially when the situation is critical or life threatening. Additional features indicating higher mortality are concepts related to bones like "coccyx" and "heel", which could indicate specific problems related to calcification, frequently observed in CKD patients. Medical terms such as "peripherally inserted central catheter", "chronic multifocal osteomyelitis" and "use of pressors (pharmacologic_substance)" could be interpreted as signs of worsening health situation. We plan to investigate these conclusions in more detail in future work. It is also interesting to note that the UMLS frequency model's most selected variable was the medical term "Anuria (non-passage or less than 100 milliliters passage of urine a day)", which was not selected in the UMLS TF-IDF model.

Further development of our model will include extensions where a shorter (e.g., 6 or 12 hours) period would be used to provide "early warning" signal to healthcare experts working in the ICU. Additional features can be extracted from MIMIC-III that would further improve the predictive performance and possibly also the interpretability of the models.

5. ACKNOWLEDGEMENT

The authors would like to acknowledge financial support from the Slovenian Research Agency (research core funding No. P2-0057 and bilateral grant ARRS-BI-US/16-17-064).

6. REFERENCES

[1] Keegan, M. T., Gajic, O., and Afessa, B. 2012. Comparison of APACHE III, APACHE IV, SAPS 3, and MPM0III and Influence of Resuscitation Status on Model Performance. *Chest.* 142, 4 (April 2012), 851–858. DOI= 10.1378/chest.11-2164.

[2] Wu, V. C., Tsai, H. B., Yeh, Y. C., Huang, T. M., Lin, Y. F., Chou, N. K., ... and Wu, M. S. 2010. Patients supported by extracorporeal membrane oxygenation and acute dialysis: acute physiology and chronic health evaluation score in predicting hospital mortality. *Artificial organs*. 34, 10 (May 2010), 828–835. DOI= 10.1111/j.1525-1594.2009.00920.x.

[3] Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.W.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A. and Mark, R.G., 2016. MIMIC-III, a freely accessible critical care database. Scientific data, 3.

[4] Marafino, B. J., Boscardin, W. J., and Dudley, R. A. 2015. Efficient and sparse feature selection for biomedical text classification via the elastic net: Application to ICU risk stratification from nursing notes. *Journal of biomedical informatics*. 54 (April 2015), 114–120. DOI= 10.1016/j.jbi.2015.02.003.

[5] MIT Laboratory for Computational Physiology-mimic-code: 2017. https://github.com/MIT-LCP/mimic-code/tree/master/concepts/severityscores. Accessed: 2017- 09- 04.

[6] Metamap: Mapping text to the umls metathesaurus. 2006. https://pdfs.semanticscholar.org/e262/a22134cca0e484be1095160cc4ec8d9e7624.pdf. Accessed: 2017- 09- 04.

[7] Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*. 34, 5, (October 2001), 301–310. DOI= 10.1006/jbin.2001.1029.

[8] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pp.267-288.

[9] R Core Team, 2013. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[10] Friedman, J., Hastie, T. and Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1), p.1.

[11] Brabrand, M., Hallas, J., and Knudsen, T. 2014. Nurses and physicians in a medical admission unit can accurately predict mortality of acutely admitted patients: a prospective cohort study. *PloS one*, 9, 7, (July 14), e101739. DOI= 10.1371/journal.pone.0101739.

[12] Stiglic, G., Kocbek, S., Pernek, I., and Kokol, P. 2012. Comprehensive decision tree models in bioinformatics. *PloS one*, 7, 3, (March 30), e33812. DOI= 10.1371/journal.pone.0033812

Indeks avtorjev / Author index

Konferenca / Conference

Uredila / Edited by

Odkrivanje znanja in podatkovna skladišča - SiKDD / Data Mining and Data Warehouses - SiKDD

Dunja Mladenić, Marko Grobelnik