Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek E

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume E

Izkopavanje znanja In podatkovna skladišča (SiKDD 2014)

Data Mining and Data Warehouses (SiKDD 2014)

Uredila / Edited by
Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

6. oktober 2014 / October 6th, 2014
Ljubljana, Slovenia

Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek E

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume E

Izkopavanje znanja In podatkovna skladišča (SiKDD 2014)

Data Mining and Data Warehouses (SiKDD 2014)

Uredila / Edited by

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

6. oktober 2014 / October 6th, 2014
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, Mitja Lasič

Ljubljana, oktober 2014

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

004.8(082)(0.034.2)

MEDNARODNA multikonferenca Informacijska družba (17 ; 2014 ; Ljubljana)
Izkopavanje znanja in podatkovna skladišča (SiKDD 2014) [Elektronski vir] : zbornik 17. mednarodne multikonference Informacijska družba - IS 2014, 6. oktober 2014, [Ljubljana, Slovenia] : zvezek E = Data mining and data warehouses (SiKDD 2014) : proceedings of the 17th International Multiconference Information Society - IS 2014, October 6th, 2014, Ljubljana, Slovenia : volume E / uredila, edited by Dunja Mladenić, Marko Grobelnik. - El. zbornik. - Ljubljana : Institut Jožef Stefan, 2014

Način dostopa (URL): http://library.ijs.si/Stacks/Proceedings/InformationSociety
ISBN 978-961-264-075-0 (pdf)
1. Gl. stv. nasl. 2. Vzp. stv. nasl. 3. Dodat. nasl. 4. Mladenić, Dunja
275925760

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2014

Multikonferenca Informacijska družba (http://is.ijs.si) s sedemnajsto zaporedno prireditvijo postaja tradicionalna kvalitetna srednjeevropska konferenca na področju informacijske družbe, računalništva in informatike. Informacijska družba, znanje in umetna inteligenca se razvijajo čedalje hitreje. Čedalje več pokazateljev kaže, da prehajamo v naslednje civilizacijsko obdobje. Npr. v nekaterih državah je dovoljena samostojna vožnja inteligentnih avtomobilov, na trgu pa je moč dobiti kar nekaj pogosto prodajanih tipov avtomobilov z avtonomnimi funkcijami kot »lane assist«. Hkrati pa so konflikti sodobne družbe čedalje bolj nerazumljivi.

Letos smo v multikonferenco povezali dvanajst odličnih neodvisnih konferenc in delavnic. Predstavljenih bo okoli 200 referatov, prireditev bodo spremljale okrogle mize, razprave ter posebni dogodki kot svečana podelitev nagrad. Referati so objavljeni v zbornikih multikonference, izbrani prispevki bodo izšli tudi v posebnih številkah dveh znanstvenih revij, od katerih je ena Informatica, ki se ponaša s 37-letno tradicijo odlične evropske znanstvene revije.

Multikonferenco Informacijska družba 2014 sestavljajo naslednje samostojne konference:

- Inteligentni sistemi
- Izkopavanje znanja in podatkovna skladišča
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Soočanje z demografskimi izzivi
- Vzgoja in izobraževanje v informacijski družbi
- Kognitivna znanost
- Robotika
- Jezikovne tehnologije
- Interakcija človek-računalnik v informacijski družbi
- Prva študentska konferenca s področja računalništva
- Okolijska ergonomija in fiziologija
- Delavnica Chiron.

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

V 2014 bomo drugič 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 je prejel prof. dr. Janez Grad. Priznanje za dosežek leta je pripadlo dr. Janezu Demšarju. V letu 2014 četrtič podeljujemo nagrado »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono je dobila nerodna izvedba piškotkov, jagodo pa Google Street view, ker je končno posnel Slovenijo. Čestitke nagrajencem!

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

FOREWORD - INFORMATION SOCIETY 2014

The Information Society Multiconference (http://is.ijs.si) has become one of the traditional leading conferences in Central Europe devoted to information society. In its 17th year, we deliver a broad range of topics in the open academic environment fostering new ideas which makes our event unique among similar conferences, promoting key visions in interactive, innovative ways. As knowledge progresses even faster, it seems that we are indeed approaching a new civilization era. For example, several countries allow autonomous card driving, and several car models enable autonomous functions such as "lane assist". At the same time, however, it is hard to understand growing conflicts in the human civilization.

The Multiconference is running in parallel sessions with 200 presentations of scientific papers, presented in twelve independent events. The papers are published in the Web conference proceedings, and a selection of them in special issues of two journals. One of them is Informatica with its 37 years of tradition in excellent research publications.

The Information Society 2014 Multiconference consists of the following conferences and workshops:

- Intelligent Systems
- Cognitive Science
- Data Mining and Data Warehouses
- Collaboration, Software and Services in Information Society
- Demographic Challenges
- Robotics
- Language Technologies
- Human-Computer Interaction in Information Society
- Education in Information Society
- 1st Student Computer Science Research Conference
- Environmental Ergonomics and Psysiology
- Chiron Workshop.

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, SLAIS and IAS.

In 2014, the award for life-long outstanding contributions was delivered in memory of Donald Michie and Alan Turing for a second consecutive year. The Programme and Organizing Committees decided to award the Prof. Dr. Janez Grad with the Michie-Turing Award. In addition, a reward for current achievements was pronounced to Prof. Dr. Janez Demšar. The information strawberry is pronounced to Google street view for incorporating Slovenia, while the information lemon goes to cookies for awkward introduction. Congratulations!

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

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

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

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

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki-Špetič
Mitja Lasič
Robert Blatnik
Mario Konecki
Vedrana Vidulin

Programme Committee

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

KAZALO / TABLE OF CONTENTS

Zbornik 17. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2014

Zvezek E

Proceedings of the 17th International Multiconference

INFORMATION SOCIETY – IS 2014

Volume E

Izkopavanje znanja In podatkovna skladišča (SiKDD 2014)

Data Mining and Data Warehouses (SiKDD 2014)

Uredila / Edited by

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

6. oktober 2014 / October 6th, 2014
Ljubljana, Slovenia

Preface / Predgovor

Data Mining and Data Warehouses (SiKDD 2014)

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 (OLAP) 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.

Odkrivanje znanja in podatkovna skladišča (SiKDD 2014)

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 (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 zakonitosti v podatkih: pristope, orodja, probleme in rešitve.

Editors and Program Chairs / Urednika

- Marko Grobelnik
- Dunja Mladenić

A HIGH-PERFORMANCE MULTITHREADED APPROACH FOR CLUSTERING A STREAM OF DOCUMENTS

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 present an efficient approach for clustering a massive stream of textual documents, with particular emphasis on parallelization by the means of multithreaded processing. The underlying clustering algorithm is adaptable to changes in the stream and includes the ability to split and merge clusters, as well as to discard old data.

1 INTRODUCTION

Streams of textual documents occur naturally in various domains, such as from news sources (e.g. Spinn3r, IJS Newsfeed [4]) and social media (e.g. blogs, twitter, etc.). Clustering is the task of arranging such documents into groups based on some perceived notion of similarity, and is often an important building block of applications. The output of the clustering process can be seen as representing a "higher-level" view of the underlying data stream and is then the basis for further processing.

The fact that data comes from a stream adds a few specific requirements to the clustering task. In particular, the clustering algorithm must allow the partition of documents into clusters to be built and updated incrementally as new documents appear in the stream; it cannot make multiple passes through the entire stream, as there is only enough storage space for documents from a relatively limited time-based window. Additionally, we assume that the contents of the stream may change over time and the set of clusters should gradually be able to adapt to that, allowing the introduction of new clusters and the splitting/merging of existing clusters.

The most important consideration for a stream-based clustering algorithm, however, is that it must on average be able to process at least as many documents per unit of time as there are new documents coming from the stream in the same unit of time, otherwise it won't be able to keep up with the incoming data. If at some point the volume of incoming data grows high enough, parallelism will need to be employed. In the present paper, we focus on the scenario where the rate of incoming data is high enough that parallelism is necessary, but not so high that it would be necessary to distribute the processing over multiple computers. Thus, our aim is to present an efficient multi-threaded clustering approach that runs on a single computer but makes good use of its multi-CPU and multi-core facilities.

The remainder of this paper is structured as follows. In Section 2, we present an architectural overview of our clustering service. In Section 3, we briefly describe our clustering approach as it would appear conceptually if its multi-threaded aspects were taken out of consideration; in Section 4 we describe the design of the multi-threaded version of the approach; Section 5 presents conclusions and discusses possible directions of future work.

2 THE ARCHITECTURE OF OUR CLUSTERING SERVICE

Our implementation runs as a web service, listening for HTTP requests which provide new documents that arrived from the source stream and now need to be added to our clustering-related data structures. The clustering service processes such requests asynchronously; it reports any changes in the assignment of documents to clusters by eventually making HTTP requests to one or more listeners, reporting such things as assignments of documents to clusters, outcomes of cluster splits/merges, and changes in cluster medoids.

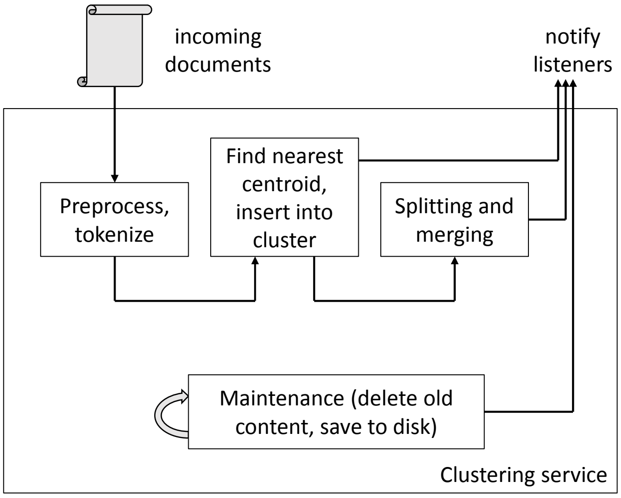


Figure 1: An architectural overview of our clustering web service.

In addition to handling the requests for adding a new document into the clustering, the service also performs maintenance operations from time to time. This includes discarding old documents and clusters, as well as saving its data structures to disk. The service keeps all its data in main memory during normal operation, but saves it to disk periodically so that it can be restarted without much loss of data in case it crashes. In earlier versions of our system, cluster merging was also performed periodically as a maintenance operation, but now it's included as a post-processing step following each addition of a new document.

3 THE UNDERLYING CLUSTERING ALGORITHM

Our approach is largely based on that of Aggarwal and Wu

[1][2]. The main principle that we use in assigning documents to clusters is to simply assign each incoming document to the cluster whose centroid is the closest (i.e. most similar) to the document. (Note that the same idea is used by popular off-line clustering algorithms such as k-means, except that k-means then performs multiple passes through the data, recalculating centroids and reassigning documents in each pass, which we cannot afford to do in an on-line setting.) For the purposes of computing centroids, measuring similarity etc., we use the bag-of-words representation (also known as the vector space model) to represent each document with a TF-IDF feature vector, normalized to unit length. After a new document is assigned to its nearest cluster, we consider splitting that cluster or merging it with another cluster. Figure 2 presents the pseudocode of our approach.

```
Input: $d$ – a document to be added to the clustering
1   If $d$ appears to be a duplicate of a document that is
    already in the clustering, stop processing $d$ and ignore it.
2   Compute $\mathbf{x}$, the TF-IDF vector representing $d$,
    normalized to unit length.
3   Find the cluster $C$ whose centroid is the closest to $\mathbf{x}$
    (in terms of cosine similarity).
4   Add $d$ into $C$, updating its various aggregated statistics
    (such as the centroid).
5   If the splitting conditions are met, consider splitting $C$:
6       Find the most promising split of $C$ into $C'$ and $C''$.
7       If this split is better than the original $C$, replace $C$
        with $C'$ and $C''$ in our clustering data structures.
8   Else, if the merging conditions are met, consider merging
    $C$ with another cluster:
9       Find a few clusters whose centroid is closest to the
        centroid of $C$, in terms of cosine distance.
        For each such cluster $C'$:
10          Let $C'' = C \cup C'$. If this cluster is better than
            keeping $C$ and $C'$ then
11              Replace $C$ and $C'$ by $C''$ in our clustering data
                structures and break.
```

Figure 2: Pseudocode describing an overview of our clustering approach.

Following the approach of [1], we maintain a set of statistics for each cluster and update them incrementally whenever the cluster changes. This includes the sum of the feature vectors of its documents, the square of this sum, per-feature variances, and a few other statistics. Our implementation also supports the option of allowing different documents to have different weights, where the weight of the document decays exponentially as the document ages, as suggested by [1]; however, this has not been found to be useful in our applications, so we currently set all weights to 1 in practice.

The fact that we're dealing with an ever-changing stream of documents requires us to introduce a few approximations. For example, in principle, whenever a new document is added to the clustering, or an old document discarded, the document frequency (DF) of any term from that document changes; as a result, the inverse document frequencies (IDF) of such terms also change, and the value of their corresponding features change accordingly, in the TF-IDF vector of any document containing any such term, as well as in the centroid of any cluster containing any such document. So theoretically, the feature vectors of most documents and the centroids of most clusters would have to be recalculated whenever a document is added to or removed from our collection. Since the costs of such an update would be prohibitive, we introduce an approximation. First, instead of storing TF-IDF vectors of individual documents, we store TF vectors instead. The IDF can be applied on the fly whenever needed, e.g. when we wish to (re)calculate the centroid of a cluster. Secondly, when a document is added to a cluster, we only recalculate the centroid of that cluster, but not the centroids of other clusters; those will get recalculated sooner or later when some new document is added to them.

To save time, a cluster is only considered for splitting (line 5 of the algorithm listing) if it is sufficiently large and if sufficiently many additions to it have been made since the last time it has been considered for splitting. The main idea during splitting is to project all members of the cluster onto a line and divide them into two groups depending on whether their projection was left or right of the projection of the centroid. This is repeated several times; in the first iteration, we project onto the principal component of the original cluster; in each subsequent iteration, we project onto the line through the centroids of the two groups from the previous iteration. In line 6, the best of these splits is chosen based on minimizing the variance; in line 7, a Bayesian Information Criterion is used to decide whether to actually accept the split.

For merging, we similarly only consider the cluster for merging (in line 8) if enough additions have been made to it since the last time it was considered for merging. Merging makes sense if two clusters are similar, e.g. as measured by the cosine similarity of their centroids. The problem here is that while the feature vectors of individual documents are sparse (i.e. they have relatively few nonzero components), the centroid of a cluster is usually fairly dense. Thus, computing a cosine between two centroids involves a dot product of two dense vectors, which is time-consuming. We resort to an approximation again: for the purposes of step 9, we temporarily make the centroid of C sparse by setting all its components to 0, except the thousand components that were highest in terms of absolute value. This substantially preserves the direction of the vector but makes the computation of dot products cheaper. In line 10, we use Lughofer's ellipsoid-overlap criterion [3] to decide whether to accept the merge; additionally, the merge is always accepted if the cosine between the two centroids exceeds a user-defined threshold.

The duplicate-detection step in line 1 is in practice somewhat custom-tailored to the particular document stream we've been using in our applications so far. This stream collects news articles from numerous websites, many of which turn out to be reprints of agency articles with few or no modifications. We declare an article to be a duplicate if an existing article has the same title (modulo capitalization and whitespace) and a sufficiently similar TF-vector. Such duplicate articles are simply discarded, rather than added into any cluster.

4 MULTI-THREADED CLUSTERING

A single-threaded, non-parallel implementation of the approach described in Section 3 is fairly straightforward. The program simply processes requests (to add a new document into the clustering) sequentially in an endless loop, finishing one request before moving on to the next one. Occasionally it can perform maintenance tasks (such as saving to disk, and discarding old clusters and documents) in between handling two requests.

Following the well-known principle that optimization should focus on those parts of the program in which the largest amount of time is spent, we timed the single-threaded implementation while performing 10^6 article additions. It turns out that approx. 54% of the time was spent in step 3, calculating the cosine similarity between the new document and the centroids of all existing clusters; 43% of the time was spent in step 9, calculating the cosine similarity between cluster centroids; all other steps together account for the remaining 3% of the time. Thus, it is clear that parallelization needs to focus on steps 3 and 9.

Another important consideration when designing a multithreaded solution involves the use of shared data structures. If a thread needs to modify some shared data structure, it requires exclusive access to it; that is, other threads shouldn't be using the data structure at all while it's being modified, even if they are content with read-only access to it. At the same time, we want to minimize the amount of time that threads spend waiting for some other thread to relinquish its exclusive lock on a shared data structure.

In the algorithm from Figure 2, modifications of shared data structures occur in step 4 (adding a new document to a cluster), step 7 (performing a split) and step 11 (performing a merge). Another modification of shared data, which is not as readily obvious from that pseudocode listing, occurs when creating a feature vector **x** corresponding to the new document *d*: a shared hash table containing the document frequencies of all terms needs to be updated, and new words might need to be added into it (if *d* contains some words that have until now never been seen in our stream of documents). Similarly, the duplicate detection in step 1 uses a hash table of document titles and needs to add the new document's title to it (if it didn't turn out to be a duplicate).

A somewhat naïve idea would be to assign each newly incoming document to one of several threads, and this thread can then execute the algorithm from Figure 2. The thread could somehow lock the cluster while it's being accessed, but we can quickly see that this is unsatisfactory. Our application calls for a large number of small clusters; eventually there will be thousands, possibly tens of thousands of clusters, and we don't want to require a thread to have to acquire and release tens of thousands of locks during step 3 or step 9.

This sort of fine-grained locking also has other inconvenient aspects. It is easy to imagine a nightmare scenario in which one thread is trying to compute the cosine between the centroid of cluster *C* and a new document; another thread has already decided that it wants to insert a different new document into *C* and now wants to update its centroid; and yet another thread is trying to split *C* into two clusters, or merge it with some other cluster.

If we don't want to have to deal with cluster-level locking, we have to accept that no thread may modify clusters (which includes adding or deleting them) while some other thread is looping through them in step 3 (or step 9, for that matter). Thus, while any thread is modifying the shared data structure (i.e. performing steps 4, 7, or 11), no other thread may be reading this data (i.e. performing step 3 or 9 – but as we saw earlier, that's where our threads will be spending 97% of their time). For nearly every new document (unless it was discarded in step 1 as a duplicate), we'll eventually have to add it to a cluster (in step 4); before our thread can do so, it must wait for all other threads to reach the end of step 3 or 9, and all those threads must then stop and wait for our thread to finish step 4. (A similar consideration applies to steps 7 and 11, but those are performed more rarely.) Clearly this has the potential to lead to an undesirable amount of waiting.

We can rewrite the pseudocode of Figure 2 in a way which emphasizes the alternation between stages which only read shared data and stages which need to modify shared data:

R1. Check if *d* is a duplicate; split it into words and prepare a TF-vector, except for any new words that aren't in our shared word table yet – these should be kept in a separate list.

M1. If R1 found *d* to be a duplicate, discard it and stop. Otherwise, add its title to the shared hash table (for future duplicate detection) and add any new words it might have contained into the shared word table; this is also the time to finalize its TF vector and update the document frequency counts in the shared word table.

R2. Compute the TF-IDF vector of our document *d* and find the cluster *C* with the nearest centroid.

M2. Insert *d* into *C*, updating *C*'s centroid and other aggregate statistics.

R3. Consider splitting *C* or merging it with other clusters, if appropriate. Do not modify any shared data structures; if the decision to split *C* is made, record what the new clusters C' and C'' would be; likewise, if the decision for a merge is made, record which cluster it would merge with and what would the resulting cluster be like.

M3. Update the shared data structures to reflect the splits or merges decided upon in step R3.

The key observation here is that there is no reason why all these steps should be performed by the same thread, as long as we maintain a small "context" data structure which helps threads keep track of the request as it passes through the stages.

Note also that stage R2 basically corresponds to step 3 of Figure 1, while step 9 is included within R3. All the M-stages are cheap, quick operations. Thus, in our multithreaded clustering implementation, we have *a single main thread which performs the M-stages for all requests*; the other threads are worker threads and perform R-stages.

The requests pass between the main thread and the worker threads until they are complete, as shown on Figure 3.

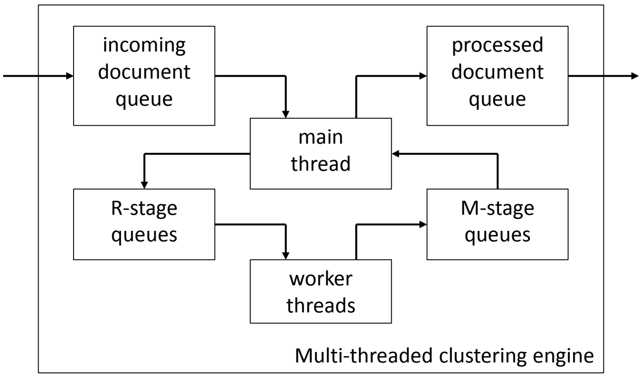


Figure 3: An overview of our multi-threaded clustering approach, showing the flow of requests through the system.

Thus, each worker thread runs in an endless loop in which it takes a job from a queue, performs the next stage (which will be either R1, R2, or R3), and deposits it into a different queue. The main thread, on the other hand, assigns jobs and periodically blocks the worker threads, performs the M-stages, and restarts them. The following is a simplified pseudocode of the main thread:

```
1   Wait a set amount of time (e.g. 1 second).
2   Set a flag which tells the worker threads to stop after they
    finish their current job.  Wait for all the worker threads
    to finish their current job.
3   Perform the M-stages of all the requests which are
    currently in progress.
4   Return to step 1.
```

Thus, most of the time the main thread sleeps (step 1) and lets the worker threads perform the R-stages of various requests. Every now and then, the main thread performs a barrier synchronization (step 2), stopping all the workers after their current job is done. Thus, at step 3, all workers are asleep, so the main thread can modify shared data.

Occasionally (e.g. once per hour), the main thread stops issuing any new R1-stage jobs to worker threads and waits for all partly completed requests to fully complete (i.e. all the way through M3). At this point, there are no partly completed requests in the system, so this is a good time to perform periodic maintenance tasks such as saving the data to disk. After this, normal processing can resume.

Since step 3 performs the M-stages of all currently open requests in one place, it is in a good position to coordinate their sometimes conflicting ideas as to what should be done. First, it performs the M1-stages, as these cannot conflict with other requests. Next it performs any splits and merges (M3) requested by recently completed R3 stages; while doing so, the main thread keeps track of which clusters have been split or merged, and ignores split/merge requests that involve clusters that have been affected by a previously processed split/merge request. Finally, the main thread performs M2-stages, inserting documents into the clusters requested by recently completed R2 stages; if any such cluster has been split, the main thread checks the centroids of the two subclusters to see which is closer to the document.

Note that this approach means that each document must pass through three iterations of the main thread before it is fully processed. Thus, if e.g. step 1 of the main thread takes 1 second, it will take at least 3 seconds before the document is processed. To use an analogy from networking, we have achieved high bandwidth at the price of also having high latency.

5 CONCLUSIONS AND FUTURE WORK

The multithreaded clustering approach presented here achieves a considerable degree of parallelism, allowing it to fully utilize a typical present-day multi-CPU multi-core PC. A further form of parallelism, not mentioned above but present in our application, comes from the fact that our stream of documents is multilingual and each language is processed separately from the others, thus each language can have its own main thread and set of worker threads.

The work presented here could be extended in several directions. For example, this approach could be applied to non-textual data with only minor modifications. The key idea behind our multithreaded approach is the multi-stage processing, concentrating all modification of shared data into one thread and using barrier synchronization for the worker threads; and there is nothing text-specific in this.

The computation of cosine similarities (which is where the algorithm still spends most of its time) could be speeded up by the means of random projections [5] into a limited (and fixed) number of dimensions. In our preliminary experiments, projecting our feature space into 1000 random projections resulted in almost no distortion (in terms of which centroid is closest to which document). After such a projection, documents and centroids become fixed-length dense vectors, and cosines can be computed very efficiently by making use of the SIMD capabilities of modern processors (as in various numerical linear algebra libraries).

Another possible direction for further work is to replace the current flat clustering with a hierarchical one. This can be desirable for some applications, and it would also speed up the assignment of documents to clusters if this is done in a top-down fashion instead of examining all the clusters.

Finally, at some point the rate of incoming documents may well grow beyond what can be processed by a single computer, so it would be interesting to investigate clustering approaches based on distributed computing.

Acknowledgments

This work was supported by the Slovenian Research Agency and the ICT Programme of the EC under under XLike (ICT-STREP-288342).

References

[1] C. C. Aggarwal, P. S. Yu. A framework for clustering massive text and categorical data streams. *SIAM Conf. on Data Mining*, 2006.

[2] C. C. Aggarwal, P. S. Yu. On clustering massive text and categorical data streams. *Knowledge and Inf. Systems*, 24(2):171–196, 2010.

[3] E. Lughofer. A dynamic split-and-merge approach for evolving cluster models. *Evolving Systems*, 3(3):135–151, 2012.

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

[5] Â. Cardoso, A. Wichert. Iterative random projections for high-dimensional data clustering. *Pattern Recognition Letters*, 33(12):1749–55, 2012.

ALGORITHM FOR CLASSIFICATION OF TEXTUAL DOCUMENTS REPRESENTED BY TANDEM ANALYSIS

Jasminka Dobša
Faculty of Organization and Informatics
University of Zagreb
Pavlinska 2, Varaždin, Croatia
Tel: +385 42 390844; fax: +385 42 213413
e-mail: jasminka.dobsa@foi.hr

ABSTRACT

In this research is presented algorithm for classification of textual documents which are represented in the space of reduced dimension in respect to original bag of words representation. Algorithm is carried out in two steps: in the first step classification is conducted for documents represented in original bag of words representation, while in the second step classification is conducted for documents represented in the space of reduced dimension. Reduction of dimensionality is obtained also in two steps: in the first step documents are represented by usage of latent semantic indexing, while in the second step this representation is projected on the space of membership matrix defining a membership of documents in classes. Evaluation of algorithm is conducted on Reuters21578 collection of documents.

1 INTRODUCTION

In this paper is represented algorithm for classification of textual documents which is carried out in two steps. In the first step documents are represented using the *bag-of-words representation*, also referred to as *vector space model.* The vector space model is implemented by creating the *term-document matrix*, which can be explained as follows. Let the list of relevant terms for a certain collection be numerated from *1* to *m* and documents be numerated from *1* to *n*. The term-document matrix is an $m \times n$ matrix $A = [a_{ij}]$ where a_{ij} represents the weight of term *i* in document *j*. In the term-document matrix, documents are represented as column vectors which dimension is the number of relevant terms. The main characteristics of such text representation are high dimensionality of input space and sparseness of the term-document matrix.

Next step in presented classification algorithm is dimensionality reduction which is also carried out in two steps. In the first step original representations of textual documents are represented by method of latent semantic indexing (LSI). In the next step representations of documents obtained by LSI are projected on the space spread by columns of membership matrix which defines membership of documents in classes. Terms occurring in the document may not be the best representation of the document content, due to the problems of synonymy (different words with similar meaning) and polysemy (one word with more meanings). By dimensionality reduction we can represent a collection of documents in a more compact way, which could save memory space, speed up the main tasks and reduce the effect of noise in the data. The method of latent semantic indexing is introduced in [2]. Today, it presents benchmark in the field of representation of documents in the space of reduced dimensionality. According to some earlier investigations [6] the method of LSI has some disadvantages in fulfilling the task of classification since its application could remove some significant information concerning structures of the classes. The idea behind second step in dimensionality reduction is to stress the structure of the classes in the data by projection on columns of class membership matrix. The inspiration for such a step comes from the method of Factorial K-means introduced by Vichi and Kiers [9]. Therefore dimensionality reduction of the original representation of documents in term-documents matrix is obtained sequentially in two steps in procedure called *tandem analysis* [7] which is frequently used by practitioners, but is not applied in this form for the task of classification of textual documents yet. Similar approach is proposed in research of Dhillon and Modha [3]. In their work is proposed reduction of dimension of term-document matrix by concept decomposition - projection of documents on centroids of groups obtained by spherical k-means algorithm. It is shown that indexing of documents obtained by concept decomposition can improve performance of information retrieval of documents [4]. Also, concept decomposition applied in its supervised form by projection of documents on centroids of classes can improve classification performance [6].

The rest of the paper is organized as follows. In the second section is given description of used techniques for dimensionality reduction (latent semantic indexing and tandem analysis). Third section gives description of methods used for automatic classification of data. In forth section are given results of experiments and the last section gives conclusion and discussion with directions for a further work.

2 DIMENSIONALITY REDUCTION TECHNIQUES

The idea of dimensionality reduction techniques is to represent documents by clustering them based on topic similarity regardless of indexing terms used. In the case of LSI the approximate representation of documents is accomplished using a truncated singular value decomposition (SVD) approximation of the term-document matrix. SVD of an arbitrary matrix **A** is given by

$$\mathbf{A} = \mathbf{U\Sigma}V^T \qquad (1)$$

where **U** is $m \times m$ orthogonal matrix where m is number of indexing terms, **V** is $n \times n$ orthogonal matrix where n is number of documents in collection and **Σ** is diagonal matrix on whose diagonal are singular values of matrix **A** in a decreasing order. For a purpose of representation of textual documents truncated SVD is used which has form

$$\mathbf{A} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k{}^T \qquad (2)$$

where $\mathbf{U_k}$ is $m \times k$ matrix whose columns consist of the first k columns of matrix **U, V**$_\mathbf{k}$ is $n \times k$ matrix whose columns consist of the first k columns of matrix **V**, and $\mathbf{\Sigma_k}$ is diagonal matrix on whose diagonal are the greatest singular values of **A** ordered in decreasing order. Representations of documents by LSI method are said to be representations in LSI space and are given by columns of matrix $\mathbf{V}_k{}^T$.

Procedure of Tandem analysis is performed by projection of matrix of document's representation by LSI on columns of membership matrix **M** in the sense of least squares. Membership matrix **M** is $n \times k$ matrix which defines a membership of documents into classes in a way that $m_{ik} = 1$ if the i^{th} document belongs to k^{th} class and $m_{ik} = 0$ otherwise. It is feasible that document belongs to multiple classes. Projection of LSI representations of documents onto column space of **M** is accomplished by solving the least square problem

$$\left\|\mathbf{V}^T - \mathbf{MZ}\right\| \to \min. \qquad (3)$$

It is known that solution of a set problem is given by

$$\mathbf{Z} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{V}^T \qquad (4)$$

and representation of documents by Tandem analysis is given by transpose of $n \times k$ matrix

$$\mathbf{B} = \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{V}.^T \qquad (5).$$

3 CLASSIFICATION METHODS

Text classification is procedure of assigning a labels of previously defined classes to a unstructured textual documents [8]. Characteristic of classes are learned on the training set of documents and tested on the test set. If **A** is term-document matrix of a set of training documents then representation of training documents by method of LSI is given by matrix $\mathbf{V}_k{}^T$, while representation of matrix **A** by method of Tandem analysis is given by transpose of matrix **B** given by formula (5). If **T** is term-document matrix of a test set of documents then representation of matrix **T** in LSI space is given by matrix **C** where

$$\mathbf{C}^T = \mathbf{T}^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}. \qquad (6)$$

Representation of test documents by method of Tandem analysis is given by transpose of a matrix

$$\mathbf{D} = \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\mathbf{T}^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1} \qquad (7)$$

where **N** is membership matrix of a test matrix. Since it is not allowed to use membership matrix of a test matrix before final evaluation, this matrix has to be approximated and here it will be applied the first step of classification. Classification in the first step will be conducted by the method of k-nearest neighbors or by support vector machines (SVMs), while classification in the second step will be conducted only by SVMs

K- nearest neighbor (k-nn) algorithm is a type of example-based classifiers. It observes class of nearest documents and assigns class c to a document if large enough proportion of nearest documents belong to that class [8]. SVMs is an algorithm that finds a hyperplane which separates positive and negative training examples with maximum possible margin [1,5]. This means that the distance between the hyperplane and the corresponding closest positive and negative examples is maximized. A classifier of the form $sign(w \cdot x + b)$ is learned, where w is the weight vector or normal vector to the hyperplane and b is the bias. Depending on which side of separating hyperplane the test example is, its prediction will be positive or negative.

4 EXPERIMENT

Experiments are conducted on the 10 largest classes of standard Reuters21578 collection using "ModApte" split having 9603 training and 3299 test documents. After stop words and words that occurred in less than 4 documents are removed, the list of 9867 terms is formed. Classification of documents is conducted by k-nn algorithm for k=10 or SVM algorithm in the first step and by SVM algorithm in the second step. LSI method is conducted for k=90, which means that documents are represented in LSI space by vectors of dimension 90. All other representation obtained by Tandem analysis also use LSI with k=90. We have treated the problem of classification for each category as a two-class problem, with members of that category being positive examples and all other documents being negative examples. Evaluation was performed using a commonly used combination of precision, recall, and the F_1 measure. Precision p is a proportion of documents predicted positive that are actually positive. Recall r is defined as a proportion of positive documents that are predicted positive. The F_1 measure is defined as $F_1 = 2pr/(p+r)$. Macroaverage is an average value of measure of evaluation for all observed classes. For classification of documents by SVM method

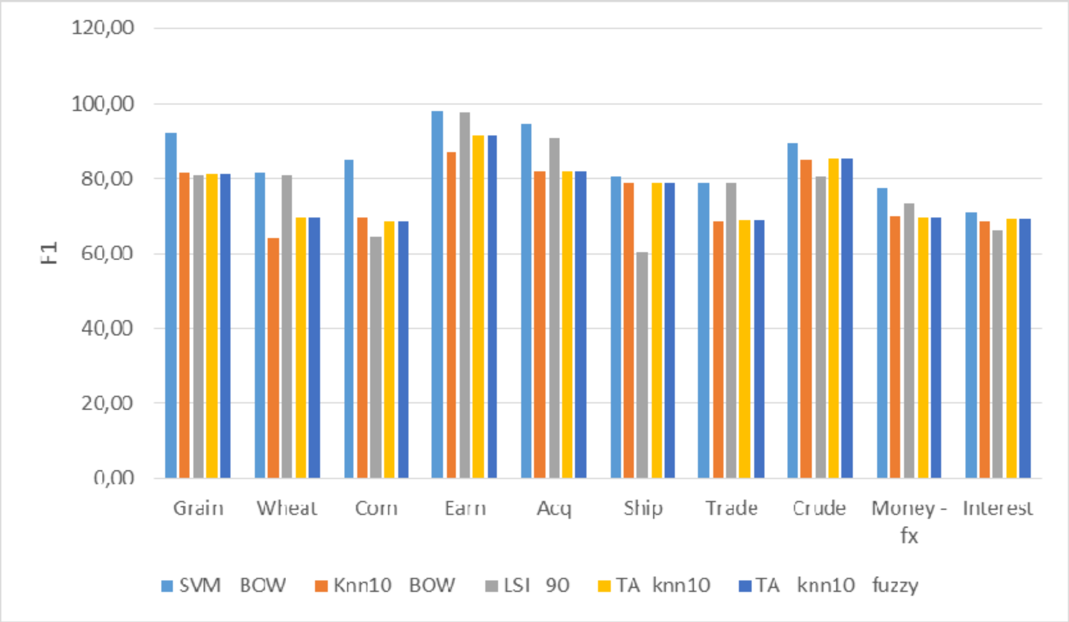


Figure 1: *F_1 measure for top 10 classes of Reuters21578 data set and for different representations of documents and used classification algorithms.*

Table 1*: Results of classification performance (precision, recall and F_1 measure) for top classes of Reuter21578 data set for different representations of documents. Classification in both steps of algorithm is conducted by SVMs.*

	Bag of words			***Tandem analysis***			***Tandem analysis modified***		
Class	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Grain	**97.04**	87.92	**92.26**	**97.04**	87.92	**92.26**	85.98	**94.63**	90.10
Wheat	**89.83**	74.65	81.54	**89.83**	74.65	81.54	78.75	**88.73**	**83.44**
Corn	**97.67**	75.00	**84.85**	**97.67**	75.00	**84.85**	81.82	**80.36**	81.08
Earn	**98.43**	97.79	**98.11**	**98.43**	97.79	**98.11**	93.36	**99.54**	96.35
Acq	**97.49**	91.66	**94.49**	**97.49**	91.66	**94.49**	89.14	**98.19**	93.45
Ship	**92.65**	70.79	80.26	**92.65**	70.79	80.26	75.00	**94.38**	**83.58**
Trade	**85.15**	73.50	**78.90**	**85.15**	73.50	**78.90**	62.21	**91.45**	74.05
Crude	**91.21**	87.83	**89.49**	**91.21**	87.83	**89.49**	76.39	**94.18**	84.36
Money - fx	**81.99**	73.74	77.65	**81.99**	73.74	77.65	72.32	**90.50**	**80.40**
Interest	**89.53**	58.78	70.97	**89.53**	58.78	70.97	73.33	**75.57**	**74.43**
Macroaverage	92.10	79.17	84.85	92.10	79.17	84.85	78.83	90.75	84.12

SvmLight v.5.0 software by Joachims (2002) with default parameters was used. For all other calculations it was used MATLAB v7.6.

On Figure 1 are shown results of classification performance in terms of F_1 measure. The first column shows F_1 measure for a classification obtained by bag of words representation with usage of SVM (SVM – BOW). The second column shows F_1 measure for classification performed by k-nn algorithm (k=10) for documents represented by bag of words representation (Knn10 – BOW). In third column are shown results of classification by SVM for LSI representation, while the fifth and the sixth column show F_1 measure for representation by Tandem analysis where classification is conducted by k-nn algorithm (k=10) in the first step and SVM algorithm in the second (TA – knn10 and TA – knn10 – fuzzy). Procedure denoted by TA – knn10 – fuzzy is a slight modification of procedure TA – knn. Namely, it can be seen from results of F_1 measure from Figure 1 that classification obtained by usage TA – knn10 representation did not improve much classification results by k-nn algorithm and bag of words representation. Since there is no need to decide categorically in the first step of classification about membership of documents to classes, the procedure TA – knn10 – fuzzy modifies procedure TA – knn10 in a way that element n_{ik} of a membership matrix contains proportion of 10 nearest documents to i^{th} document contained in k^{th} class. For example, if there is 3 documents from k^{th} class among 10 nearest documents to i^{th} document then element of membership matrix for test set of documents is n_{ik} =0.3. In the case of TA – knn10 procedure n_{ik}=0, since decision that i^{th} document is in k^{th} class is made if there is at least 4 documents among 10 nearest documents to i^{th} document in the k^{th} class. From results shown in Figure 1 it can be seen that modification of TA – knn10 procedure did not result in significant improvement (there is improvement of macroaverage of approximately 1%) in terms of F_1 measure. Nevertheless, there are more differences in terms of precision and recall which is not elaborated here. Analysis of differences obtained by modifications of the predictions of classification obtained in the first step of algorithm will be discussed through results shown in Table 1. It shows results of classification performance (precision, recall and F_1 measure) for top 10 classes of Reuter21578 data set for three different representations of documents: bag of words representation, representation in reduced dimension space by Tandem analysis and representation in reduced dimension space by Tandem analysis with modifications of predictions obtained by classification in the first step. Classification is conducted by method of SVM in both steps. From Table 1 it can be seen that results of precision, recall and F_1 measure are exactly the same for a bag of words representation and representation obtained by Tandem analysis. Hence, by usage of Tandem analysis classification performance can be improved in comparison to LSI method (Figure 1), but apparently it is limited by approximation of membership matrix obtained in the first step of classification. In Tandem analysis with modification predictions obtained by SVM are modified in a following way: if prediction for i^{th} document belonging to a k^{th} class is greater than 0.6 then element of membership matrix for test set of documents is n_{ik} =1, if prediction is less than -0.6 then n_{ik} =0, otherwise n_{ik} =0.5. Such a modification resulted in significant improvement of classification recall, but precision of classification dropped at the same time resulting in a similar macroaverage of F_1 measure.

5 CONCLUSION AND DISSCUSION

In the paper is introduced a novel algorithm for a classification of textual documents represented in a space of reduced dimension obtained by Tandem analysis which consist of two steps. In the first step is performed LSI and in the second step representations in the LSI space are projected on a space spread by membership matrix of a train and test data set. Classification is performed twice, firstly to get approximate membership matrix of a test set of documents and secondly to classify documents represented by Tandem analysis. Although results of F_1 measure are the best for bag of words representation, it is shown that F_1 measure of classification is improved for 5 out of 10 top classes of Reuters21578 data set in comparison to LSI when k-nan algorithm is used in the first step of classification. Results of classification including precision, recall and F_1 measure when SVM algorithm is used in both steps are exactly the same for a bag of words representation and representation by Tandem analysis. This guides to a conclusion that classification performance is limited by the first step of classification. It is important to stress that in the case of bag of words representation documents are represented in space of dimension of almost 10 000 (number of indexing terms), while representation by Tandem analysis is of dimension 90 (dimension of LSI space). Hence, representation by Tandem analysis requires much less memory space.

In the further work method will be tested for a task of information retrieval and cross-lingual information retrieval. Also, algorithm of Tandem analysis will be compared with algorithm of simultaneous performance of both dimensionality reduction steps (reduction of variables/terms and reduction of objects/documents).

References

[1] Cristianini, N., Shave-Taylor, J., Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[2] Deerwester, S., Dumas, S., Furnas, G., Landauer, T., Harsman, R.., Indexing by latent semantic analysis, J. American Society for Information Science, 1990, 41: 391-407.

[3] Dhillon, I.S., Modha, D.S., Concept decomposition for large sparse text data using clustering, Machine Learning, 2001, 42(1): 143-175.

[4] Dobša, J., Dalbelo-Bašić, B. , Concept decomposition by fuzzy k-means algorithm, Proceedings of the IEEE/WIC International Conference on Web Intelligence, WI 2003, 2003, 684-688.

[5] Joachims, T., Text categorization with support vector machines: Learning with many relevant features, In Proc. of the European Conference on Machine Learning, 1998, Springer, 137-142.

[6] Karypis, G., Hong, E., Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval & categorization, Technical Report TR-00-016, Department of Computer Science, University of Minnesota, Minneapolis, 2000.

[7] Rocci, R., Gattone, S.A., Vichi, M., A new dimension reduction method: factor discriminant K-means, Journal of Classification, 28:210-226.

[8] Sebastiani, S., Machine learning in automated text categoriyation, ACM Computer Surveys, 2002, 34: 1-47.

[9] Vichi, M., Kiers, H.A. L., Factorial K-means analysis for two-way data, Computational Statistics and Data Analysis, 2001, 37:49-64.

MINING DEMAND AND SUPPLY DATA WITH METHODS OF ECONOMIC STATISTICS

Melita Hajdinjak
Faculty of Electrical Engineering
University of Ljubljana
Tržaška cesta 25, 1000 Ljubljana, Slovenia
Tel: +386 1 4768385
e-mail: melita.hajdinjak@fe.uni-lj.si

ABSTRACT

This paper presents a data mining approach for demand and supply data arising from economic statistics. It is based on demand and supply functions with different types of price elasticities, which can be used to identify interesting patterns and data dependencies. Constant, directly proportional and linear price elasticities are considered.

1 INTRODUCTION

In economics, the amount of a good that consumers are willing to buy at a given price, holding constant the other factors that influence purchases, is the *quantity demanded*. The relationship between the unit price and the total quantity desired by consumers is described by the demand curve. It is downward sloping (with some theoretical exceptions, such as Veblen and Giffen goods) because of the *law of demand* [1], which says that people will buy more of a service, product, or resource as its price falls. On the other hand, the *quantity supplied* is the amount of a good that firms want to sell at a given price, holding constant other factors that influence firms' supply decisions, such as costs and government actions. The relationship between the unit price and the total quantity offered by producers is described by the supply curve, which is upward sloping (with some important exceptions, such as if the seller is badly in need of money or if he wants to get rid of his products) because of the *law of supply* [1,2], which says that the price increases with the quantity. In the case of oil, this can be explained by the fact that small quantities will be supplied using the most efficient oil plant available, but that as quantity increases, producers will have to use less efficient oil plants with higher production costs.

When we plot the supply and demand curves for a product on the same graph, they intersect at the price where the amount producers are willing and able to supply equals the amount consumers are willing and able to purchase. The price and quantity where supply and demand meet are called the *equilibrium price* and *equilibrium quantity,* respectively [2]. If suppliers ignore demand and continue to produce units and price them too high, they will not be purchased but instead sit in the warehouse. If they produce too few, demand will go unmet and consumers will claim for more. Of course, the equilibrium varies with the observed good, service, or resource. Moreover, the equilibrium changes if a 'shock' occurs such that one of the variables we were holding constant changes, causing a shift in either the demand curve or the supply curve.

2 PRICE ELASTICITY

A measure used in economics to show the responsiveness of the quantity demanded (or supplied) of a good or service to a change in its price is called *price elasticity* of demand (or supply). It was defined by Alfred Marshall [3] as the percentage change in quantity in response to a one percent change in price, holding all the other factors constant:

$$\varepsilon = \frac{\frac{dQ}{Q}}{\frac{dP}{P}}.$$

Here, Q is the quantity demanded (or supplied) and P is the price. If Q is the quantity demanded, then ε is the price elasticity of demand and the above formula yields a negative value for exactly those goods that conform to the law of demand, which is due to the inverse nature of the relationship between price and quantity. On the contrary, while price elasticity of demand is usually negative, price elasticity of supply is usually positive, due to the direct proportion between price and quantity supplied as stated by the law of supply. Depending on whether the elasticity is greater than, equal to, or less than -1 for demand (and 1 for supply), the price elasticities for a good, service, or resource are described as relatively inelastic, unit elastic, or relatively elastic [2].

For analytical convenience in theoretical demand (and supply) analyses, a special kind of functions for which the price elasticity is constant for all price levels and all price changes is frequently used. Since the percentage changes depend on both amount of the change, or the unit change, and the starting point, or base value, of the change, slope and elasticity are different concepts. For instance, even though the slope of a linear demand (or supply) curve is constant, the elasticity is *not* constant along this curve, a straight line. This is because unit changes are identical for each segment on the line, but the base values are not. The special type of demand or supply curve for which the price elasticity is the same at every point along the curve is given by

$$Q(P) = AP^{\varepsilon},$$

where *A* is an arbitrary constant. Constant elasticity assumes that consumers and providers are equally sensitive to price changes whatever the price may be. Several studies, however, indicate that this is not always true by showing that price elasticity can be non-constant [4,5,6]. In particular, while the elasticity of demand usually decreases with price, the elasticity of supply usually increases with price. We shall thus consider elasticity that is directly proportional to the price level,

$$\varepsilon = aP \text{ with } Q(P) = Ae^{aP},$$

as well as elasticity that is linearly dependent on the price level,

$$\varepsilon = aP + b \text{ with } Q(P) = Ae^{aP}P^{b},$$

where a and b are constants. Linear price elasticity decreases with price if $a<0$ and it increases if $a>0$. In the case of oil, when the price increases, most consumers become more sensitive to price changes, which can be modelled using constants $a>0$ and $b<0$, in contrast to the suppliers, which become less sensitive since they have more freedom to use less efficient oil plants with higher production costs, which can be modelled using $a<0$ and $b>= 0$. In addition, a and b having the same sign are characteristic neither for demand nor for supply. Since price elasticity of demand is usually negative, $a<0$ and $b<0$ would mean that the consumers become more and more insensitive to price changes when the price increases, which is highly unrealistic. Similarly, since price elasticity of supply is usually positive, $a>0$ and $b>0$ would mean that the suppliers have unlimited production capacities.

For constant-elasticity functions demand and supply are perfectly reversible with respect to prices, which is analytically convenient since it allows us to uniquely associate the amount of demand or supply with price (injectivity). Although elasticity that is proportional to the price level does not cause irreversible demand (or supply) effects, linear elasticity causes them for some pairs of constants a and b (i.e., as soon as $ab<0$ or $a=b=0$). Linear or piecewise linear price elasticity models have been used in economics, marketing and business before but only to a limited extent [7,8,9,10]. This may be due to the fact that, until very recently, the class of all demand and supply functions with linear price elasticity has not been sufficiently theoretically studied [11].

We can use demand functions with different types of price elasticities (constant, directly proportional, linear) to search for or extract interesting patterns and dependencies from a chosen data set.

3 MINING U. S. OIL IMPORTS DATA

We focus on the data set named *Monthly U. S. Oil Imports*, which contains the data about 1000s of oil barrels purchased, the total value of oil and the unit price in $ for every month of the years 1973-2007. It was published by the Foreign Trade Division of the U. S. Census Bureau (http://www.census.gov/foreign-trade/). The monthly movements of purchased quantity and the unit price are shown in Figure 1.

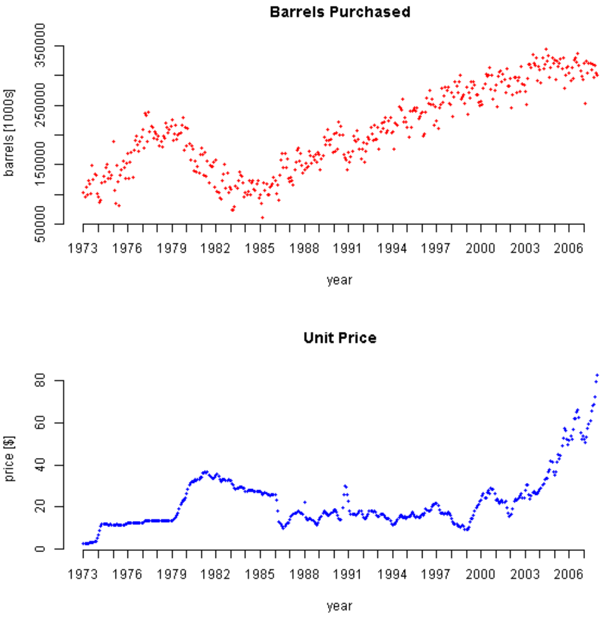


Figure 1: *Monthly U. S. oil imports in the years 1973-2007. The unit price is not adapted to inflation.*

In 2012, the United States produced 60 % of the petroleum (crude oil and petroleum products) it used, the remainder being imported. The largest sources of imported oil were Canada, Saudi Arabia, Mexico, Venezuela, and Russia. Oil imports into the US peaked in 2005 when imports supplied 60 % of US consumption. They have declined since, due to increased domestic oil production and reduced consumption [12]. However, in Figure 1, the most visible peak of imported oil quantity is when the U. S. created the strategic petroleum reserve to augment supply in case of a national emergency. It started in 1975 after oil supplies were cut off during the 1973-74 oil embargo.

We have divided the oil imports data in groups by years. For every year, least-squares regression analysis with three different regression functions has been performed: (1) constant price-elasticity function, (2) directly proportional price-elasticity function, and (3) linear price-elasticity function. The regression results for oil imports in the year 1980 are shown in Figure 2.

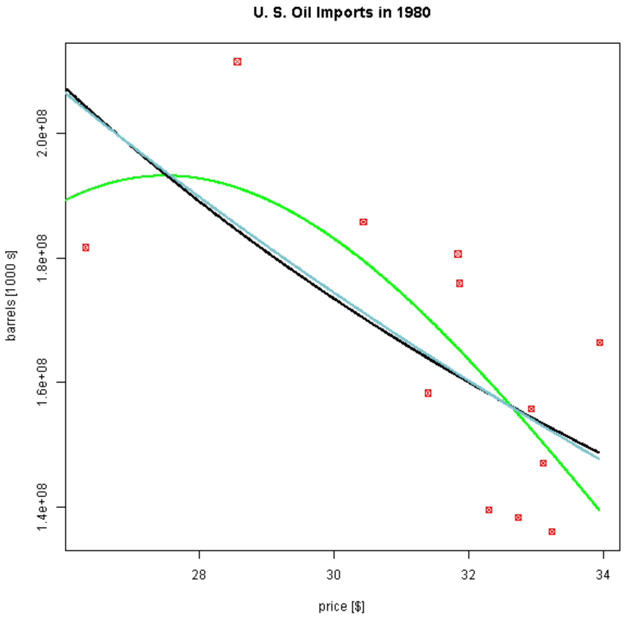


Figure 2: *Regression curves for U. S. oil imports in 1980. The black curve has constant price elasticity (R^2=0.43), the light blue curve has directly proportional price elasticity (R^2=0.44), and the green curve is the most general curve with linear price elasticity (R^2 =0.51).*

Of course, the best fit is obtained by the most general linear price-elasticity functions, but only for 11/35 of the observed years the goodness of fit and the statistical significance of the estimated parameters are high enough (*$p<0.05$*) to derive valid conclusions. The constant price-elasticity functions and the directly proportional price-elasticity functions, on the other hand, both produce 9/35 statistically significant regression models. The models' price elasticities are given in Table 1.

	constant $(\varepsilon = b)$	directly prop. $(\varepsilon = aP)$	linear $(\varepsilon = aP + b)$
1973	-	-	-1.44P + 5.57
1974	0.71	0.08P	0.95P - 8.06
1980	-1.25	-0.04P	-0.50P + 13.69
1987	1.58	0.10P	1.36P - 20.07
1990	-0.41	-0.02P	-0.00P - 0.40
1992	0.77	0.05P	-0.40P + 7.23
1994	0.64	0.05P	0.20P - 2.15
1997	-0.73	-0.04P	-0.23P + 3.65
2001	-	-	-0.36P + 6.93
2003	-0.83	-0.03P	-0.28P + 6.88
2006	0.43	0.01P	0.09P - 4.74

Table 1: *Yearly elasticities of U. S. oil imports.*

Interestingly, some of the determined price elasticities are characteristic for demand ($\varepsilon<0$), others are characteristic for supply ($\varepsilon>0$). Linear price-elasticity models with *$a>0$* and *$b<0$* indicate that the U. S. government was more sensitive to price changes when the price increased, which corresponds to the law of demand. On the other hand, models with *$a<0$* and *$b>0$* indicate that the U. S. government was less sensitive to price changes when the price increased, which is atypical for consumers. This behaviour may be explained by continuously high prices or an extreme need for oil (e.g., in years 1980, 1990, 1992, 1997, 2003).

The means of the models' yearly coefficients of determination R^2 are *0.56*, *0.57* and *0.63*, respectively, and the corresponding mean p-values for the F-test are *0.008*, *0.007* and *0.017*. Another interesting regression observation is a sequence of highly insignificant models between 1975 and 1985, which indicates an atypical, unpredictable oil purchase during the creation of the U. S. petroleum reserve.

Furthermore, assuming linear price elasticities, we have defined *monthly price elasticities* as

$$\varepsilon_i = \frac{\frac{Q_i - Q_{i-1}}{Q_i}}{\frac{P_i - P_{i-1}}{P_i}}$$

and *monthly elasticity slopes* as

$$a_i = \frac{\varepsilon_i - \varepsilon_{i-1}}{P_i - P_{i-1}} .$$

The monthly elasticity values for the oil imports data are shown in Figure 3.

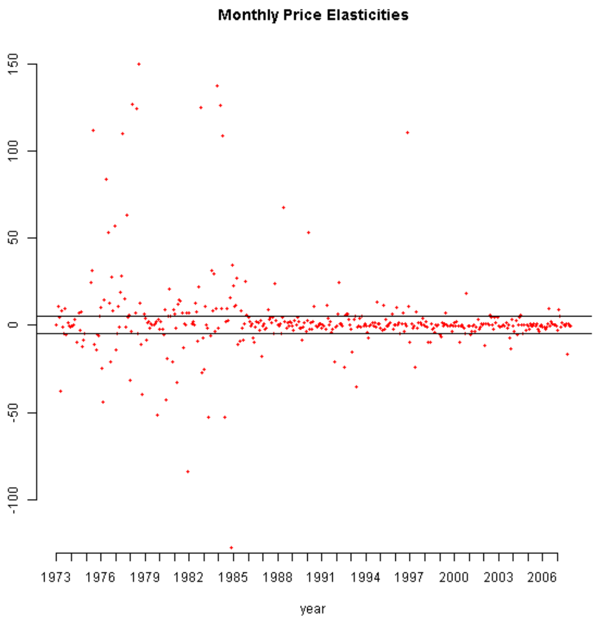


Figure 3: *Monthly price elasticities in the years 1973-2007. The black horizontal lines are the limits of the region with* $|\varepsilon_i| \le 5$.

A similar picture aroused for the monthly elasticity slopes. From this we have concluded that the responsiveness of the imported oil quantity to a change in its price stabilized with the years. No other interesting patterns have been observed.

The monthly price elasticities (and the monthly elasticity slopes) could even be used to automatically detect *stable* time intervals in which the purchased quantity changes as directed by the price change and in accordance with an assumed type of price elasticity. Hence, the advantage of this approach compared to regression is that we do not need to analyze observations in predetermined time intervals, but only monitor the changes that happen from observation to observation. When the changes of monthly price elasticities and/or monthly price elasticity slopes are substantial, boundaries between stable time periods may be set. Substantial changes or instabilities can occur for various reasons, including oil embargos, wars and other drastic changes in the population size or governmental policy. On the other hand, usual changes in the population size or undramatic inflations should not cause substantial differences in the price elasticities or price elasticity slopes.

4 CONCLUSION

We have proposed a data mining approach for demand and supply data arising from economic theory and applied statistics. It is based on demand/supply functions with different types of price elasticities, which can help us searching for interesting patterns and data dependencies.

In this paper, we have applied the approach on U. S. oil imports data, and it has given some useful results. For instance, we have detected an 11-year period of highly unstable oil imports data, which indicates radical changes (the creation of the U. S. petroleum reserve) in the governmental actions. Also, local peaks of purchased quantity have been detected by linear price-elasticity models atypical for demand. In addition, monthly price elasticities and monthly price elasticity slopes have been introduced as a method for detecting stable time intervals. Within stable time intervals, all the other factors except quantity and price may be considered constant, and price elasticities as defined by Marshall [3] may be determined using statistical regression.

Unfortunately, due to a large number of factors that influence oil imports the data analysis was quite complex and not as clear as it could be. A thorough empirical analysis on simpler demand and supply data thus needs to be performed to see the real potentials of the approach.

References

[1] Parkin, M., Economics, 6th ed. Pearson Education, Inc., London, 2003.

[2] Perloff, J. M., Microeconomics: Theory and Applications with Calculus, 3rd ed. Prentice Hall, Englewood Cliffs, 2013.

[3] Marshall, A., Principles of Economics, Revised Edition. Macmillan, London, 1890.

[4] Evans, L., "On the Restrictive Nature of Constant Elasticity of Demand Functions." International Economic Review 35 (4) (1994) 1015-1018.

[5] Homburg, C. and Hoyer, W. D. and Koschate, N., "Customers Reactions to Price Increases: Do Customer Satisfaction and Perceived Motive Fairness Matter?" Journal of the Academy of Marketing Science 33 (1) (2005) 36-49.

[6] Dargay, J. M., Are Price and Income Elasticities of Demand Constant? The UK experience. Oxford Institute for Energy Studies, England, 1992.

[7] Fibich, G. and Gavious, A. and Lowengart, O., "The dynamics of price elasticity of demand in the presence of reference price effects." Journal of the Academy of Marketing Science 33 (1) (2005) 66-78.

[8] Dreze, X. and Vanhuele, M. and Laurent, G., "Consumers' immediate memory for prices." Journal of Consumer Research 33 (2) (2006) 163-172.

[9] Carvalho, M., Price Recall, Bertrand Paradox and Price Dispersion with Elastic Demand. Discussion Paper 2009-69, Tilburg University, Center for Economic Research, 2009.

[10] Allenby, G. M., "Modeling marketplace behavior." Journal of the Academy of Marketing Science 40 (1) (2012) 155-166.

[11] Hajdinjak, M., Linear Price Elasticity of Demand and Supply, sent to The Econometrics Journal.

[12] U. S. Energy Information Administration, “How dependent are we on foreign oil?”, May 10, 2013.

NELL: The Never-Ending Language Learning System[1]

Estevam R. Hruschka Jr.[2]
Federal University of Sao Carlos, Sao Carls SP – Brazil
e-mail: estevam@dc.ufscar.br

ABSTRACT

Never-Ending Language Learner (NELL)[1] is a computer system that runs 24/7, forever, learning to read the web. extract (read) more facts from the web, and integrate these into its growing knowledge base of beliefs; and ii) learn to read better than yesterday, enabling it to go back to the text it read yesterday, and today extract more facts, more accurately. This system has been running 24 hours/day for over four years now. The result so far is a collection of 70 million interconnected beliefs (e.g., servedWith(coffee, applePie), isA(applePie, bakedGood)), that NELL is considering at different levels of confidence, along with hundreds of thousands of learned phrasings, morphological features, and web page structures that NELL uses to extract beliefs from the web .

1 INTRODUCTION

Despite tremendous progress in machine learning over the past decades, we still have very limited number of atempts to build machine learning systems that learn cummulatively forever, using what they learned yesterday to improve their ability to learn tomorrow, and improving indefinitely. We seek to build such a system, in the domain of natural language understanding (reading the web). We call this system NELL (Never-Ending Language Learner).

The main motivation for building NELL is based on the belief that we will never really understand machine learning until we can build machines that, like people, have the following characteristics: i) *learn many different types of knowledge or functions;* ii) *from years of diverse, primarily self-supervised experience;* iii) *in a staged curricular fashion, where previously learned knowledge enables learning further types of knowledge;* iv) *where self-reflection and the ability to formulate new rep- resentations and new learning tasks enable the learner to avoid stagnation and performance plateaus*

The domain choice (*reading the web*) is mainly motivated because we believe AI community will not be able produced *Natural Language Understanding* systems (which should go beyond *Natural Language Processing*) until we have computer systems that react to arbitrary sentences by saying one of the following: *I understand, and I already knew that;* or, *I understand, and I didn't know, but I accept that;* or *I understand, and I disagree because ...*

The input to NELL is an initial ontology, with categories and relations, as well as instances (labeled training examples) of these categories and relations. Given this input, our goal is for NELL to run 24 hours/day, 7 days/week, forever, interacting with human trainers for up to an hour per day. On each day, NELL must accomplish two things:

1. **Performance task**: Each day it must extract more factual beliefs from the web in order to further populate its knowledge base, according to the given ontology.
2. **Learning task**: Each day it must learn to read better than it could the previous day.

NELL is evaluated by its success in achieving both of these tasks. To evaluate its success at the first task, we evaluate the correctness and breadth of the beliefs it extracts. To evaluate its success at the second task, we measure the change over time in its competence at task one. For example, we can send it to a sample of the same web sources it visited yesterday, and measure whether it extracts more facts more accurately today than it did yesterday.

2 NELL KEY FEATURES AND CURRENT STATE

Based on the main characteristics mentioned in the previous section, NELL has been built and is running since January, 2010. Currently the system follows its Cumulative, Staged Learning showing that performing a learning task *X* can help improving the ability to learn a different task *Y.* NELL's current ability to learn includes: i) classify noun phrases (NPs) by category; ii) classify NP pairs by relation; iii) discover rules and patterns to predict new relation instances; iv) learn which NPs (co)refer to which latent concepts; v) discover new relations to extend the initial ontology; vi) learn to assign temporal scope to beliefs; vii) learn to *microread* single sentences; viii) vision: co-train text and visual object recognition; ix) goal-driven reading: predict, then read to corroborate/correct. In addition we have started working on making NELL a conversational agent on Twitter and we plan to add a robot body to NELL.

[1] http://rtw.ml.cmu.edu

[2] This paper describes a collaborative research project with significant contributions from the following people: *Tom Mitchell, William Cohen, Partha Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, Jayant Krishnmurthy, Ni Lao, Kathryn Mazaitis, Tahir Mohammad, Ndapa Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, Joel Welling.*

COMPLEX EVENT DETECTION AND PREDICTION IN TRAFFIC

Blaž Kažič, Dunja Mladenić, Luka Bradeško
Artificial Intelligence Laboratory,
Jožef Stefan Institute and Jožef Stefan International Postgraduate School,
Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: blaz.kazic@ijs.si

ABSTRACT

When dealing with large amounts of heterogeneous traffic data streams, how a Complex Event Processing (CEP) system, which can efficiently process and predict complex events in traffic, is set up is a crucial matter. In this paper, several issues and methods related to finding different rules that can be used to develop such a system are presented.

Statistical methods to detect complex events from traffic data are first described. Two types of techniques are used to research relations between complex events: descriptive and predictive data mining. First, association rules are used to analyze data and express regularities in data. Second, decision trees and decision rules algorithms are used for the prediction of complex events.

All the algorithms were tested with regards to how different social events affect the traffic system near the Stozice stadium in Ljubljana, Slovenia. The results show that methods described in this paper are feasible and can be used for developing an advanced traveler information system.

1 INTRODUCTION

This work is inspired by the need for the development of methods for real time complex event detection and processing in urban mobility networks, as proposed is the Mobis [1] project. This is usually done with specialized programs for complex event processing (CEP), such as ESPER [2], ETALIS [3], VANET [4]. These programs are capable of receiving different on-line data streams from which they detect changes in real time. These changes can be specified as events and complex events, and can be used for event processing and stream reasoning in real time. However, all these programs require some background knowledge (e.g. an ontology) and rules preprogrammed into the system.

The main concept of this paper is to investigate methods for extracting an ontology from heterogeneous data and finding useful rules for processing and predicting complex events in traffic. Since traffic anomalies are usually caused by series of other unpredictable events, they are usually extremely difficult to explain and even harder to predict. Nevertheless, there are some cases in which it is obvious that one event can affect the other. Here, we have attempted to determine how large social events in Stozice stadium in Ljubljana, Slovenia, influence other complex events in nearby traffic systems.

2 DATA

In this work, three different types of data were used:

- **Traffic loop sensor data** from a sensor near the stadium. These data contain information on how many cars have passed by in the last hour (for every 5 minutes, from 2011 to 2013).
- **Parking spots sensor data** at Stozice stadium were used for the same period as loop sensors. The sensors returns information about availability of free parking spots every 5 minutes between 8am and 10pm.
- **Data on 50 major social events** at Stozice stadium were collected for the same period as the other data.

2.1 Detecting Complex Events

Complex events had to be extracted from this time series dataset. In this research, a complex event is defined as an anomaly in traffic. Therefore, congested traffic can be considered to be a complex event. For example, traffic congestion in the evening, when there is usually low traffic, can be considered to be a complex traffic event. In contrast, traffic congestion during the morning and afternoon rush hours on workdays are quite common; therefore, they are not considered to be complex traffic events.

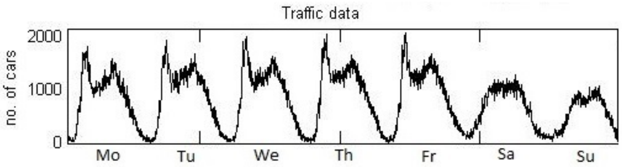


Figure 1: *Traffic loop sensor data for one week.*

There has been a great deal of research in detecting anomalies in data sets [5]. One of the basic methods is to capture the basic statistics of sensor data, for every 30-minute segment of time within each day of the week [6]. This marginal statistic is used to describe “usual” traffic. In order to identify complex events, marginal statistics are compared to real time data. If the difference is greater than a certain set threshold, these states can be marked as anomalous and therefore complex events.

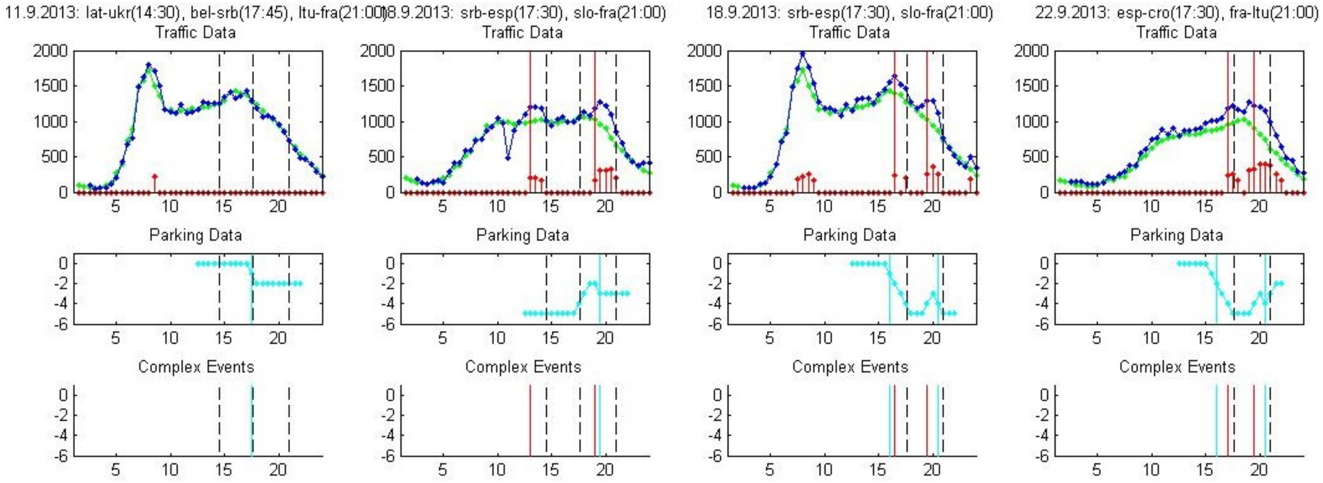


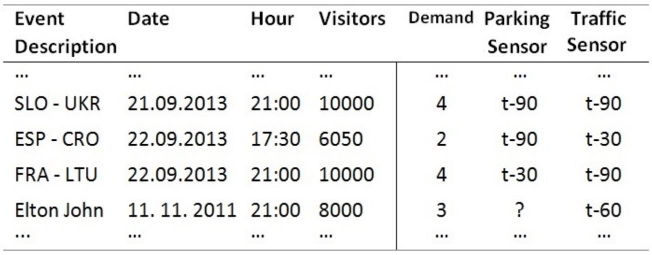
Figure 2: *Complex event detection from traffic loop sensor data and parking sensor data for four bigger social events.*

This relatively straightforward method can be used for anomaly detection in any count data, such as web access logging, security logs in buildings, etc. In the case of this paper, this technique was used to detect congested traffic from traffic loop sensors, as can be seen on Figure 2. Unfortunately, this method could not be used for parking sensor data, because of an inaccurate operation. Complex events for parking data are, therefore, detected as the start of decreasing free parking spots before a social event.

2.2 Data Base

The database was created from extracted complex events. Part of it is seen in Figure 3. Every instance represents one social event in Stozice, which is described with several attributes. For further work in this paper, only the last three attributes were used, since they contain information from all other attributes and are derived from them.

- **Demand attribute** is derived from information on how many visitors had visited one event. It has four possible values, with "1" indicating less than 50% occupancy and "4" indicating a sold out event.
- **Parking sensor attribute** contains information on how soon before a social event, the first complex event happened. It has five possible values; "no" indicating no complex event and "t-0", "t-30", "t-60" and "t-90", meaning complex events happened 0, 30, 60 and 90 minutes before the social event, respectively.
- **Traffic sensor attribute** contains information on events in traffic and has the same possible attribute values as parking sensor. For the predictive data mining part, this attribute is also marked as a target value.



Event Description	Date	Hour	Visitors	Demand	Parking Sensor	Traffic Sensor
...	...	...	...	...	...	...
SLO - UKR	21.09.2013	21:00	10000	4	t-90	t-90
ESP - CRO	22.09.2013	17:30	6050	2	t-90	t-30
FRA - LTU	22.09.2013	21:00	10000	4	t-30	t-90
Elton John	11. 11. 2011	21:00	8000	3	?	t-60
...	...	...	...	...	...	...

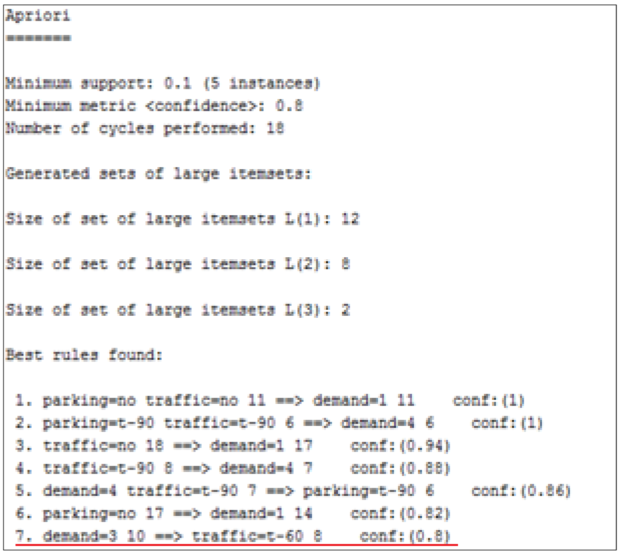
Figure 3: *Part of extracted complex events dataset*

3 DESCRIPTIVE DATA MINING

For descriptive data mining part, a well-known method of association rules was used to express regularities in complex events in the database. With these rules, we can better understand our problem and find some interesting unknown correlations between items in dataset. Two main algorithms (Apriori and Predictive apriori) were used with the help of Weka software [7].

3.1 Association Rules

The Apriori algorithm iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence that can be set as an input parameter. In this work, minimum confidence was set to 0.8.

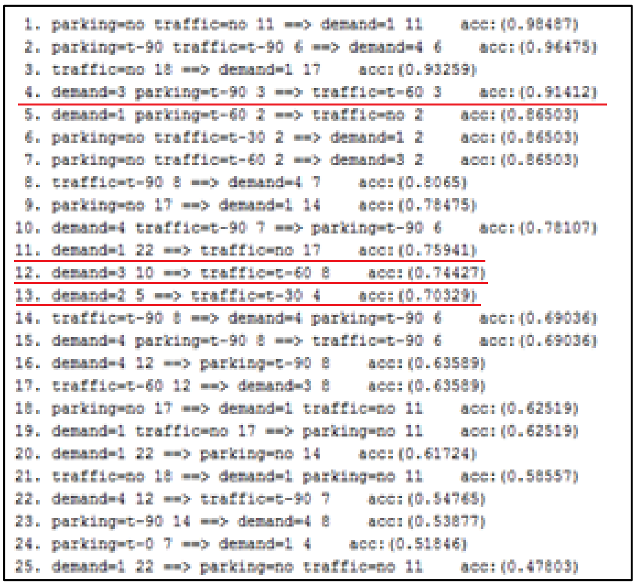


```
Apriori
=======

Minimum support: 0.1 (5 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 8

Size of set of large itemsets L(3): 2

Best rules found:

 1. parking=no traffic=no 11 ==> demand=1 11    conf:(1)
 2. parking=t-90 traffic=t-90 6 ==> demand=4 6    conf:(1)
 3. traffic=no 18 ==> demand=1 17    conf:(0.94)
 4. traffic=t-90 8 ==> demand=4 7    conf:(0.88)
 5. demand=4 traffic=t-90 7 ==> parking=t-90 6    conf:(0.86)
 6. parking=no 17 ==> demand=1 14    conf:(0.82)
 7. demand=3 10 ==> traffic=t-60 8    conf:(0.8)
```

Figure 4: *Association rules from Apriori algorithm.*

From the Apriori algorithm results on Figure 4, we can see that the rules with the highest confidence level are the trivial ones. We read this rules like this: if we know that there is a social event in Stozice tonight, and if there is no complex event in parking sensor stream, and there is no complex event in traffic sensor stream two hours or less before the game, then we can assume that the demand of the game is “1”. From Rule 7, we can explain a complex traffic event. If there is a social event with demand 3 in Stozice, it is likely that there will be a congested traffic 60 minutes before the event.

The second algorithm used was Predictive Apriori. It searches for n number of rules (that we define as an input parameter) with an increasing support threshold, concerning a support-based corrected confidence value.



```
 1. parking=no traffic=no 11 ==> demand=1 11    acc:(0.98487)
 2. parking=t-90 traffic=t-90 6 ==> demand=4 6    acc:(0.96475)
 3. traffic=no 18 ==> demand=1 17    acc:(0.93259)
 4. demand=3 parking=t-90 3 ==> traffic=t-60 3    acc:(0.91412)
 5. demand=1 parking=t-60 2 ==> traffic=no 2    acc:(0.86503)
 6. parking=no traffic=t-30 2 ==> demand=1 2    acc:(0.86503)
 7. parking=no traffic=t-60 2 ==> demand=3 2    acc:(0.86503)
 8. traffic=t-90 8 ==> demand=4 7    acc:(0.8065)
 9. parking=no 17 ==> demand=1 14    acc:(0.78475)
10. demand=4 traffic=t-90 7 ==> parking=t-90 6    acc:(0.78107)
11. demand=1 22 ==> traffic=no 17    acc:(0.75941)
12. demand=3 10 ==> traffic=t-60 8    acc:(0.74427)
13. demand=2 5 ==> traffic=t-30 4    acc:(0.70329)
14. traffic=t-90 8 ==> demand=4 parking=t-90 6    acc:(0.69036)
15. demand=4 parking=t-90 8 ==> traffic=t-90 6    acc:(0.69036)
16. demand=4 12 ==> parking=t-90 8    acc:(0.63589)
17. traffic=t-60 12 ==> demand=3 8    acc:(0.63589)
18. parking=no 17 ==> demand=1 traffic=no 11    acc:(0.62519)
19. demand=1 traffic=no 17 ==> parking=no 11    acc:(0.62519)
20. demand=1 22 ==> parking=no 14    acc:(0.61724)
21. traffic=no 18 ==> demand=1 parking=no 11    acc:(0.58557)
22. demand=4 12 ==> traffic=t-90 7    acc:(0.54765)
23. parking=t-90 14 ==> demand=4 8    acc:(0.53877)
24. parking=t-0 7 ==> demand=1 4    acc:(0.51846)
25. demand=1 22 ==> parking=no traffic=no 11    acc:(0.47803)
```

Figure 5: Association r*ules extracted with Predictive Apriori algorithm.*

From Figure 5, we can see even more rules that are explaining complex traffic events, e.g. Rule 4 also includes complex event regarding parking places. We can understand this rule this way: if we know that there is a social event with demand “3”, and we know that there is a complex event in parking sensor streams, it is highly likely that there will be traffic congestion 60 minutes before the game. We can also see some other interesting rules (11, 12, and 13) that can explain complex traffic events.

4 PREDICTIVE DATA MINING

Rules can also be extracted with certain predictive data mining methods. In this work, decision trees and rule learner methods were used and evaluated.

4.1 Decision Trees

It is possible to transform any decision tree into a set of rules. In this research, decision tree was built with the help of Weka’s J48 algorithm. Complex traffic events were set as the target class. The visualized tree is seen in Figure 6. From this decision tree, we simply derive rules for every leaf. For example, we can assume that traffic will be congested 90 minutes before the game if the game is labeled with demand 4, and so on.

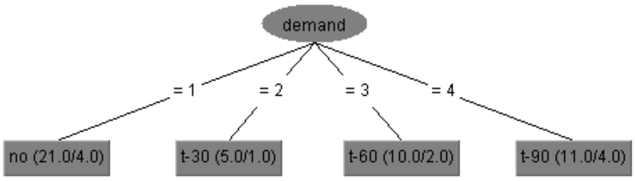


Figure 6: *Pruned decision tree, from which we can extract decision rules*

How complex the rules will be depends on where the tree is pruned. For demonstrating purposes, an unpruned tree was built, which can be seen in Figure 7. In this case, it is seen that parking sensor complex events are also included; therefore, the rules are more complex. Because of the very small dataset (only 50 data samples), we can also see that the number of instances in particular leafs is very low or even zero. Therefore, these rules are not very useful and are shown solely for demonstration of how pruning affects the complexity of rules.

4.2 Rule Learner

Another predictive method that was tested is the decision rule learner. In this case, Weka’s JRip (RIPPER) algorithm was used, according to which classes are examined in increasing size, and an initial set of rules for the class is generated using incremental reduced error. The algorithm proceeds by treating all the examples of a particular judgment in the training data as a class, and finding a set of rules that covers all the members of that class. Thereafter, it proceeds to the next class and does the same, repeating this until all classes have been covered.

Figure 8 shows the rules extracted by JRip Algorithm. It can be seen that the rules JRip are the same as those obtained with pruned decision tree.

An unpruned version of JRip algorithm was also tested, and the results are shown in Figure 9. It is seen that, like in an

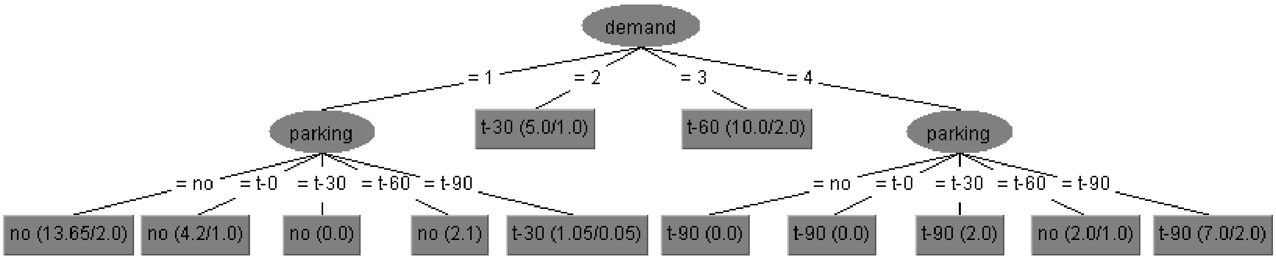
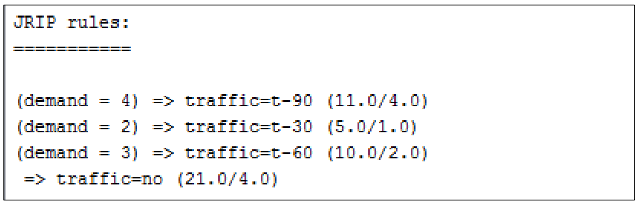


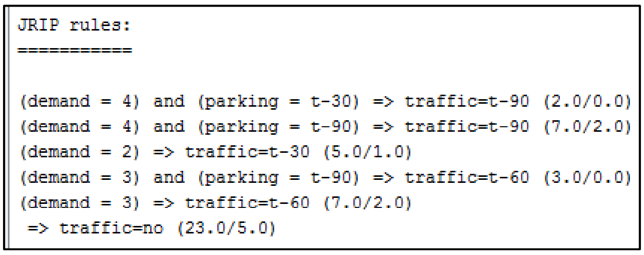
Figure 7: *Unpruned decision tree. Rules are more complex, but number of instances in the leaves is very low because of small dataset.*

unpruned decision tree, the rules are more complex, but also with very few instances in the rules (see Figure 9) because of the small dataset.



```
JRIP rules:
===========

(demand = 4)  => traffic=t-90 (11.0/4.0)
(demand = 2)  => traffic=t-30 (5.0/1.0)
(demand = 3)  => traffic=t-60 (10.0/2.0)
 => traffic=no (21.0/4.0)
```

Figure 8: *Rules extracted with pruned JRip algorithm.*



```
JRIP rules:
===========

(demand = 4) and (parking = t-30) => traffic=t-90 (2.0/0.0)
(demand = 4) and (parking = t-90) => traffic=t-90 (7.0/2.0)
(demand = 2) => traffic=t-30 (5.0/1.0)
(demand = 3) and (parking = t-90) => traffic=t-60 (3.0/0.0)
(demand = 3) => traffic=t-60 (7.0/2.0)
 => traffic=no (23.0/5.0)
```

Figure 9: *Unpruned JRip rules.*

4.3 Evaluation

We can evaluate predictive algorithms and see how the rules extracted with these methods are useful for prediction. Algorithms were compared to the ZeroR algorithm that represents a type of baseline. From the results in Figure 10, we can see that all mentioned algorithms performed better than ZeroR. It is also seen that the evaluation results for the pruned decision tree and pruned JRip rules are entirely the same. This makes sense, since both algorithms extracted the same rules. However, unpruned versions of both algorithms are also the same, but this is more a coincidence since the rules are different. This is probably due to the small dataset. We can also see that pruned algorithms performed better than unpruned ones, which was expected since unpruned models were used only to demonstrate how to extract more complex rules.

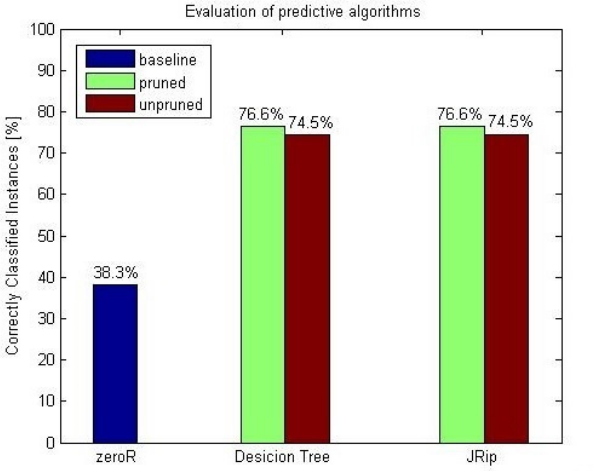


Figure 10: *Decision tree rules and JRip rules evaluation.*

5 CONCLUSIONS

We can conclude that both the descriptive data mining and the predictive data mining methods described in this paper are useful for better understanding of datasets as well as for extracting rules from the given dataset. The author is aware of the problems with the small dataset used in the presented work. Consequently, the next plan is to enlarge the dataset, if possible, and to use these methods on other larger datasets. Regardless of the small dataset, this paper shows that the suggested methods return feasible results and, therefore, can be used to develop advanced travel information systems that can detect, process and predict complex events in traffic.

ACKNOWLEDGEMENTS
This work was supported by the Slovenian Research Agency and the ICT Programme of the EC under MobiS (FP7-318452).

References

[1] Project Mobis, *Personalized Mobility Services for energy efficiency and security through advanced Artificial Intelligence techniques, FP7-ICT*

[2] E. Olmezogullari, Online Association Rule Mining over Fast Data. *Proc. 2013 IEEE International Congress on Big Data (BigData Congress), Santa Clara, CA*. 2013.

[3] D. Anicic, S. Rudolph, P. Fodor, N. Stojanovic. Stream Reasoning and Complex Event Processing in ETALIS, *Proc. Semantic Web – Interoperability, Usability, Applicability, Vol. unpublished: under review (2010), pp. 1-10*

[4] F. Terroso-Saenz, M Valdes-Vela, C. Sotomayor-Martinez, A cooperative approach to traffic congestion detection with complex event processing and VANET, *Proc. IEEE Transactions on Intelligent Transportation Systems, Vol. 13,* 2012.

[5] A. Ihler, J Hutchins, P Smyth, Adaptive Event Detection with Time-Varying Poisson Processes, *Proc. KDD '06 Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 207-216*

[6] E. Horvitz. J. Apacible, R. Sarin, L. Liao, Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service, *Proc. Twenty-First Conference on Uncertainty in Artificial Intelligence, UAI-2005, Edinburgh, Scotland, July 2005.*

[7] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten (2009); The WEKA Data Mining Software: *An Update; SIGKDD Explorations, Volume 11, Issue 1.*

GOVERNMENT TRANSPARENCY THROUGH TECHNOLOGY

Matej Kovačič[*]*, Gaber Cerle*
Centre for Knowledge Transfer in Information Technologies
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773814; fax: +386 1 4773935
e-mail: matej.kovacic@ijs.si, gaber.cerle@ijs.si

ABSTRACT

This paper presents usage of Supervizor, an online application that provides information on financial transactions of the public sector bodies. Supervizor contains by now 50 mio. transactions from both government and local agencies to government contractors from 2003 to 2014 and matches such transactions to company records from the Business Register including director lists and corporate leadership. The application, which has been designed and developed by the Commission for the Prevention of Corruption, won the UN Public Service Award in 2013, an important recognition of excellence in public service. The data on transactions from Supervizor are also provided in machine readable form.

1 INTRODUCTON

Government transparency has been recognized as an important tool empowering citizens, limiting risks for illicit management and corruption and increasing the level of responsibilities of public office holders for effective and efficient use of public finance. This paper presents possible findings of using technology as a tool to increase transparency of public spending and to detect misuse of public funds.

The application called Supervizor [1] is a web application that shows financial transactions of the public sector bodies (direct and indirect budget users, including state and a local level) with legal entities registered in Republic of Slovenia. It has been designed and developed by the Slovenian Commission for the Prevention of Corruption. It received international recognition and won the UN Public Service Award in 2013.

The rest of this paper is structured as follows. Section 2 describes the various data sources for Supervizor, Section 3 briefly comments of legal questions, especially regarding protection of personal data, while examples of usage and comments on the potential impact are given in Section 4. Section 5 gives conclusions.

2 SUPERVIZOR AND ITS DATASETS

Supervizor combines a large amount of data from the following sources:

- data about financial transactions of budget users from January 2003 onwards provided by Public Payments Administration;
- Slovenian Business Register, Register of legal entity‘s bank accounts and the public posting of company's annual reports;
- database of securities and their owners from Central Securities Clearing Corporation;
- registry of taxpayers;
- registry of direct and indirect budget users, which includes the bodies of all three branches of power, independent judicial and state bodies, local communities and their parts with legal personality, public institutes, public funds, public agencies etc.;
- database of public procurements including small value procurement;
- accounting entries of payments for direct budget users from January 2003;
- data about tax debtors;
- data about financial transfers to the so-called favorable tax environments („tax havens“).

Public Payments Administration is providing payment services for direct and indirect users of central and local government budgets. It manages all the financial transaction's flows of public finance within the single treasury system, so Public Payments Administration keeps all data about financial transactions from budget users to private companies in one central database. This database is

[*] Matej Kovačič is a former employee of the Commission for the Prevention of Corruption, where he has been leading development of Supervizor application.

in fact the most important source of data for Supervizor application and is updated daily.

The AJPES (Agency of the Republic of Slovenia for Public Legal Records and Related Services) is keeping records on business entities in Slovenia and manages ePRS Slovenian Business Register as a central public database on all business entities, their subsidiaries and other organization segments located in Slovenia which perform profitable or non-profitable activities. AJPES is also maintaining eRTR - Register of legal entity‘s bank accounts, including the information about the type of the account. AJPES is also managing annual reports of Slovenian business entities. In Slovenia companies (including banks, insurance companies, investment funds and co-operatives), sole proprietors, legal entities governed by public law, non-profit organizations and associations have to present their annual reports to AJPES. These data are published on the Internet and also used for tax and statistical purposes. For companies with a mandated statutory audit, AJPES publishes audited annual reports. Data from Slovenian Business Register and Register of legal entity‘s bank accounts are also updated on a daily basis.

The third main source of data is Database of public procurements in Slovenia. This database is maintained by the public company Official Gazette and Ministry of Finance and contains information about all public procurements, procurements of a small value and notices of awards of the contracts under a framework agreements. Data also includes information of which company received public procurement and the financial value of certain business. This data is also updated daily.

Next important source is MFERAC database, which contains accounting entries of payments for direct budget users. MFERAC is unified application for accounting management and is used by all direct budget users in Slovenia. It is maintained by the Slovenian Ministry of Finance. This data is updated on a daily basis too.

Data about tax debtors is published monthly on the Internet by the Tax administration of the Republic of Slovenia. Published are companies which have their payments delayed for 90 days. Supervizor application scrawles this data from the Tax administration‘s list, OCR's it and imports it into the Supervizor's database. Then it analyses it to identify those tax debtors which were receiving funds from the public sector at the time they have an outstanding tax debt (90 days before they were published on Tax administration‘s list). This list of tax debtors is then published.

Data about financial transfers to the so-called favorable tax environments („tax havens“) are published on the Internet by the Office for Money Laundering Prevention. Supervizor application transfers this data into it's database and shows the transactions on it's website.

3 LEGAL QUESTIONS RELATED TO DATA

It is important to note, that almost all of the data is public and therefore accessible on the basis of the Slovenian Access to Public Information Act or commercially available, so there was no need for any legislative changes regarding Supervizor. Records on business entities, annual reports and information about public procurements are been completely public, however data about financial transactions contain personal and classified data (for instance some data related to salaries, transactions to intelligence and security services, returns of taxes, etc.). These data were requested by the Commission for the Prevention of Corruption on the basis of the Integrity and Corruption Prevention Act, which gives the Commission powers for acquisition of data and documents. Commission then developed algorithms for detection and elimination of those protected data.

Technically, the main challenge was to clean the protected data, link all the data, especially accounting entries and cash flow (financial transactions going through Public Payment Administration) and to optimize the database management and create a web application which is working fast (Supervizor has around 100.000 users per day and contains about 50 mio. of financial transactions).

4 APPLICATION'S USE AND IMPACT

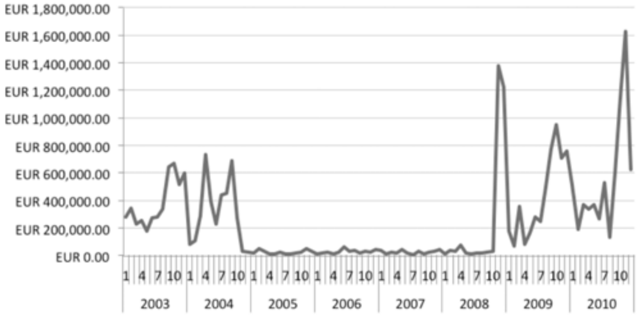
Supervizor was launched on 23rd of August 2011. There was a significant interest by the general public, journalists, researchers and mostly positive response from other budget users, local level and abroad. Public and government bodies are recognizing it as an important tool for transparency, mitigation of corruption risks, tool which decreases risks for illicit management, abuse of functions and unfair competitiveness and clientelism in public procurement procedures. Since financial transactions and financial flow analyses are a vital part of the evidence-gathering process when investigating economic crime, public finance crime and corruption, Supervizor is also used by Police, Court of Audit, Commisiion for the Prevention of Corruption, Tax Administration, Office for Money Laundering Prevention, etc.

Supervizor allows oversight and visualization of all financial transactions of Slovenian public sector in a simple and understandable way, including date and amount of transactions and for the payments over 2000 EUR also the purpose of money transfers. Application also shows also ownership and management structure of the Slovenian companies and some data from their annual reports, granted public procurements and all other relevant information from the used data sources. The application provides simple user interface for searching information about public expenditures and companies/recipients of public funds.

Since 2014, the data on transactions from Supervizor are being provided as open data in machine-readable form, in order to encourage researchers and interested public to perform their own analysis.

Commission already performed some interesting analysis of financial transactions which aim was to detect if there is a link between individual governments and disbursement of funds to particular companies.

Graph 1: Payments to a selected subset of 65 companies between 2003 and 2010 [2]

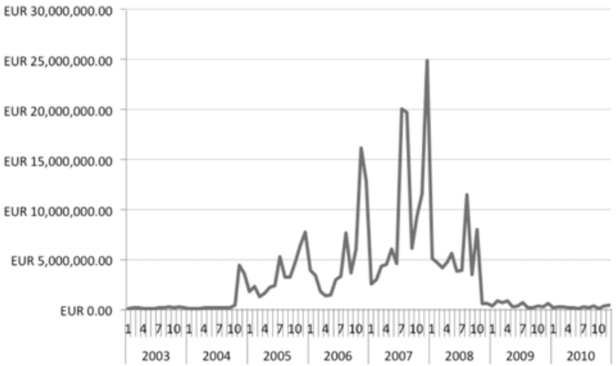


Graph 1 shows the payments to a selected subset of 65 companies between 2003 and 2010. We can clearly see drop in payments from November 2004 till November 2008, which time wise corresponds to government's change in that period.

The analysis [2] has shown high correlation between the change of government in power and money disbursements from budget users to a limited number of companies, high inflexibility of the market for certain services (namely IT services, pharmaceutical products, construction works, etc.) and the existence of a group of companies which are highly dependent on the financial transfers from direct budget users (they receive a great amount of their income from budget users only), which constitutes a noticeable risk of corruption.

The key benefit of the project is that money flows from the public to private sector are accessible to the public quickly and in a simple way. Also, the use of application is completely free and requires no user registration. Supervizor's proactive approach to transparency of public finances is making both government expenditure and business environment more transparent. The project has proved that it is possible to increase the transparency with a minimal financial input and in accordance with the existing legislation.

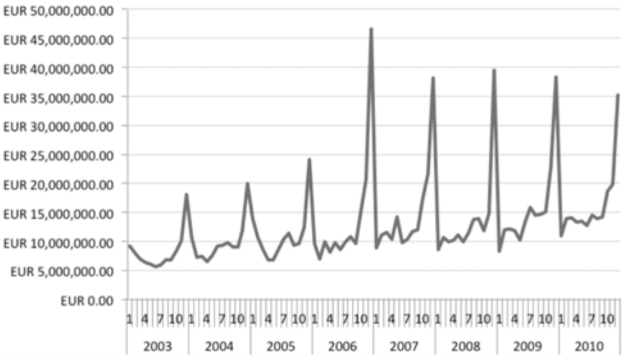
Graph 2: Payments to a selected subset of 252 companies between 2003 and 2010 [2]



Graph 2 shows the payments to a selected subset of 252 companies between 2003 and 2010. We can clearly see increase in payments from November 2004 till November 2008, which time wise corresponds to government's change in that period.

In 2013, the online application received the United Nations award for the excellence in public administration. With the amendments to the Slovenian Access to Public Information Act in 2014, the application Supervizor will be able to also include the transactions of the enterprises wholly owned by the state or local authorities.

Graph 3: Payments (between 2003 and 2010) to all companies registered for IT related activities [2]



Graph 3 shows dynamics of payments to all companies, registered for IT related activities between 2003 and 2010. Graph clearly shows increase of received payments at each December, which corresponds to end of a tax year when budget users should refund unused funds to a central budget.

5 CONCLUSION

The data used in Supervizor application is highly structured and therefore quite simple to analyze and link together. However, the main challenge was to obtain the data, to clean personal and other protected data and to open it, and to create an application with necessary database optimizations which visualizes money flows from the public to private sector quickly and in a simple way.

However, the project has shown that it is possible to increase the transparency with a minimal financial input and in accordance with the existing legislation, only with a good idea and devotion to the ideal of transparency.

References

[1] Supervizor. <http://supervizor.kpk-rs.si>. Ljubljana. 2014. In Slovenian.

Komisija za preprečevanje korupcije. Letno poročilo 2010 [z dodatkom do vključno maja 2011]. <https://www.kpk-rs.si/download/t_datoteke/9349>. Ljubljana, Slovenia, 2010, pp. 43-46. In Slovenian.

TOWARDS SOCIAL MEDIA MINING: TWITTEROBSERVATORY

Inna Novalija, Miha Papler, Dunja Mladenić
Artificial Intelligence Laboratory
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773144; fax: +386 1 4251038
e-mail: inna.koval@ijs.si

ABSTRACT

This paper presents an approach to social media mining based on a pipeline that implements observing, enriching, storing, modeling and presentation techniques. While social media mining solutions have been used for information extraction and sentiment identification about certain topics, applying social media for modeling and nowcasting is a promising new research direction. A novel tool TwitterObservatory that allows observing, searching, analyzing and presenting social media is introduced as a part of the research. Illustrative examples of using the proposed pipeline show how the TwitterObservatory implementing the pipeline can support the user in interaction with the social media data.

1 INTRODUCTION

Social media mining refers to data mining of content streams produced by people through interaction via Internet based applications. Social media mining is usually associated with noisy, distributed, unstructured and dynamic data, as well as with informal text processing.

In this paper we propose a pipeline for social media mining that includes observing, enrichment, storage, modeling and user interface components.

In this research we introduce a novel TwitterObservatory tool for observing, searching, analyzing and presenting information obtained from social media and in particular, from Twitter[1].

The paper is structured as follows: Section 2 contains the related work on social media mining; Section 3 describes the social media mining pipeline at a high level; Section 4 introduces the observing techniques for social media; Section 5 provides the insights into enriching and storing techniques for social media; Section 6 presents the user interface; Section 7 introduces modeling as a part of social media mining pipeline and finally, Section 8 concludes the paper.

2 RELATED WORK

The related work in the area of social media mining covers a number of interesting and relevant topics. In particular, researchers discussed summarization of tweets according to the given query [1], summarization of YouTube comments with sentiment detection and tag cloud [2], identification of the main headlines for the day with language modeling [3]. A number of approaches to classification of the informal text have been suggested by Irani et al. [4], Ramage et al. [5], Lambert et al. [6] and Sriram et al. [7]. And while some researchers have been dealing with spam versus non spam classifications [4], other clustered twitter stream into several topics [5] or classes, such as news, events, opinions [7]. Retrieval of the relevant tweets based on trained language model for each hash-tag on tweeter has been covered by [8]. Rupnik et al. [9] suggest a method for multilingual document retrieval through hub languages, which have alignments with many other languages. A special attention should be dedicated to the approaches dealing with sentiment detection in social media streams. Sentiment detection has been performed at different levels, starting with user sentiments about certain topics [10]. Štajner et al. [11] addressed the problem of sentiment analysis in an informal setting in different domains and two languages.

3 SOCIAL MEDIA MINING PIPELINE

In this paper we present a pipeline that implements a complete mechanism for mining social media. The pipeline (Figure 1), uses EventRegistry [12] mechanisms and includes observing, enrichment, storage, modeling and user interface components.

The pipeline finds its practical implementation as a TwitterObservatory[2] tool. TwitterObservatory uses data observation, enrichment and storage techniques for social media data presentation, search and analytics. In addition,

[1] ***twitter***.*com*

[2] ***twitterobservatory***.*net*

TwitterObservatory provides a suitable user interface that allows users to:

- observe upcoming tweets, search by keywords,
- search social media data by keywords, hashtags etc.
- analyze social media data by volume, sentiment etc.

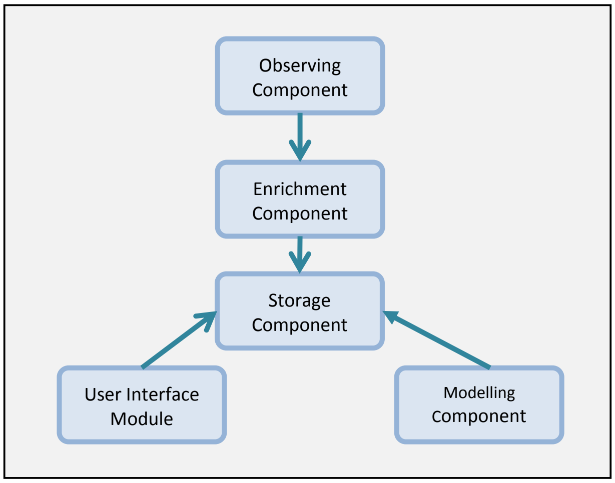


Figure 1: Social Media Mining Pipeline

4 OBSERVING SOCIAL MEDIA

The observing functionalities and mechanisms behind social media mining pipeline and TwitterObservatory are provided in subsections 4a Observing Tweets by Locations and 4b Observing Tweets by Keywords.

a. OBSERVING TWEETS BY LOCATION

This subsection provides description of approaches behind gathering social media data.

For obtaining social media data from Twitter we have used REST Twitter API[3].

The Twitter API allows observing tweets by geo coordinates.

Location can be considered as an important parameter of social media data, since modeling, analyzing and nowcasting is often a location based task.

In this research we have used geo coordinates from United Kingdom. Ten largest cities (by population) have been picked out, then according to Twitter API requirements we have formed a geo coordinates boxes around each city coordinate and set them as filters into Twitter API requests.

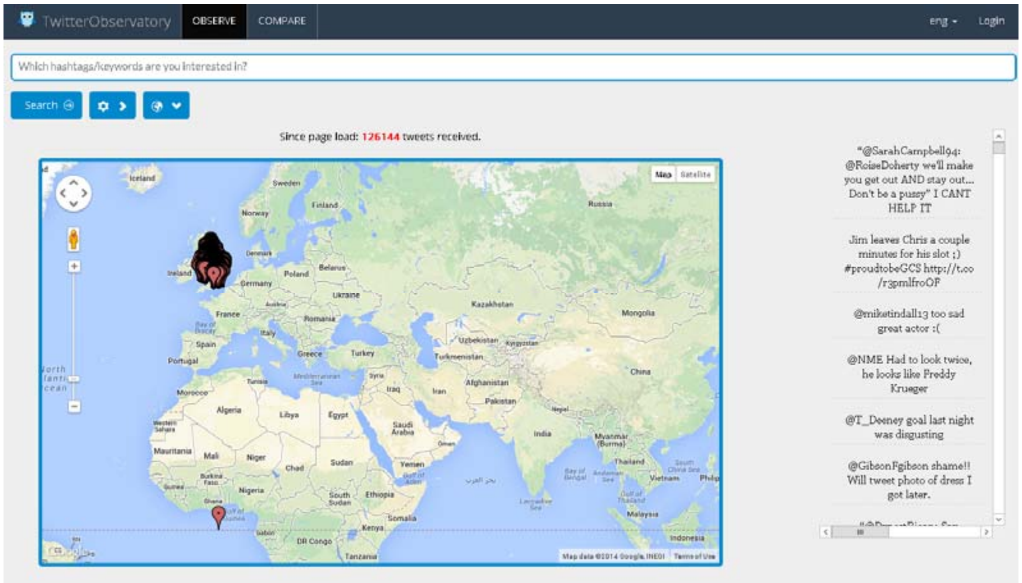
Figure 2 demonstrates upcoming tweets from UK – the location of each upcoming tweet is pinned up on the map and the textual content of the tweet is provided on the right panel.

Overall, we have obtained 31 GB of location based tweets from United Kingdom for a period from November 2013 until July 2014.

As possible to notice, the approach used for data gathering can be easily adapted for other geographical places.

b. OBSERVING TWEETS BY KEYWORDS

Another technique for obtaining tweeter data is to filter social media data by keywords. Up to 400 keywords can be used in one application that uses REST Twitter API. For implementation of this approach we have used the following procedure:



Since page load: 126144 tweets received.

"@SarahCampbell94: @RoiseDoherty we'll make you get out AND stay out... Don't be a pussy" I CANT HELP IT

Jim leaves Chris a couple minutes for his slot ;) #proudtobeGCS http://t.co/r3pmlfroOF

@miketindall13 too sad great actor :(

@NME Had to look twice, he looks like Freddy Krueger

@T_Deeney goal last night was disgusting

@GibsonFgibson shame!! Will tweet photo of dress I got later.

Figure 2: Observing Tweets by Locations (UK)

[3] *https://**dev.twitter**.com/docs/api/1.1*

- select most common words from Wikipedia,
- set up to 400 words in the filter.

An example of most common words in English, according to ranks, are the following: “the”, “be”, “to”, “of”, “and”, “a”, “in”, “that”, “have”, “I”.
Over 500 GB of tweets data have been observed using keyword filters.

5. ENRICHING AND STORING SOCIAL MEDIA DATA

In this section enrichment and storage components of social media data are briefly discussed.
In order to generate additional features that can be used for modeling and nowcasting, we perform enrichment of social media data. The most typical enrichment tasks include sentiment and cross-lingual topic identification of social media data. Enrycher[4] and XLing[5] tools are used for these purposes.
Storage and analytics of social media data is one of the main tasks of the social streams processing infrastructure. Storage component of social media mining pipeline is based on QMiner[6] tool functionalities. QMiner is a data analytics platform for processing real-time streams of structured or unstructured data.

6. USER INTERFACE

In order to give the users a possibility to observe social media data and perform simple analytics tasks based on their experience and intensions, social media mining pipeline contains a user interface module.
In particular, TwitterObservatory provides a suitable user interface that allows user to view upcoming social media data (tweets), search tweets by different queries and analyze the search results within different dimensions.
One of the TwitterObservatory functionalities demonstrated at Figure 3 is the possibility to filter the stored social media data by keywords. Keyword “job” is provided as a filter for our storage.
The users can view the text of tweets, the author of the tweet and the publishing date/time.
Figure 4 presents a possibility to obtain a tag cloud for tweets filtered by keyword “job”. The most relevant tags are: “good”, “today”, “time”.
Figure 5 shows a sentiment graph for tweets filtered by keyword “job”. Sentiment varies on a daily basis.
Figure 6 shows a timeline (or volume) for tweets filtered by keyword “job”. Volume of tweets varies on a daily basis.

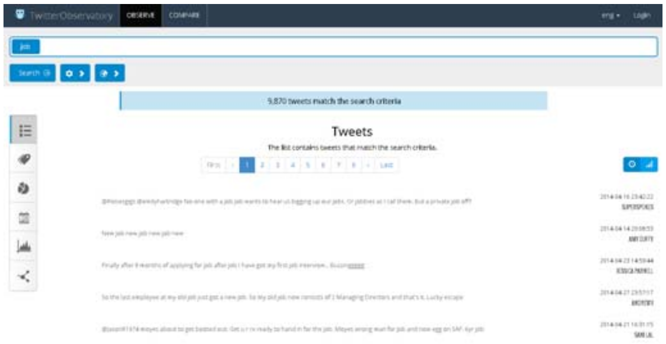

TwitterObservatory
OBSERVE
COMPARE
eng
Login
job
Search
9,870 tweets match the search criteria
Tweets
The list contains tweets that match the search criteria.
First
1
2
3
4
5
6
7
8
Last

Figure 3: Observed Tweets with Details (Filter: “job”)

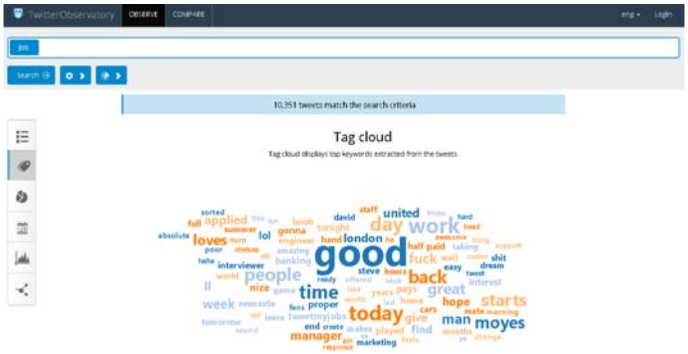


Figure 4: Tag Cloud for Tweets (Filter: ”job”)

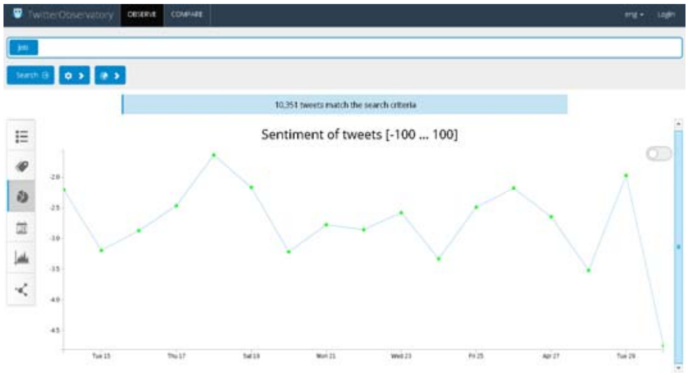
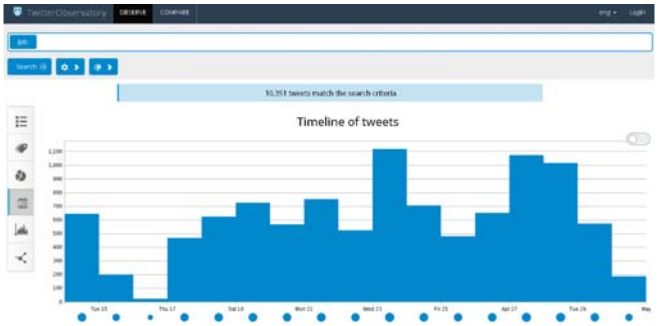


Figure 5: Sentiment for Tweets (Filter: ”job”)

7. INTRODUCTION TO DATA MODELING

An important part of social media mining pipeline is a modeling component. Modeling and nowcasting functionalities are intended to connect social media with external datasets, such as macroeconomic data.
In particular, the goal of modeling and nowcasting is to relate micro-signals coming from social media (such as

[4] ***enrycher***.*ijs.si*
[5] ***xling***.*ijs.si*
[6] ***qminer***.*ijs.si*



10,351 tweets match the search criteria

Figure 6: Tweets Timeline (Filter: ”job”)

micro-signals related to stocks, micro-signals related to labor, micro-signals related to consumers, micro-signals related to real estate and credit, micro-signals related to energy) with macro-economic variables.

In order to perform the modeling and nowcasting, a set of features extracted from social media (features, such as volume, sentiment, trending concepts or hashtags etc) should be applied.

First test will be done on data such as NTSF indices and other stock indices relevant to regional based crawling of tweets (most twitter data crawled so far is from UK). Also historic stock market data provides relatively unbiased, general, frequent and free to use data, which we think will be a good starting point for building models. What we are hoping to see in the initial steps when correlating twitter and stock market data is spikes in volume of published tweets with relevant keywords and hashtags. Through further analysis of different correlations we would like to find a map (maybe even a graph) of keywords that best responded to events in each macroeconomic data.

Later we will expand the model for a wide variety of macroeconomic data from different fields as mentioned before. Frequency is an important factor in what data will be used, and so is diversity. Ideally we want to make a model which will cover all aspects of economic environment, so we could study how events in different areas of economy influence public opinions which we hope to see mirrored in twitter data.

Combined features from social media should be correlated with macroeconomic time series, with a number of operators for time series analysis used (moving average (MA), exponential moving average (EMA), moving average convergence/divergence (MACD), moving norm, variance, moving variance, standard deviation, moving standard deviation, differential, derivative, skewness, kurtosis, volatility).

8. CONCLUSION AND FUTURE WORK

In this paper we presented an approach for social media mining based on a pipeline that implements observing, enriching, storing, modeling and presentation techniques. A novel tool TwitterObservatory that allows observing, searching, analyzing and presenting social media has been introduced.

The developed software components enable monitoring of social media stream including enrichment and storing of the data.

The future work will be based on implementing additional functionalities for social media mining pipeline and on developing extensive modeling and nowcasting functionalities for social media and external datasets.

9. ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and SYMPHONY EU FP7 project (Contract no: 611875).

References

[1] Sharifi, B., Hutton, M.-A., & Kalita, J. (2010). Summarizing Microblogs Automatically. NAACL HLT 2010.

[2] Potthast, M., & Becker, S. (2010). Opinion Summarization of Web Comments. ECIR 2010.

[3] Lee, Y., Jung, H.-Y., Song, W., & Lee, J.-H. (2010). Mining the blogosphere for top news stories identification. SIGIR 2010.

[4] Irani, D., Webb, S., & Pu, C. (2010). Study of Static Classification of Social Spam Profiles in MySpace. ICWSM 2010.

[5] Ramage, D., Dumais, S., & Liebling, D. (2010). Characterizing Microblogs with Topic Models. ICWSM 2010.

[6] Lampert, A., Dale, R., & Paris, C. (2010). Detecting Emails Containing Requests for Action. NAACL HLT 2010.

[7] Sriram et al, B. (2010). Short text classification in twitter to improve information filtering. SIGIR 2010.

[8] Efron, M. (2010). Hashtag retrieval in a microblogging environment. SIGIR 2010.

[9] Rupnik, J., Muhič, A., & Škraba, P. (2012). Multilingual Document Retrieval through Hub Languages. SiKDD 2012.

[10] O'Connor et al, B. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. ICWSM 2010.

[11] Štajner, T., Novalija, I., & Mladenić, D. (2012). A service oriented framework for natural language text enrichment. Informatica Journal, 34:3, 307-313.

[12] Leban, G., Fortuna, B., Brank, J., & Grobelnik, M. Event Registry – learning about world events from news, In Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion.

A System For Large Scale Data Exploration and Organization

Luis Rei, Blaz Fortuna, Marko Groblenik, and Dunja Mladenić
Artificial Intelligence Laboratory, Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia
Tel: + 386 14773528; fax: + 386 14773851
e-mail: {luis.rei, blaz.fortuna, marko.grobelnik, dunja.mladenic}@ijs.si

ABSTRACT

We present a semi-automatic data exploration and organization tool. The system integrates machine learning and text mining algorithms into an simple user interface and a Client/Server architecture. The main features of the systems include unsupervised and supervised methods for concept suggestion, visualization and ability to make both data and methods available to other applications as a service.

1 INTRODUCTION

This paper presents an approach and system intended to support the user in managing textual information overload and expert scarcity by leveraging ontologies and machine learning to semi-automatically organize information. The developed system is focused on the ease of use and simple integration of discovered models with other software. It can interactively handle large datasets in the gigabyte range. Small modifications to its interactive interface could easily allow it to handle datasets over one terabyte.

The tool is meant for use by anyone who wants to rapidly understand unlabeled data and/or build classifiers and services from it. Regardless of whether they are machine learning experts, domain experts or not. Some of the methods used in the proposed tool were first proposed, explored and evaluated in OntoGen [1, 2], a desktop application built for semi-automatic construction of topic ontologies from text corpora.

We will begin by providing an overview of the architecture, its relative advantages. Following, we describe some of the methods and algorithms used. Next, we detail an example use case. Lastly we will present our planned future work.

2 ARCHITECTURE OVERVIEW

The developed system, Elycite[1] , is an Open Source Client/Server application. We built it on top of QMiner[2] , an open source data analytics platform for processing large-scale real-time streams of structured and unstructured data. QMiner is used as a database and server where the data is stored and where all the machine learning algorithms are executed. The current Elycite version works with textual data, however QMiner allows the user to load any data that can be represented as a series of JSON records. Elycite allows the user to choose which field to use for different steps in the exploration process. The client component of the software is a browser based interface. This Client/Server architecture provides several important characteristics:

- **Computational capability** - the increase in the amount of data to be analyzed requires an increase computational capability. The previously mentioned OntoGen focused on smaller datasets which could be handled by a personal computer. Elycite focus on much larger datasets while allowing it's interface to run on portable computers and tablets.

- **Collaborative** - the increase in interest of data analysis and large amounts of data to analyze, in some cases, means that more people need to work together to sift through it. With Elycite, several people can work together, possibly simultaneously, to organize the same data, and build an ontology and classifiers collaboratively by sharing the same server, while using different clients.

- **Services** - the server component provides REST [3] services which can be used by either the client component or other applications. For example, a concept can be easilty changed into a classifier and exposed through a web service. Other applications can use this web service to classify documents into the developed ontology without the need to implement any additional steps.

- **Web Interface** - Web interfaces became popular during the early 2000s. A lot of traditional desktop software has or is moving to the Web. The adoption of smartphone and tablets has left the Web as the common interface across all platforms. Our tool provides a simple, interface, with multiple visualizations available and hidden-by-default *Advanced options*.

[1] Elycite: https://github.com/lrei/elycite
[2] QMiner: http://qminer.ijs.si/

3 METHODS

Elycite works on documents, more precisely, QMiner records. Each individual document is known as an instance. The user organizes the documents into sets of related documents which form concepts. Concepts can have sub-concepts. For example, we can think of all news articles that are related to sports and group them together under the concept *Sports*. If we then group all articles related to football and call it the concept *Football*, we naturally we consider the latter to be a sub-concept of the former. The concepts and the relationships between them together form an Ontology. Concepts can be created manually or with the aid of machine learning techniques. We have combined a set of unsupervised, semi-supervised and supervised techniques for organizing and exploring a dataset. All algorithms work on a common feature representation. The features are extracted using preprocessing techniques for textual data such as stemming and stop-word removal.

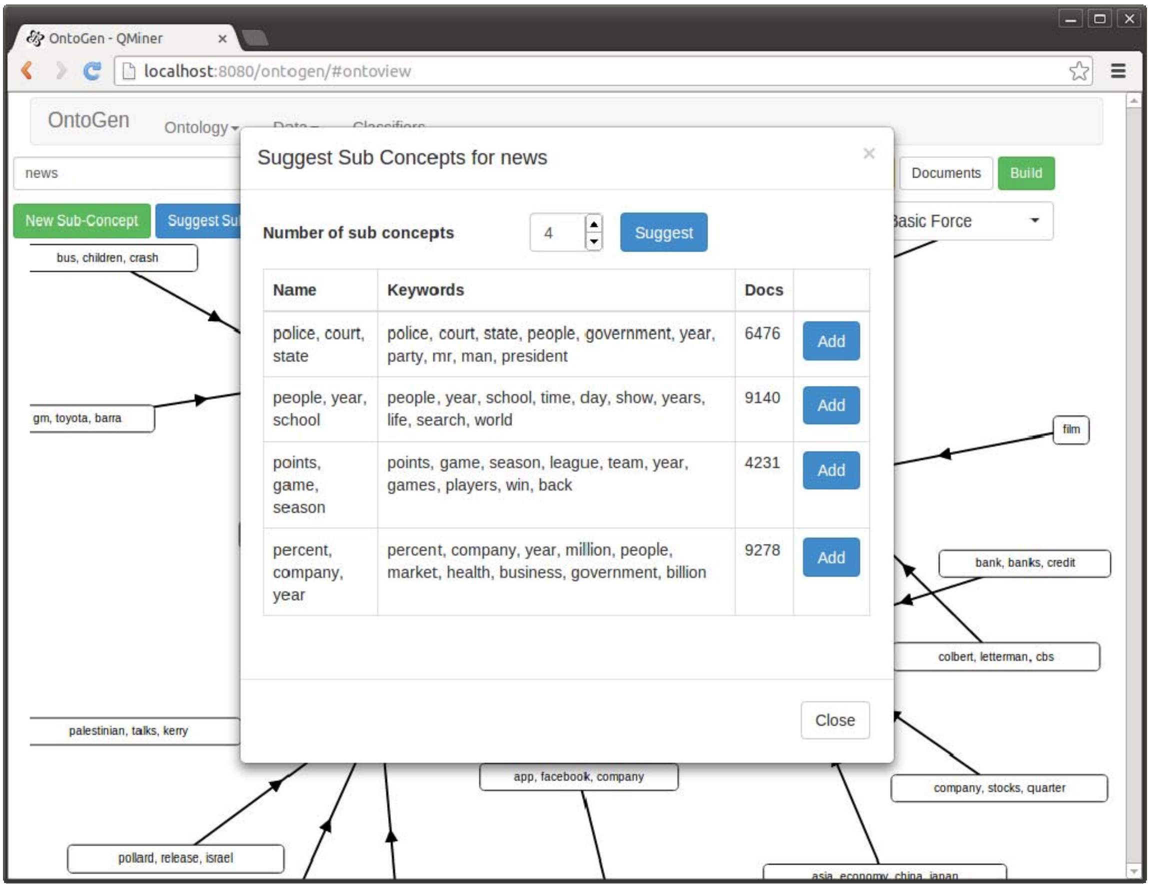
3.1 Unsupervised Methods

We use TF/IDF keyword extraction throughout the application to provide a human readable summary of a set of documents such as those that constitute a concept, which is really just a set of documents grouped together.

Unsupervised concept suggestions are based on the KMeans++ clustering method from [1, 2] where clusters of instances from the selected concept are treated as sub-concept suggestions. The main advantage of the unsupervised method offered is that it requires very little input from the user – only the number of sub-concepts (and even that would be possible to omit in further development of the system is required). Figure 1 shows an example of clustering based sub-concept suggestion in a news dataset.

3.2 Semi-supervised Methods

For semi-supervised method we use SVM based active learning method [4], where the user supervision is provided first by a query describing the concept that the user has in mind and followed by an initial sequence of Yes or No questions. This process is shown in Figure 2. The system refines the suggested concept after each reply from the user and the user can decide when to stop the process based on how satisfied he is with the current concept suggestion (based on keywords and number of documents). A user could start by providing the query *election* on a news article dataset and create a concept that includes all articles related to a specific election, all elections or even politics in general, depending on the answers given to the questions.



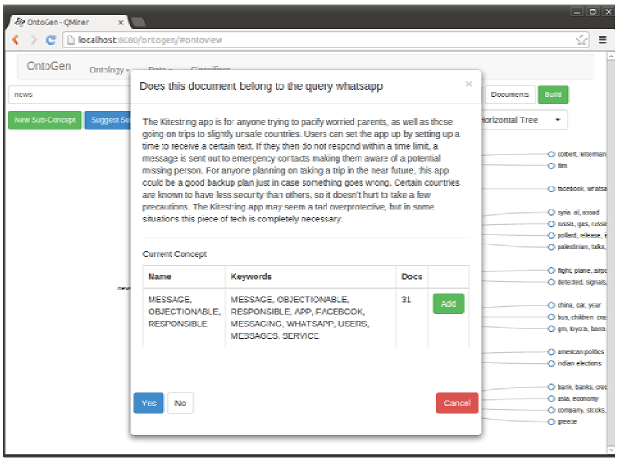
Suggest Sub Concepts for news

Number of sub concepts: 4 Suggest

Name	Keywords	Docs	
police, court, state	police, court, state, people, government, year, party, mr, man, president	6476	Add
people, year, school	people, year, school, time, day, show, years, life, search, world	9140	Add
points, game, season	points, game, season, league, team, year, games, players, win, back	4231	Add
percent, company, year	percent, company, year, million, people, market, health, business, government, billion	9278	Add

Close

Figure 1: C*lustering based sub-concept suggestion.*



Does this document belong to the query whatsapp

The Kitestring app is for anyone trying to pacify worried parents, as well as those going on trips to slightly unsafe countries. Users can set the app up by setting up a time to receive a certain text. If they then do not respond within a time limit, a message is sent out to emergency contacts making them aware of a potential missing person. For anyone planning on taking a trip in the near future, this app could be a good backup plan just in case something goes wrong. Certain countries are known to have less security than others, so it doesn't hurt to take a few precautions. The Kitestring app may seem a tad overprotective, but in some situations this piece of tech is completely necessary.

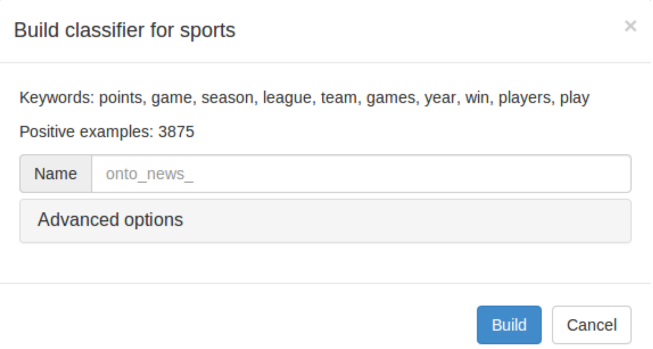
Current Concept

Name	Keywords	Docs	
MESSAGE, OBJECTIONABLE, RESPONSIBLE	MESSAGE, OBJECTIONABLE, RESPONSIBLE, APP, FACEBOOK, MESSAGING, WHATSAPP, USERS, MESSAGES, SERVICE	31	Add

Figure 2: *Active Learning based sub-concept suggestion.*

3.3 Supervised Methods

The user can also select a concept, created manually, semi-automatically or fully automatically, and press a button to build a classifier from it. Figure 3 shows the resulting dialog which allows naming the classifier to be built and allows the user to specify additional parameters, such as classifier hyperparameters, hidden by default.


Build classifier for sports

Keywords: points, game, season, league, team, games, year, win, players, play

Positive examples: 3875

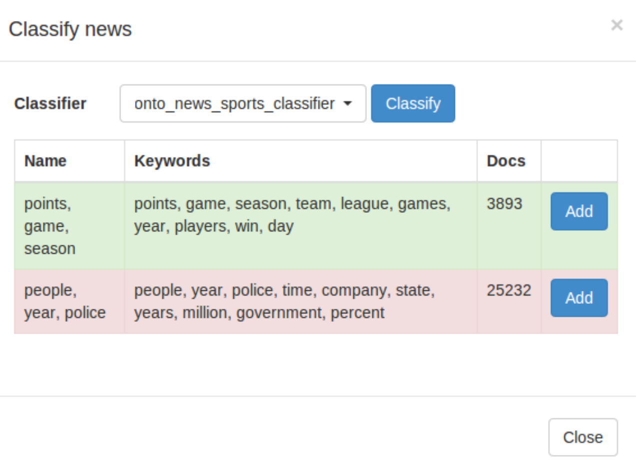
Name: onto_news_

Figure 3: *Creating a classifier from a concept.*

This feature allows the user to create concepts in a new ontology based on concepts from a previous ontology or use a classifier built for one concept to create sub-concepts in a different concept. This is shown in Figure 4. Consider a set of classifiers created from an ontology developed on a news article dataset. These could be used to classify datasets of web pages or tweets into the same ontology concepts e.g. *Politics*, *Sports*, etc. Now consider working on a dataset of publicly traded company descriptions. It is possible to have the concept of *Research* as a sub-concept of the concepts of *Energy*, *Medical* and *Technology*. In this case, a classifier that identifies technology companies that are research focused could also be used for indentifying energy or medical companies that are also research oriented.

We use binary classifiers and thus it is possible, though not necessary, to create a positive concept and a negative concept. This divides a set of documents into documents that belong to a concept and concepts that do not. Figure 4 shows the positive concept with a green background and the negative concept with a red background.

Another interesting possibility is to create a concept in the current corpus using the semi-supervised techniques described above, build a classifier for it and use that classifier in a different application thanks to Elycite's ability to provide classifiers as services.


Classify news

Classifier: onto_news_sports_classifier

Name	Keywords	Docs	
points, game, season	points, game, season, team, league, games, year, players, win, day	3893	Add
people, year, police	people, year, police, time, company, state, years, million, government, percent	25232	Add

Figure 4: *Using a classifier to create sub-concepts: positive (green) and negative (red).*

4 EXAMPLE USE CASE

A typical workflow begins with an exploratory phase (1), where the user tries to understand the content, topics and their distribution in the corpus. In the subsequent development phase (2), the user develops models for some or all of the previously discovered topics. The final phase (3) consists of applying, the developed models to new data. Let us now consider the case of analyzing comments from marketing questionnaires, where customers provide feedback in a form of a unstructured text.

1. The analyst identifies topics and subtopics in the comments using clustering, with automatically extracted keywords as summaries for each topic. Using active learning, the analyst can drill down to a particular issue or topic he is interested in. This step is illustrated in Figure 5 where a dataset of 80,000 twitter profiles is explored based on the user's description.

2. The analyst creates classifiers for particular topics and exposes them as services. The process of creating a classifier was shown in Figure 3.

3. Services are integrated with another application to provide real-time monitoring, personalized feedback or promotional offers.

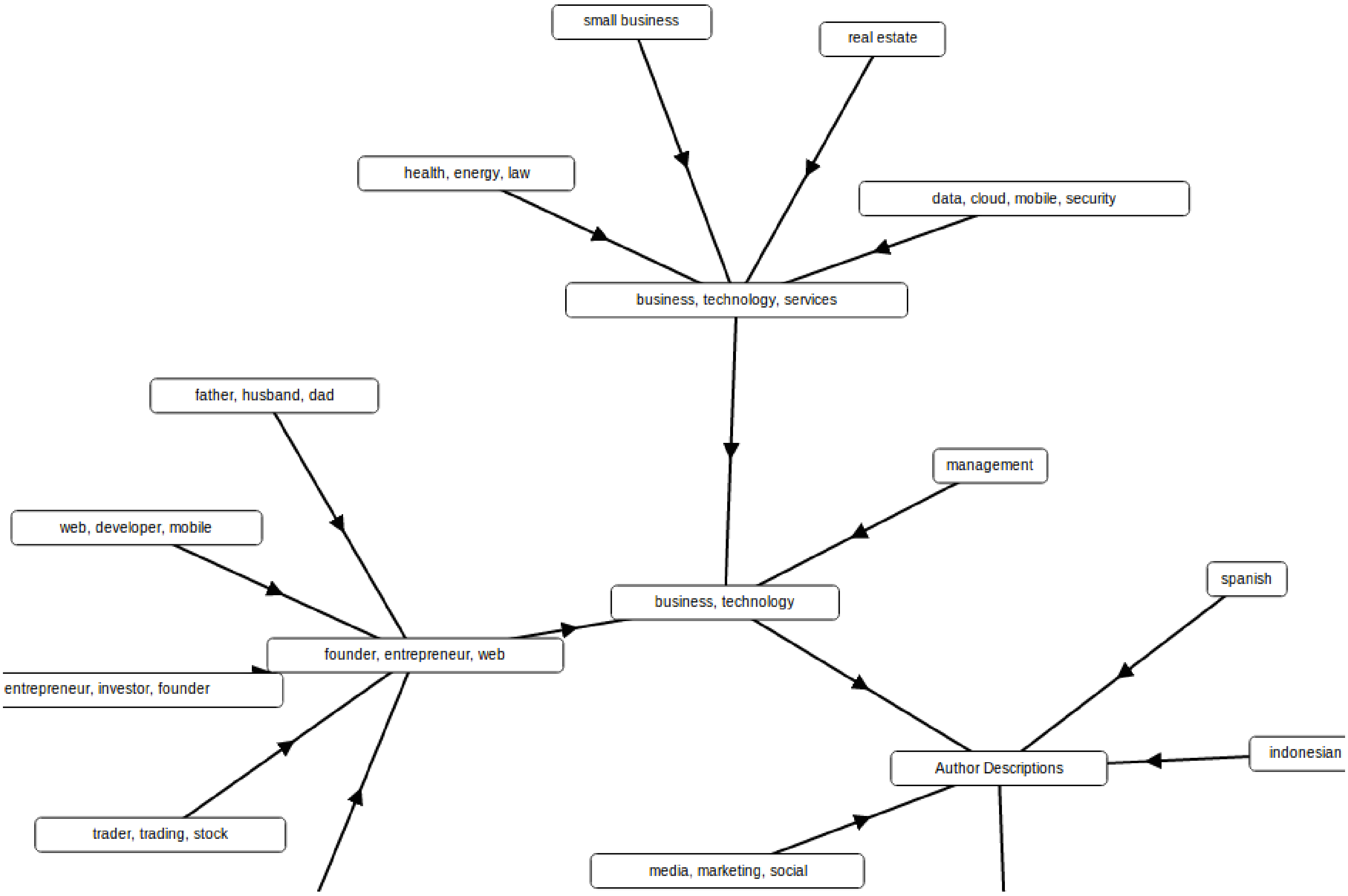


Figure 5: *Visualization of part of an ontology created from a dataset of twitter profile descriptions with Elycite.*

5 FUTURE WORK

Some of the first ideas for future work include addressing the previously mentioned issues related to the interactive use of datasets larger than a few gigabytes and by making incremental improvements in preprocessing such as n-gram generation and the use of the hashing trick [5]. We plan to implement guided learning [6], where the objective is to find documents belonging to a very rare class (i.e. concept) semi-automatically. We also want to provide high quality pre-built classifiers for common tasks such as sentiment classification and topic categorization. The most ambitious part of our planned future work is providing support for numerical, graph, image and time series data. This implies the addition of new preprocessing, feature exatraction or learning, and concept discovery methods and visualizations.

ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT Programme of the EC under XLime (FP7-ICT-611346).

References

[1] Fortuna, B., Grobelnik, M., Mladenic, D.: OntoGen: Semi-automatic Ontology Editor. Human Interface and the Management of Information. Interacting in Information Environments 4558(Chapter 34), 309–318 (2007)

[2] Fortuna, B., Mladenic, D., Grobelnik, M.: Semi-automatic construction of topic ontologies. Semantics, Web and Mining 4289, 121-131 (2006)

[3] Fielding, R. T., Taylor, R. N.: Principled design of the modern Web architecture. ACM Transactions on Internet Technology 2, 115-150 (2002)

[4] Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research 2, 45–66 (2002)

[5] K. Weinberger, A. Dasgupta, J. Attenberg, J. Langford, A. Smola: Feature hashing for large scale multitask learning. In Proceedings of the International Conference on Machine Learning (2009)

[6] Attenberg, J., & Provost, F.: Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining* (2010)

TWEETVIZ: TWITTER DATA VISUALIZATION

Dario Stojanovski, Ivica Dimitrovski, Gjorgji Madjarov
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Rugjer Boshkovikj 16, 1000 Skopje, Macedonia
e-mail: stojanovski.dario@gmail.com,
{ivica.dimitrovski, gjorgji.madjarov}@finki.ukim.mk

ABSTRACT

Twitter is the leading micro-blogging and social network service and is attracting an enormous amount of attention in recent years. Users on Twitter generate an abundance of information every day, establishing Twitter as the focal point for analyzing and visualizing social media data.

In this paper, we present a web tool for visualizing Twitter data, TweetViz. TweetViz offers several different kinds of visualizations that can pertain to a Twitter user or any keyword or hashtag entered through the interface. TweetViz also includes a so called Streamgraph visualization that represents topic distribution in a set of tweets. The topic distributions are created using LDA (Latent Dirichlet Allocation).

1 INTRODUCTION

Increasing popularity of social media has led to micro-blogging services becoming one of the most popular methods of information consumption. Twitter is a social networking and micro-blogging service that enables users to send and read short messages. These messages, also known as tweets, are maximum 140 characters long. As of this year, Twitter has over 271 million monthly active users and over 500 million tweets sent per day. As a result of the massive amount of data generated on a daily basis, Twitter has become the main focus of many researchers involved in data mining.

The Twitter community uses the service as a way of sharing personal thoughts and ideas, posting news and discussing popular topics. It is also used in marketing purposes by companies, institutions, politicians etc.

Some of the research goals so far have been to extract knowledge about user interests and behavior, detect trends amongst group of users, analyze information dissemination in the network etc.

Most of the work focuses on analyzing words, word pairs and hashtags, with little attempts made to leverage some more advance Natural Language Processing (NLP) techniques such as LDA (Latent Dirichlet Allocation) or LSA (Latent Semantic Analysis).[1]

In this short paper, we present our web tool TweetViz for Twitter data analysis and visualizations. TweetViz incorporates several user-orientated and hashtag or search term visualizations. In addition, we explore an approach where tweets are presented as a mixture of topic distributions over some time interval using LDA.

2 RELATED WORK

There has been significant work done in the field of visualizing and analyzing Twitter activity. Many scientific papers and web tools explore different approaches on visualizing data generated from Twitter. Approaches range from visualizing temporal and spatial data to representing network data. Great portion of the conducted research studies trending topics and general Twitter activity about some subject.

When it comes to visualizing tweets from a single user, the tool proposed in [2] offers visualizations that can aid to understanding the user behavior. As in our approach they explore the frequency of the user's activity in separate days and times of the day. They provide a timeline visualization that presents tweets on a graphic, where the x-axis represents days and the y-axis represents time. Users can also classify tweets by subject, clustering tweets that contain a set of user-defined tags in the category that holds these tags. Their tool DeepTwitter also offers a tag cloud visualization, but it only presents tags specified by the user as opposed to our approach which visualizes all frequent words the user tweeted.

Some of the proposed tools analyze and visualize topic distribution in a collection of tweets. In [1], they attempt to achieve topic alignment between sets of tweets over time. As this is still an open issue in NLP, they aim at solving this problem by using their visualization tool TopicFlow. TopicFlow is an extension of NodeXL, a network visualization tool that offers tweets retrieval. This approach analyses topics at discrete time slices separately. The LDA algorithm is used to provide scaffolding for temporal analysis of Twitter trends. In this approach, similarity between topics is calculated using cosine similarity metric in order to achieve topic alignment. They too provide information for the topics, specifically the words that the topic is consisted of and their statistical importance for the respective topic.

The Streamgraph visualization technique used in TweetViz is also explored in [3] and [4]. ViralViz [3] is another tool that uses LDA and extends NodeXL. This approach differs from the one mentioned before [1] in that way that it takes a more network orientated aspect of topic evolution analysis. ViralViz uses GraphML files generated by NodeXL and then presents topic distribution as a Streamgraph. In addition to

LDA, they provide an approach that extracts keywords based on their statistical significance. In [4], the Streamgraph is not build on data from Twitter and they don't use LDA, but the same principles for creating the visualization apply. The utilization of the Streamgraph and LDA in the presented related work confirms our motivation to use this technique to visualize topic changes in a set of tweets in our tool as well.

3 TWEETVIZ

In this paper we present our Twitter analysis and visualization tool TweetViz. TweetViz offers several different interactive visualizations that can provide insight into user interests and activity as well as information about certain keywords and hashtags. TweetViz visualizations can be divided into two separate modules. The first is user-centric and focuses on analyzing user behavior from different aspects. The second module, on the other hand, is more search term orientated, where a user can explore Twitter activity surrounding a specific hashtag or keyword. Moreover, TweetViz incorporates the LDA algorithm for visual representation of topic distribution, from tweets, both from a specified user or tweets containing a search term. Figure 1 depicts the architecture of the web tool. TweetViz is consisted of separate modules for collecting tweets, preprocessing the content of the tweets, and transforming the data to an acceptable format for the visualization modules in the user interface.

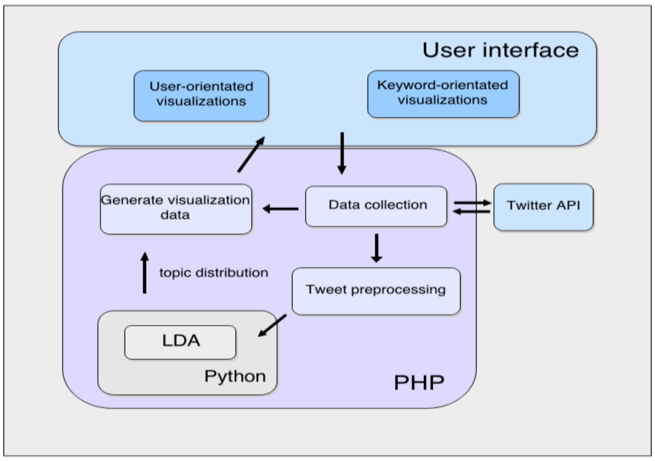

User interface
User-orientated visualizations
Keyword-orientated visualizations
Generate visualization data
Data collection
Twitter API
topic distribution
Tweet preprocessing
LDA
Python
PHP

Figure 1 System architecture

The frontend of TweetViz is built using standard HTML, CSS and JavaScript. The client sends asynchronous request to the PHP based backend which returns the data needed to create the visualizations. In TweetViz we use few third party libraries. Google Charts and d3 (Data Driven Documents) are used to build the different types of visualizations. For generating topic distributions with LDA we used the Python gensim[1] framework.

3.1 Data collection

Twitter enables third party applications and developers to get access to the enormous amount of data generated by users every day. This is done using the Twitter REST API which offers a lot of different endpoints for retrieving this data. Of our interest when building the web tool was the endpoint that allows us to retrieve tweets from a single user. We also leverage the search capabilities offered by the Twitter Search API. This is used to get tweets which contain a given search term, both keyword and hashtag.

Although it offers extensive functionality, the Twitter API has certain limitations. First of all is the rate limit window which limits the number of requests that can be sent. Another deficiency is that the search service provided by Twitter does not index all tweets and as a result, not all tweets are available for retrieval from Twitter Search. Nevertheless, it can provide sufficient number of tweets for the purposes of our web tool.

3.2 User-orientated visualizations

As mentioned before, TweetViz is a web tool that offers different types of visualizations, many of which are focused around a specific user. In order to understand what the user is interested in tweeting about and to provide insight into his behavior on Twitter, TweetViz explores different approaches to creating interactive visualizations.

First of all, TweetViz plots a chart depicting the number of tweets the user posted on a daily basis. Although it is a simple chart, it still can be used to analyze change in user activity and possibly, in combination with other visualizations, why those changes occur.

Twitter enables users to manually label the topic of the tweets they publish with keywords, or also known as hashtags. Many research papers, both in visualizing and analyzing Twitter data revolve around hashtags, proving their usefulness when extracting knowledge from Twitter data.

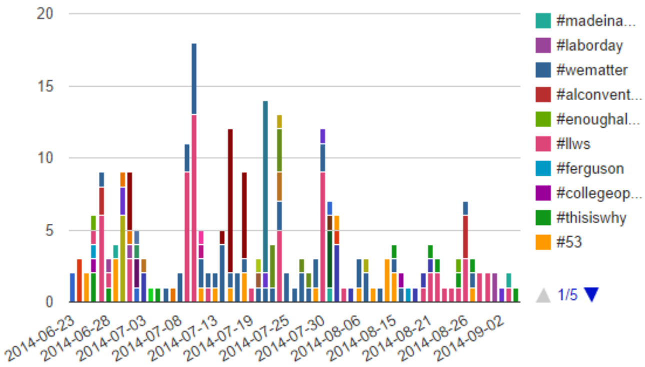


Figure 2: User-hashtag distribution

In Figure 2 is presented a stacked column chart that displays the different hashtags the user tweeted about over some time interval. This provides a nice visual way of seeing what the user is interested in and even detect what sort of topics he tends to combine.

Another interesting approach is visualizing user activity in different parts of the day and even analyzing if this affects the topics he tweets about. This approach to analyzing user activity is certainly not well explored, and this chart can bring

[1] http://radimrehurek.com/gensim/

some information as to its usefulness. Although overwhelming at first, the interactive characteristic of the chart enables users to easily get around this visualization.

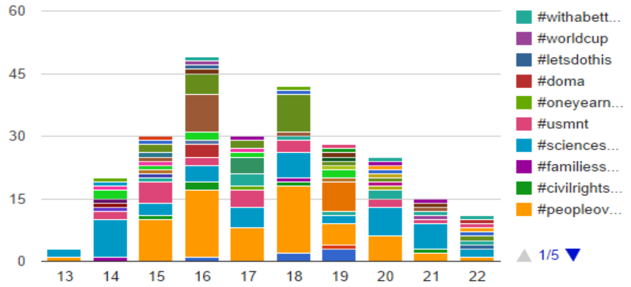


Figure 3: User-hashtag distribution in different time of day

There are some other types of visualizations where one can see the popularity of the user by observing the number of retweets and favorites his tweets get. This is a simple visualization that can give information about the user's influence in the Twitter community.

Generating these visualizations, as well as those similar to these in the keyword-orientated module does not require any special processing of the data.

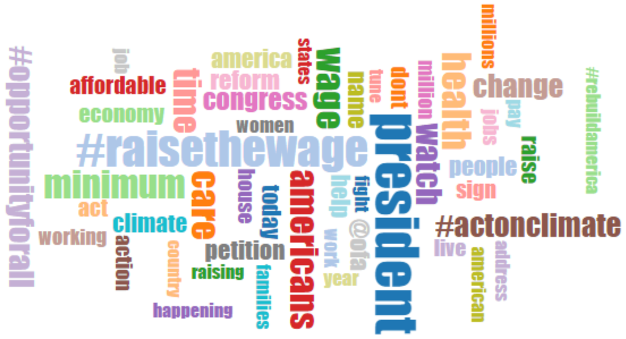


Figure 4: Word cloud

One common way of visualizing frequent keywords in any type of text analysis is creating a so called *word cloud* or *tag cloud*. Before proceeding with generating this *word cloud* (Figure 4), some preprocessing steps need to be made. Almost mandatory, when processing natural language, we need to remove stopwords from the text. Because of the specific domain, the list of stopwords needs to be extended with some specific Twitter words and abbreviations such as "RT", "retweet", "cc" etc. The rest is a simple weighting process, where more common words get larger dimensions in the *word cloud* as opposed to less frequent ones. This is a nice way of observing what a user tweets about that is not concentrated to hashtags only.

3.3 Keyword-orientated visualizations

TweetViz offers users to visualize Twitter activity surrounding a given term. They can search for a specific hashtag or any given keyword. Keyword-orientated visualizations in TweetViz are somewhat similar to the user-orientated. Users can view a chart showing the number of tweets sent containing the search term per day. This is a simple way of detecting spikes in Twitter activity about that term. There is also a chart that shows popularity of a hashtag in different times of the day, again useful for discovering patterns in activity around a search term. The *word cloud* visualization is also available when a user enters a keyword or a hashtag, contributing to a better understanding of the context surrounding the search term.

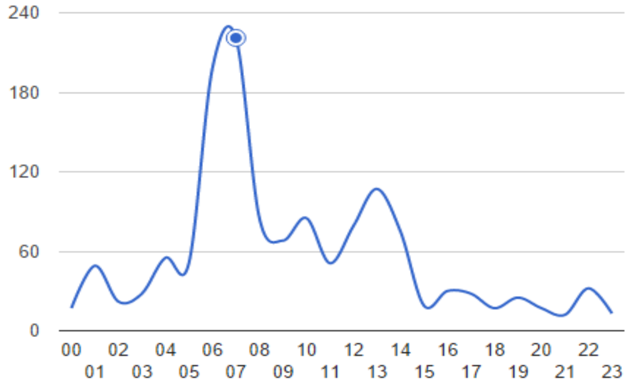


Figure 5 Temporal distribution of a keyword or hashtag

3.4 Visualizing topic distribution

Most of the approaches to analyzing data generated from Twitter revolve around simple techniques such as word count, hashtag count etc. In this paper we leverage a more advance NLP algorithm, LDA. Latent Dirichlet Allocation [5] is a three layer hierarchical Bayesian model in which each text document is modeled as a finite mixture of a set of topics. Every topic is modeled as an infinite mixture over an underlying set of topic probabilities. Simply put, a tweet is represented as a set of topics accompanied with appropriate probabilities, and each topic is made up of words with respective probability distributions. For example, a topic can be represented by a set of words ["mobile", "wear", "watch"]. When generating the models, again we preprocess the textual data, by removing stopwords and additionally, stemming the words.

This interactive visual representation of topic distribution can provide insight into how user interests change over time. An appropriate way of visualizing topic distribution in a time interval is by utilizing a Streamgraph (Figure 6). The Streamgraph visualization technique was proposed by [6] as a more aesthetic alternative to stacked graphs. A Streamgraph is consisted of a finite number of layers, each layer presenting a time series. There a lot of different aspects to be considered when creating Streamgraphs, such as algorithms for generating the graph, coloring and ordering of the layers, all of which are detailed in [6]. For the purposes of this paper, we use an implementation of a Streamgraph in d3[2], which implements the techniques suggested in [6].

In our case, each layer represents a topic, and we track user interest in the topic along the time interval. We tried with a few different color schemes, keeping in mind to use a broader

[2] http://bl.ocks.org/WillTurman/4631136

color range to better distinguish different layers. We use the *silhouette* algorithm for generating the Streamgraph.

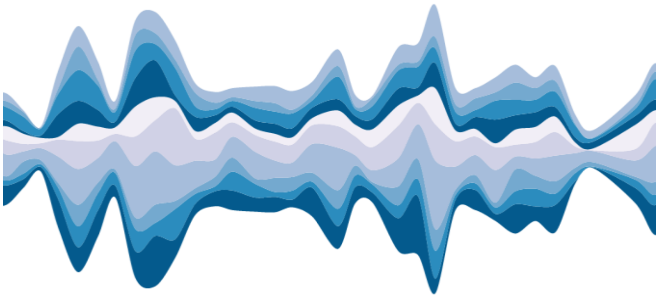
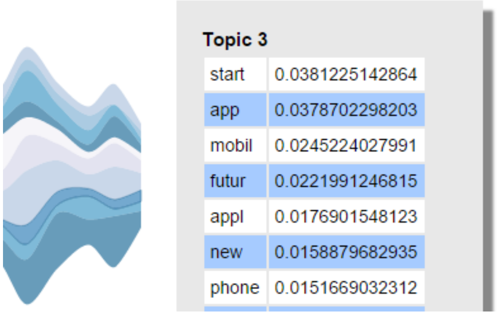


Figure 6: Streamgraph displaying topic distribution

The LDA model was trained on a corpus of over 1.4 million tweets with 3 passes so as to get a better representation of the topic distributions. One limitation of the LDA algorithm is the fact that the number of topics has to be predefined. We choose this parameter to be 20, though we only show 10 topics at a time in the visualization. We decided to use a smaller number of topics in order not to overwhelm the users in the Streamgraph visualization. [1]

The Twitter data that is presented on the Streamgraph is separated into time slices. Each time slice is consisted of a set of tweets. As a result, time slices containing more tweets will have larger y-axis values. A layer's height in a certain time interval is dependent on the presence of the related topic in the set of tweets. As was done in [3], we tend to bring topics with greater differences in distribution to the top and bottom of the Streamgraph as oppose to those with lower differences that end up in the middle. This adds to a clearer way of presenting the layers and differentiating between them.



Topic 3

start	0.0381225142864
app	0.0378702298203
mobil	0.0245224027991
futur	0.0221991246815
appl	0.0176901548123
new	0.0158879682935
phone	0.0151669032312

Figure 7 Detail information for a specific layer

This visualization offers interactivity by allowing users to hover over a layer and get some additional information about it. Users are presented with the words that the topic is consisted of and their respective probabilities as shown in Figure 7. The words appear in a stemmed form, but still, they are informative and can be used to understand what the layer is representing.

3.5 User interface

When designing the user interface, we strived for simplicity. Users are not overwhelmed with many controls and options. There is only one input control in which the user can enter a valid Twitter username, keyword or a hashtag. Then, he is presented with all of the available visualizations, which are interactive in order to improve user experience. The users can hover over certain parts of the visualizations to get additional information. Also, when clicking on those parts, users are presented with the tweets relevant to that part of the visualization. For example, in Figure 3, if a user wants to see the tweets that contain hashtag "#letsdothis" posted at 7pm, he only needs clicking on that particular part of the graph. This is a feature present in most of the other visualizations.

4 CONCLUSION

Analyzing data can be greatly simplified by visualizing it first, which is more appealing to the eye. In this paper, we present our web tool for analyzing and visualizing data generated from the micro-blogging service Twitter. TweetViz offers a set of user-orientated and keyword-orientated visualizations. We show how this web tool can be used to understand user behavior and interests from different aspects as well as general Twitter activity connected to some keyword or hashtag.

We also propose a not so well explored approach of visualizing topic distribution in a set of tweets over some time interval. Topic distributions are generated using the LDA algorithm. Our web tool TweetViz can be of use to anyone interested in exploring Twitter activity and provide for a nice visual way of analyzing data from Twitter.

Acknowledgement

We would like to acknowledge the support of the European Commission through the project MAESTRA – Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References:

[1] S. Malik, A. Smith, P. Papadatos, J. Li. TopicFlow: Visualizing Topic Alignment of Twitter Data over Time. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

[2] G. Rotta, V. de Lemos, F. Lammel, I. Manssour, M. Silveira, A. Pase. Visualization Techniques for the Analysis of Twitter Users' Behavior. ICWSM, 2013.

[3] J. Bradley, N. Fung, I. Julien, M. Malu, M. Mauriello. ViralViz: Visualizing Temporal Content Flow in Social Networks.

[4] S. Havre, B. Hetzler, L. Nowell. ThemeRiver: Visualizing Theme Changes over Time. In Proceedings of the IEEE Symposium on Information Vizualization 2000 (INFOVIS '00). IEEE Computer Society, Washington, DC, USA, 115-.

[5] D. M. Blei, A. Ng, M. I. Jordan. Latent dirichlet allocation. The Journal of Machine Learning Research, 3, 993-1022.

[6] L. Byron, M. Wattenberg. Stacked Graphs-Geometry & Aesthetics. IEEE Trans. Vis. Comput. Graph., 14(6), 1245-1252.

RAZŠIRITEV ISKALNIKA Z ORODJI ZA SEMANTIČNO ISKANJE V SLOVENSKI POLICIJI

Mladen Tomaško[1], *Dunja Mladenić*[2]
[1]Služba generalnega direktorja policije, Ministrstvo za notranje zadeve RS
Litostrojska cesta 54, 1000 Ljubljana, Slovenia
Tel: +386 1 4773419; fax: +386 1 4251038
e-mail: mladen.tomasko@policija.si
[2]Institut Jožef Stefan in Mednarodna Podiplomska šola Jožefa Stefana,
Jamova 39, 1000 Ljubljana, Slovenia
e-mail: dunja.mladenic@ijs.si

POVZETEK

Spreminjanje katere koli komponente v obsežnih informacijskih sistemih z velikim številom uporabnikov, je predvsem zaradi inercije takšnih sistemov in pogosto tudi tehnične zahtevnosti, težavna naloga. Uspešnost rešitve je odvisna od natančne preučitve navad uporabnikov, preproste uporabe, čim bolj neopazne integracije in kvalitetne podpore (usposabljanja, pomoči v realnem času, …) Pri tem je še posebej pomembno, da je rešitev vgrajena v obstoječi sistem tako, da čim manj spreminja dosedanji način dela.Pri tehnični izvedbi lahko naletimo na težave s podatki, ki so neustrezno razvrščeni, vsebujejo veliko napak, ali pa so nepopolni. To predstavlja dodatne izzive pri uporabi orodij semantičnih tehnologij. Po drugi strani pa je takšna rešitev dodana vrednost, katere prednosti se pokažejo na daljši čas. Zahteva pa začetno privajanje uporabnikov na spremembe (v tem času storilnost lahko celo upade), dolgoročno pa računamo na opazen prihranek časa in kakovostnejše poslovanje. Prispevek predlaga pristop za razširitev standardnega iskalnika s pomočjo metod semantičnih tehnologij. Predstavljena rešitev je trenutno pred fazo testiranja v vsakdanjem delovnem procesu.

1 UVOD

Ministrstvo za notranje zadeve republike Slovenije uporablja za svoje pisarniško poslovanje IBM Lotus Notes[5]. Sistem je razbit na več kot 40 samostojnih strežnikov, ki so do pred kratkim omogočali iskanje le po lokalnih vsebinah. Zaradi tega smo razvili iskalnik, ki išče po vseh Lotus Notes zbirkah in omogoča tudi ustrezno implementacijo varnostnega pravilnika, ki omogoča za vsakega posameznika definiranje ustreznih dostopov do dokumentov[3]. Dostop do večine dokumentov je namreč omejen, ker so v njih podatki, ki jih lahko obdelujejo le osebe, ki za to imajo zakonsko podlago. Že pri integraciji novega iskalnika se je pokazala potreba po dopolnitvi obstoječega iskalnika z orodji, ki bi omogočala učinkovitejše razvrščanje dokumentov, oz. bi omogočila prikaz tudi tistih dokumentov, ki so neposredni povezavi z uporabnikovim iskanjem, čeprav jih uporabnik ni direktno iskal. Pomembno pri vsem tem je, da nam nova rešitev omogoča nadaljnje izboljšave sistema zato smo v predlagani rešitvi za dopolnitev funkcij iskalnika uporabili odprtokodni program Lucene/Sorl [6][7], za naprednejše delo s podatki pa prav tako odprtokodno rešitev iz področja semantičnih tehnologij OntoGen[1]. V prispevku so na kratko opisani koraki pri razvoju te rešitve.

2 PREDSTAVITEV PODATKOV

Iskalnike uporabljamo za preiskovanje različnih zvrsti podatkov, ki so lahko zapisani v podatkovnih bazah, pa tudi v manj urejenih zbirkah podatkov ali na svetovnem spletu. V prispevku smo za primarni vir podatkov vzeli zbirke IBM Lotus Notes in to tisti del, ki sestavlja sistem SPIS (SRC.SI pisarniški informacijski sistem) v katerem so shranjeni vsi dokumenti, katerih pomembnost je takšna, da morajo biti zabeleženi in shranjeni v informacijskem sistemu MNZ in Policije.

2.1 Lastnosti podatkov

Sistem IBM Lotus Notes je razpršen na 42 strežnikih. V njem so združene podatkovne zbirke za pisarniško poslovanje, podatkovne zbirke elektronske pošte in veliko namenskih podatkovnih zbirk. V naši rešitvi smo se omejili le na podatkovne zbirke, ki vsebujejo dokumente, ki nastajajo pri pisarniškem poslovanju, se pravi izhodne, vhodne in lastne dokumente. Količina podatkov v sistemu pa se že meri v stotinah terabajtov. Podatkovne zbirke vsebujejo več kot pet milijonov dokumentov. Podatki v njih so shranjeni v strukturirani in nestrukturirani obliki. Strukturirani podatki vsebujejo vse osnovne podatke o avtorju dokumenta, naslovniku, področju, ki ga dokument zajema, datumu in času nastanka, podatke, ki jih rabi varnostni pravilnik. Pri takšni količini podatkov se pojavlja tudi veliko napačno razvrščenih dokumentov, ki otežujejo iskanje s klasičnim iskalnikom. Te napake so shranjene v strukturiranih poljih,. Po navadi je napačno vneseno področje, včasih tudi naslovnik, posebej ko je dokument naslovljen na službo ki je sestavljena iz več sektorjev in oddelkov. V nestrukturiranih podatkih – nas zanima predvsem polje v katerem se nahaja celotno besedilo dokumenta – je teh napak praviloma manj, čeprav je besedilo zapisano skupaj z glavo, naslovnikom, in ostalimi podatki, ki sestavljajo dokument pripravljen za tiskanje in pošiljanje po navadni pošti. Prav ta podvojenost podatkov, ki so drugače že zapisani v strukturiranih poljih, vsebuje pa jih tudi besedilo, otežijo pravilno grajenje ontologije in jih je bilo treba na določen način izločiti. Del smo jih izločili že pri izvozu podatkov iz podatkovnih zbirk. Tu smo izločili strukturirana polja, pustili smo le tista, ki smo jih pozneje rabili zato, da smo lahko preverjali kako uspešno je OntoGen razvrščal dokumente.

Izločanje podatkov iz dokumentov, ki so bili v nestrukturiranih poljih je bilo bolj zahtevno. V pomoč nam je bilo, da je v večini dokumentov besedilo urejeno na enak način. Tako smo lahko določili točki med katerima je območje besedila dokumenta in smo to besedilo shranili v datoteke. Del podatkov smo pa očistili tako, da smo pripravili seznam besed (predvsem kratic), ki niso relevantne za naš problem (n.pr. MNZ, UKP, UUP, …) in jih OntoGen ni upošteval pri postopku razvrščanja.

2.2 Varnostni pravilnik

V Policiji je varnost podatkov izrednega pomena. Zato mora vsaka rešitev upoštevati stroga varnostna merila, ki omogočajo nadzor dostopov do dokumentov in preprečujejo dostop nepooblaščenim uporabnikom. Sistem Lotus Domino vsebuje zmogljiv varnostni model, ki dostope do podatkov preverja na več ravneh [4]. Varnostni model sestoji iz šestih nivojev:

1. **varnost omrežja,**
2. **overovitev uporabnikov,**
3. **dostop do strežnika,**
4. **zaščita zbirke podatkov,**
5. **varnost oblikovnih elementov,**
6. **omejitev dostopa do dokumentov.**

Močan varnostni model je nujen zaradi velike količine osebnih podatkov, ki jih lahko vidijo le pooblaščeni uporabniki. Poleg tega pa zbirke vsebujejo tudi tajne podatke, ki so še posebej varovani z močnim šifriranjem.

3 PREDLAGANA REŠITEV

Na Sliki 1 je grafično podan predlog rešitve. Že prej smo omenili, da je osnovna zahteva bila dodati obstoječemu iskalniku rešitev, ki bo uporabljala semantična orodja in ponudila uporabnikom podobne dokumente oz. jim pomagala pri njihovem razvrščanju. Kot je iz Slike 1 razvidno je rešitev zgrajena tako, da prvi del iskanja opravi osnovni iskalnik v Lotus Notes, ki je zgrajen na podlagi odprtokodne rešitve Lucene/Sorl [6][7]. Komercialne rešitve so se izkazale za prezapletene in prezahtevne za prilagajanje specifičnemu okolju Policije. Zato se je prej omenjena odprtokodna rešitev izkazala kot ekonomsko najbolj upravičena. Enostavna implementacija in učinkovito delovanje Lucene sta glavna razloga, da je Lucene uporabljen kot rešitev za iskanje po zbirkah Lotus Notes. Iskalnik išče po indeksiranih podatkih, indeksiranje pa se ves čas odvija v ozadju. Tukaj je treba omeniti, da je bilo treba v iskalnik vgraditi varnostni pravilnik, ki omogoča nadzorovan dostop do dokumentov.

Drugi del “iskanja” oz. pravilneje rečeno razvrščanja podatkov opravi prav tako odprtokodna rešitev OntoGen [2]. OntoGen je polavtomatski in podatkovno usmerjen urejevalnik ontologij. Osredotoča se na urejanje tematskih ontologij (sklop tem, povezanih z različnimi vrstami odnosov). Sistem združuje metode podatkovnega rudarjenja, z učinkovitim uporabniškim vmesnikom kar pomeni manj porabljenega časa in preprostejše delo. Na ta način zapolnjuje vrzel med zapletenimi orodji za urejanje ontologij in ga lahko uporabljajo strokovnjaki na posameznih področjih, ki nimajo nujno znanja potrebna za gradnjo ontologij. S pomočjo OntoGena smo pripravili ontologijo, ki nam pomaga pri razvrščanju dokumentov, predlaga podobne dokumente ko pripravljamo nov dokument, ideja pa je, da bi nam ponujal besede, ki jih lahko uporabimo v iskalniku, da bi dobili boljše zadetke. Na nek način, bi nas učil kako bolje izrabiti iskalnik.

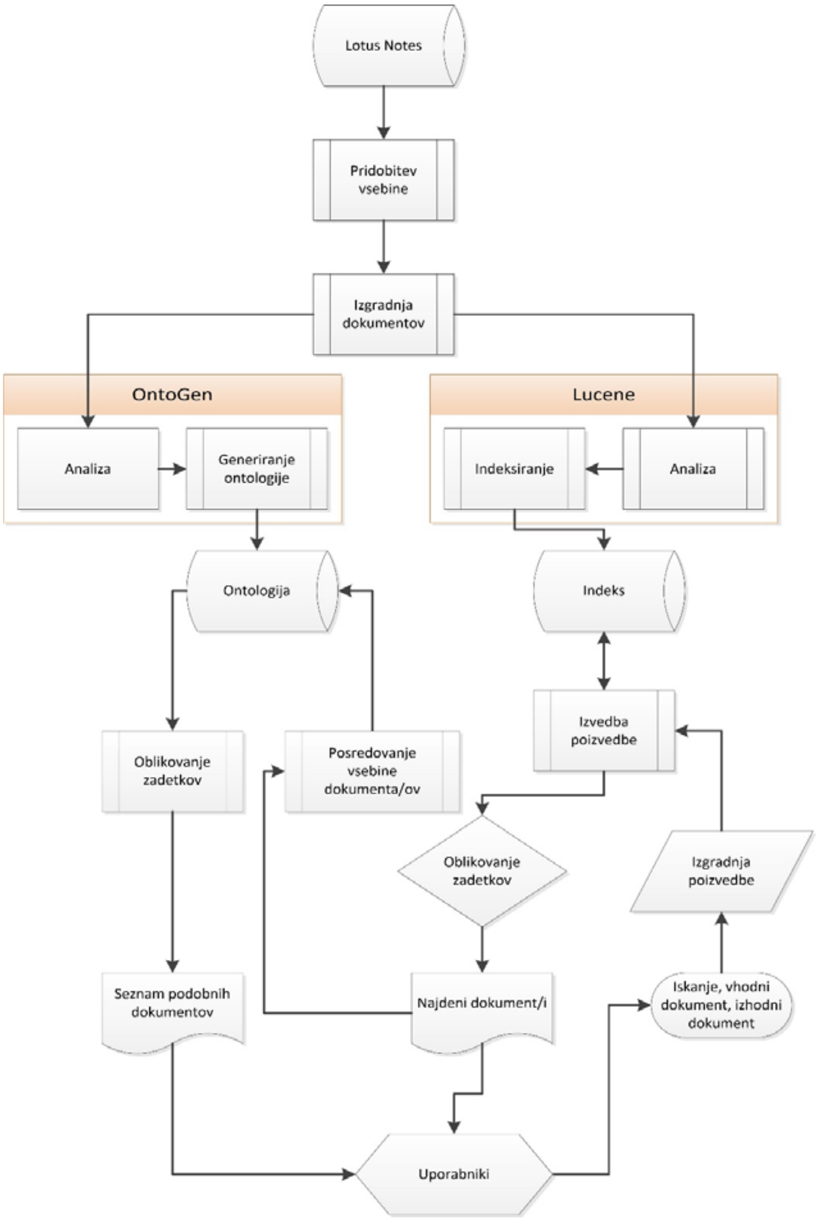
3.1 Priprava podatkov

Zbirka Lotus Notes vsebuje več kot pet milijonov dokumentov. Za pripravo ontologije smo izmed njih naključno izbrali 3000 dokumentov, izmed katerih smo 300 uporabili za gradnjo ontologije, na preostanku pa smo preverjali kako uspešen je naš model.

V dokumentih se pojavlja veliko število kratic, ki večinoma predstavljajo organizacijske enote ministrstva ali so okrajšave nekaterih mednarodnih organizacij. Te kratice so velikokrat povzročile napačno razvrščanje dokumentov in je nekatere izmed njih bilo treba izločiti. Začetek gradnje ontologije je bil zaradi tega poln poskusov priprave ustrezne oblike dokumentov in iskanja besed, ki jih je bilo treba uvrstiti na stop seznam (izločiti iz obdelave). Pokazalo se je, da ključ k uspešni gradnji ontologije predstavljajo dobro izbrani in dobro pripravljeni podatki.

3.2 Gradnja ontologije

Gradnja ontologije je potekala v več korakih. Na začetku se je pokazalo, da podatki, ki smo jih ponudili OntoGenu niso bili dovolj dobro izbrani (kljub temu, da smo jih izbirali naključno niso vsebovali dokumentov iz vseh področij, ki jih zajema delo Policije), zato smo morali postopek izbire dokumentov nekajkrat ponoviti. Tudi kakovost priprave podatkov na začetku ni bila ustrezna in smo morali izločati neustrezne besede in izraze. Težava je bila tudi s pretvorbo različnih kodnih zapisov. Vsi dokumenti niso bili v enakih kodnih zapisih in jih je bilo treba poenotiti. Po večkratnih



Slika 1. Prikaz arhitekture predlagane rešitve, pri čem je iskanje razdeljeno na dva dela: Lucene opravi osnovno iskanje po bazi, OntoGen omogoča semantično iskanje za namen pridobivanja podobnih dokumentov.

poskusih, so se pričeli kazati zametki ontologije, ki jo je bilo treba še ročno optimizirati. Težavo predstavlja velika razvejanost organizacije MNZ in prepletenost delovnih področij. Npr. kaznivo dejanje lahko obravnava navaden policist na policijski postaji, policist kriminalist, kriminalist iz policijske uprave ali kriminalisti generalne policijske uprave. Razlika je predvsem v tem ali kaznivo dejanje zajema lokalno, regionalno ali državno raven. Ontologija precej natančno sledi tem pravilom in omogoča uspešno razvrščanje dokumentov glede na vsebino.

Naša rešitev v osnovi sledi organizacijski strukturi MNZ in delovnim procesom na vseh nivojih s tem, da so pri gradnji ontologije bile upoštevane določene specifičnosti posameznih področij (predvsem v primerih, ko gre za prepletanje delovnih procesov na različnih nivojih). Pravila, ki so vgrajena v ontologijo omogočajo ustrezno identifikacijo dokumentov in v prihodnje bodo upoštevali tudi umeščenost uporabnika, ki je sprožil iskanje, v organizacijsko strukturo.

3.3 Oblikovanje zadetkov poizvedbe

Naša rešitev in iskalnik delujeta neodvisno drug od drugega. Iskalnik Lucene razvršča zadetke po tem kako ustrezajo iskanim besedam in kako pogosto se te besede pojavijo v najdenih dokumentih. Pri iskanju ne upošteva druge ključne besede in izraze, ki se pojavljajo v najdenih dokumentih. Naša rešitev ponudi podobne dokumente, ki niso neposredno vezani na iskane izraze, temveč se na posebnem seznamu pokažejo še dokumenti, ki se tematsko ujemajo s tistimi, ki jih je našel iskalnik. Poleg tega ima uporabnik možnost prikaza podobnih dokumentov v času, ko sam sestavlja nek dokument, oziroma mu program sam predlaga umestitev novega dokumenta v klasifikacijsko strukturo.

Pravila zapisana v ontologiji nam pomagajo poiskati podobne dokumente, oziroma predlagati razvrstitev obstoječih. Uspešnost razvrščanja je odvisna od kompleksnosti delovnega področja iz katerega je dokument. Na bolj kompleksnih je stopnja učinkovitosti nekoliko nižja kar pomeni, da bo treba ontologijo še nekoliko dodelati. Tega se bomo lotili po zaključenem testiranju, ko bomo imeli na razpolago več podatkov.

4 TESTIRANJE UČINKOVITOSTI MODELA

Merjenje učinkovitosti smo zasnovali na dveh nivojih: vprašalnik (bolj subjektivno) in samodejno spremljanje uporabnikov (bolj objektivno). Struktura testiranih uporabnikov odraža strukturo celotne organizacije. Vzorec obsega 30 do50 uporabnikov, ker bi pri manjšem številu težko dosegli pravilno zastopanost vseh razredov.

4.1 Anketiranje uporabnikov

Vprašalnik je sestavljen iz dveh sklopov. V vsakem sklopu so še kontrolna vprašanja, ki omogočajo, da nadziramo kakovost odgovorov. Testno obdobje bo trajalo od 14 – 21 dni, tako, da bodo v tem obdobju zajeta najpomembnejša opravila, ki se pojavljajo na mesečni ravni

Vprašalnik je, vsaj deloma, subjektivna ocena uporabnika kako je doživljal delo z novimi funkcionalnostmi pri svojem vsakdanjem delu, zato je treba rezultate, ki preveč odstopajo od povprečja na ustrezen način dodatno analizirati in ugotoviti zakaj je do teh odstopanj prišlo.

Analiza rezultatov je izredno pomembna ker nam omogoča ugotoviti ustreznost rešitve, oceniti prihranke zaradi izboljšanih delovnih procesov in v končni fazi oceniti ali je rešitev zrela za implementacijo v celoten sistem. Če se izkaže, da so na določenem področju pomanjkljivosti bo treba pristopiti k dodatnim izboljšavam rešitve in ponoviti postopek testiranja.

4.2 Samodejno spremljanje obnašanja uporabnikov

Za potrebe spremljanja obnašanja uporabnikov bomo pripravili dodatno rešitev, ki bo zapisovala klike na povezave, število iskanj, število obdelanih dokumentov, čas potreben za posamezen dokument, morebitne neustrezne klike in podobno.

Ti podatki nam bodo v pomoč pri končnem ocenjevanju rešitve in smiselnosti njene implementacije v sedanji obliki. Dobljene rezultate bo treba pred končno analizo ustrezno ovrednotiti in jih pripraviti za medsebojno primerjavo. Vsekakor bo treba upoštevati tudi zahtevnost delovnega mesta, zahtevnost pripravljenih dokumentov in ne samo njihovo število ali čas potreben za obdelavo.

5 ZAKLJUČEK

Predlagali smo razširitev obstoječega iskalnika po internih bazah Slovenske policije. Pri tem smo uporabili semantične tehnologije, s pomočjo katerih smo vnaprej zgradili ontologijo vsebin dokumentov, ki so shranjeni v obstoječih bazah. Priprava ontologije za organizacijo s tako zapleteno organizacijsko strukturo zahteva veliko ročnega dela in dobro poznavanje tako organizacijske strukture kot delovnih procesov. Posebno težavo predstavlja prepletanje delovnih procesov med različnimi nivoji, tako da je včasih težko izluščiti kateri način dela je najbolj pravilen. Zato ko je ontologija enkrat v grobem zgrajena, sledi zahtevno dopolnjevanje in optimiziranje. Kljub temu, da je ta proces časovno precej potraten se na koncu obrestuje z izboljšanimi delovnimi procesi, povečano učinkovitostjo in prihranki na nivoju celotne organizacije.

Kljub temu, da je predlagana rešitev še v fazi testiranja, brez natančno definiranega varnostnega pravilnika, se že vidijo razlike v uspešnosti razvrščanja dokumentov. Koliko se bo to pokazalo kot izboljšanje delovnih procesov in s tem povezanih finančnih učinkov bo pokazalo spremljanje rezultatov dela v daljšem časovnem obdobju. Takrat bo narejena tudi analiza, ki bo ovrednotila razmerje med vloženimi sredstvi in morebitnimi prihranki. Po drugi strani pa je vsak poskus uporabe naprednih tehnologij v tako kompleksnih sistemih dobrodošel, ker na praktičnih primerih pokaže na katerih vse področjih te tehnologije lahko uporabimo. S tem lahko pri menedžmentu dobimo možnost nadaljnjega razvoja informacijskega sistema, ki bo primernejši tako z vidika zaposlenih, kot z vidika uporabnikov uslug policije.

Viri

[1] B. Fortuna, M. Grobelnik, D. Mladenic: Semi-automatic Data-driven Ontology Construction System. In: *Proc. of the 9th International multi-conference Information Society IS-2006*, Ljubljana, Slovenia (2006).

[2] B. Fortuna, M. Grobelnik, D. Mladenic: Semi-automatic Ontology Editor, In*: Proc. of the 12th International Conference on Human-Computer Interaction,* Beijing , China (2007).

[3] P. Skale. Načrtovanje in razvoj iskalnika za potrebe policije. *Msc Thesis*. Fakulteta za računalništvo in informatiko. Univerza v Ljubljani. 2012.

[4] IBM, *Inside Lotus: The Architecture of Notes and the Domino Server*, IBM Press, 2000.

[5] Zgodovina razvoja Lotus Notes in Domino. http://www.ibm.com/developerworks/lotus/library/ls-NDHistory/. (dostop avgust 2014)

[6] Lucene. http://lucene.apache.org. (dostop julij 2014)

[7] Apache Solr. http://lucene.apache.org/solr. (dostop avgust 2014)

[8] Zgodovina razvoja Lotus Notes in Domino. http://www.ibm.com/developerworks/lotus/library/ls-NDHistory/ (dostop julij 2014)

CONSISTENCY AND COMPLETENESS OF MULTIWORD EXPRESSIONS DURING TRANSLATION

Katerina Zdravkova[1]*, Aleksandar Petrovski*[2]*, Tomaž Erjavec*[3]
[1]Faculty of Computer Science and Engineering, University of Skopje, Macedonia
[2]Faculty of Informatics, Slavic University, Sveti Nikole, Macedonia
[3]Department of Intelligent Systems, Jožef Stefan Institute, Slovenia
e-mails: katerina.zdravkova@finki.ukim.mk; a.petrovski.sise@gmail.com; tomaz.erjavec@ijs.si

ABSTRACT

One of the crucial challenges of statistical machine translation is the lexical consistency of manually translated words and multiword expressions (MWEs) with multiple occurrences in the source language. In this paper, we present the degree of translation inconsistency and we introduce the index of translation completeness of fixed MWEs. The research was based on the recently developed system that intends to extract the entire candidate MWEs from Orwell's 1984 parallel corpora and to predict their translations between English, Macedonian, and Slovene.

1 INTRODUCTION

Since the early 1990s, traditional rule-based machine translation (MT) has been enhanced and replaced by the statistical MT [1]. The efficiency of, at that time rather revolutionary approach has been proved, and many tools and parallel corpora (many of them collected in http://www.statmt.org/) have been developed to enable an effective translation of written texts, no matter the languages involved it the process.

Statistical MT of MWEs can be successfully performed using non-hierarchical phrase-based SMT, which exploits only the continuous phrases [2]. In an absence of relevant parallel MWE corpora between English, Macedonian and Slovene, we decided to create an own system, which extracts all the candidate MWEs from sentence aligned corpora and then predicts their translations. The proposed system consisted of four complementary phases:

- *extraction* of all candidate continuous sequences of words that appear in each language at least twice,
- *syntactical filtering* of obtained candidates, using a predefined set of eligible syntactic expressions,
- prediction of *potential translation equivalents* from corresponding pairs of aligned sentences where MWEs appear, and
- cross-evaluation of candidate translations, interchanging the source and the target language.

In this paper we evaluate the efficiency of the system and try to determine the key causes of wrong expressions and inaccurate translation. The structure of the paper is the following: The analysis and research of the document-level consistency is presented in the second section. The typical examples of translation inconsistency and incompleteness are illustrated in the third section. The consistency index and the degree of translation completeness are introduced in the fourth section. Following the same section, the lexical consistency and the completeness of English to Macedonian and English to Slovene translation of Orwell's 1984 are calculated. The paper concludes with the ideas that might improve the quality of document-based statistical MT.

2 ANALYSIS OF PREVIOUS RESEARCH

Multiword expressions, which are defined as combinations or strings of words without a unique syntactic or semantic property, are among the crucial obstacles of machine translation [4]. Many lexicons include significant amounts of MWEs, including lists of phrasal verbs (*think of / мисли на / misliti na*), nominal multiwords (*dark-haired girl / темнокоса девојка / temnolaso dekle*), pronouns (*almost nothing / скоро ништо / skoraj ničesar*), adverbs (*during his childhood / за време на неговото детство / med njegovim otroštvom*) and other phrases.

Multiword expressions are extremely frequent. Jackendoff estimated that they appeared in the speaker's lexicon with a comparable frequency with the simple words [5]. In addition, they are very heterogeneous [4]. Even when MWEs are restricted to fixed strings, their treatment in MT is one of the most challenging NLP tasks [6]. In order to become useful for further MT research, MWEs should be extracted out of a parallel and aligned corpus. Their automatic identification and acquisition have been exhaustively researched by many authors. Most of the proposed techniques identify MWEs using different statistical measures, for example, the mutual information, permutation frequency, and Pearson's chi-square [4]. Statistical measures can be extended with various additional information concerning the word alignment [4, 8].

The detection of missing lexical entries for MWEs based on error mining methods and maximum entropy model was recommended by Zhang et al [7]. Apart from proposing their approach, they list the ten most frequent and least frequent MWEs using Google search engine. Statistical properties were also efficient during the extraction of non-compositional compounds [4].

Once extracted from parallel aligned corpora, MWEs can undergo through the translation process. The typical recent SMT tools, such as Moses (http://www.statmt.org/moses/) are phrase-based models [8]. Moses used the Bayes rule to initially calculate the probability for translating a foreign sentence into English. The same approach was very soon implemented for many other languages, including Macedonian [9]. Numerous experiments have shown that Moses performs much better that word-based models, and more significantly, it appeared that the use of syntax doesn't lead to better performance.

Caseli and her collaborators combined phrase-based and word-based model, creating the first alignment based MWE extraction method [4]. For each language, they created an output of the aligner or the tagger along with the target words that were aligned to them. Inspired by this project, we suggest a new, slightly less rigorous approach [10]. Its intention is to identify all the MWEs appearing in the multilingual sentence aligned Multext-East corpus [11]. The effectiveness of the system will be illustrated with the examples of aligned English to Macedonian and Slovene translation of 968 multiword expressions existing in the English original of Orwell's novel 1984.

The extraction process in these two projects passed through a pre-processing phase, which produced parallel, sentence aligned and PoS tagged multilingual corpora [4, 12]. Furthermore, some MWEs were word-aligned to be associated with semantics [4]. False positive examples were syntactically filtered using patterns or syntactic constraints. Many inadequate candidates were further eliminated using the cross-evaluation mentioned in the introduction of this paper [11]. In our system, we eliminated the syntactically ineligible MWEs using different patterns [10]. In many occasions, the filtering process using the cross-evaluation offered a very good result.

The implementation of mutual cross-evaluation among English, Macedonian and Slovene revealed that in many occasions:

- manual translator of Orwell's 1984 was either inconsistent or had "an artistic freedom",
- inflectional paradigms, which are richer in the Slavic languages can influence the translation,
- the context in which the same target MWE appeared can also influence its translation.

As a result, many MWEs were translated with an MWE that is shorter than the real target, up to the extreme not to be translated at all. Partial incompleteness or the entire absence of the target MWEs were the main drawback of our system.

3 EXAMPLES OF INACCURATE TRANSLATIONS

The English version of Orwell's 1984, which serves as a base for the parallel corpus contains 6701 sentences and 104302 words. Macedonian translation consisted of 6712 sentences with 98846 words. The amount of continuous word sequences with multiple occurrences in both languages exceeded 15000. The English candidate MWEs were matched with the translated Macedonian MWEs. As a result, the extraction phase ended up with 968 English MWEs, the majority of which produced a target Macedonian MWE [10].

Due to the abundance of Slovene nominal inflections, the amount of omitted Slovene translations was higher. Here are some typical examples that explain the deficiency of the statistical machine translation without a morphological extension we created. The cross-evaluation, which reverted the source and the target language revealed that in some occasions two different English MWEs were translated with the same MWE. This can be treated as a revert inconsistency. Table 1. presents several cases of inconsistencies across three languages. The omitted parts of the most acceptable translations in the corresponding language are presented in the parentheses. The MWEs in bold are the starting points for the translation.

Language	English	Macedonian	Slovene
Multiword expression	**the seconds were ticking by**	секундите минуваа (отчукувајќи)	sekunde so tiktakale mimo
Mac 1: секундите минуваа отчукувајќи ...			
Mac 2: секундите минуваа бескрајно долги ...			
Multiword expression	**almost on a level with**	речиси на исто (со)	no translation
Mac 1: ... речиси на исто ниво со ...			
Mac 2: ... речиси на исто рамниште со ...			
Slov 1: ... skoraj na ravni z ...			
Slov 2: ... skoraj v isti višini z ...			
Multiword expression	the first thing	(прва работа што мора) да ја сфатиш е	**prva stvar ki jo moraš ...**
Eng 1: the first thing for you to understand ...			
Eng 2: the first thing you must realize ...			

Table 1: *Incompleteness due to lexical inconsistency*

Slavic incomplete or missing translations of English due to inflections are presented in the Table 2. The parentheses in the Macedonian example are given to describe the MWE.

Language	English	Macedonian	Slovene
Multiword expression	**smell of her hair**	(мирисот) на нејзината коса	vonj njenih las
Mac 1: ... (пријатниот) мирис на нејзината коса			
Mac 2: мирисот на нејзината коса			
Multiword expression	**ideologically neutral**	идеолошки неутрален	ideološko nevtralen/na
Slov 1: ... (nobena beseda ... ni bila) ideološko nevtralna			
Slov 2: ... (predmet govora ni bil) ideološko nevtralen			
Multiword expression	**against us**	против нас	po robu (proti nam)
Slov 1: ... (nikdar ne) postavi po robu			
Slov 2: ... (in se nam) postavila po robu			

Table 2: *Incompleteness due to inflections*

In many occasions, the context was the crucial reason of the pruned or missing translations. It is worth mentioning that there were several examples with shorter multiword expression even in the English original, as presented in the Table 3. They are a result of the reverse order of source and target extraction due to cross-evaluation.

In order to distinguish the importance of the context, the original source contexts in parallel with the target ones is also specified. Although the absence of Slovene translation in the last example is mainly due to the inflections, it is presented here, because the different grammatical cases (genitive in *prepisovalne ekipe* and locative in *prepisovalni ekipi*) themselves also arise from the context.

Language	English	Macedonian	Slovene
Multiword expression	**for more than half an hour**	(за) повеќе од половина час	za več kot pol ure
Eng 1: ... and never for more than half an hour at a time Mac 1: ... и никогаш повеќе од половина час			
Eng 2: ... to turn off the telescreen for more than half an hour Mac 2: ... да го држат исклучен телекранот повеќе од половина час			
Multiword expression	**definitive edition**	дефинитивното издание	no translation (dokončna izdaja)
Eng 1: ... (the eleventh edition is the) definitive edition ... Slov 1: ... (enajsta izdaja je) dokončna			
Eng 2: ... (we were producing a) definitive edition ... Slov 2: ... (pripravljali smo) končno izdajo			
Multiword expression	(in) the rewrite squad	**во одделот за препишување**	no translation (prepisovalna ekipa)
Mac 1: ... до завршните работи во одделот за препишување Eng 1: ... (down to final touching-up by) the rewrite squad Slov 1: ... (pa do končne obdelave) prepisovalne ekipe			
Mac 2: ... никогаш не работев во одделот за препишување Slov 2: ... (nikdar nisem bila v) prepisovalni ekipi Eng 2: (i was never in) the rewrite squad			

Table 3: *Incompleteness due to the context*

In the next section, the inconsistency index, which was proposed by Itagaki et al. [3] will be introduced and calculated for those MWEs that existed in all the three languages. In parallel with the inconsistency index, we also propose the degree of incompleteness, which is the direct consequence of the inconsistent translation.

4 CONSISTENCY AND COMPLETENESS OF MULTIWORD EXPRESSIONS

Human translators usually work with very large translation units. Without a large list of own translated phrases, or an automated translation tool, the possibility to inconsistently generate the translation is high.

In 2007, Itagaki, Aikawa and He decided to devise an index to assess the terminology translation consistently [4]. They discovered that the estimation could be effectively done using the Herfindahl-Hirschman Index (*HHI*), which was previously used to measure the market concentration. The index is calculated as:

$$HHI = \sum_{i=1}^{n} S_i^2$$

where *S* is the ratio of each translation (*i*) to the total number of translations (*n*) within a product. To simplify the definition, whenever one word is translated with *n* different words, each one with a frequency S_i, in such case, the consistency of the translation is the sum of squared frequencies within the document [4, 13].

HHI is applicable to multiword expressions, replacing the single words to lexical units. For example, the multiword expression *the dark-haired girl*, which appears twice in the source language was uniquely translated to Macedonian (*темнокосата девојка*) and Slovene (*temnolaso dekle*), so its consistency is 1. The English MWE *during his childhood* also appeared twice, with two Macedonian translations: *за време на неговото детство* and *во текот на неговото детство*, and a unique Slovene translation *med njegovim otroštvom*. The consistency of the Macedonian translation is 0.5^2 + 0.5^2 = 0.25., while the Slovene consistency is 1. The translation of the phrases that appear in the English original more than once was always perfectly consistent (*yes said Winston* / *да рече Винстон* / *da je rekel Winston*; *how many fingers Winston* / *колку прсти Винстоне* / *koliko prstov Winston*).

By adopting the consistency index of lexems to lexical units, i.e. to multiword expressions, we also propose to calculate their relative consistency as a ratio between *HHI* and the cardinality of the set of all multiword expressions appearing in the target corpus at least twice:

$$RC = \frac{HHI}{|MWE|}$$

In the Macedonian version, 48 out of 968 English MWEs had no translation due to inconsistent translation, or the translation consisted of only one word, which was excluded from the MWE corpus. Further 127 were partially inconsistent, thus the consistency index was 836.75, or relatively 86.44%.

The translation to Slovene had a relative consistency of 80.40%, due to 162 MWEs without a translation, and 91 with partial inconsistency, or a total consistency index of 778.25.

The examples presented in the tables above indicate that the key outcome of human inconsistency used as a source in the statistical machine translation systems is the incompleteness of generated target expressions. To measure the degree of incompleteness of MWE translations, we propose the index of completeness *DG* of a single MWE calculated as:

$$DG = \frac{length(generated\ MWE)}{length(complete\ MWE)}$$

For example, the English expression *almost on a level with* is translated with *речиси на исто* instead of *речиси на исто ниво со*. Its completeness is 0.6. But, whenever the short MWE is not a subset of the complete MWE, such as the translation of *against us* to Slovene, which was *po robu* (see Table 2.), in such case the completeness is 0. This estimation can be done after a manual inspection of the translated MWEs.

We also define a combined completeness *CC* of all *m* MWEs extracted from the source corpus as:

$$CC = \frac{1}{m}\sum_{j=1}^{m} S_j^2 DG_j^2$$

The combined completeness of the MWE *almost on a level with* is 0.25 * 0.36 = 0.09. The combined translation of a consistent translations is 1.

Due to the higher consistency, Macedonian translations had a higher combined completeness of 83.42%, compared to 74.97% for Slovene translations.

5 CONCLUSIONS AND FURTHER RESEARCH

The proper identification of MWEs that appear multiple times in the parallel sentence aligned corpora offers an opportunity to improve the quality of statistical machine translation.

In the research presented in this paper, we tried to define a framework for effective treatment of lexical units across languages. It passed through four complementary phases presented in the introduction of the paper. In order to measure the correctness of MWE extraction process, as well as the translation prediction, we measured the consistency and completeness of generated translations of MWEs existing in the small parallel Multext-East corpus. We intend to implement the same approach to measure the same parameters in the raw material obtained when Moses SMT toolkit, which was implemented over SETimes corpus [9].

In order to improve the quality of the created translation system, we will first incorporate MWE lexical entries, which are currently created for the Macedonian language [14]. They will consist of fixed MWE lexical entries used in the current stage of the system, and extended with semi-fixed and flexible MWEs. We will also intend to study the lexical cohesion, and extend the document-level translation to a larger collection. Inspired by Ben et al., the final goal in this direction will be the integration of the model into a hierarchical phrase-based SMT system [15]. Most current SMT systems translate sentences individually, assuming that the sentences in a text are independent [16]. A further extension of the system is directed towards the extraction of the common knowledge about multiword expressions out of a continuous context and its incorporation into a translation system capable to competently deal with them.

References

[1] P. F. Brown et al. A statistical approach to machine translation, *Computational linguistics 16.2*, pp. 79-85, 1990.

[2] M. Galley, C. D. Manning. Accurate non-hierarchical phrase-based translation, *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 966-974, 2010.

[3] M. Itagaki, T. Aikawa, X. He. Automatic Validation of Terminology Translation Consistency with Statistical Method, *Proceedings of MT summit XI, 269-274*, pp. 269-274, 2007.

[4] H. Caseli, C. Ramish, M. Nunes, A. Villavicencio. Alignment based extraction of multiword expressions, *Language Resources & Evaluation 44*, pp. 59-77, 2010.

[5] R. Jackendoff. 'Twistin' the night away, *Language 73*, pp. 534-559, 1997.

[6] J. Tiedemann. To cache or not to cache, Experiments with Adaptive Models in Statistical Machine Translation, *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pp. 189-194, 2010.

[7] Y. Zhang, V. Kordoni, A. Villavicencio, M. Idiart. Automated Multiword Expression Prediction for Grammar Engineering, *Proceedings of the 5th Workshop on Important Unresolved Matters*, pp. 44-52, 2006.

[8] P. Koehn, F. J. Och, D. Marcu. Statistical phrase-based models, *Proceedings of NAACL 2003*, pp. 48-54, 2003

[9] M. Stolikj, K. Zdravkova. Resources for Machine Translation of the Macedonian Language, *online Proceedings of ICT Innovations 2009*.

[10] K. Zdravkova, A. Petrovski. System for extraction of potential multi-word expressions and prediction of their translations from a multilingual corpus, *PARSEME 2^nd^ general meeting*, poster 43, 2014.

[11] T. Erjavec. MULTEXT-East: morphosyntactic resources for Central and Eastern European languages, *Language Resources and Evaluation*, Vol. 46 / 1, pp. 131-142, 2012.

[12] V. Vojnovski, S. Džeroski, T. Erjavec. Learning PoS tagging from a tagged Macedonian text corpus, Proceedings of SiKDD 2005, Ljubljana, Slovenia, pp. 199-202, 2005.

[13] L. Guillou. Analysing Lexical Consistency in Translation, *Proceedings of DiscoMT*, Sofia, Bulgaria, pp. 10-18, 2013.

[14] A. Petrovski, K. Zdravkova. How to create a MWE lexical entry?. *PARSEME 3^rd^ general meeting*, poster 8, group B, 2014.

[15] G. Ben, D. Xiong, Z. Teng, Y. Lu, Q. Liu. Bilingual Lexical Cohesion Trigger Model for Document-Level Machine Translation, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, pp. 382-386, 2013.

[16] S. Stymne, J. Tiedemann, C. Hardmeier, J. Nivre. Statistical Machine Translation with Readability Constraints, *Proceedings of the 19th Nordic Conference of Computational Linguistics*, pp. 375-474, 2013.

Indeks avtorjev / Author index

Konferenca / Conference

Uredili / Edited by

Izkopavanje znanja In podatkovna skladišča (SiKDD 2014) / Data Mining and Data Warehouses (SiKDD 2014)

Dunja Mladenić, Marko Grobelnik