IMMM 2013

The Third International Conference on Advances in Information Mining and Management

ISBN: 978-1-61208-311-7

November 17 - 22, 2013

Lisbon, Portugal

IMMM 2013 Editors

Andreas Schmidt, Karlsruhe University of Applied Sciences & Karlsruhe Institute of Technology, Germany

IMMM 2013

Foreword

The Third International Conference on Advances in Information Mining and Management (IMMM 2013), held between November 17-22, 2013 in Lisbon, Portugal, continued a series of academic and industrial events focusing on advances in all aspects related to information mining, management, and use.

The amount of information and its complexity makes it difficult for our society to take advantage of the distributed knowledge value. Knowledge, text, speech, picture, data, opinion, and other forms of information representation, as well as the large spectrum of different potential sources (sensors, bio, geographic, health, etc.), led to the development of special mining techniques, mechanisms support, applications and enabling tools. However, the variety of information semantics, the dynamic of information update and the rapid change in user needs are challenging aspects when gathering and analyzing information.

We take here the opportunity to warmly thank all the members of the IMMM 2013 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to IMMM 2013. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the IMMM 2013 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that IMMM 2013 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of information mining and management.

We are convinced that the participants found the event useful and communications very open. We hope that Lisbon, Portugal, provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

IMMM 2013 Chairs:

IMMM General Chairs

Philip Davis, Bournemouth and Poole College - Bournemouth, UK
David Newell, Bournemouth University - Bournemouth, UK

IMMM Advisory Chairs

Petre Dini, Concordia University, Canada & IARIA, USA
Andreas Holzinger, Institute for Medical Informatics, Statistics and Documentation (IMI) / Medical University Graz (MUG), Austria
Kuan-Ching Li, Providence University, Taiwan
Abdulrahman Yarali, Murray State University, USA

IMMM Industry Liaison Chairs

George Ioannidis, IN2 search interfaces development Ltd., UK
Johannes Meinecke, SAP AG / SAP Research Center Dresden, Germany

IMMM Special Area Chairs on Data Management

Robert Wrembel, Poznan University of Technology, Poland

IMMM Special Area Chair on Special Mining

Yulan He, Knowledge Media Institute / The Open University, UK

IMMM Special Area Chair on Semantic Data Handling

Stefan Brüggemann, OFFIS - Institute for Information Technology, Germany

IMMM Special Area Chair on Databases

Lena Strömbäck, Linköpings Universitet, Sweden

IMMM Special Area Chair on Cloud-based Mining

Roland Kübert, High Performance Computing Center Stuttgart / Universität Stuttgart, Germany

IMMM Publicity Chairs

Zaher Al Aghbari, University of Sharjah, UAE
Alejandro Canovas Solbes, Polytechnic University of Valencia, Spain

IMMM 2013

Committee

IMMM General Chairs

Philip Davis, Bournemouth and Poole College - Bournemouth, UK
David Newell, Bournemouth University - Bournemouth, UK

IMMM Advisory Chairs

Petre Dini, Concordia University, Canada & IARIA, USA
Andreas Holzinger, Institute for Medical Informatics, Statistics and Documentation (IMI) / Medical University Graz (MUG), Austria
Kuan-Ching Li, Providence University, Taiwan
Abdulrahman Yarali, Murray State University, USA

IMMM Industry Liaison Chairs

George Ioannidis, IN2 search interfaces development Ltd., UK
Johannes Meinecke, SAP AG / SAP Research Center Dresden, Germany

IMMM Special Area Chairs on Data Management

Robert Wrembel, Poznan University of Technology, Poland

IMMM Special Area Chair on Special Mining

Yulan He, Knowledge Media Institute / The Open University, UK

IMMM Special Area Chair on Semantic Data Handling

Stefan Brüggemann, OFFIS - Institute for Information Technology, Germany

IMMM Special Area Chair on Databases

Lena Strömbäck, Linköpings Universitet, Sweden

IMMM Special Area Chair on Cloud-based Mining

Roland Kübert, High Performance Computing Center Stuttgart / Universität Stuttgart, Germany

IMMM Publicity Chairs

Zaher Al Aghbari, University of Sharjah, UAE
Alejandro Canovas Solbes, Polytechnic University of Valencia, Spain

IMMM 2013 Technical Program Committee

Aseel Addawood, Cornell University, USA
Zaher Al Aghbari, University of Sharjah, UAE
Riccardo Albertoni, Consiglio Nazionale delle Ricerche - Genova, Italy
César Andrés Sanchez, Universidad Complutense de Madrid, Spain
Annalisa Appice, Università degli Studi di Bari Aldo Moro, Italy
Avi Arampatzis, Democritus University of Thrace, Greece
Liliana Ibeth Barbosa Santillán, University of Guadalajara, Mexico
Barbara Barricelli, University of West London, U.K.
Shariq Bashir, National University of Computer and Emerging Sciences, Pakistan
Grigorios N. Beligiannis, University of Western Greece - Agrinio, Greece
Jorge Bernardino, ISEC - Polytechnic Institute of Coimbra, Portugal
Konstantinos Blekas, University of Ioannina, Greece
Gloria Bordogna, CNR - National Research Council / IDPA - Institute for the Dynamics of Environmental Processes - Dalmine, Italy
Stefan Brüggemann, Astrium GmbH - Bremen, Germany
Olivier Caelen, Atos Worldline, Belgium
Alain Casali, Aix Marseille Université, France
Nadezda Chalupova, Mendel University - Brno, Czech Republic
Sukalpa Chanda, Gjøvik University College, Norway
Chi-Hua Chen, National Chiao Tung University, Taiwan R.O.C.
Yili Chen, Monsanto Company, USA
Been-Chian Chien, University of Tainan, Taiwan
Sung-Bae Cho, Yonsei University, Korea
Ronan Cummins, National University of Ireland - Galway, Ireland
Frantisek Darena, Mendel University - Brno, Czech Republic
Andre Ponce de Leon F. de Carvalho, University of Sao Paulo at Sao Carlos, Brazil
Sébastien Déjean, Université de Toulouse & CNRS, France
Juan José del Coz Velasco, Universidad de Oviedo - Gijón, Spain
Mustafa Mat Deris, University of Tun Hussein Onn, Malaysia
Emanuele Di Buccio, University of Padua, Italy
Qin Ding, East Carolina University - Greenville, USA
Aijuan Dong, Hood College - Frederick, USA
Nikolaos Doulamis, National Technical University of Athens, Greece
Anass Elhaddadi, University of Paul Sabatier - Toulouse, France
Christian Ernst, Ecole des Mines de St Etienne - Gardanne, France
Ingrid Fischer, Universität Konstanz, Germany
Paolo Garza, Dipartimento di Automatica e Informatica Politecnico di Torino, Italy
Alessandro Giuliani, University of Cagliari, Italy
Eloy Gonzales, National Institute of Information and Communications Technology - Kyoto, Japan
Luigi Grimaudo, Politecnico di Torino, Italy
Richard Gunstone, Bournemouth University, UK
Fikret Gurgen, Bogazici University, Turkey
Nima Hatami, University of California - San Diego, USA

Kenji Hatano, Doshisha University, Japan
Ourania Hatzi, Harokopio University of Athens, Greece
Yulan He, Aston University, U.K.
Andreas Holzinger, Medical University Graz (MUG), Austria
Gilles Hubert, IRIT - University of Toulouse / Université Paul Sabatier, France
Masoumeh Izadi, McGill University Health Center - Montreal, Canada
Mansoor Zolghadri Jahromi, Shiraz University, Iran
Bartłomiej Jefmański, Wrocław University of Economics, Poland
Heng Ji, City University of New York, USA
Wei Jin, Amazon.com, Seattle, USA
Tahar Kechadi, University College Dublin, Ireland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Etienne Kerre, Fuzziness and Uncertainty Modelling Research Unit - Gent, Belgium
Frank Klawonn, Ostfalia University of Applied Sciences - Wolfenbuettel, Germany
Roumen Kountchev, Technical University of Sofia, Bulgaria
Piotr Kulczycki, Polish Academy of Science | Cracow University of Technology, Poland
Rein Kuusik, Tallinn University of Technology, Estonia
Cristian Lai, CRS4, Italy
Giuliano Lancioni, Roma Tre University, Italy
Mariusz Łapczyński, Cracow University of Economics, Poland
Georgios Lappas, Technological Institute of Western Macedonia, Greece
Hao Li, The City University of New York, USA
Kuan-Ching Li, Providence University, Taiwan
Shuying Li, University of Science and Technology of China (USTC), China
Tao Li, Florida International University, USA
Qing Liu, CSIRO, Australia
Xumin Liu, Rochester Institute of Technology, USA
Elena Lloret Pastor, Universidad de Alicante, Spain
Corrado Loglisci, University of Bari "Aldo Moro", Italy
Flaminia Luccio, Università Ca' Foscari Venezia, Italy
Qiang Ma, Kyoto University, Japan
Shuai Ma, Beihang University, China
Stéphane Maag, TELECOM SudParis, France
Thomas Mandl, Universität Hildesheim, Germany
Ioannis Manolopoulos, Aristotle University of Thessaloniki, Greece
Francesco Marcelloni, University of Pisa, Italy
Elena Marchiori, Radboud University - AJ Nijmegen, The Netherlands
Ali Masoudi-Nejad, University of Tehran, Iran
Artura Mazeika, Max Planck Institute for Informatics - Saarbrücken, Germany
Johannes Meinecke, SAP AG, Germany
Dia Miron, Recognos, Romania
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Charalampos Moschopoulos, Katholieke Universiteit Leuven, Belgium
Katarzyna Musial-Gabrys, King's College London, UK
Peter Mutschke, GESIS - Leibniz Institute for the Social Sciences, Germany
Ulrich Norbisrath, BIOMETRY.com / University of Tartu, Estonia
Kok-Leong Ong, Deakin University, Australia
Samia Oussena, University of West London, UK

Yoseba Penya Landaburu, University of Deusto - Basque Country, Spain
Nathalie Pernelle, Université Paris-Sud, France
Jürgen Pfeffer, Carnegie Mellon University, USA
Katja Pfeifer, SAP AG, Germany
Ioannis Pratikakis, Democritus University of Thrace - Xanthi, Greece
Nishkam Ravi, NEC Labs - Princeton, USA
Arno H.P. Reuser, Reuser's Information Services, Netherlands
Daniel Romero, Cornell University, USA
Paolo Rosso, Universidad Politécnica Valencia, Spain
Igor Ruiz-Agundez, University of Deusto - Basque Country, Spain
Alessia Saggese, University of Salerno, Italy
Jörg Scheidt, University of Applied Sciences Hof, Germany
Gyuzel Shakhmametova, Ufa State Aviation Technical University, Russia
Mingsheng Shang, University of Electronic Science and Technology of China, China
Armin Shams, University of Tehran, Iran
Josep Silva, Universitat Politècnica de València, Spain
Simeon Simoff, University of Western Sydney, Australia
Cristina Solimando, University Roma Tre, Italy
Theodora Souliou, National Technical University of Athens, Greece
Jaideep Srivastava, University of Minnesota, USA
Vadim Strijov, Computing Centre of the Russian Academy of Sciences, Russia
Tatiana Tambouratzis, University of Piraeus, Greece
Tõnu Tamme, University of Tartu, Estonia
Mehmet Tan, TOBB University of Economics and Technology, Turkey
Yi Tang, Chinese Academy of Sciences, China
Xiaohui (Daniel) Tao, The University of Southern Queensland, Australia
Olivier Teste, Université de Toulouse, France
Vincent S. Tseng, National Cheng Kung University, Taiwan, R.O.C.
Chrisa Tsinaraki, Technical University of Crete Campus, Greece
Pavel Turcinek, Mendel University - Brno, Czech Republic
Franco Turini, University of Pisa, Italy
Lorna Uden, Staffordshire University, UK
Eli Upfal, Brown University - Providence USA
Domenico Ursino, Università Mediterranea di Reggio Calabria, Italy
Nico Van de Weghe, Ghent University, Belgium
Julien Velcin, Université de Lyon 2, France
Zeev Volkovich, ORT Braude College Karmiel, Israel
Michael N. Vrahatis, University of Patras, Greece
Stefanos Vrochidis, Information Technologies Institute - Centre for Research and Technology Hellas, Greece
Baoying (Elizabeth) Wang, Waynesburg University, USA
Qi Wang, University of Science and Technology of China, China
Hao Wu, Yunnan University - Kunming, P.R.China
Feng Yan, Facebook Inc., USA
Zhenglu Yang, University of Tokyo, Japan
Jan Zizka, Mendel University - Brno, Czech Republic

Copyright Information

Table of Contents

Analysis of Patents for Prior Art Candidate Search

Sébastien Déjean*, Nicolas Faessel†, Lucille Marty‡, Josiane Mothe†, Samantha Sadala‡ and Soda Thiam‡
*IMT, Toulouse, France
Email: sebastien.dejean@math.univ-toulouse.fr
†IRIT, Toulouse, France
Email: {faessel,mothe}@irit.fr
‡INSA, Toulouse, France
Email: {lmarthy,sadala,thiam}@etud.insa-toulouse.fr

***Abstract*—In this paper, we describe a method for analyzing a collection of patents in order to help prior art candidate search in an interactive and graphical way. The method relies on the use of two data mining methods: hierarchical agglomerative clustering and principal component analysis, which are applied successively. The correlation between the application patent and the other patents is a good indicator to help decide the classes of patents to look at.**

***Index Terms*—Information retrieval, Patent retrieval, Prior art retrieval, Visualization, Information mining.**

I. Introduction

Patents correspond to one type of intellectual property right that plays an important role in innovation and in the economy. Patent retrieval is crucial considering the amount of existing information and economic issues associated with patents. The number of patents available make mandatory to have effective and efficient ways of searching and browsing them. For example, Thomson Scientific provides the Derwent World Patents Index® , which ”contains over 21.85 million patent families covering more than 45.2 million patent documents, with coverage from over 47 worldwide patent authorities” [1]. Additionally, when applying for a new patent, prior art should be verified both by the applicant, to fill in the application, and by the certifier to analyze the application. 1.98 million applications were filed in worldwide in 2011 according to the World Intellectual Property Organization (WIPO) [2]. Verifying prior art is thus crucial and has been investigated in the relevant literature. International evaluation forums also define tasks to evaluate prior art search.

The Conference and Labs of the Evaluation Forum Initiative (CLEF), formerly known as Cross-Language Evaluation Forum, launched the Intellectual Property track (CLEF-IP) in 2009 to investigate Information Retrieval (IR) techniques for patent retrieval. In 2011, CLEF-IP specifically focuses on prior art candidate search. The Text REtrieval Conference Chemical Track 2011 (TREC-CHEM 2011) also considers prior art candidate search [3]. Prior art search is not the only task evaluation forums investigates, there are many information retrieval and analysis tasks and other challenges related to patents [4], [5]. However, in this paper, we will focus on this task.

Prior art candidate search aims at querying and retrieving the patents in order to discover any knowledge existing prior to the analyzed patent application [6]. The underlying objective that is pursued is to find patents, which can invalidate a given patent application. Most of the approaches to this task use standard information retrieval (IR) processes [7], [8].

Results are evaluated using standard IR measures that consider the rate of relevant documents that have been retrieved and the rate of retrieved documents that are relevant. A relevant document in the case of prior art candidate search is a patent that is evaluated as potentially invalidating the current patent application. More specifically, evaluation measures such as Mean Average Precision (MAP), recall, precision, recall at 100 (when 100 retrieved documents are considered) and precision at 100 have been used.

In this paper, we make post evaluation analysis. More specifically, this study aims at analyzing patents knowing the relevance judgments. We show that considering patent title and patent abstract is complementary. We also show that it is possible to iteratively reduce the number of patents to be analyzed while keeping a high level of recall. Patent analysis is done using clustering and factorial analysis methods. The analysis results in a graphical visualization the user can interact with. The rest of this paper is organized as follows. Section II presents related works regarding prior art search, and browsing and clustering document collections. Section III describes the way patents are indexed and the resulting representations to be analyzed. Section IV presents the methodology of analysis and Section V the results of the patent analysis. Finally, Section VI discusses the results and concludes this paper.

II. Related Work

A. Prior art candidate search

Prior art candidate search has been studied in CLEF-IP as finding ”patent documents that are likely to constitute prior art to a given patent application” [9]. Figures 1 and 2 show an application patent, and a previous patent potentially invalidating the application.

The results obtained in the evaluation campaign show that the task is quite hard. For example, the best run in 2010 obtained a MAP of about 0.26 [7].

Mainly, standard IR methods have been used in the literature to solve this task. Madgy et al. [10] have used standard information retrieval techniques (stop word removal, stemming

```
<patent−document ucid=”EP−1236420−A1” lang=”FR”>
  <bibliographic−data>
    <technical−data status=”new”>
      <invention−title lang=”DE”>Brste zum Auftragen eines
           Produktes auf keratinische Fasern
       </invention−title>
      <invention−title lang=”EN”>Brush for applying a
           product on keratinous fibres</invention−title>
      <invention−title lang=”FR”>Brosse pour l’application
           d’un produit sur les fibres kratiniques
       </invention−title>
    </technical−data>
  </bibliographic−data>
  <abstract lang=”EN”>
    <p>The applicator comprises a rod with a brush at one
         end. The portion of the rod adjacent to the brush
         has an axis (Y) and the brush comprises a core (11)
          from a portion of which bristles extend. The core
         is curved over a part of its length and the angle
         between the rod axis and the core axis is less than
          90o and the brush free end is not aligned with the
          rod axis.</p>
  </abstract>
  <abstract lang=”FR”>
    <p>La prsente invention concerne un dispositif pour l’
         application d’un produit sur les fibres kratiniques
         , notamment pour l’application de mascara sur les
         cils, comportant une tige munie  une extrmit d’une
         brosse, la portion de la tige adjacente  la brosse
         ayant un axe (Y).</p>
  </abstract>
  <abstract lang=”FR”>...</abstract>
  <description lang=”FR”>...</description>
  <claims>...</claims>
</patent−document>
```

Fig. 1. Patent application

```
<patent−document ucid=”EP−0792603−A1” lang=”FR”>
  <bibliographic−data>
    <technical−data status=”new”>
      <invention−title lang=”DE”>Brste zum Anbringen von
           Kosmetika und insbesondere Mascara</invention−
           title>
      <invention−title lang=”EN”>Brush for applying
           cosmetics and in particular mascara</invention−
           title>
      <invention−title lang=”FR”>Brosse progressive pour
           appliquer un produit cosmtique, notamment du
           mascara</invention−title>
    </technical−data>
  </bibliographic−data>
  <abstract lang=”EN”><p>The applicator brush consists of a
        cylinder of bristles radiating from a core in the
       form of a twisted metal wire spiral, and has at least
        one concave curved recess (107) in the cylindrical
       surface. The recess is oval, circular or elliptical
       in shape, or it can be made from two sections of a
       circle which intersect. It is located in a zone (109)
        of the bristle cylinder which is less dense than the
        two ends (112,113). The ends of the cylinder have
       alternating long and short bristles. The applicator
       brush can be attached to the inside of a cap which
       screws onto the neck of a cosmetic product container
       .</p>
  </abstract>
  <description lang=”FR”>...</description>
  <claims>...</claims>
</patent−document>
```

Fig. 2. Prior art candidate to invalidate patent in Figure 1

using Porters stemmer) and obtained a Mean Average Precision of 0.1216, a recall at 100 of 0.3036 and precision at 100 of about 0.228. They used the Indri system that ranks the results using a language model and inferred networks. Becks et al. [11] used the Okapi system and BM25 weighting. Most of the other participants also used either the Lemur/Indri system or the Apache/Lucene system. However, these approaches do not allow browsing and interactive visualization.

B. Browsing and clustering document sets

One frequently cited work regarding document browsing is Scatter/Gather [12]. The principle developed in this approach is an interactive refinement of the target document set. From an initial clustering of a document collection in k-clustering, the users select the clusters they are interested in; then, the system re-clusters this subset of documents, and so on.

Many works have investigated search result clustering either to re-rank results initially retrieved by a search engine or to group the results and provide users with clusters of documents they can choose [13].

Classification in the context of patents is generally concerned with classifying patents using the International Patent Classication (IPC) codes or other classification schema. However, some interesting work has been conducted in this context for patent mapping. Kim et al. [14] for example cluster patent document contents using k-means. From the clustering results, they extract a semantic network that helps having an overview of the patent subset. Sharma [15] uses k-means clustering algorithms in order to structure a patent sub-collection. Jun et al. [16] uses patent clustering in order to predict technology trends. Djean and Mothe [17] also presents various visual clustering methods and applications.

Our work also uses document clustering and interactive refinement of clusters. More specifically, we cluster patents according to their content, which is automatically extracted. In addition, our work studies the effects of the clustering parameters on the results.

III. Patent representation for text analysis

Since the collection is composed of more than three million patents, we decided to first select a sub-part of these patents, the ones that are more likely to be prior art candidates for each topic. To select this subset of patents, we use a standard information retrieval process. Then, we built a representation that fits the type of analysis we intend to investigate. We generate a patent representation that keeps both the section part from which each term occurs and the language in which it is used. The collection and these two steps are described in this section.

A. Collection

The collection we used is the one used in CLEF-IP forum 2011 [18]. It consists of more than three million patent documents from European Patent Office sources with contents in French, German and English. Several documents can be related to the same patent and correspond to versions (e.g. application phase and granted patent). The documents contain bibliographic data, a title, an abstract, a description, and claims. The patent title is provided in three languages (English, German and French). For some of the patents, the abstract is also provided in the three languages, for others in one of the languages only. The description and claims are in one language only.

Topics correspond to patents and thus have the same structure. In this study, we used the 1000 official topics, to which are associated the patents relevant to the task. Relevance judgements are produced automatically, using patent citations from seed patents.

B. First patent selection

To make a first targeted sub-collection for each topic, we used the Terrier system [19]. We built an index considering the English titles and English abstracts. We used a stop-word list and Porter's stemmer and the default Terrier parameters. Then, from a topic, we query the index and retrieve, at most, the 1000 first patents. Those patents are then indexed more precisely.

C. Patent indexing and representation

Since patents are multilingual, or to be more precise, a language is associated to each patent's parts, we built three indexes, one per language for the patents from the targeted subset (in reality we build those indexes for the entire corpus once). In the English index, we consider the English patent parts only; in the German index, we consider the German parts only, and so on. This solution allowed us to use appropriate stop-word list and stemmer.

Each indexing term was associated with the patent part it comes from, the associated language and its frequency and becomes a variable that represents the patent (in the statistical sense). For example, the word "test", found in the abstract in English, will be counted in the variable named A_EN_test and its frequency will be kept. If it occurs in the French title too, a second variable will be defined named T_FR_test.

The indexing result can be defined as a matrix in which lines correspond to patents (individuals) observed according to the variables (indexing terms as defined previously) which are the columns of the matrix. The values inside the matrix are the term frequencies in the various parts and languages. This representation allows us to easily fuse lines or columns. Fusing lines is mandatory to fuse the various patent versions into a single one (in that case we use the mean of the frequency in each version to calculate the new term frequency). Fusing columns is useful when one wants to calculate the frequency of a term, independently to which patent parts it occurs.

IV. Analysing prior art

Prior art candidate search aims at finding patents related to the topic. It can be considered as a search problem or as a clustering problem. In this study, we chose the latter solution. In addition, the representation we chose is highly dimensional. Factorial analysis is a class of methods that aims at reducing the dimension of data whilst keeping its structure. We use Principal Component Analysis (PCA) in this study.

A. Analysing method

1) Clustering method: We chose to consider the ascending agglomerative hierarchical clustering (AHC) to group similar patents. Indeed, the target number of clusters is unknown. Unlike k-means or other methods, AHC does not require specifying the desired number of clusters. Rather, the number of clusters is chosen by looking at the graph on the decay of the node heights.

The AHC requires choosing the aggregation measure and the dissimilarity measure to use. The best aggregation measure is the Ward measure as this method makes homogeneous clusters. We use this method. With regard to the dissimilarity measure, we could use the Euclidean distance (which is the most commonly used) or other distances such as the Minkowski distance (for which the power parameter p as to be chosen) or the Manhattan distance, which are the most popular. The Manhattan distance corresponds to the norm 1, this distance will select too many patents and not the most correlated. We compared the Euclidean distance and the Minkowski distance using p=1/2 and kept the latter (the detailed results are not presented in this paper).

2) Dimension reduction: PCA is a very popular method in multidimensional statistical analysis. It makes it possible to produce graphical representations of the rows or the columns of the considered matrix, and does so in a reduced dimension space. The method is defined so that the dispersion obtained in this reduced dimension space is the largest (Jolliffe, 2002 [20]; Mardia et al., 1979 [21]). The aim of PCA is to replace a p-dimensional observation by a q linear combination of the variables, where q (the dimension of the reduced space) is much smaller than p. The linear combination defined by PCA are the eigenvectors related to the first q greatest eigenvalues.

B. Methodology

We promote a way to browse the sub-collection of patents in order to better detect prior art candidate, which is based on clustering and PCA. The method we suggest is iterative, as shown in Figure 3. First, we used a PCA in order to plot the patents and look for groups and dispersion of patents. Then, we cluster the patents in order to refine the target set of patents: the cluster containing the topic is selected. Finally, we consider the correlations between the query and the patents belonging to the cluster selected at the previous stage in order to evaluate the number of relevant patents this method selects. This methodology is applied first to patents represented by titles and abstracts, then to titles only and finally to abstracts only.

V. Results

The results we present in this section are based on the topic EP-1236420-A1 from the corpus. The experiments have been conduced on a single query topic, which is not corresponding to the CLEF-IP task. The results we are presenting here should be considered as preliminary results.

When considering the EP-1236420-A1 topic, there are 11 relevant patents associated to this topic identified as follows: EP0663161, EP0728427, EP0792603, EP0808587, EP0811336, EP0811337, EP0832580, EP0842620, EP0895734, EP1020136, EP1177745. In order to simplify the writing, we will refer to the these patents by using the

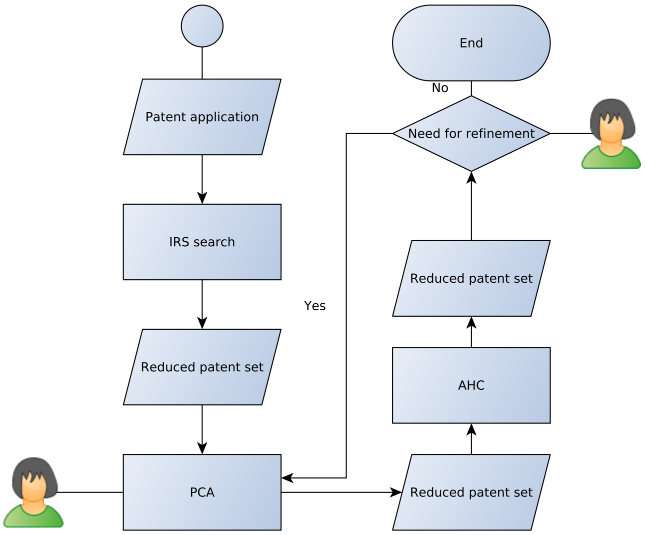


Fig. 3. Process description

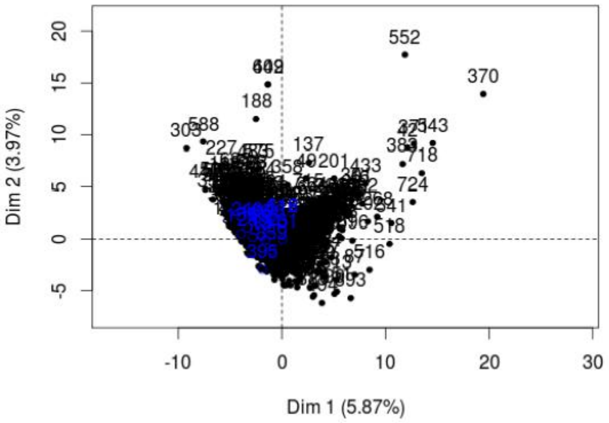


Fig. 4. Patent factor map (PCA) based on titles and abstracts using the two first dimensions

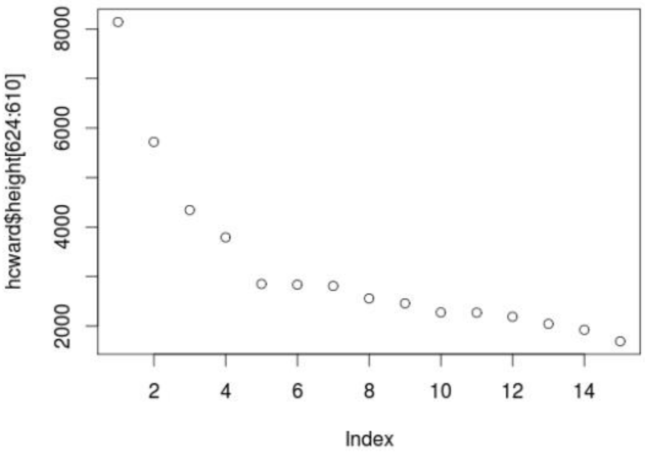


Fig. 5. Number of clusters (AHC)

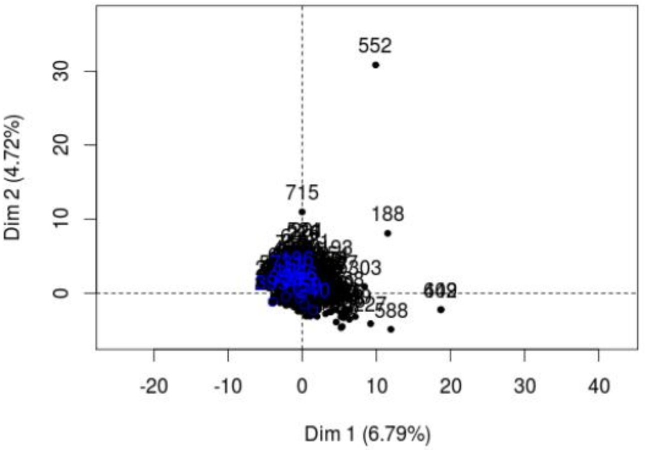


Fig. 6. Patent factor map (PCA) based on titles and abstracts on a reduce patent set

names A, B, C, D, E, F, G, H, I, J, and K respectively. The most invalidating patents are C, D, G and H.

A. English words from titles and abstracts

Figure 4 shows the representation of the patents after the first PCA. The relevant documents and the query are represented in blue, and the query is represented by the number 731. We can see three groups of patents: one in the top-left, one in the top-right and the other in the center. The query is between these three groups; from this visualization, it is difficult to know which group it belongs to. This can be explained by the fact that all these patents belong to the sub-collection related to the query 731. Moreover, we cannot identify the number of patents because the number of individuals is very large. But we can observe a ”blue” group which means that the relevant documents are close to the query. However, this graph is not very relevant to our analysis.

We will now focus on the results of the clustering which is an AHC as depicted in Section IV-A1. Figure 5 shows the graph of heights according to the number of clusters.

At this stage, we would like to obtain a sufficient number of patents in each cluster, in order to find the maximum of prior art candidates. According to Figure 2, a relevant pruning is 5 as the number of clusters. When analyzing the patent clusters, the query is in a cluster formed by 276 patents. Among these patents, we found the relevant patents labelled A, C, D, G, H, I, J and K, which means 8 out of 11.

We apply PCA on the topic cluster. Figure 6 shows the representation of the 276 patents after PCA. The query and the relevant patents are in blue; we can see one group in the center. Patents from this cluster are much related to the query. We looked to the correlations between the query and the patents belonging to the query cluster. The results are presented in the Table I. The threshold should be chosen according to the number of prior art candidates we would like to have. We can see, for example, that 0.25 is a good compromise to have a sufficient number of prior art candidates (the four most

TABLE I
RETRIEVED PATENTS ACCORDING TO QUERY-PATENT CORRELATION – TITLES AND ABSTRACTS

Threshold	Number of patents	Relevant patents
0	101	A, B,G, H, I,K
0.2	42	A,G, H, I, K
0.25	39	A,G, H, I, K
0.3	31	A,G, H, I, K
0.4	19	A,G, I, K
0.5	12	I,K
0.6	7	K
0.7	3	

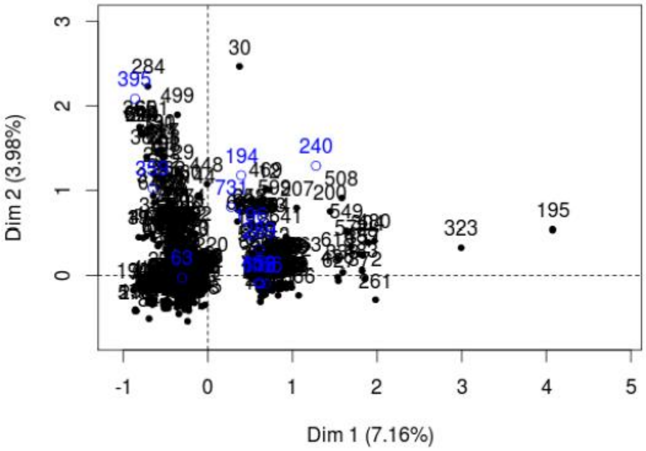


Fig. 7. Patent factor map (PCA) based on titles only using the two first dimensions

invalidating patents C, D, G and H are in the cluster).

B. English words from titles only

In this section, we consider the English words from the title only and reproduce the same analysis: first PCA, then a clustering on the patents and finally looking at the correlations between the query and the patents belonging to that cluster. In Figure 7, three distinct groups of patents can be observed, as well as three isolated patents. Following the method we promote, we apply clustering using AHC. The best pruning is obtained when considering 4 clusters.

The topic cluster is composed of 101 patents; it contains the relevant patents: A, B, G, H, I, K. There are 6 relevant patents out of 11. Figure 5 represents the patents after PCA was applied to the topic cluster. We can see a group of patents very close to the topic, for example the patents 359, 182, 108, 253.

Finally, we looked to the correlations between the query and the patents belonging to the query cluster. The results are presented in the Table II.

C. English words from abstracts only

The analysis is the same as previously, except that it is applied to patent abstracts only. The PCA on the sub-collection is presented in Figure 8.

We can see three groups of patents, as it was the case when considering the English words from titles and abstracts: one in the top-left, one in the top-right and the other in the center.

TABLE II
RETRIEVED PATENTS ACCORDING TO QUERY-PATENT CORRELATION – TITLES ONLY

Threshold	Number of patents	Relevant patents
0	276	A, C, D, G, H, I, J, K
0.2	144	A, C, D, G, H, I, J
0.25	79	A, C, D, G, H, J
0.3	36	A, H, J
0.4	5	J
0.5	1	

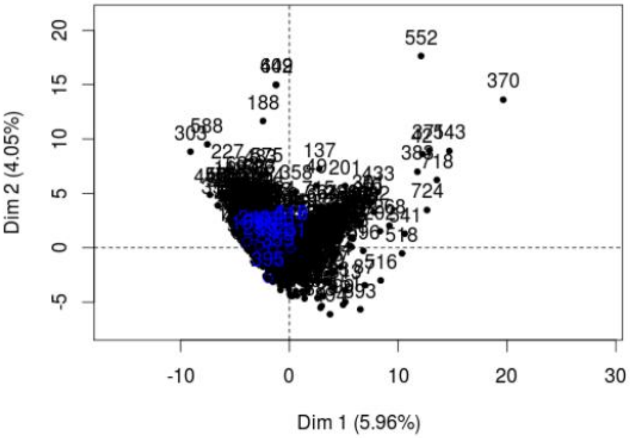


Fig. 8. Patent factor map (PCA) based on abstracts only using the two first dimensions

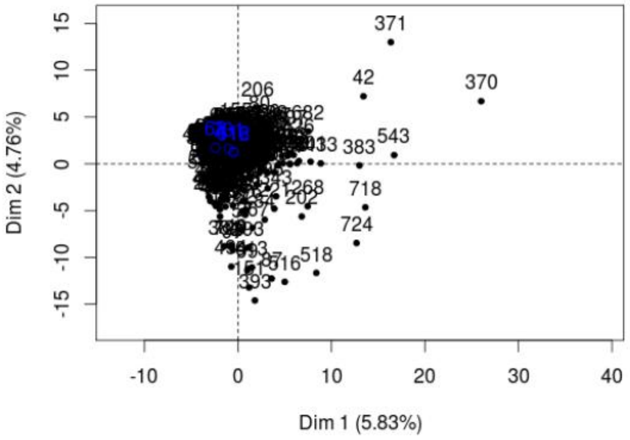


Fig. 9. Patent factor map (PCA) based on abstracts only on a reduce patent set

The query is always between these three groups so from this visualization, it is not possible to know which group it belongs to, but we still have a ”blue” group, this means that the relevant patents are highly correlated to the query.

When applying AHC, the best pruning is obtained when considering 4 clusters. The topic is in a cluster formed by 348 patents and containing the relevant patents B, E, F.

We apply a PCA on this cluster, and represent the individuals in Figure 9.

TABLE III
RETRIEVED PATENTS ACCORDING TO QUERY-PATENT CORRELATION
ABSTRACTS ONLY

Threshold	Number of patents	Relevant patents
0	348	B, E, F
0.2	208	B, E, F
0.25	115	B, E, F
0.3	58	B, E, F
0.4	10	B, E, F
0.5	1	

TABLE IV
RETRIEVED PATENTS ACCORDING TO QUERY-PATENT CORRELATION
THRESHOLD 0.25

	Number of patents	Relevant patents
Titles & Abstracts	39	A, **G**, **H**, I, K
Titles only	79	A, **C**, **D**, **G**, **H**, J
Abstracts only	115	B, E, F

The query is in the center of the group. Some patents around the group are not relevant to the query. We have lost much more relevant patents using the abstract only for clustering than when using the titles only.

When looking at the correlations between the query and the patents belonging to the query cluster we found the results presented in Table III.

VI. DISCUSSION AND CONCLUSION

The results we obtained when either considering titles and abstracts or titles or abstracts only are interesting in various ways. If we consider one particular threshold for the correlation between patents and the topic patent, patents that are obtained in the topic cluster are noticeably quite different depending on the patent part that is analyzed. Table IV presents the results for a correlation threshold of 0.25. One can see that the results obtained using title only and abstract only are disjointed. Moreover, the 4 most invalidating patents (in bold in Table IV) are retrieved and, in the case of this topic, title only is enough to find out them.

These results have been obtained using a single query patent. The process we use is iterative and interactive ; for this reason it could be difficult to compare to other methods. However, we could compare the mean average precision (or other performance measure) we obtain using our method with the values obtained by the CLEF-IP participants. That will be done in future work.

In this paper, we have shown that analysis methods can be used in an interactive way to visualize patents. This method can be used to browse a patent collection when searching for prior art candidate in an original way. As future work, we will study multilingual aspects. We think that clustering and factorial analysis could help in grouping together patents that are preliminary written in different languages thanks to the shared terms in the titles and abstracts.

REFERENCES

[1] "Derwent World Patents Index," 2012, accessed: 2013-11-10. [Online]. Available: http://www.eval-inno.eu/wiki/index.php/Derwent_World_Patents_Index

[2] "World Intellectual Property Organization," 2011, accessed: 2013-11-10. [Online]. Available: http://www.wipo.int/

[3] M. Lupu, J. Zhao, J. Huang, H. Gurulingappa, J. Fluck, M. Zimmermann, I. V. Filippov, and J. Tait, "Overview of the trec 2011 chemical ir track." in TREC. National Institute of Standards and Technology (NIST), 2011.

[4] M. Lupu, K. Mayer, J. Tait, and A. Trippe, Current Challenges in Patent Information Retrieval, ser. The Information Retrieval Series. Springer, 2011, vol. 29.

[5] J. Tait and B. Diallo, "Future patent search," in Current Challenges in Patent Information Retrieval, ser. The Information Retrieval Series. Springer, 2011, vol. 29, pp. 389–407.

[6] H. Gurulingappa, B. Müller, R. Klinger, H.-T. Mevissen, M. Hofmann-Apitius, C. M. Friedrich, and J. Fluck, "Prior art search in chemistry patents based on semantic concepts and co-citation analysis," in TREC, E. M. Voorhees and L. P. Buckland, Eds. National Institute of Standards and Technology (NIST), 2010.

[7] J. T. Florina Piroi, "CLEF-IP 2010: Retrieval Experiments in the Intellectual Property Domain," IRF, Tech. Rep., 2010, tech.Rep IRF-TR-2010-00005.

[8] F. Piroi, M. Lupu, A. Hanbury, and V. Zenz, "Clef-ip 2011: Retrieval in the intellectual property domain," in CLEF (Notebook Papers/Labs/-Workshop), V. Petras, P. Forner, and P. D. Clough, Eds., 2011.

[9] "CLEF-IP 2011 overview," 2011, accessed: 2013-09-04. [Online]. Available: http://www.ir-facility.org/clef-ip

[10] W. Magdy and G. J. F. Jones, "Applying the kiss principle for the clef-ip 2010 prior art candidate patent search task," in CLEF (Notebook Papers/LABs/Workshops), M. Braschler, D. Harman, and E. Pianta, Eds., 2010.

[11] D. Becks, T. Mandl, and C. Womser-Hacker, "Phrases or terms? the impact of different query types," in CLEF (Notebook Papers/LABs/-Workshops), M. Braschler, D. Harman, and E. Pianta, Eds., 2010.

[12] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/gather: a cluster-based approach to browsing large document collections," in Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, ser. SIGIR '92, 1992, pp. 318–329.

[13] C. Carpineto, S. Osinski, G. Romano, and D. Weiss, "A survey of web clustering engines," ACM Computer Survey, vol. 41, no. 3, 2009.

[14] Y. G. Kim, J. H. Suh, and S. C. Park, "Visualization of patent analysis for emerging technology," Expert Systems with Applications, vol. 34, no. 3, pp. 1804–1812, Apr. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2007.01.033

[15] A. Sharma, "A survey on different text clustering techniques for patent analysis," International Journal of Engineering, 2012.

[16] S. Jun, S.-S. Park, and D.-S. Jang, "Technology forecasting using matrix map and patent clustering," Industrial Management and Data Systems, vol. 112, no. 5, pp. 786–807, 2012.

[17] S. Déjean and J. Mothe, "Visual clustering for data analysis and graphical user interfaces," in Handbook of Cluster Analysis, C. Hennig, M. Meila, F. Murtagh, and R. Rocci, Eds. http://www.crcpress.com: Chapman & Hall, CRC Press, 2014, to appear.

[18] "CLEF-IP 2011, download data, University of Technology Vienna," 2011, accessed: 2010-09-04. [Online]. Available: http://www.ifs.tuwien.ac.at/∼clef-ip/download/2011/index.shtml

[19] I. Ounis, G. Amati, P. V., B. He, C. Macdonald, and Johnson, "Terrier Information Retrieval Platform," in Proceedings of the 27th European Conference on IR Research (ECIR 2005), ser. Lecture Notes in Computer Science, vol. 3408. Springer, 2005, pp. 517–519.

[20] I. T. Jolliffe, Principal Component Analysis, 2nd ed. Springer-Verlag, 2002.

[21] K. V. Mardia, J. T. Kent, and J. M. Bibby, Multivariate Analysis. Academic Press, 1979.

Clustering Algorithms and Weighted Instance Based Learner for User Profiling

Ayse Cufoglu
Dept. of Computing and Technology
Faculty of Science and Technology
Anglia Ruskin University
Cambridge, UK
a.cufoglu@ieee.org

Mahi Lohi, Colin Everiss
Dept. of Electronic, Networks and Computer Engineering
Dept. of Business Information Systems
University of Westminster
London, UK
lohim@wmin.ac.uk, c.g.everiss@wmin.ac.uk

***Abstract*—User profiling has created opportunities for service providers to make available a channel for user awareness as well as to achieve high user satisfaction. Apart from traditional collaborative and content-based methods, a number of classification and clustering algorithms have been used for user profiling. In our previous work, a weighted classification method, namely Weighted Instance Based Learner (WIBL), was proposed and evaluated for user profiling. In this paper, we aim to compare the performance of a WIBL algorithm with well known clustering algorithms for user profiling. Simulations showed that a WIBL is capable of outperforming the clustering algorithms.**

Keywords-User Profiling; Weighted Instance Based Learner (WIBL); Clustering Algorithms.

I. Introduction

Personalization of services, is an opportunity to help improve the quality of service. The success of these services relies on how well the service provider knows the user requirements and how well this can be satisfied. The user profile is the representation of the user and holds information about the user such as personal profile data (demographic profile data), interest profile data and preference profile data. These profiles are the outcome of the user profiling. In user profiling applications a major challenge is to build and handle user profiles. In the literature, two fundamental user profiling methods have been proposed for this purpose. These are the collaborative and the content-based methods. It is also possible to use a hybrid of these two methods [1]-[3].

The collaborative method has been built on the assumption that similar users, with respect to the age, sex, and social class, behave similarly, and therefore have similar profiles [1][4]. The content-based method, on the other hand, has been built on the concept of content similarity and assumes that users behave similarly under the same circumstances [1][4]. Apart from the traditional profiling methods, well known data mining and machine learning algorithms have found applications within the user profiling process in personalization [5]-[7]. This paper is the first in the literature to compare the performance of Weighted Instance Based Learner (WIBL) [8], with selected algorithms for user profiling.

The paper is organized as follows: Section II, provides related works for this study. Section III, provides information about the algorithms, while Section IV, presents the simulation results. Finally, Section V, concludes this paper.

II. Related Works

Various research works have been carried out with user profiling methods [9]-[13]. For example, the moreTourism, mobile recommendations for tourism [9], uses a hybrid method. The proposed recommended system takes into account the tags, provided by the users, to provide tourist information profiled for users with similar likes depending on the user profile (user tag cloud), location in time and space, and the nearby context (e.g., nearby historical places and museums). In [10], Fernandez et al. proposed a tourism recommendation system that offers tourist packages (i.e., include tourist attractions and activities), that best matches with the user's social network profiles. Different from [9], the proposed hybrid system provides recommendations based on both the user's viewing histories (in this instance, Digital Television (DTV) viewing histories received from the user's set-top boxes via a 2.5/3G communication network) and the preferences in the social network (i.e., preferences of the user's friends). In [11], collaborative filtering was employed together with techniques from the Multi Criteria Decision Analysis (MCDA) for item recommendation. In this system, the MCDA was used to find the similar users while collaborative filtering was used to recommend items. In another work, a hybrid TV program recommendation system, gueveo.tv [12], has been proposed. According to Martinez et al. [12], the proposed system works well because both methods complement each other in a way, that the content-based method recommends usual programs while collaborative method provides the discovery of new shows.

The significance of user profiles for various personalization applications has triggered the use of classification and clustering algorithms in user profiling [5]-[7]. In [5], Irani et al. focused on the social spam profiles in MySpace. In their work, they compared well known machine learning algorithms (AdaBoost algorithm, C4.5 Decision Tree (DT), Support Vector Machine (SVM), Neural Networks (NNs), and Naive Bayesian(NB)) with respect to their abilities to distinguish spam profiles from legitimate profiles. According to the simulations carried out on over 1.9 million MySpace profiles, the C4.5 DT algorithm achieved the highest accuracy of 99.4% in finding the spam profiles, while NB achieved 92.6% accuracy. Simulations were performed on the Weikato Environment for Knowledge Analysis (WEKA) platform where classifiers' default settings were used with 10 fold-cross validation. Paireekreng and Wong [6] investigated the use of clustering and classification of user profile at the client-side mobile. Here, the authors focused on content personalization to help mobile users retrieve information and services efficiently. In their proposed two phase framework, clustering was used to construct a user profile, while classification was classifying user profile based on the class information from clustering. In this work, K-means, TwoStep, Anomaly and Kohenen clustering algorithms were compared for clustering. Moreover, Locally Weighted Learning (LWL), RepTree, Decision Table and SVMReg classifiers were compared for classification. According to simulations, authors state that, for this framework, K-means and RepTree were the best options for classification and clustering respectively.

In our previous work [8], we have proposed a weighted classification method, WIBL. In this paper, however, we aim to compare the performance of WIBL with well known clustering algorithms on user profile.

III. Algorithms

A. Weighted Instance Based Learner (WIBL)

Instance Based Learner (IBL), is a comprehensive form of the Nearest Neighbour (NN) algorithm which normalizes the range of its attributes, processes instances incrementally and has a simple policy for tolerating missing values [14]. In contrast to IBL, the WIBL [8] assigns weights to the attributes and considers the weighted distance of the instances for classification. Here, relevant attributes are aimed to have more influence on classification than irrelevant attributes. In WIBL the function that calculates the distance between test instance (new user) X_i and the training instance (existing user) Y_j is $dist(X_i, Y_j) = \sqrt{\sum_{k=1}^{A} w_{k,l}(C_m)\; g(x_i(k), y_j(k))}$, where $w_{k,l}(C_m) = P(C_m|f_k(l))$ [8]. Here, l is equal to the value of the $x_i(k)$. Therefore, the selection of which weight is to be used for a particular attribute value is based on k and $x_i(k)$. Note that $g(x_i(k), y_j(k))$ is evaluated as it is in IBL [8].

B. Clustering Algorithms

Clustering, also called unsupervised classification, is the process of segmenting heterogeneous data objects into a number of homogenous clusters [15]. Each cluster is a collection of data objects that are similar to one another and dissimilar to the data objects in other cluster/s [16]. A successful clustering algorithm has clusters with high intra-class similarity and low inter-class similarity [16] (see Figure 1 [17]).

Each clustering algorithm uses a different method to cluster the information. In the literature, the most popular clustering methods can be categorized into three subsections. These are Hierarchical, Partitional and Density-Based Clustering (DBC).

1) Hierarchical Clustering: Hierarchical clustering, is the process to create a hierarchical decomposition (dendogram) of the set of data objects [16]. The well known hierarchical clustering algorithms are Single-Linkage, Complete Linkage and Average-Linkage.

In Single-Linkage Clustering (SLC), the resulted distance between two clusters is equal to the shortest distance from any member of one cluster, to any member of the other cluster [18]. Here, the shortest distance reflects the maximum similarity between any two data objects in two different clusters.

The Complete-Linkage Clustering (CLC), is the opposite form of the single-linkage clustering since, in complete-linkage, the link between two different clusters is expected to be the maximum distance from any data object of one cluster to any data object of the other cluster [18]. The maximum distance reflects the minimum similarity between two data objects in two different clusters.

The Average-Linkage Clustering (ALC), can be considered as a combination of single and complete-linkage algorithms. The link between two clusters is equal to the average greatest distance of all paired data objects of these clusters.

2) Partitional Clustering: Partitional clustering is a non-hierarchical clustering method. This method creates disjoint clusters in one step, by decomposing the dataset. Therefore, there is no relationship among the clusters [19].

K-means, is the most representative algorithm of partitional clustering [17]. In this algorithm, the number of clusters, Q, is defined by the user. Then, randomly selected Q data objects become the center (cluster centroid) of the Q clusters. The rest of the data objects are assigned to the closest clusters. The cluster center is represented by the mean values of the data objects within the cluster. Therefore, every time that the cluster centroid is being updated, a new data object becomes a member of a cluster. This process is repeated until no change can occur. Figure 2 [4], summarizes the convergence of the K-means clustering algorithm.

3) Density-Based Clustering: Clusters have various sizes and shapes. Clustering based on the similarity distance between the data objects, results in only spherical shaped objects. To find clusters with complex shapes, requires a more comprehensive method than partitional clustering methods. DBC methods have been developed to find the clusters with arbitrary shapes. Such methods use connectivity and density

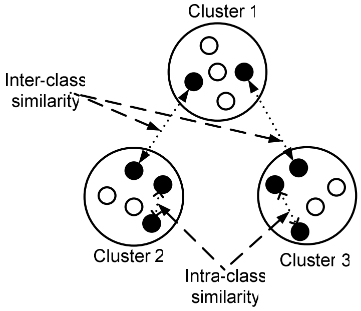


Figure 1. Intra and inter cluster similarity

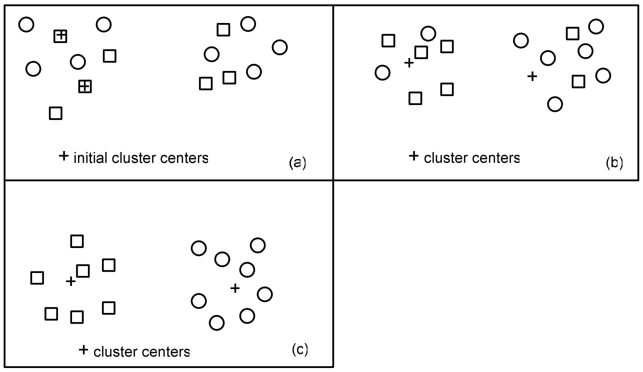


Figure 2. Convergence of K-means partitional clustering: (a) first iteration; (b) second iteration; (c) third iteration

functions to find arbitrary shape clusters [16]. In the data space, these methods consider clusters as dense regions of data objects which are separated by low density regions [20].

IV. Simulations

In this paper, we are comparing the accuracy performance of SLC, CLC, ALC, and K-means clustering algorithms with WIBL for user profiling. The following two sections provide detailed information about the dataset for the simulations and the results of these simulations.

A. Dataset

For the simulations, the dataset used was provided in [21], named ‘Adult Data Set’. This dataset was created by Becker via extracting information from the 1994 census dataset and denoted to UCI (University of California, Irvine) Machine Learning Repository [21] by Kohavi and Becker for data mining applications. In this dataset, the demographic information of 303894 users is listed. 2000 selected users were adopted from this dataset. Some of the demographic information has been discarded and new information has been added to create a complete dataset of user profiles for the simulations.

In this study, each user is represented with three sets of profile information; demographic, interest and preference data. These profiles include information such as Age, Annual Income, Sex, Sport Interest, Music Interest, Book Interest, Marital Status, Employment, Education and Profession. Simulations were carried out with two sets of datasets, which were training and test datasets. Both datasets have been selected from the complete user profile dataset and both has 1000 instances and 15 (A=15) attributes respectively. It is also worth mentioning that the content of both datasets are different from the ones which were used in [8].

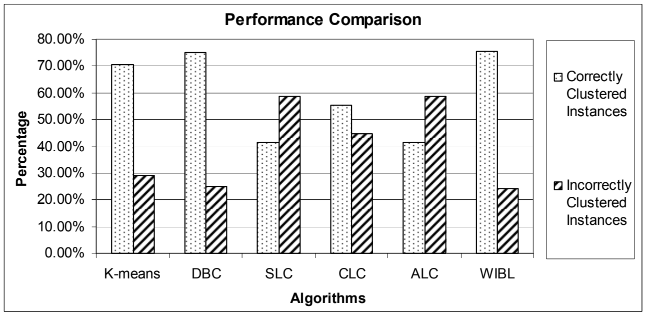


Figure 3. Performance comparison of the algorithms

Table I
Performance comparison of the algorithms on user profile dataset where N=1000 and M=1000

Algorithms	Correctly Clustered	Incorrectly Clustered
K-means	707(70.7%)	293(29.3%)
DBC	751(75.1%)	249(24.9%)
SLC	413(41.3%)	587(58.7%)
CLC	552(55.2%)	448(44.8%)
ALC	414(41.4%)	586(58.6%)
WIBL	756(75.6%)	244(24.4%)

Clustering algorithms were tested on the WEKA machine learning platform providing a benchmarking, consisting of a collection of popular existing learning schemes that can be used for practical data mining and machine learning applications [22].

B. Simulation Results

This subsection discusses the simulation results of SLC, CLC, ALC, K-means and WIBL, conducted on the above defined user profile dataset. The simulation parameters were set to be A = 15, Q = 5, N =1000 and M= 1000. Other simulation parameters (i.e., distance algorithm (Euclidean Distance)) were set as the default by WEKA, except the ‘number of iterations’ value for K-means, being taken as 7. Here, dissimilar training and test datasets have been used that includes information of different users. All algorithms were evaluated on the same training dataset and tested on the same test dataset, to obtain the classification/clustering accuracy results.

Table 1 and Figure 3 show the results of each simulation. From Figure 3, we can clearly see the performance comparison of the algorithms. Here, it can be seen that the lowest incorrectly clustered instance percentage, was archived by

WIBL and DBC. On the other hand, highest incorrectly clustered percentage is performed by the SLC and ALC hierarchical clustering algorithms. From the table, it can be seen that the best result is achieved by the WIBL algorithm, with 756 correctly clustered instances out of 1000 instances. The DBC follows the WIBL algorithm, with 751 correctly and 249 incorrectly clustered instances. The third best result is achieved by the K-means algorithm, with 707 correctly clustered instances. From Table 1, it can also be observed that the lowest performance was achieved by the SLC algorithm. The SLC clustered 413 instances correctly. With 414 correctly clustered instances, the ALC performs the second lowest result. Here, simulations revealed that the SLC and ALC algorithms perform similar, with user profile dataset, by clustering more than half of the instances incorrectly. The CLC algorithm performs better than the SLC and ALC algorithms by clustering 552 instances correctly.

In general, simulations showed that hierarchical clustering algorithms do not perform very well with user profiles. On the other hand, the DBC, with arbitrary clusters, gives one of the best results with user profiles. Moreover, using feature weighting to emphasize the relevancy of features during user profiling, has resulted in the WIBL achieving the highest performance among all the used algorithms.

V. Conclusion and Future Works

This paper aimed to evaluate the performance of the Weighted Instance Based Learner (WIBL) together with the well known clustering algorithms on a user profile dataset. The simulations were conducted on user profile dataset that reflects the user's demographic, interest and preferences information. Two sets of user profile dataset, training and test datasets, were used for the simulations. Here, all algorithms were trained on the same training dataset and tested on the same test dataset. According to the simulation results, Single-Linkage Clustering (SLC) has the lowest performance. The best performance, on the other hand, is achieved by the WIBL. The WIBL algorithm outperformed all the algorithms by classifying 75.6% of the instances correctly. Hence, it can be conclude that, compared to the well known clustering algorithms, with WIBL we can achieve the highest accuracy in user profiling.

This work is the first in the literature to present the comparison of classification accuracy of the WIBL and well known clustering algorithms with user profiles. Future studies could compare the performance of WIBL with well known classifiers. It would also be interesting to test and evaluate the performance of these algorithms with different real word user profile dataset.

References

[1] G. Araniti, P. D. Meo, A. Iera, and D. Ursino, “Adaptive controlling the QoS of multimedia wireless applications through user profiling techniques”, IEEE Journal on Selected Areas in Communication, 21(10), December 2003, pp. 1546-1556.

[2] S. Henczel, “Creating user profiles to improve information quality”, Factiva, 23(3), May 2004, p. 30.

[3] D. Poo, B. Chng, and J. M. Coh, “A hybrid approach for user profiling”, Annual Hawaii International Conference on System Sciences, 4(4), January 2003, pp. 1-9.

[4] A.K. Jain and R.C. Dubes, “Algorithms for clustering data” , 1st Edition, Prentice-Hall Advanced Reference Series, Prentice Hall, Inc., Upper Saddle River, NJ, 1998, pp. 1-304.

[5] D. Irani, S. Webb, and C. Pu, “Study of static classification of social spam profiles in MySpace”, International Conference on Weblogs and Social Media, May 2010, pp. 82-89.

[6] W. Paireekreng and K. W. Wong, “Client-side mobile user profile for content management using data mining techniques”, International Symposium on Natural Language Processing, October 2009, pp. 96-100.

[7] W. Paireekreng, K. W. Wong, and C. C. Fung, “A model for mobile content filtering on non-interactive recommendation systems”, IEEE International Conference on Systems, Man and Cybernetics, October 2011, pp. 2822-2827.

[8] A. Cufoglu, M. Lohi, and C. Everiss, “Weighted Instance Based Learner (WIBL) for user profiling”, IEEE International Symposium on Applied Machine Intelligence and Informatics, January 2012, pp. 201-205.

[9] M. R. Lopez, A. B. B. Martinez, A. Peleteiro, F. A. M. Fonte, and J. C. Burguillo, “MoreTourism:mobile recommendations for tourism”, IEEE International Conference on Consumer Electronics, January 2011, pp. 347-348.

[10] Y. B. Fernandez, M. L. Nores, J. J. P. Arias, J. G. Duque, and M.I.M. Vicente, “TripFromTV+:Exploiting social networks to arrange cut-price touristic packages”, IEEE International Conference on Costumer Electronics, January 2011, pp. 223-224.

[11] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukias, “Multicriteria user modelling in recommender systems”, IEEE Intelligence Systems, 26 (2), April 2011, pp. 64-76.

[12] A. B. B. Martinez et. al., “A hybrid content-based and item-based collaborative filtering to recommend TV programs enhanced with singular value decomposition”, Elsevier Information Sciences, 180(22), November 2010, pp. 4290-4311.

[13] D. Xu, Z. Wang, Y. Zhang, and P. Zong, “A Collaborative tag recommendation based on user profile”, International Conference on Intelligent Human-Machine Systems and Cybernetics, August 2012, pp. 331-334.

[14] D. W. Aha, D. Kibler, and M. K. Albert, Instance-based learning algorithms, Machine Learning journal, 1(6), 1991, pp. 37-66.

[15] M. J. A. Berry and G. S. Linoff, “Data mining techniques: for marketing, sales and customer relationship management”, 2nd Edition, Electron. Book, John Wiley and Sons, Inc. (US), 2004, pp. 11, 165-167, [Retrieved: June, 2013].

[16] M. Sushmita and A. Tinku, “Data Mining: Multimedia, soft computing and bioinformatics”, Electron. Book, Hoboken, John Wiley and Sons, Inc.(US), 2003, pp. 18-19, [Retrieved: July, 2013].

[17] A. L. Symeanidis and P. A. Mitkas, “Agent intelligence through data mining multiagent systems, artificial societies and simulated organizations; 14”, Electron. Book, New York: Springer Science and Business Media, 2005, pp. 21-23, 27-28, [Retrieved: August, 2013].

[18] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, “Cluster Analysis”, 5th Edition, John Wiley and Sons, Ltd.(London), 2011, pp. 73-78.

[19] M. Khosrowpour, “Encyclopaedia of information science and technology”, Electron. Book, Hershey, PA Idea Group Reference, 2005, pp. 2063- 2067, [Retrieved: August, 2013].

[20] J. Wang, “Encyclopaedia of data warehousing and mining”, Electron. Book, Hershey, PA Information Science Reference, 2006, p. 144, [Retrieved: June, 2013].

[21] A. Asuncian and D. J. Newman, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2007, [Retrieved: July, 2013].

[22] H. Mark et al., “The WEKA data mining software: an update”, SIGKDD Explorations, 11(1), June 2009, pp. 10-18.

Comparative Visual Analysis of Large Customer Feedback Based on Self-Organizing Sentiment Maps

Halldór Janetzko, Dominik Jäckle and Tobias Schreck
University of Konstanz
Konstanz, Germany
Email: Halldor.Janetzko@uni.kn, Dominik.Jaeckle@uni.kn, Tobias.Schreck@uni.kn

***Abstract*—Textual customer feedback data, e.g., received by surveys or incoming customer email notifications, can be a rich source of information with many applications in Customer Relationship Management (CRM). Nevertheless, to date this valuable source of information is often neglected in practice, as service managers would have to read manually through potentially large amounts of feedback text documents to extract actionable information. As in many cases, a purely manual approach is not feasible, we propose an automatic visualization technique to enable the geospatial-aware visual comparison of customer feedback. Our approach is based on integrating geospatial significance calculations, textual sentiment analysis, and visual clustering and aggregation based on Self-Organzing Maps in an interactive analysis application. Showing significant location dependencies of key concepts and sentiments expressed by the customer feedback, our approach helps to deal with large unstructured customer feedback data. We apply our technique to real-world customer feedback data in a case-study, showing the capabilities of our method by highlighting interesting findings.**

Keywords*—*customer relationship management, review analysis, self-organizing maps, sentiment analysis.

I. Introduction

Many companies with business in the world wide web collect reviews and customer feedback of their products and services. One common way of assessing customer satisfaction are grading schemes (e.g., one to five stars) and free text forms allowing more detailed customer comments. But aside from showing the average rating or the distribution of ratings, more sophisticated and consequently also more expressive analyses are performed very rarely. This is surprising, as the free text provided by customers is a valuable source of hints with respect to customer needs and satisfaction levels, but a manual inspection is often not feasible. Modern approaches of text processing and visualization can help at this end, by summarizing important themes and sentiments in large amounts of text.

An effective analysis of textual customer feedback should involve and examine different aspects of the text content. The most obvious one is the *frequency of statements or terms*. Simple statistics and visualization methods like word clouds may help to get a first impression of most important keywords. But simple statistics do not help to analyze, whether the customers liked or disliked these points. The next important aspect is the *sentiment* extracted from the context of the addressed keywords occurring in the text. E.g., customers may complain or praise products or services, and by using sentiment analysis, we aim at capturing this notion. From a company's point of view, negative statements are in many cases more important to analyze than the positive ones, to improve customer satisfaction. But the computation of one single sentiment score is not very expressive as customers might review more than one aspect, and different customers may have different opinions. Therefore, the challenge is to arrive at a fine-grained analysis in this complex data. The sentiment analysis should assign sentiment scores with respect to the attributes of the product or service, instead of computing one value. Customers, for instance, could like a certain bought product, at the same time complain about a too complicated ordering process. Yet another key aspect holding valuable information in customer feedback data is the *geospatial location*. Customer feedback can be geolocated by several ways, including having the customer address in a corporate database, or by geo-resolving the IP address an anonymous web feedback was provided. From that we can derive the geospatial distribution of customer feedback, which is important for two reasons. First, for global companies, cultural differences may influence the customers' conception and country specific products or services should be offered. Second, besides cultural differences there is another aspect which may change customer's needs. The geographic location determines for instance the climate and may also impose delivery obstacles resulting from the geographic topology. In very dry areas, for example, it may be reasonable to leave a parcel outside the customer's housing, but in rainy areas the customer may complain about a soaked product. Concerning the topology, hard to reach customers (e.g., islands or exclaves) may complain about long delivery times, but there may be nothing the company could do about it.

Our main motivation for this work was the following starting hypothesis, to be explored on a real-world CRM data set: "The geographic position of reviewing customers correlates to their satisfaction levels and needs." We wanted to see, whether there are differences in customer preferences caused by the geospatial location. The result of this analysis could help to improve the customer satisfaction by detecting differences in customer needs. Companies can therefore differentiate better among their customers and can easily focus and channel their efforts.

In this paper, we perform customer feedback analysis based on *sentiment maps*. Sentiment maps are the result of preceding opinion mining steps, where the occurrence of a term is drawn on a geographic map. The color used hereby depicts the sentiment and the sentiment map consequently shows not only the geospatial distribution of the term but simultaneously also

ISBN: 978-1-61208-311-7

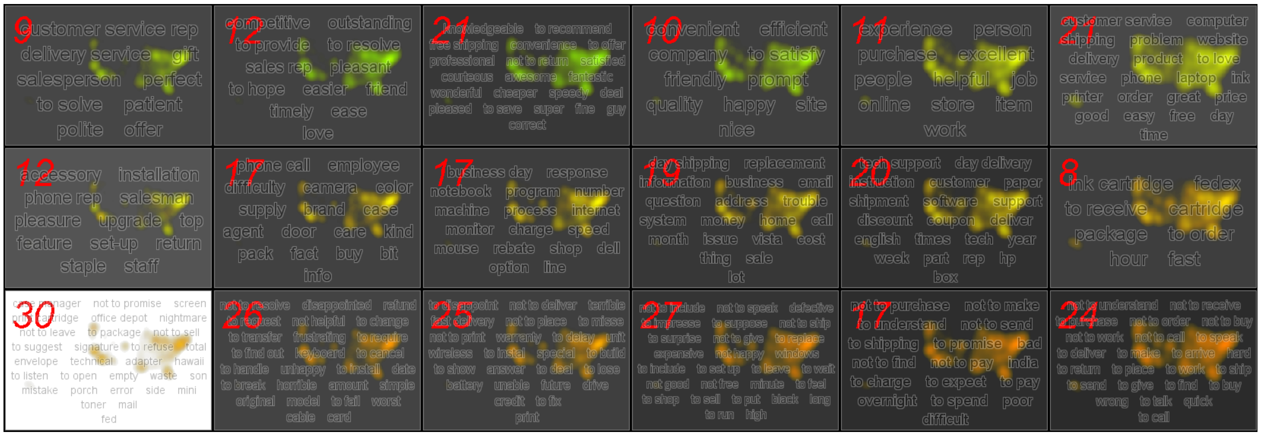

9
customer service rep
delivery service gift
salesperson perfect
to solve patient
polite offer
12
competitive outstanding
to provide to resolve
sales rep pleasant
to hope easier friend
timely ease
love
21
knowledgeable to recommend
free shipping convenience to offer
professional not to return satisfied
courteous awesome fantastic
wonderful cheaper speedy deal
pleased to save super fine guy
correct
10
convenient efficient
company to satisfy
friendly prompt
quality happy site
nice
11
experience person
purchase excellent
people helpful job
online store item
work
21
customer service computer
shipping problem website
delivery product to love
service phone laptop ink
printer order great price
good easy free day
time
12
accessory installation
phone rep salesman
pleasure upgrade top
feature set-up return
staple staff
17
phone call employee
difficulty camera color
supply brand case
agent door care kind
pack fact buy bit
info
17
business day response
notebook program number
machine process internet
monitor charge speed
mouse rebate shop dell
option line
19
day shipping replacement
information business email
question address trouble
system money home call
month issue vista cost
thing sale
lot
20
tech support day delivery
instruction customer paper
shipment software support
discount coupon deliver
english times tech year
week part rep hp
box
8
ink cartridge fedex
to receive cartridge
package to order
hour fast
30
case manager not to promise screen
print cartridge office depot nightmare
not to leave to package not to sell
to suggest signature to refuse total
envelope technical adapter hawaii
to listen to open empty waste son
mistake porch error side mini
toner mail
fed
26
not to resolve disappointed refund
to request not helpful to change
to transfer frustrating to require
to find out keyboard to cancel
to handle unhappy to install date
to break horrible amount simple
original model to fail worst
cable card
25
to disappoint not to deliver terrible
fast delivery not to place to misse
not to print warranty to delay unit
wireless to instal special to build
to show answer to deal to lose
battery unable future drive
credit to fix
print
27
not to include not to speak defective
to impresse to suppose not to ship
to surprise not to give to replace
expensive not happy windows
to include to set up to leave to wait
not good not free minute to feel
to shop to sell to put black long
to run high
17
not to purchase not to make
to understand not to send
to shipping to promise bad
not to find not to pay india
to charge to expect to pay
overnight to spend poor
difficult
24
not to understand not to receive
to purchase not to order not to buy
not to work not to call to speak
to deliver to make to arrive hard
to return to place to work to ship
to send to give to find to buy
wrong to talk quick
to call

Fig. 1. Visualization of customer feedback data using sentiment analysis and self organizing maps. Term groups with different sentiment scores are visible and additionally the geospatial distribution of the terms is displayed. The brightness of the background represents the coherence of the geographic distribution for each SOM node.

the sentiment distribution. Following this approach leads to one sentiment map for each term. Further details of this approach can be found in the beginning of the analysis results section of this paper. A result of our technique is depicted in Figure 1.

We present in this paper our methodology analyzing customer feedback with respect to sentiment and geospatial customer location. Our contributions are the combined text and geospatial analysis of customer feedback data and the visual representation allowing a comparative analysis. Furthermore, we show that there are indeed frequent feedback terms (concepts) with a high geospatial dependency. The paper is structured as follows. First, we will give an overview to existing and related work in section II, and then detail our approach in section III. Findings from an application to a real-world data set will be discussed in section IV. We will conclude with an outlook to future work.

II. Related Work

Our work relates to a number of areas, which we briefly review in the next paragraphs.

Self-Organizing Maps for Visual Data Analysis. Many problems in visual data analysis require the reduction of data to perform meaningful analysis on a reduced version of data. Clustering reduces the data to a smaller number of groups to more easily analyze and compare; and dimensionality reduction reduces the number of dimensions of data items to consider, and to project data to 2D displays. The Self-Organizing Map (SOM) algorithm [1] is a well-known method, which provides both data reduction and projection in an integrated framework. As a neural-network type method it learns a set of prototype vectors arranged on a regular grid, typically embedded in 2D. The method typically provides robust results in both data clustering and 2D layouting. Using regular 2D grids as neural structures for the SOM training, visualization in form of heatmaps, component planes, and distance distributions comprise basic methods for visual exploration of data using SOM processing [2]. SOM-based Visual Analysis to date has considered different application domains, including financial data analysis based on multivariate data models [3], analysis of web clickstream data using Markov Chain models [4], trajectory-oriented data [5], or time-oriented data [6]. Image Sorter [7] proposed to visually analyze collections of images by training a SOM over color features extracted from the images. We here follow that idea, in that we analyze geospatial heatmaps of sentiment scores using SOM of respective color features as well.

SOM-Based Visual Analysis of Geospatial Data. Many application problems involve georeferenced data items, and visual analysis approaches have been identified as very helpful also for geospatial data analysis processes [8]. Choropleth (or thematic) maps are a basic, popular technique to show the distribution of a scalar value over a land-covering map [9]. Also, SOM-based approaches have been studied in context of geospatial data analysis, and proven useful to this end. When considering georeferenced data with SOM, basically two approaches exist. First, in the *joint data model*, one single data representation is formed by combining spatial and other multivariate data into a single vector representation which is input to the SOM method. Examples include [10], where a joint vector representation for both geolocation and demographic data was formed for census data analysis. More methods can be found in [11]. As a second approach, *linked views* integrate visual data analysis of each data aspect (geolocation, time, multivariate measures, etc.) in separate views combined by linking and brushing. One example system is [12], where a linked view system proposed the joint visual analysis of geospatial and multivariate data. Also, in [13], we proposed to jointly analyze geospatial and temporal phenomena by a linked view. There, SOM clusters can be computed for either data perspective, and the correspondence of clusters to the other perspective is shown by an auxiliary view. In our approach we do not consider geolocation data explicit for the SOM generation, but implicitly by the spatial-sensitive color features extracted from sentiment heatmaps generated from text data (cf. also Section III for details).

Feature-based Text Visualization. Finally, we relate to a body of work in visual document analysis. In general, *feature-based document analysis* abstracts a document (or collection, or stream) by a set of features which are more easy to visualize, as compared to the content of the documents. Numerous document features for different applications have been studied

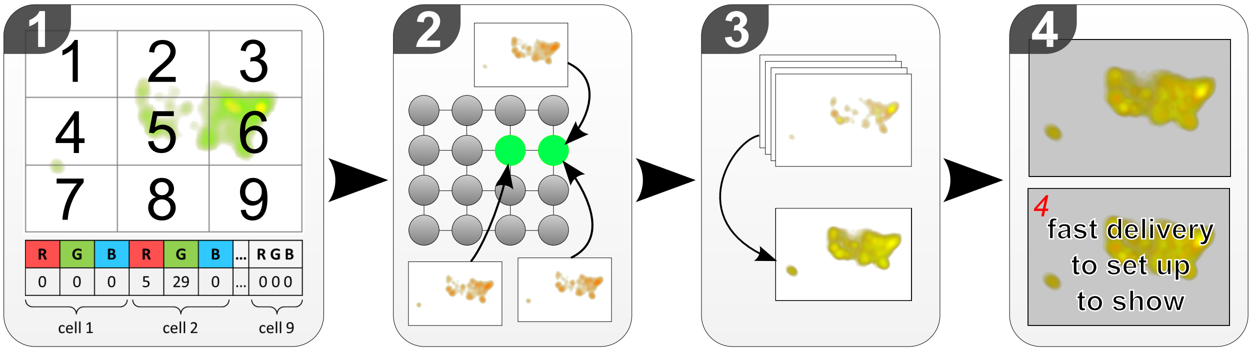


Fig. 2. Process pipeline to create self-organizing sentiment maps. The self-organizing sentiment map is determined by (1) extracting corresponding feature vectors, (2) calculating the best matching unit, (3) aggregating similar images according to the SOM, and finally (4) mapping the coherence and the additional information like the used terms to the images.

to date. For example, features scoring the readability of documents have been proposed in [14], and features applicable to classify authorship of documents have been surveyed in [15]. Sentiment features rate the polarity (in terms of positiveness of negativeness of statements) in a given text. In combination with time-series analysis, sentiment features can be used, e.g., to detect critical customer opinions in near real time, as possibly arising from some feedback channel [16]. In [17], we applied sentiment analysis to customer feedback data and analyzed it by means of geospatial heatmaps generated for the sentiments. While in [17], we considered only small sets of such heatmaps which we sequentially inspected, the focus of our work here is the comparative analysis of large numbers of sentiment maps, based on the SOM method.

III. Technique

Our approach enables the geospatial visual comparison of customer feedback sentiments by using a Self-Organizing overview display. Figure 2 shows the overall process that is divided into four steps: (1) First, we extract a color feature vector for each sentiment map. (2) Second, we train the SOM and assign every sentiment map exactly one node. In step (3) we aggregate all sentiment maps that are located on the same map node. (4) Finally, we calculate the coherence and enhance the aggregated sentiment map with the content terms from the represented customer review texts. We next detail these steps.

Feature Vector Extraction. The feature vector we use as input to the SOM computation consists of localized RGB color values. We create a grid overlay for each sentiment map and calculate the color mean value for each cell. The mean value is determined by the color value of each single pixel contained in the corresponding grid cell. The representative feature vector for any sentiment map is created using the RGB color model. All RGB mean values are forming the feature vector:

$$\text{Picture1} = (R_{1,1}\, G_{1,1}\, B_{1,1}\, R_{1,2}\, G_{1,2}\, B_{1,2}\, R_{1,3}\, G_{1,3}\, B_{1,3} \ldots)$$

$R_{i,j}$ represents the value of R for picture i and cell j. This format is used as feature vector representing one sentiment map; each picture is assigned exactly one vector. Then, the extracted feature vectors are used to train the SOM using the SOMPAK implementation [18] (see also Figure 2 (1)).

Sentiment Map Classification. We apply a standard SOM training process following best practices suggested in [18]. Based on the defined SOM grid resolution, the prototype vectors are linearly initialized. Then, two learning phases are applied. First, a coarse learning is performed with a larger training radius, so that every considered node has a wide impact factor. Then, a fine-tuning training step is performed with a smaller training radius. Once the SOM-training has finished, the best matching prototype vector on the SOM grid (best matching unit, or bmu) is determined for each sentiment map by finding the node with the minimal distance (1).

$$bmu(SM) = \min_{k=1}^{M} \left(\sqrt{\sum_{i=1}^{N} (v(SM).i - v(node_k).i)^2} \right) \quad (1)$$

We iterate all sentiment maps and calculate the best matching unit for each sentiment map SM. Then, we iterate all M trained SOM nodes and calculate the minimal Euclidean distance between the sentiment map and the trained SOM node. Therefore, the feature vector of the sentiment map and the vector of the SOM node are used. The corresponding vector is determined via the function $v()$ with size N. The control variable i addresses every single vector entry. Finally, the sentiment map is assigned to the SOM node with the minimal distance (see also Figure 2 (2)). The grid size can be chosen individually for each application.

Similarity-based Sentiment Map Aggregation. As the outcome of the SOM and bmu mapping, multiple sentiment images may share the same SOM node. Therefore, we need to provide aggregation of such sets of maps. To find a representative image for those sentiment maps different approaches are possible. We here chose to apply visual aggregation and merge all similar images into one. Therefore, every sentiment map is assigned a transparency value, so that we are able to create one image by lying one sentiment map upon each other. The resulting image visualizes all aggregated sentiment areas. By adding multiple pictures on top of each other, the last added picture on top has the highest impact according to the process of alpha composition in terms of occlusion [19]. For that reason, we calculate the intersection of sentiment maps on our own based on the color, shown in Figure 2 (3).

Coherence Mapping and Map Enhancement. The last step of the pipeline is twofold: First, we map the background of the aggregated sentiment map to its coherence. Second, we enhance the aggregated sentiment map with additional information.

Aggregating multiple sentiment maps may result in an image showing a constant distribution. But the single pictures might be very diverse regarding the positions where the identified sentiment was mapped to. In order to understand the composition of those aggregated sentiment clusters it is important to define a quality criterion: the coherence of the sentiment maps. Thus, we make use of the background and define a coherence measure. The coherence measure (2) expresses how similar two sentiment maps are according to its feature vector. The coherence is mapped to the color range from black (high coherence) to white (very low coherence).

$$coherence(SMS) = \frac{\sum_{i=1}^{N}\sum_{j=i}^{N} dist(SMS_i, SMS_j)}{\frac{N \cdot (N+1)}{2}} \qquad (2)$$

The $coherence$ is calculated for N sentiment maps (SMS) addressing the same SOM node. Summarizing, we build the average of all pictures including a distance function. We then sum the distance value of each sentiment map to all other sentiment maps. The distance function $dist$ between two sentiment maps is defined in equation (3).

$$dist(p,q) = \frac{\sqrt{\sum_{k=1}^{M}(v(p).k - v(q).k)^2}}{|\{i \in 1..M : \neg(v(p).i = v(q).i = 0)\}|} \qquad (3)$$

The distance function requires two sentiment maps as parameters with dimension M. The Euclidean distance is normalized by the number of vector dimensions M excluding all dimensions with zero values in both dimensions. The problem of a possible low coherence raises with the algorithm of the SOM: To calculate the similarity in the second step the Euclidean distance combines sentiment maps that are very sparse, as the exact locations do not matter.

Sentiment Keyword Visualization. Every sentiment map corresponds to one term. As a consequence, if multiple sentiment maps are aggregated, the resulting image corresponds to multiple terms. Hence, we combine the aggregated sentiment maps with a simple but effective text representation: All terms are drawn semi-transparent with a gray border on top of the aggregated image. Also, the amount of sentiment maps that have been aggregated is indicated by a red number on the top left corner. Using an intelligent text layout algorithm, the analyst can easily identify the terms corresponding to the image; the text uses the full width and height to be easy to read. Figure 2 (4) illustrates the automatic labeling result.

Depending on the chosen grid size in the first step (feature vector extraction), the final result may differ. To allow data abstraction and overview large data sets, we typically chose a relatively small grid size, where the amounts of nodes is significantly smaller than the amount of considered sentiment maps.

IV. Analysis Results

We applied our methods described above to sentiment maps of a real-world data set of collected customer reviews. The reviews were collected after online purchases via an online survey. The data set consists out of 86.812 customer reviews with an average of 18.4 words per review (the median is 12 words per review). In this section, we will first describe the input images resulting from a technique called sentiment maps more in detail. Afterwards, we will discuss interesting findings with respect to the geographic distribution of frequently reported review terms.

Sentiment Maps. Sentiment maps allow the user to inspect the geospatial sentiment distribution of individual terms and are introduced in [17]. After collecting all terms of all reviews excluding stop words one visualization for each of these terms is created. More specifically, first all occurrences of the respective term are determined and the sentiment value for these occurrences are retrieved. The data is then used to generate the sentiment map as illustrated in Figure 3. The data is first partitioned into two subsets: the occurrences with positive sentiment in Figure 3(a) and occurrences with negative sentiment in Figure 3(c). The two partitions are processed separately. A Gaussian blurring function is applied in order to spatially extend the occurrences and increase the visual salience of the geospatial distribution patterns. The result is a blurred representation for both sentiments showing the respective geospatial occurrences as depicted in Figures 3(b) and 3(d). Finally, a combined image is created by using the RGB channels of the RGB color model. The blurred image of the negative occurrences is put in the red channel and the green channel is used for the positive occurrences. Consequently, locations with both positive and negative sentiments will result in yellow colors, while pure positive sentiments will result in green colors. We did not differentiate between within negative sentiments or positive sentiments respectively as this differentiation is highly user and application dependent. But sentiment maps could be extended by this possibility. The final result of this technique can be seen in Figure 3(e).

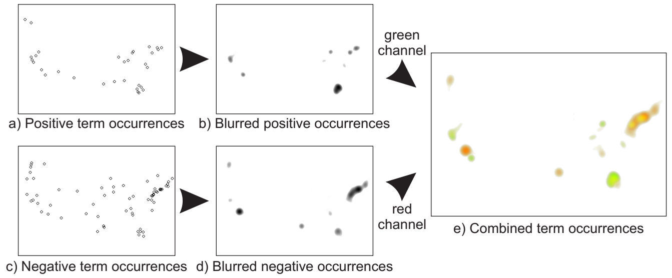

green channel
a) Positive term occurrences
b) Blurred positive occurrences
red channel
e) Combined term occurrences
c) Negative term occurrences
d) Blurred negative occurrences

Fig. 3. The location maps of positive (a) and negative (c) review terms are blurred to increase the visual saliency and to give a visual aggregation (b,d). Both blurred location maps are then combined (e) by using the RGB channels and show the distribution of positive (green), neutral (yellow) and negative (red) term occurrences.

Discussion of Findings. We applied the technique described in section III to a dataset consisting of 327 sentiment maps. These terms were found in preceding document mining steps and contains the words being nouns, verbs, and adjectives. Note that some of these terms are even compound nouns like ”phone call“ or negated verbs like ”not to send“. The resulting overview visualization can be seen in Figure 4.

On an abstract level there is a clear grouping and ordering of the sentiment maps visible. Terms with only negative

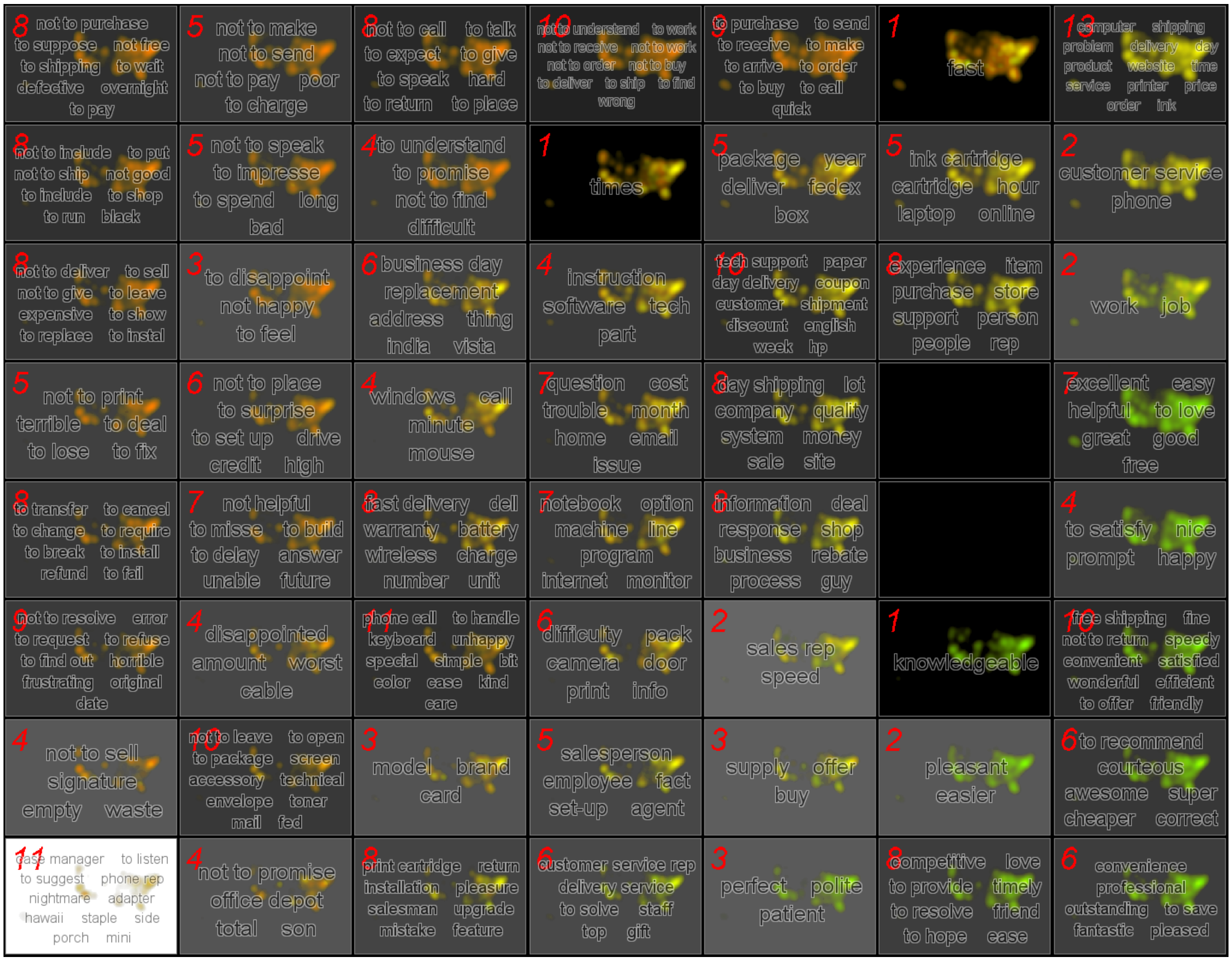


Fig. 4. Resulting overview visualization of the SOM analysis process. There are SOM nodes like the one in the lower left being highlighted with a bright background showing a high internal diversity.

occurrences (reddish images) are located in the upper left while positive terms (greenish sentiment maps) are located in the lower right. The first diagonal consists of terms being either mentioned negatively and positively equally often (upper right) and terms with a geospatial, diverse distribution (lower left). The SOM analysis enables the analyst to get a fast overview over terms being mentioned always positive or negative.

The strongly highlighted, white node of Figure 4 in the lower left contains eleven terms showing a very diverse geospatial distribution. As this is the node being highlighted most we will investigate this node in the following paragraphs. Detailed analysis via drill-down techniques are possible in our system and reveal the geospatial distribution for each single term. The visualization of all eleven contained terms is depicted in Figure 5.

Inspecting the sentiment maps more in detail reveals that this node mainly contains sentiment maps with sparse and diverse geospatial distributions.

The most obvious sentiment map contained in this SOM node is the term ”hawaii“. It is occurring mostly positively and collocated with the geospatial position of the Hawaiian islands. Inspecting the customer comments in detail, we found that customers liked the free shipping possibilities to Hawaii,

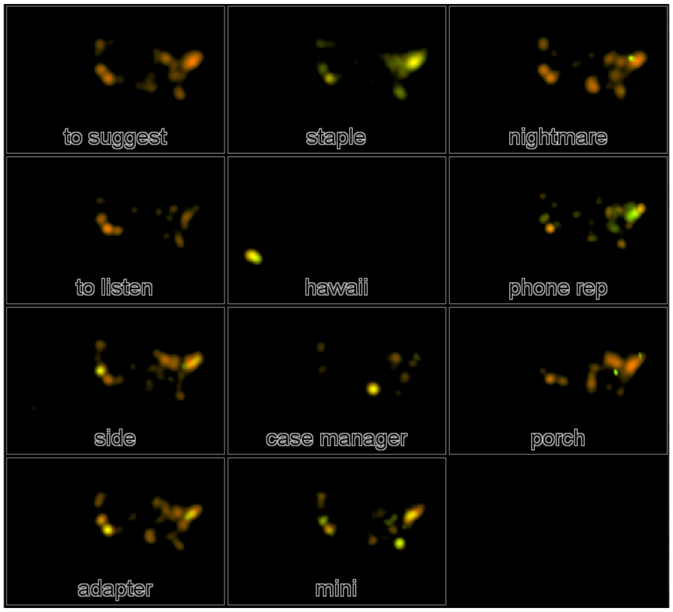


Fig. 5. Visual representation of all sentiment maps contained in the lower left node of Figure 4.

ISBN: 978-1-61208-311-7

which seems not to be taken for granted. Service managers can learn from this information that (Hawaiian) customers do care about the shipping procedure and that free shipping might be an advantage over competitors.

Also, the term ”case manager“ (third row, second column in Figure 5) shows an interesting pattern. Although mostly mentioned negatively because of language issues – the customer support was hard to understand because of foreign accents – there are many positive occurrences in Houston, Texas where customers liked the support regarding their printers. Service managers should now investigate further what the characteristics about the problems in Houston were.

Two further interesting sentiment maps are the ones of ”nightmare“ and ”porch“. Investigating the underlying reviews shows that the preceding sentiment analysis did not work correctly as all the reviews were purely negative. This is not a drawback of the method per se, but exemplifies the uncertainty of any sentiment analysis and the sensitivity of our method to the input data. The comments regarding the term ”porch“ were mentioning that the parcel was left unattended on the porch. The term ”nightmare“ was used in cases where the process of ordering and returning products did not go smoothly.

V. Conclusion

We presented our approach to visually compare and inspect large sets of textual customer feedback with respect to sentiment expressed regarding key concepts, and geographic distribution. For each concept, a sentiment map was rendered, and set of all maps was visually clustered and aggregated by the SOM approach. Interaction methods allow to navigate the overview visualizations and drill down for detailed inspection and relation of feedback topics in geospatial context. Application findings presented indicate that key concepts and their sentiment scores being highly dependent on geographic position. Such findings can be very helpful in analyzing service levels across locations, products, and customers, and similar applications in CRM.

We have several ideas to extend our work in future for improved analysis. One possibility improving the visual representation is the integration of semantic zoom approaches. Semantic zoom can allow to merge neighboring SOM nodes to reduce the level of detail. Additionally, semantic zoom can be applied to the shown terms by using an ontology grouping terms, showing only the common parent of a set of related concepts. The ontology also leads to another extension possibility we are going to integrate in future. We plan to show the hierarchic relationships between terms directly on the SOM representation. Last but not least, we want to consider more detailed map visualizations concerning production facilities and income distributions among different cities correlating geospatial dependent properties with the text features.

Acknowledgment

This work has been supported by the Consensus project and has been partly funded by the European Commission’s 7th Framework Programme through theme ICT-2013.5.4 ICT for Governance and Policy Modelling under contract no.611688.

We are thankful to Ming Hao of Hewlett-Packard Labs for fruitful collaboration on the analysis of geo-referenced customer review report data. We also thank the Helsinki University of Technology for providing the SOMPAK implementation which we used in this study.

References

[1] T. Kohonen, *Self-Organizing Maps*. Springer-Verlag, 2001.

[2] J. Vesanto, “SOM-based data visualization methods,” *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111–126, 1999.

[3] G. J. Deboeck and T. K. Kohonen, Eds., *Visual Explorations in Finance: with Self-Organizing Maps*. Springer, 1998.

[4] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma, “Visual cluster exploration of web clickstream data,” in *IEEE VAST*, 2012, pp. 3–12.

[5] T. Schreck, J. Bernard, T. von Landesberger, and J. Kohlhammer, “Visual cluster analysis of trajectory data with interactive kohonen maps,” *Information Visualization*, vol. 8, no. 1, pp. 14–29, 2009. [Online]. Available: http://dx.doi.org/10.1057/ivs.2008.29

[6] J. Bernard, J. Brase, D. Fellner, O. Koepler, J. Kohlhammer, T. Ruppert, T. Schreck, and I. Sens, “A visual digital library approach for time-oriented scientific primary data,” *Springer International Journal of Digital Libraries, ECDL 2010 Special Issue*, pp. 111–123, 2011.

[7] K. U. Barthel, “Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics,” *Image Analysis for Multimedia Interactive Services, International Workshop on*, vol. 0, pp. 227–230, 2008.

[8] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag, 2005.

[9] A. M. MacEachren, *How Maps Work - Representation, Visualization, and Design*. Guilford Press, 2004.

[10] F. Bao, V. Lobo, and M. Painho, “The self-organizing map, the geo-som, and relevant variants for geosciences,” *Computers & Geosciences*, vol. 31, no. 2, pp. 155 – 163, 2005, geospatial Research in Europe: AGILE 2003. [Online]. Available: http://www.sciencedirect.com/science/article/B6V7D-4F0851H-2/2/ac6bd99e54c5ff81db9a173604b0d3aa

[11] P. Agarwal and A. Skupin, Eds., *Self-Organising Maps: Applications in Geographic Information Science*. Wiley, 2008.

[12] D. Guo, J. Chen, A. M. MacEachren, and K. Liao, “A visualization system for space-time and multivariate patterns (VIS-STAMP),” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1461–1474, 2006.

[13] G. Andrienko, N. Andrienko, S. Bremm, T. Schreck, T. von Landesberger, P. Bak, and D. A. Keim, “Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns,” *Computer Graphics Forum*, vol. 29, no. 3, pp. 913–92, 2010.

[14] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim, “Visual Readability Analysis: How to Make Your Writings Easier to Read,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 5, pp. 662–674, May 2012.

[15] P. Juola, “Authorship attribution,” *Found. Trends Inf. Retr.*, vol. 1, no. 3, pp. 233–334, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1561/1500000005

[16] C. Rohrdantz, M. C. Hao, U. Dayal, L.-E. Haug, and D. A. Keim, “Feature-based Visual Sentiment Analysis of Text Document Streams,” *ACM Transactions on Intelligent Systems and Technology, Special Issue on Intelligent Visual Interfaces for Text Analysis*, vol. 3, no. 2, pp. 26:1–26:25, 2012.

[17] M. C. Hao, C. Rohrdantz, H. Janetzko, D. A. Keim, U. Dayal, L.-E. Haug, M. Hsu, and F. Stoffel, “Visual sentiment analysis of customer feedback streams using geo-temporal term associations,” *Information Visualization*, Jun. 2013.

[18] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen, “SOM_PAK: The self-organizing map program package,” Helsinki University of Technology, Tech. Rep. A31, 1996.

[19] T. Porter and T. Duff, “Compositing digital images,” in *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’84. New York, NY, USA: ACM, 1984, pp. 253–259. [Online]. Available: http://doi.acm.org/10.1145/800031.808606

Multicore Mining of Correlated Patterns

Christian Ernst
Ecole des Mines de St Etienne
CMP - Site Georges Charpak
Gardanne, France
ernst@emse.fr

Alain Casali
Aix-Marseille Universit,
CNRS, LIF UMR 7279,
Marseille, France
alain.casali@lif.univ-mrs.fr

***Abstract*—We present a new approach related to the discovery of correlated patterns based on the use of multicore architectures. Our work rests on a full Knowledge Discovery in Databases system allowing one to extract Decision Correlation Rules based on the Chi-squared *criterion* from any database that includes a target column. We use a levelwise algorithm as well as contingency vectors, an alternate and more powerful representation of contingency tables. The goal is to parallelize the extraction of relevant rules by invoking the Parallel Patterns Library which allows a simultaneous access to the whole available cores on modern computers. We finally present first results and performance gains.**

Keywords*—Data Mining, Decision Correlation Rule, Multicore Architecture, Parallel Pattern Library.

I. Introduction and Motivation

Innovations in multicore architectures have begun to allow parallelization on inexpensive desktop computers. Many standard software products will soon be based on recent parallel computing concepts implemented on such hardware. Consequently, there is a growing interest in the field of parallel data mining algorithms, especially in Association Rules Mining (ARM). By exploiting multicore architectures, parallel algorithms may improve both execution time and memory requirement issues, two main objectives of ARM.

Independently of this framework, we developed a Knowledge Discovery in Databases (KDD) system based on the discovery of Decision Correlation Rules (DCRs) with large and specialized databases [1]. The rules are functional in semiconductor fabs: the goal is to discover the parameters that have the most impact on a specific parameter, the yield of a given product. DCRs are close to Association Rules, but present huge technical differences. After implementing DCRs using “conventional sequential algorithms”, we adapted our approach to multicore implementation issues.

This paper is organized as follows: In Section II, we expose current aspects of multicore programming. Section III is dedicated to related work: we present (i) an overview of ARM over a multicore architecture and (ii) what DCRs are. Section IV describes the concepts used for multicore decision rules mining and our algorithms. In Section V, we show first results of experiments. Last Section summarizes our contribution and outlines some research perspectives.

II. New features in multicore programming

In the last two decades, parallelization on personal computers has consisted to develop multithreaded code layers. What required complex co-ordination of threads, due to the interweaving of shared data processing. Although threaded applications added limited performance on single-processor machines, the extra overhead of development has been difficult to justify. But with Intel and AMD introducing commercially multicore chips in 2005, non exploiting the resources provided by multiple cores will now quickly reach performance ceilings. At that last date, no simple software environment able to take advantage of the different processors have been simultaneously proposed. New opportunities appeared in 2010, presented in the C++ language used in our developments.

Multicore processing influenced actual computational software development. Many modern languages do not support multicore functionality. Therefore, different conceptual models deal with the problem, such as using a coordination language (programming libraries and/or higher order functions). Users program using these abstractions, and an “intelligent” compiler then chooses the best implementation based on the context [2]. Cilk++, OpenMP, OpenHMPP, TBB, *etc.*, are examples of such models having been recently proposed for use on multicore platforms. A comparison of some OpenX approaches can be found in [3]. Nevertheless, the majority of these models rests on an intelligent transformation of general code into multithreaded code.

A novel albeit simple idea was proposed by the Open Multiprocessing consortium in 1997, based on the fact that looping functions are the key area where splitting parts of a loop across all available hardware resources may increase application performance. The OpenMP Architecture Review Board became an API that supports shared memory multiprocessing programming now also in C++. It consists of a set of compiler directives, library routines, and environment variables that influence run-time behavior. In order to schedule a loop across multiple threads, the OpenMP `pragma` directives were introduced in 2005 to explicitly relay to the compiler more about the transformations and optimizations that should take place. To illustrate our purpose, we compute in parallel an approximation of the value of π using a Riemann Zeta function ($\pi^2/6 = 1/1^2 + 1/2^2 + 1/3^2 + ...$, see Listing 1):

```
double pi2 = 0.0;
#pragma omp for
for (int i = 1; i < 1000000; i++) {
 #pragma omp atomic
 pi2 += 6.0 / (i * i);
}
```

Listing 1: Computing π using basic multithreaded parallelization

The first directive requests that the *for* loop should be executed on multiple threads, while the second is used to prevent multiple simultaneous writes to the *pi2* variable.

The example also shows the limits of parallelism. It is widely agreed that applications that may benefit from using more than one processor necessitate (i) operations that require a substantial amount of processor time, measured in seconds rather than milliseconds and (ii) one or more loops of some kind, or operations that can be divided into discrete but significant units of calculation that can be executed independently of one another. So the chosen example with a single instruction at each iteration does not fit parallelization, but is used nevertheless to illustrate in a simple way the new features introduced by actual multicore programming techniques.

These were first developed by Microsoft through an own "parallel" approach in the 2000s. Since 2010, and the relevant versions of the .NET framework and Visual Studio, Microsoft enhanced support for parallel programming by providing a runtime tool and a class library among other utilities. The library is composed of two parts: Parallel LINQ (PLINQ), a concurrent query execution engine, and Task Parallel Library (TPL), a task parallelism component of the .NET framework. What is particularly advanced is that this component entirely hides the multithreading activity on the cores: the job of spawning and terminating threads, as well as scaling the number of threads according to the number of available cores, is done by the library itself. The main concept is here a Task, which can be executed independently.

The Parallel Patterns Library (PPL) is the corresponding available tool in the Visual C++ environment, and is defined within the Concurrency namespace. The PPL operates on small units of work (Tasks), each of them being defined by what is called a λ expression (see below). The PPL defines almost three kinds of facilities for parallel processing: (i) algorithm templates for parallel operations, (ii) class templates for managing shared resources, and (iii) class templates for managing and grouping parallel tasks.

Listing 2 rewrites our example using the *parallel_for* algorithm, equivalent to a *for* loop that executes loop iteration in parallel on multiple cores:

```
float pi2 = 0;
parallel_for(1, 1000000, [&pi2](long n)
 {
   // share <pi2> between the cores
   pi2 += 6.0 / (n * n);
 }
);
```

Listing 2: Computing π using multi-core parallelization

The PPL also proposes the *parallel_for_each* algorithm (for repeated operations on a STL container), and the *parallel_invoke* algorithm (which executes a set of two or more independent Tasks in parallel).

As mentioned when discussing the OpenMP `pragma` directives, if the computation on each iteration in the *parallel_for* is very short and it is the case here, there will be important overhead in allocating the task to a core on each iteration, which may severely erode any reduction in execution times. This will also be the case if the overall loop integrates important shared resources management, as will be shown in Section IV.

The second main novelty introduced by the PPL is the use of λ expressions: A computational model invented by Alonzo Church in the 1930s, which directly inspired both the syntax and the semantics of most functional programming languages [4]. The λ calculus in its most basic form has two operations: (i) Abstractions, which correspond to anonymous functions, and (ii) Applications, which exist to apply the function. Anonymous functions are often called "lambdas", "lambda functions" or "lambda expressions": They remove all need for scaffolding code, allowing a predicate function to be defined in-line in another statement.

The syntax of a λ function is reasonably straight-forward, of the form:

```
[lambda-capture] (parameter-list) {->} return-type
    {statement-list}
```

In our example (Listing 2), the element of the lambda in the square brackets is called the capture specification: It relays to the compiler that a lambda function is being created and that the local variable *pi2* is being captured by reference. The final part is the function body.

Lambdas behave like function objects (as did previously functors), except for that we cannot access the class that is generated to implement a lambda in any way other than using the lambda. Consequently, any function that accepts functors as arguments will accept lambdas, but any function only accepting function pointers will not.

These and many other features of λ functions have been included in the C++11 language norm, allowing a more declarative programming style, taking for example advantage of STL algorithms in a much streamlined and cleaner form. λ functions allow the inline definition of a function body in the code section in which it is to be logically used. As well as providing strong hints to the compiler about potential real time optimizations, λ functions make discerning the intent about what a section of code is doing much easier.

III. Related Work

Due to the variety of the algorithms (and their specific internal data structures) no general model allowing parallel ARM computation exists. Main techniques based on A-Priori algorithms [5] are described in Section 3.A. Other multicore ARM approaches are based either on vertical mining [6] or on FP-Growth [7]. Each model consists in an multicore optimized architecture built upon specific thread managers [4]. Finally, Section 3.B presents the main results about what Decision Correlation Rules are and how can we compute them using a single processor.

A. A-Priori based algorithms

Most of the parallel ARM algorithms are based on parallelization of A-Priori that iteratively generates and tests candidate itemsets from length 1 to k until no more frequent itemsets are found. These algorithms can be categorized into *Count Distribution*, *Data Distribution* and *Candidate Distribution* methods [8]. The *Count Distribution* method divides the database into horizontal partitions, that are independently scanned, in order to obtain the local counts of all candidates on each process. At the end of each iteration, the local counts are summed up a into the global counts so that frequent itemsets can be found. The *Data Distribution* method partitions both the database and the candidate itemsets in the main memory of parallel machines. Since each candidate is counted by only one process, all processes have to exchange database partitions during each iteration in order, for each process, to obtain the global counts of the assigned candidate itemsets. The *Candidate Distribution* method also partitions candidate itemsets but replicates, instead of partitioning and exchanging, the database transactions. Thus, each process can proceed independently.

B. Decision Correlation Rules

Brin et al. [9] have proposed the extraction of correlation rules using the Chi-Squared (χ^2) statistic instead of the support and the confidence measures. The χ^2 (i) is a more significant measure in a statistical way than an association rule, (ii) takes into account the presence but also the absence of the items and (iii) is non-directional, highlighting thus more complex existing links than implications. A correlation rule is represented by an itemset.

Let r be a binary relation over a set of items $\mathcal{R} = \mathcal{I} \cup T$. $\mathcal{I}$ represents the items of the binary relation used as analysis *criteria* and $\mathcal{T}$ is a target attribute which may not necessarily have a value. The computation of the value for the χ^2 function for an item $X \subseteq \mathcal{R}$ is based on its contingency table. In order to simplify the notation, we introduce, in a first step, the lattice of the literalsets associated with $X \subseteq \mathcal{R}$. This set of cardinality $|X|$ contains all the literalsets that can be built up given X.

Definition 1 (Literalset Lattice): Let $X \subseteq \mathcal{R}$ be a pattern, we denote by $\mathbb{P}(X)$ the literalset lattice associated with X. This set is defined as follows: $\mathbb{P}(X) = \{Y\overline{Z}$ such that $X = Y \cup Z$ and $Y \cap Z = \emptyset\} = \{Y\overline{Z}$ such that $Y \subseteq X$ and $Z = X \backslash Y\}$.

Definition 2 (Contingency Table): For a given pattern X, its contingency table, noted $CT(X)$, is a $2^{|X|}$ matrix. Each cell yields the support of a literalset $Y\overline{Z} \in \mathbb{P}(X)$: the number of transactions including Y and containing no 1-item of Z.

In order to compute the value of the χ^2 function for a pattern X, we apply the following formula:

$$\chi^2(X) = \sum_{Y\overline{Z} \in \mathbb{P}(X)} \frac{(Supp(Y\overline{Z}) - E(Y\overline{Z}))^2}{E(Y\overline{Z})} \qquad (1)$$

Brin et al. [9] have shown that there is a single degree of freedom between the items. A table giving the centile values in function of the χ^2 value for X can be used in order to obtain the correlation rate for X.

Definition 3 (Correlation Rule): Let $MinCor$ (≥ 0) be a given threshold and $X \subseteq \mathcal{R}$ a pattern. If $\chi^2(X) \geq MinCor$, then X is a valid correlation rule. If X contains an item of $\mathcal{T}$, then the obtained rule is called a Decision Correlation Rule (DCR).

Moreover, in addition to the previous constraint, the Cochran *criteria* [10] are used to evaluate whether a correlation rule is semantically valid: all literalsets of a contingency table must have an expectation value different to zero and 80% of them must have a support larger than 5%. This last *criterium* has been generalized as follows: $MinPerc$ of the literalsets of a contingency table must have a support larger than $MinSup$, where $MinPerc$ and $MinSup$ are given thresholds.

Definition 4 (Equivalence Class): We denote by $[Y\overline{Z}]$ the equivalence class associated with the literal $Y\overline{Z}$: it contains the set of transaction identifiers including Y and containing no value of Z ($i.e.$, $[Y\overline{Z}] = \{i \in Tid(r)$ such that $Y \subseteq Tid(i)$ and $Z \cap Tid(i) = \emptyset\}$).

Definition 5 (Contingency Vector): Let $X \subseteq \mathcal{R}$ be a pattern. The contingency vector of X, denoted $CV(X)$, groups the set of the literalset equivalence classes belonging to $\mathbb{P}(X)$ ordered according to the lectic order.

Since the union of the equivalence classes $[Y\overline{Z}]$ of the literalset lattice associated with X is a partition of the Tids, we ensure that a single transaction identifier belongs only to one single equivalence class. Consequently, for a given pattern X, its contingency vector is an exact representation of its contingency table. To derive the contingency table from a contingency vector, it is sufficient to compute the cardinality of each of its equivalence classes. The following proposition shows how to compute the CV of the $X \cup A$ pattern given the CV of X and the set of Tids containing pattern A.

Proposition 1: Let $X \subseteq \mathcal{R}$ be a pattern and $A \in \mathcal{R}\backslash X$ a 1-item. The contingency vector of the $X \cup A$ pattern can be computed given the CVs of X and A as follows:

$$CV(X \cup A) = (CV(X) \cap [\overline{A}]) \cup (CV(X) \cap [A]) \qquad (2)$$

In order to mine DCRs, we have proposed [1] the LHS-Chi2 algorithm (see Algorithm 1) based both (i) on a double recursion in order to browse the search space according to the lectic order and (ii) on CVs.

The `CREATE_CV` function is an implementation of formula 2, while the `CtPerc` predicate checks the relaxed Cochran *criteria*.

IV. Parallel Extraction of Correlated Patterns

The development of multicore applications raises two difficulties in terms of (i) application design and (ii) shared resource management. The second aspect is rather “normal” when coping with parallelism. And will constitute the aim of this section, illustrated through specific mechanisms provided by the PPL. But application design must not be underestimated because, amongst other points, of its impact on resource management. Parallelizing existing algorithms is an important consideration as well [2]. The use of recursive

Algorithm 1: LHS-CHI2 Algorithm.

```
input  : $X$ and $Y$ two patterns
output: $\{ Z \subseteq X$ such that $\chi^2(Z) \geq MinCorr\}$
if $Y = \emptyset$ and $\exists t \in \mathcal{T} : t \in X$ and $|X| \geq 2$ and $\chi^2(X) \geq MinCorr$ then
    Output X, $\chi^2(X)$
end
$A := max(Y)$ ;
$Y := Y \backslash \{A\}$ ;
LHS-CHI2 (X,Y) ;
$Z := X \cup \{A\}$ ;
CV(Z) := CREATE_CV (CV(X),Tid(A)) ;
if $CtPerc\ (CV(Z), MinPerc, MinSup)$ and $|Z| \leq MaxCard$ then
    LHS-CHI2 (Z,Y) ;
end
```

algorithms in a multicore environment is here a sufficient challenge. This because recursion cannot be measured in terms of number of loops to perform: We first tried to replace the recursive calls by calls to appropriate threads, which quickly appeared “impossible’. Another approach was based on the well known fact that each recursive algorithm can be rewritten in a iterative way. However, the *while* loop used to run over the used stack may not be evaluated in terms of a *for* loop due to the absence of explicit boundaries.

In order to solve the problem, we recalled that we first compared our LHS-CHI2 algorithm to a LEVELWISE one, based on the same monotone and anti-monotone constraints but which did not include Contingency Vectors management. The main reason of the obtained performance gains is that pruning the search space using the lectic order is much more “elegant” than using the LEVELWISE order but has no impact nor on the results nor on the performances. On the other hand, generating the candidates at a given level is a bounded task, limited by the number of existing 1-items. So we decided to (i) use the LEVELWISE order to prune in a parallel way the search space and (ii) to keep the CVs in order to manage the constraints.

The corresponding result is presented hereafter through different functions. The overall algorithm (see Listing 3), called *PLW_Chi2*, where PLW stands for Parallel LEVELWISE, demonstrates first parallel features of our method in order to generate (and then to test) the candidates.

In order to simplify the notations, the following Listings use uc, ui, ul, us to substitute to, respectively, $unsigned\ char$, $unsigned\ int$, $unsigned\ long$ and $unsigned\ short$ standard declarations.

```
void PLW_Chi2 (us X[], us sX, us sI)
// X[] : set of computed 1-items
// sX : number of valid 1-items within X[]
// sI : total number of 1-items in X[]
{
 // number of candidates at level cl and (cl+1)
 ul cit, nit;
 cit = sX;
 for (uc cl = 2; cl <= MaxLv && cit > 0; cl++)
 {
   nit = 0L;
   T_Res aRes;
   combinable<ul> lnit;
   parallel_for(0u,(ui) cit,[cl, X, sX, &aRes, &lnit](int i)
    {
     dowork_level (cl, X, sX, i, sI, &aRes);
     lnit.local() += aRes.nit;  // ...
    });
   nit = lnit.combine(plus<ul >());
   // ...
   cit = nit;
   update_shared_resources ();
 }
}
```

Listing 3: The simplified PLW_CHI2 method

Parallelization takes place at each *cl* level of the LEVELWISE search algorithm. The number of launched Tasks at level *cl* directly depends of the number of existing candidates at level *(cl - 1)*, *e.g.*, *cit*. Each Task corresponds to a call to the *dowork_level ()* function, which performs the work it is intended to do (see below), and collects some statistics during the call through the *aRes* object. Let us mention here that database access is performed through global objects.

In this paragraph, we only focus on the signification of a particular statistic, the *lnit* member of the *aRes* object: It sums the number of discovered candidates to be examined at the next level. Because each Task computes its own candidates for the next level, the method has to pay attention to the possible interference which could take place during the overall parallel computation on such a “shared” variable, which can be seen as an aggregation pattern. A two-phase approach is therefore used: First, partial results are locally computed on a per-Task basis. Then, once all of the per-Task partial results are at disposal, the results are sequentially merged into one final accumulated value. The PPL provides the *combinable* class data structure that creates per-Task local results in parallel, and merge them as a final sequential step. In the above code, the final accumulated object is the *lnit* object, which decomposes into local to each Task *lnit.local()* sub-objects. After the *parallel_for* loop achieves, the final sum is produced by invoking the *combine()* method on the global object.

Listing 4 partially shows the implementaton of the *dowork_level ()* function:

```
void dowork_level (uc nc, us pX[], us X, ul nel,
    us sIX, T_Res& pRes)
{
 us vmin, tCand[MaxLv + 1]; // a candidate
 ul j, k;
 uc *theCV;  // a CV
 // other declarations and initializations
 // get current itemset
 vmin = get_pattern (nc, tCand, pX, cX, nel, sIX);
 //j is the index of the first 1-item to add
 j = 0;
 while (j < cX && pX[j] <= vmin)  j++;
 for (k = j; k < cX; k++)
 {
  // add a 1-item to current itemset to produce a candidate
  tCand[0] = pX[k];
  // compute its CV if the constraints are valid
  theCV = compute_CV (tCand, nc,  ...);
  // memorize the candidate and add it to results
  // if applies
  store_CV (tCand, nc, theCV, pRes, ...);
  // update statistics
  pRes.nit++;  //...
 }
}
```

Listing 4: Code of main method called by PLW_CHI2

We shall not enter into the implementation details of this function. First because the code is most C likely and is easy to understand. And second because it does not include any specific parallel or shared memory features. So, we shall only explain its overall functionalities. The *for* loop is used to produce all the candidates at the current stage (the *tCand* variable). This is done by "adding" the possible existing 1-items to the base itemset managed by the function, and identified by the *nel* "number" (we shall discuss this aspect later). Once having generated such a candidate, we verify first if the different constraints underlying to our method are verified or not by the candidate. If it is the case, we compute its Contingency Vector using the *compute_CV ()* function. We second (try to) memorize the candidate in order to reuse it at the next level, and we add the candidate to the results if it contains one item of the target attribute.

Finally, the *store_CV ()* function (see Listing 5) describes, in a very simplified way, a specific section of code dedicated to the storage of results:

```
bool store_CV (us X[], us cardX, ...)
{
 //add X to the result file if X contains the target
 if (...)
 {
  critical_section cs;
  cs.lock();
  if (...)
    write_llhsp_to_file (X, cardX, ...);
  else
    write_pattern_to_file (X, cardX, ...);
  cs.unlock();
  }
// ...
}
```

Listing 5: Sharing ressources with the PPL using a critical section

Let us focus on the functionality involved in the first *if* statement: k-itemsets verifying the whole defined constraints and including an item belonging to the target column must be included into the results. This is done through their insertion into data files (one is associated to each value of k). During the parallellization process, each Task may write to one of these files each time it discovers a new valid itemset. What raises another shared resource problem, addressed by the PPL by the use of critical sections (a well-known concept in multithreading developments), as shown in the above code. When encountering such an instruction at run-time, the OS will not authorize any other Task to execute before the "lock" has been released.

To finish this Section, we explain the way we manage the memorization of candidates and associated information such as CVs. The main shared data structure in our developments is a tree storing the k-itemsets of "interest" (see Listing 6). The corresponding node structure, given in the C language, is:

```
typedef struct pattern_node
{
 unsigned short *Pattern; // the pattern
 unsigned char *pCV; // pointer to the CV
 T_NM *brother; // pointer to next node at same level
 T_NM *son; // pointer to next node at lower level
 ...
} T_NM;
```

Listing 6: Main index structure used by the PLW_CHI2 method

Each time a Task discovers a candidate verifying the whole constraints, the candidate is inserted into the tree. The insertion by itself uses the critical section concept we just introduced. Because the stored itemsets (patterns) are lexicographically organized within the tree, each of them can be referred to by a node number (what explains the *nel* "number" introduced above). Finally, after evaluating the candidates, the exploring process will retain them or not. In the latter case, the tree structure may be garbaged, which is done by the *update_shared_resources ()* function called at the end of our global *PLW_Chi2* method.

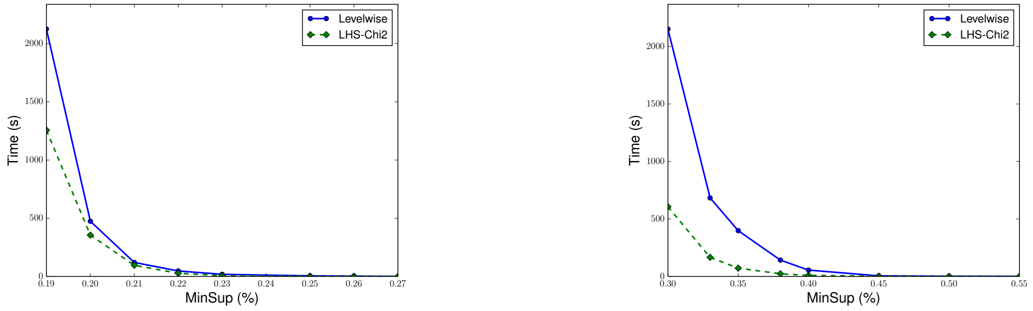
V. EXPERIMENTAL ANALYSIS

As briefly mentioned in the Introduction Section, this work has been initially applied on raw data measurement files provided by two industrial manufacturing partners in the area of Microelectronics: STMicroelectronics (STM) and ATMEL (ATM). The results of the realized experimental series are presented on 2 plans to be followed. They are associated with an analysis of 2 files among those supplied by both manufacturers. The first one (STM) contains 1241 columns and 296 lines. The second (ATM) consists of 749 columns and 213 lines. We chose a target attribute among a few possible columns. In both cases, the presented diagrams show the execution times of two methods when $MinSup$ varies while $MinPerc$ (0.34 for the STM file and 0.24 for the ATM one) and $MinCor$ (1.6 resp. 2.8) are fixed (see section III-B for more details).

Figures 1(a) and 1(b), extracted from [1], show the execution times of a standard LEVELWISE algorithm and the LHS-CHI2 algorithm on a non core computer (a HP Workstation with a 1.8 GHz processor and 4 Gb RAM, working under a Windows XP 32 bits OS). The difference between the two methods is that the LEVELWISE method uses no contingency vectors but standard computation of contingency tables. As the graphs point it out, the response times of the LHS-CHI2 method are between 30% and 70% better than LEVELWISE.

Figures 2(a) and 2(b) show the same execution times using the LHS-CHI2 algorithm and the presented PLW-CHI2 algorithm on a 4 core computer (a DELL Workstation with a 2.8 GHz processor and 12 Gb RAM working under the Windows 7 64 bits OS).

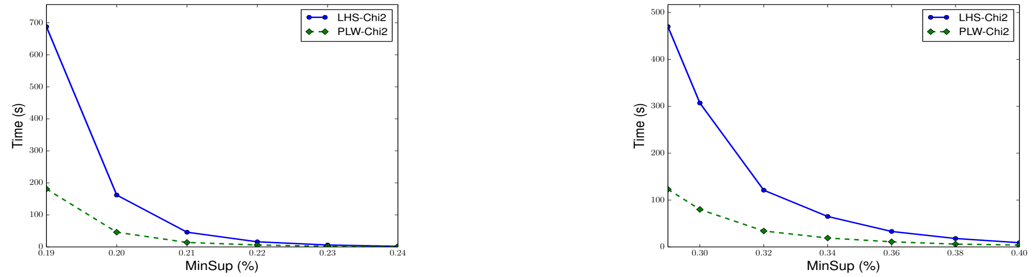
As it is easy to understand, the LHS-CHI2 method works here about two times faster on the multicore architecture, this is not because of the number of cores (which are not used) but because of the computer basic enhanced capabilities. When regarding to the performances of the PLW-CHI2 method, there is a gain factor of about 3.5, which is to compare to the number of available cores, which is 4. In other words, the parallelization of the LHS-CHI2 algorithm raises performance gains practically equals to the number of cores, the (little) loss being due to the shared memory management issues. Let us underline here that we do not integrate in these amounts that one core remains dedicated to system management ...



(a) Results for STM File

(b) Results for ATM File

Fig. 1: Execution times with a single processor.



(a) Results for STM File

(b) Results for ATM File

Fig. 2: Execution times with 4 cores.

VI. Conclusion and future work

In this paper, we present a new approach to discover correlated patterns, based on the usage of multicore architectures. Our approach is based on two concepts: Contingency Vectors, an alternate representation of contingency tables, and the Parallel Patterns Library. One advantage of Contingency Vectors is that they allow the Chi-squared computation of a k-itemset directly from one of its subsets. However, the usage of this library has a disadvantage: The parallelization of recursive algorithms is hard (we do not control neither the number of cores, nor the depth of the tree), even if we derecursify the algorithm. That is why we have chosen to implement a Levelwise algorithm which implements these two concepts. Experiments are convincing because our *PLW_Chi2* algorithm gains a time factor of about 3.5 (when using 4 cores) in comparison with the recursive version. For future works, we intend to develop a new version of the recursive algorithm using Contingency Vectors and to build our own thread/core manager.

References

[1] A. Casali and C. Ernst, "Discovering correlated parameters in semiconductor manufacturing processes: A data mining approach," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 25, no. 1, pp. 118–127, 2012. I, III-B, V

[2] J. Darlington, M. Ghanem, Y. ke Guo, and H. W. To, "Guided resource organisation in heterogeneous parallel computing," 1996. II, IV

[3] G. Krawezik and F. Cappello, "Performance comparison of mpi and openmp on shared memory multiprocessors," *Concurrency and Computation: Practice and Experience*, vol. 18, no. 1, pp. 29–61, 2006. II

[4] H. P. Barendregt, "Functional programming and lambda calculus," in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, 1990, pp. 321–363. II, III

[5] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 1994, pp. 487–499. III

[6] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "Parallel algorithms for discovery of association rules," *Data Min. Knowl. Discov.*, vol. 1, no. 4, pp. 343–373, 1997. III

[7] E. Li and L. Liu, "Optimization of frequent itemset mining on multiple-core processor," in *VLDB*, C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, and E. J. Neuhold, Eds. ACM, 2007, pp. 1275–1285. III

[8] R. Agrawal and J. C. Shafer, "Parallel mining of association rules," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 962–969, 1996. III-A

[9] S. Brin, R. Motwani, and C. Silverstein, "Beyond market baskets: Generalizing association rules to correlations," in *SIGMOD Conference*, 1997, pp. 265–276. III-B, III-B

[10] D. Moore, "Measures of lack of fit from tests of chi-squared type," *Journal of statistical planning and inference*, vol. 10 (2), no. 2, pp. 151–166, 1984. III-B

Detecting Command and Control Channels of a Botnet Using a N-packet-based Approach

Félix Brezo, José Gaviria de la Puerta and Pablo G. Bringas
DeustoTech Computing – University of Deusto
Bilbao, Spain
Email: {felix.brezo, jgaviria, pablo.garcia.bringas}@deusto.es

***Abstract*—The botnet phenomenon is one of the major threats in nowadays cyberspace. The ability of malware writers to code profitable applications with a softened learning curve is forcing public and private organisms to take measures against these infections. In this paper, we propose a method to identify traffic belonging to the Command & Control channels from a botnet. Our method takes into account the attributes of the packets captured from a connection to build vectorial representations of the connection by appending them into sequences of packets. Thus, we provide an empirical study of how these representations can be used to detect such a communicative behaviour by considering the issue as a supervised classification problem and comparing the results obtained by more than 20 machine learning algorithms.**

***Keywords*—botnet detection; n-packets; supervised learning; traffic analysis**

I. The botnet threat

The origin of the term *botnet* is commonly set in the fusion of the concepts of *robot networks*. In this way, botnets, as collections of infected machines remotely controlled by cybercriminals, are to naturally evolve into more complex entities that will have to be taken into account by private and public organizations. We can define this phenomenon as the new generation of malware that brings to light the profitable business of obscure economies in the deep web.

They have become a relevant issue to the different organizations internationally dedicated to computer security. Europol and NATO/OTAN are just two revealing examples. The former started to train their professionals to face the threats that these threads bring with them to privacy, anonymity and company's and public administration's security. The latter was forced in 2008 to create an observatory so as to watch for the rights of the allies in the cyberespace after the systematic attack suffered by Estonian cyberfacilities in 2007 [1].

What is certain is the fact that controlling the vast number of computers kidnapped by some of the biggest botnets and their potential computing power have not passed over neither for computer's professionals dedicated to code distributed solutions [2] nor for malware writers specific targets: monetizing massive infections [3]. In this way, botnets, as collections of infected machines remotely controlled by cybercriminals, are to evolve in complexity to constitute a threat even bigger in the near future.

To achieve this goal, the communications between bots have suffered major changes since its origins. Malware designers have found a hot topic on being able to cope with scalability and fault-tolerance. Thus, it is precisely a mechanism capable of maintaining a continuous communication with zombies what would determine the topology of the network, its capacity to avoid detection and disruption and the complexiy of the protocols defined to face this issues [4].

That is how the old-fashioned IRC (Internet Relay Chat) clients were brought in, often created on-the-fly by malware writers so as to open the doors to more complex solutions to boost anonymity and usability. This very last issue is one of the main catalysts of the concept *Hacking as a Service* —also known by some authors as *Crime as a Service* — as a malicious evolution from its benign and profitable branch, *Software as a Service* philosophy. Malware writers, conscious of the profitability of coding malicious applications for third parties, are adapting their tools so as to soften as much as possible the learning curve of the final user to widen their target market. That was the case of *Mariposa* botnet (Butterfly botnet in Spanish), dismantled in 2010 in the framework of of an operation conducted by the Spanish Guardia Civil and coordinated with different European organisms and Panda Software. In them, three people with little technical formation were arrested accused of being the botmasters behind a network of almost four million computers.

Against this background, we have advanced the state of the art with the following contributions:

- We show a method to model traffic connections by using the attributes inherent to the subsequent packets from such connections.
- We provide empirical validation of our method with a study that explores the capability of such representation model to identify Command & Control communications performed by some HTTP botnets.
- We show how the proposed method achieves high detection rates and how these could be used to identify infections. At the same time, we also discuss the shortcomings of the proposed approach and suggest future lines of work that might be explored in the near future.

The remainder of this paper is structured as follows. Section II describes the representation model used in this article. Section III states the methodology applied to evaluate the method as well as the results obtained in the experiments performed. Section IV analyses the implications of the aforementioned results as well as outlines some future lines of work.

Finally, Section V collects the conclusions to be extracted from this article.

II. Traffic modelling

Although other approaches have tried to fight botnets by applying transformations to the data fields from a given connection [5], in this paper we are going to perform a more accurate atomization of the characteristics of a connection by grouping representations as sequences of consecutive packets from a connection and analysing their detection capabilities depending on the length of these sequences.

In previous work [6], we have demonstrated that the analysis of the characteristics of packets on their own could be used as indicators of the kindness of a communication. While we considered the observation of a single packet as the obtention of a snapshot of the state of the connection at a time t_0, we also suggested that the monitoring of the evolution of these characteristics of a connection over time would be pretty more accurate. As a result, we have developed a method to represent such an evolution by employing the attributes of the packets observed over time to create a time-dependant representation model. To achieve this objective, we are going to build representations that take into account the concatenation of the individual characteristics of each packet in a chronologically ordered connection. However, to avoid biases we would perform some previous transformations to the traffic sniffed by hiding information relative to IP addresses, MAC addresses or ports as their inclusion would lead to a problem which is trivially solved once an infected address is identified. In Algorithm 1, we show the pseudo-code relative to the construction of sequences of up to n packets that lead to the obtention of the combined representations.

Given a dataset of connections C compounded by c connections from which we have observed a series of packets, the representation generation process of up to $n = 12$ will go through each and every of the c connections creating representations by appending the attributes of n consecutive packets. Additionally we have added as part of the representation some complementary temporal variables of different measures of central tendency and dispersion. The idea is to use them to represent the temporal intervals that passed between the reception of one packet and the following used in the same representation.

```
Input  : A bidimensional array $A$ that contains a list of
         the attributes of the available packets for a
         connection $c$ in the dataset $C$
$muestras \leftarrow []$;
foreach $c \in C$ do
    $nPaq \leftarrow len(c)$;
    // The process will be performed as
       many times as the length of the
       representations was desired. In
       this case, up to $n = 12$
    foreach $i \in range(2, n)$ do
        // In a connection $c$ we could
           obtain $nPaq - i + 1$ sequences of
           length $i$
        foreach $j \in c$ do
            $rep \leftarrow []$;
            $tiempos \leftarrow []$;
            // We select the $i$ packets that
               will compound the
               representation
            foreach $k \in i$ do
                $rep.append(j + k)$;
                $tiempos.append(j + k)$;
            end
        end
        // We work out the internal times
           $t_i$
        $attM \leftarrow calculoTi(tiempos)$;
        $rep.append(attM)$;
        // We add the type of the sample
           as the last attribute of it
        $rep.append(attTipo)$;
        // We add the whole sample to our
           list of full representations
           of the dataset
        $muestras.append(rep)$;
    end
end
Output: A bidimensional array containing a list of the
        attributes of the available sequences for each
        connection
```

Algorithm 1: Obtention of the characteristics of a single packet.

We benefit from having the calculation of such gap as a trivial task, taking into account that any sniffing tool provides the timestamp of when a packet was observed. However, considering this value in absolute terms may lead us to an error. If we only considered the time by itself we could end up introducing a bias in the dataset that can use this parameter in a way that we do not like: identifying connections by the *distance* (understood as *travelling time*) to/from the server. This is not an interesting point because we may be developing a system capable of detecting *distances* instead of using the rest of characteristics inherent to the packets. We want our approach to be more general, avoiding the analysis of the content of the studied connections. The way in which we have addressed this issue is the following: we have considered relative times instead of absolute times. There are two special cases to this rule. For a sequence length of $n = 1$ the spatio-temporal characteristics are not taken into account, whilst for a sequence length of $n = 2$, the values obtained will always be $\{1\}$ as a result of the application of the general formula in Equation 1:
$t_1 = \frac{T_2 - T_1}{T_2 - T_1} = 1$.

To perform these calculations, we have normalized this spatio-temporal measure: for representations using sequences of n packets, we have considered the time passed between the first and the second packet as a unit, while the rest of spatio-temporal measures observed would be weighted accordingly. Given a sequence of n packets observed in the moments $\{T_1, T_2, \cdots, T_n\}$, we would obtain the $n - 1$ spatio-temporal coefficients $\{t_1, t_2, \cdots, t_{n-1}\}$ by applying the following:

$$t_i = \frac{T_{i+1} - T_i}{T_2 - T_1} \qquad (1)$$

obtaining a vector of coefficients as the following:

$$t = \{1, \frac{T_3 - T_2}{T_2 - T_1}, \frac{T_4 - T_3}{T_2 - T_1}, \cdots, \frac{T_n - T_{n-1}}{T_2 - T_1}\} \qquad (2)$$

To evaluate the evolution of these parameters, we take into account two different points of view: a central measure and a dispersion one. Central measures are tools used to resume the information of a dataset in a single scalar that tries to represent the central point of a dataset. Taking into account that this information corresponds to relative data, the most appropriate central measure is not the arithmetic mean, but the geometric mean. In spite of being commonly used as the most known central measure, the former is not always a strong statist, as it can be broadly influenced by the appearance of atypical values. At the same time, the most appropriate central measure to be used with relative values is the geometric mean [7]. Mathematically, the geometric mean $\bar{x}$ of a collection of n elements $\{a_1, a_2, \cdots a_n\} \in R$ can be expressed as in equation 3:

$$\bar{x} = \sqrt[n]{\prod_{i=1}^{n} x_i} = \sqrt[n]{x_1 * x_2 * \cdots * x_n} \qquad (3)$$

This tool is used to determine the variability of the dataset. This kind of measures determines to what extent the different values in a dataset are distant from the central point of the distribution [8], being higher for very disperse values and smaller for those that vary less. In this paper, we are using the standard deviation. For a collection of n elements $\{a_1, a_2, \cdots a_n\} \in R$, the standard deviation σ can be expressed as equation 4:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad (4)$$

By including these values to our representations, we claim we are avoiding biases that would lead to the sole identification of the distance of the servers expressed in time-to-destiny.

III. Experimental Validation

To evaluate what can be considered a good reference for botnet traffic detection, we have conducted a series of experiments that apply machine learning algorithms. In the following subsections we are describing the methodology applied, the different algorithms selected and the comparison metrics used to evaluate their performance, as well as the results achieved for each and every representation.

A. General Methodology

The proposal developed in this article considers the identification of HTTP communications from a botnet as a supervised classification problem. This election is not arbitrary. The authors will adapt a philosophy similarly applied to the identification of IRC botnets in the past [9], [10]. The latter performed a supervised approach to detect IRC-controlled botnets identifying the problem as a supervised classification task. In this type of problems, we study phenomenons represented by a d-dimensional vector X in R^d that can be classified in K different ways according to a vector Y of *labels* or *classes*. The application of classification algorithms in supervised learning approaches make use of a previously labelled dataset [11] which, in our case, will correspond to the legitimate or botnet labelled traffic samples.

With such objective, we have defined a training dataset D_n as $D_n = \{(X_i, Y_i)\}_{i=1}^{n}$ where X_i represents the events corresponding to phenomenon X while Y_i is the label that classifies it in the category that the classifier assumes as correct. For instance, for the case of sequence lengths of $n = i(\forall 1 <= i <= 12)$ packets, we can identify a sequence X_i defined by a series of attributes that represent it, being Y_i the category assigned to that sequence in accordance to the estimations of each classifier. As the authors know the concrete characteristics (IP, MAC, etc.) of the connections associated to malicious traffic, we were able to easily label the class of each and every packet as *bot* or *legitimate* to build the training datasets used in Section III.

Below, we go through the different supervised learning algorithms used to face the problem of detecting Command & Control traffic as a supervised classification problem. We have opted to compare the performance of the different classification algorithms given the notable differences in similar experiments depending on the approach used such as the detection of errors in software quality models [12] or the automatic classification of commentaries in social websites [13]. The tool used to perform these experiments is the Waikato Environment for Knowledge Analysis (WEKA)[1]. This software plataform written in Java was conceived to experiment with machine learning and data mining while remaining widely expandable with a variety of official and community-developed plugins.

In the case of this paper, the algorithms employed for this experiments are the ones that follow:

- **Support Vector Machines**. We have used the Sequential Minimal Optimisation (SMO) algorithm [14] employing for the experiments different kernels: a polynomial kernel [15], a normalized polynomial kernel [15], a Pearson VII kernel [16] and a Radial Basis Function (RBF) kernel [15].
- **Decision Trees**. We have selected *Random Forest* [17] and the implementation of the C4.5 [18] performed by the WEKA developers [19], J48.
- **Bayesian Networks**. With Bayesian Networks, we have used different algorithms of structural learning: K2 [20], *Hill Climbing* and *Tree Augmented Naïve* (TAN) [21], as well as testing the effectiveness of the *Naïve Bayes* algorithm [22].
- **Bagging**. We have used the implementation of the Bagging Method with the fast learning algorithm REPTree, a decision tree method used in the past, for example, to successfully evaluate the performance of individuals in online gaming platforms [23].
- **Perceptrons**. We have used different types of perceptrons that try to face the traditional problems of this technique to label classes which are not lineally separable: the time consuming Multi Layer Perceptrons (MLP) [24] and the Voted Perceptron [25].

[1] http://www.cs.waikato.ac.nz/ml/weka/

- **K-Nearest Neighbour (KNN)**. We have conducted experiments in the range from $k = 1$ to $k = 10$ neighbours. The goal is to check if the hypothesis already raised by the authors in the past regarding the inefficiency of the inclusion of more neighbours [26] could also be applied to the n-packet sequences.

The method employed to compare these algorithms is making use of a cross-validated series of experiments and balanced training datasets as detailed below.

1) Cross Validation: One of the ways of validating the behaviour of the classifiers consists of using cross validation techniques. These techniques divide the data collected into k training datasets compounded by the $[100 - (k-1) \cdot \frac{100}{k}]\%$ of the samples. Thus, after the training, the model would be validated with the rest of the $(\frac{100}{k})\%$ samples to evaluate its efficiency. The system error E can be defined as follows:

$$E = \frac{1}{k} \cdot \sum_{i=1}^{k} E_i \tag{5}$$

being E_i the error of each and every iteration. The k value used in WEKA for this research is $k = 10$, so that each dataset used is divided into ten training and testing datasets: the former would be compounded by the 90% of the representations whilst the latter, the ones used to analyse the capabilities of the model, by the remaining 10%.

2) Resampling and balance of the training data: As previously suggested, the number of packets of each session may vary noticeably depending on the data analysed, a point which has its importance if we take into account that it is much easier for us to generate legitimate traffic than to monitor botnet connections. However, the usage of unbalanced data may introduce skews in the classification producing over-fitting in certain cases [27]. Because of that, we have decided to use resampling techniques that readjust the number of packets of each instance as a previous step to the appliance of the different classification algorithms. In this case, the algorithm will opt to reduce the dataset to a number of samples per class equal to the number of the class with less samples in the initial dataset: that is to say, in the case of using sessions where the number of legitimate samples was greater than the number of malicious samples ($b >= m$), the algorithm will proceed to select only m benign samples to train the models. WEKA implements this technique with the Spread Subsample algorithm.

B. Comparison Metrics

The evaluation of each and every method is going to be performed according to different parameters commonly used in the evaluation of the performance of the classification methods [28]. As a first step, we have to take into account the significance of the confusion matrix shown in Table I. Thus, we can define *True Positives* TP as the number of botnet packets correctly identified, *False Positives* FP as the number of legitimate packets incorrectly classified as relative to a botnet, *True Negatives* TN as the number of legitimate packets correctly identified as legitimate and *False Negatives* FN as those packet generated by botnets that the system was unable to correctly identify. In this way, by the combination of the aforementioned data, we have defined the following comparison metrics so as to evaluate the performance of each classifier:

TABLE I. GENERIC CONFUSION MATRIX.

		prediction outcome		
		p	**n**	**total**
actual value	**p′**	True positive	False negative	P′
	n′	False positive	True negative	N′
	total	P	N	

- ***Accuracy* (Acc.)**. The $Acc.$ —see equation 6— is worked out by dividing the total number of correct labels by the total number of instances that compose the full dataset.

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

- ***Possitive Predictive Value* (PPV)**. The PPV —also known as *precision*— is a value that presents the possibility of finding a positive result representing the tested condition, *id est*, a sample labelled as a positive which effectively is a TP. It is mathematically defined in equation 7 as follows:

$$PPV = \frac{TP}{TP + FP} \tag{7}$$

- ***Area Under ROC Curve* ($AURC$ o AUC)**. The geometric meaning of the ROC curve derives from the establishment of a relationship amongst the false negatives and the false positives [12]. This value is obtained representing, for each possible election of cut values, the TPR in the y-axis and the FPR for the x-asis. Originally used by the American army in researches after the Pearl Harbor attack in 1941 so as to detect radar signatures of Japanese aircraft [29], this measure has been used sin the last part of the XX Century as a comparison metric to evaluate the performance of classification algorithms [30]. It is broadly used to generate statistics that represent the performance of a classifier in a binary system as a tool to select those probably optimal models from the suboptimal ones. Although this measure does not provide information about the good behaviour of a model, the $AURC$ helps to determine the validity of the data distribution given a series of predictory conditions [31].

C. Results

The benign dataset used in this experiment corresponds to traffic obtained from different sessions run by the students of the *Máster Universitario de Seguridad de la Información*

—in English, Master of Information Security— held by the University of Deusto. Thus, the authors could proceed to the monitoring of the connections during the following browsing hours by using a sniffing tool to monitor the traffic. As an additional remark, and because of the legal implications that would imply the fact of storing such kind of traffic data, it was necessary to obtain the students agreement to participate in the experiment being held. This restriction has some implications experimentally speaking: it may introduce a sort of skew in the data analysed regarding the final content of the communications as a user having being advised of being monitored for an experiment is substantially less inclined to visit, for instance, banking sites or webpages with pornographic content. However, the authors consider this naturally introduced error as acceptable since the students were told to perform standard browsing sessions following the patterns stated by a recent study which tried to provide data on the average usage of the Internet [32].

Meanwhile, malware traffic samples were obtained after infecting different machines with samples of Flu, Prablinha and Warbot. All of them are controlled from a web-based Command & Control panel from which the botmaster can execute the different attacks. For the purpose of this research, all along the infected sessions we performed different attacks requesting the infected node to execute different tasks: performing DDoS attacks, downloading and executing a file, transferring files, storing the keystrokes from the user, etc. The number of Command & Control packets (the only ones analysed in this paper) stored ranged from 1,000 to 5,000 in each session.

With this information, we built 12 datasets for each piece of malware (a total of 36 different datasets) generating representation samples of $n = \{1, 2, 3, \cdots, 12\}$ packets. Each dataset was trained with each and every of the 23 previously defined classification algorithms to complete the 828 experiments conducted whose results are explained below. Note that the results of each individual family have been grouped into a single graph by calculating the arithmetic mean of the values separately obtained.

In Figure 1, we can see the evolution of the percentage of correct classifications performed by every algorithm for each proposed representation. Generally, we can see an important improvement of the classifications performed for those representations taking into account longer sequences. This improvement, however, is not linear. For all the algorithms, even for those with worse behaviour such as SMO RBF, Naïve Bayes and Voted Perceptron, we can observe a very important jump from sequences of single packets to those using $n = 2$ and $n = 3$ packets, after what the performance of the classifier starts to stabilise its improvements. As occurred in similar research performed in the past, the authors have noticed that taking into account more neighbours in the KNN classifier does not improve the classification. On the contrary, we have observed that the more neighbours are considered, the worse the results obtained. We can affirm that the best results have been obtained by Bayes Net K2 classifier for a sequence length of $n = 12$ (being the best example of a bett), while we can consider the second best algorithm Random Forest for sequence lengths of $n = 6$ and $n = 7$. Random Forest shows signs of stagnation for sequences longer than $n = 8$. As a negative remark, we have to pinpoint the little efficiency of some classifiers in comparison to the rest of the algorithms tested. *Voted Perceptron* and *Naïve Bayes* achieved the worst results for all lengths.

The use of the PPV puts in context the results shown in Figure 1. The PPV graphs (see Figure 2) show the relation between the true positives and the false positives encountered. This measure is important taking into account that a high detection rate would be useless if the False Positive Ratio is also high. If this was the case and a system was implemented in a real environment, users could feel frustrated and the detection efforts would have been in vain.

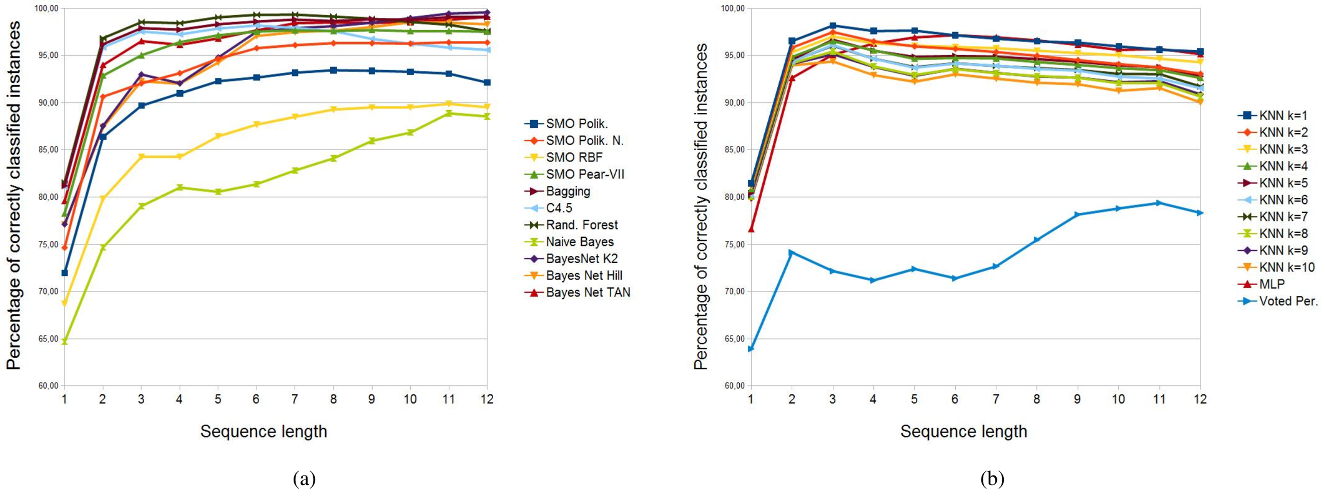
In this research, we have found out that 12 out of the 23 classifiers have obtained values over 0.95 at least once. In this case, the best overall results were obtained by SMO using the Pearson VII kernel, achieving almost the maximum score for lengths of $n = 10$ and $n = 11$. We have to highlight here the Bayesian classifiers (excepting Naïve Bayes), Bagging and Random Forest, although this last one shows again symptoms of lightly losing efficiency for the longest lengths of packets. This indicator confirms what we have exposed before about the consideration of more neighbours on KNN, while Multi Layer Perceptron does not obtain significant results taking into account the extra time needed to perform the experiments for such classifier. Once again, the worst classifiers are Naïve Bayes, SMO with RBF kernel (though they could cut the gap towards the rest of the classifiers, specially for the longest sequences) and Voted Perceptron.

Finally, in Figure 3 we show the evolution of the values under the Roc Curve ARC for the detection of Command & Control traffic depending on the number of packets included in the representation. These values represent the kindness of the dataset used while we increase the number of packets included per sequence.

The results are especially positive in the case of Bayes Net (with K2 and TAN search methods), Random Forests and Bagging. At the same time, most KNN versions of the algorithm obtain very high values. With regard to this group of algorithms, we find interesting to pinpoint an inversion of the trends observed in the aforementioned indicators showing that KNN with $k = 1$ is the worst classifier amongst them. The worst overall results are obtained again by Voted Perceptron, Naïve Bayes and SMO RBF, which are unable to cope with the problem in an efficient way.

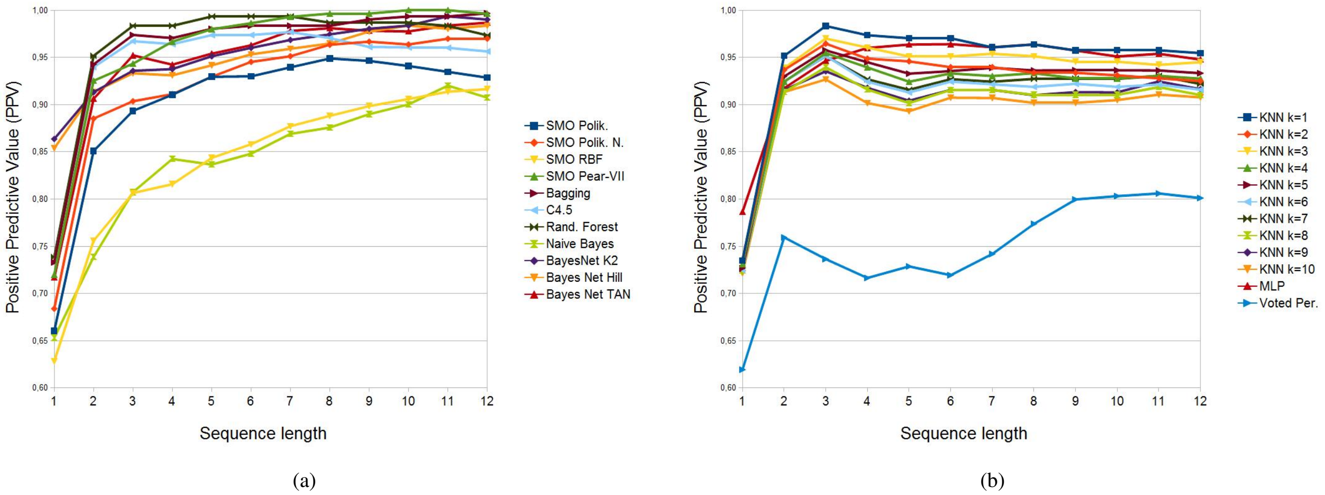
IV. Discussion and future work

One of the main problems supervised learning brings into the field of traffic classification is the immense amount of data that has to be processed. At the same time, researchers will have to deal with some important legal difficulties associated to the extraction of traffic samples which could serve to avoid any kind of biases in the training datasets. We are applying different algorithms assuming the additional efforts required in the usage of supervised approaches [33] as a first step towards the application of other philosophies. In this line, the legal —and logical— need of having the consent of the users that will donate their user sessions, forces us to assume certain risks associated to the inexistence of browsing habits that in real environments would be considered as legitimate ones —such



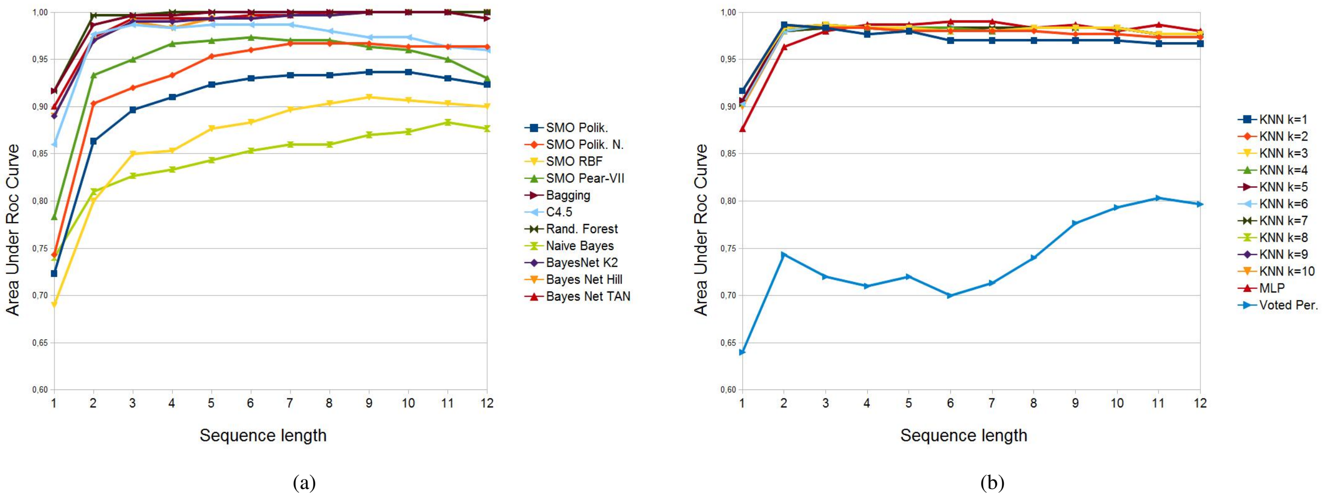
(a) (b)

Fig. 1. Evolution of the $Acc.$ while using different sequence lengths as the representations used by the classification algorithms to detect C&C traffic.



(a) (b)

Fig. 2. Evolution of the PPV while using different sequence lengths as the representations used by the classification algorithms to detect C&C traffic.



(a) (b)

Fig. 3. Evolution of the ARC while using different sequence lengths as the representations used by the classification algorithms to detect C&C traffic.

as the visit to websites with adult content—. However, the appearance in testing environments is little or none because of the fact that the user knows that he is surfing through a network that is being monitored.

The virtue of the method defined in this paper resides in the fact of classifying traffic independently of the content that flows through the network, using as detection metrics the values relative to the length of the headers, the connection frequency or the periodicity of the polls between the infected machine and the machine from where the botmaster issues the orders.

One of the most important issues is the fact that the malicious charge of a packet (or even of a short sequence of packets) is relatively low. Accordingly, the appearance of True Negatives is less harmful than in other fields such as malware detection, where being unable to detect a malicious instance could have dramatic consequences. As the number of packets generated by a single connection is, normally, pretty big, we can think of a detection method which could employ the historical records of the representations connected to a single point. By doing this, we would be able to assign a value that, at any time, could let the system know the reliability of the current connection taking into account how the packets with certain source or destiny have been classified. The fact of being able to tolerate certain True Negatives would provide resilience to a system which would be able to inform the user the establishment of suspicious connections even without having a traditional malware detection solution on its computer.

V. Conclusions

Botnets are going to remain as one of the most important cyberthreats in the near future. In this regard, we have proposed in this article a methodology to validate a new representation model that could be used to identify the Command & Control communications performed by a botnet. We have proved that the attributes obtained from the observation of the packets in a connection could be used to fit the initial objectives. We have defined a series of transformations to perform to the observed packets to build vectorial representations sensible to both, atomic attributes of the packets and spatio-temporal variables. To face this issue, we have considered the problem as a binary classification problem in which the algorithms employed would have to be able to differentiate benign samples from malicious ones.

The comparison metrics employed have guided us to compare the performance of the different representations employed and the different algorithms. We can state that the inclusion of more packets in our representations improve the results obtained by a given algorithm for shorter representations. At the same time, we have proved that some algorithms, such as Bayes Net and Random Forest obtained very high detection ratios. Anyway, the characteristics of the packet classification problem has some advantages that could be exploited in the favour of the security analyst: the payload of a single packet is pretty less relevant to the payload of a malicious binary, so a greater False Negative ratios would be tolerated with little or no harm for the final user, letting the system take less borderline decisions by waiting for more pieces information.

The increasing amount of data to be analysed in current corporate networks will undoubtedly force us to face congestion problems. Thus, future work should lead us to the management of fewer amounts of data, making the inclusion of an unsupervised approach which would need less labelled instances and the consideration of more botnet families should be studied to give more robustness to a methodology whose implementation could complement the traditional honeypot solutions and content-based detection techniques.

References

[1] S. Blank, “Web war i: Is europe’s first information war a new kind of war?” *Comparative Strategy (2008)*, vol. 27, no. 3, p. 227, 2008. [Online]. Available: http://www.informaworld.com/10.1080/01495930802185312

[2] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, “Seti@ home: an experiment in public-resource computing,” *Communications of the ACM*, vol. 45, no. 11, pp. 56–61, 2002.

[3] Z. Li, Q. Liao, and A. Striegel, “Botnet economics: uncertainty matters,” in *Managing Information Risk and the Economics of Security*. Springer, 2009, pp. 245–267.

[4] N. Cornhill and M. Morris, “Communication structures of botnets with case studies,” 2012.

[5] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, “Disclosure: detecting botnet command and control servers through large-scale netflow analysis,” in *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012, pp. 129–138.

[6] F. Brezo, J. Gaviria de la Puerta, X. Ugarte-Pedrero, I. Santos, P. G. Bringas, and D. Barroso, “Supervised classification of packets coming from a http botnet,” in *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*. IEEE, 2012, pp. 1–8.

[7] B. Carlson, “Algorithms involving arithmetic and geometric means,” *The American Mathematical Monthly*, vol. 78, no. 5, pp. 496–505, 1971.

[8] J. M. Bland and D. G. Altman, “Statistics notes: measurement error,” *Bmj*, vol. 312, no. 7047, p. 1654, 1996.

[9] G. Gu, J. Zhang, and W. Lee, “BotSniffer: Detecting botnet command and control channels in network traffic,” in *Proceedings of the 15^{th} Annual Network and Distributed System Security Symposium (NDSS’08)*. Citeseer, 2008.

[10] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, “Using machine learning techniques to identify botnet traffic,” in *In 2nd IEEE LCN Workshop on Network Security (WoNS2006*, 2006, pp. 967–974.

[11] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” in *Proceeding of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007, pp. 3–24.

[12] Y. Singh, A. Kaur, and R. Malhotra, “Comparative analysis of regression and machine learning methods for predicting fault proneness models,” *International Journal of Computer Applications in Technology*, vol. 35, no. 2, pp. 183–193, 2009.

[13] I. Santos, J. de-la Peña-Sordo, I. Pastor-López, P. Galán-García, and P. G. Bringas, “Automatic categorisation of comments in social news websites,” *Expert Systems with Applications*, 2012.

[14] J. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” *Advances in Kernel Methods-Support Vector Learning*, vol. 208, 1999.

[15] S. Amari and S. Wu, “Improving support vector machine classifiers by modifying kernel functions,” *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.

[16] B. Üstün, W. Melssen, and L. Buydens, “Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel,” *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 29–40, 2006.

[17] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] J. R. Quinlan, *C4. 5 programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[19] S. R. Garner, “Weka: The Waikato environment for knowledge analysis,” in *Proceedings of the New Zealand Computer Science Research Students Conference*, 1995, pp. 57–64.

[20] G. F. Cooper and E. Herskovits, “A bayesian method for constructing bayesian belief networks from databases,” in *Proceedings of the* 7^{th} *conference on Uncertainty in artificial intelligence*, 1991.

[21] D. Geiger, M. Goldszmidt, G. Provan, P. Langley, and P. Smyth, “Bayesian network classifiers,” in *Machine Learning*, 1997, pp. 131–163.

[22] D. D. Lewis, “Naive (Bayes) at forty: The independence assumption in information retrieval,” *Lecture Notes in Computer Science*, vol. 1398, pp. 4–18, 1998.

[23] K. J. Shim, K.-W. Hsu, and J. Srivastava, “Modeling player performance in massively multiplayer online role-playing games: The effects of diversity in mentoring network,” in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*. IEEE, 2011, pp. 438–442.

[24] M. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)–a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[25] Y. Freund and R. E. Schapire, “Large margin classification using the perceptron algorithm,” *Machine learning*, vol. 37, no. 3, pp. 277–296, 1999.

[26] F. Brezo, J. Gaviria de la Puerta, X. Ugarte-Pedrero, I. Santos, P. G. Bringas, and D. Barroso, “Supervised classification of packets coming from a http botnet,” in *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*. IEEE, 2012, pp. 1–8.

[27] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, “Evolving diverse ensembles using genetic programming for classification with unbalanced data,” 2012.

[28] D. Powers, “Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation,” *School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE-07-001*, 2007.

[29] D. Green, J. Swets *et al.*, *Signal detection theory and psychophysics*. Wiley New York, 1966, vol. 1974.

[30] K. Spackman, “Signal detection theory: Valuable tools for evaluating inductive learning,” in *Proceedings of the sixth international workshop on Machine learning*. Morgan Kaufmann Publishers Inc., 1989, pp. 160–163.

[31] J. Lobo, A. Jiménez-Valverde, and R. Real, “Auc: a misleading measure of the performance of predictive distribution models,” *Global Ecology and Biogeography*, vol. 17, no. 2, pp. 145–151, 2007.

[32] F. Lera-Lpez, M. Billon, and M. Gil, “Determinants of internet use in spain,” *Economics of Innovation and New Technology*, vol. 20, no. 2, pp. 127–152, 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/10438590903378017

[33] A. Teichman and S. Thrun, “Tracking-based semi-supervised learning,” *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 804–818, 2012.

Performance Evaluation of Scale-out NAS for HDFS

Makoto Sato
Kanagawa Institute of Technology
Japan
s1021079@cce.kanagawa-it.ac.jp

Hiroki Kanzaki
Graduate School of Kanagawa Institute of Technology
Japan
s1285025@cce.kanagawa-it.ac.jp

Shoji Kawata
Graduate School of Kanagawa Institute of Technology
Japan
s1285014@cce.kanagawa-it.ac.jp

Shun Sugiyama
Zuken NetWave, Inc.
Japan
shun.sugiyama@znw.co.jp

Shingo Otsuka
Kanagawa Institute of Technology
Graduate School of Kanagawa Institute of Technology
Japan
otsuka@ic.kanagawa-it.ac.jp

***Abstract*— Isilon of EMC Corporation, which is the major company of the storage production, announces the correspondence to Hadoop Distributed File System (HDFS). Hadoop is parallel distributed processing base for large-scale data constructed in HDFS and MapReduce. In addition, it can treat huge files using plural computers link in Hadoop. However, the tendency of detailed performance and various parameters is not known. Therefore, in this paper, we perform the comparison in the case with normal HDFS and Isilon for HDFS using a benchmark about writing performance and reading performance.**

Keywords-Hadoop; HDFS; Scale-out NAS; Isilon.

I. INTRODUCTION

Scale-out NAS (Network Attached Storage) has extensibility and the management characteristics that conventional scale-up NAS does not have. And it can expand capacity and the performance seamlessly. Therefore, we are able to be gradually expanded from small constitution as needed. Scale-out NAS manages the cluster as single file system, and it is not necessary to be conscious of the real physical position of data [1]. Isilon of EMC Company announces that it native supports Hadoop Distributed File System (HDFS) in latest OS (OneFS) [2].

Hadoop is parallel distributed processing base for the large-scale data constructed in HDFS and MapReduce. In addition, it can treat huge files by letting plural computers link in Hadoop. HDFS is constructed in Namenode and cluster of Datanode. Datanode manages the data in HDFS divided into the fixed length called the block [3] [4].

Namenode manages the file attribute information called metadata and the information of the file system. Namenode is a single obstacle point in HDFS. In one of the faults of HDFS, The file system becomes offline when Namenode falls. Isilon solves this problem of single obstacle points in HDFS.

In addition, HDFS (Hadoop) maintains a replica function for HDD trouble of Datanode, and the user can decide the number of the replicas depending on the use situation. If there is much number of the replicas, redundancy improves, but the processing capacity decreases because the access to an HDD increases. Generally, the number of the replicas of Hadoop is around three. In the case of Isilon for HDFS, it can handle the number of the replicas with one because Isilon secures redundancy. Therefore, we can make use of HDFS performance to the maximum.

This way, Isilon solves a part of the problems in HDFS and maintains high processing capacity. However, the tendency of detailed performance and various parameters is not known. Therefore, in this paper, we perform the comparison in the case with normal HDFS and Isilon for HDFS using a benchmark about writing performance and reading performance.

II. RELATED WORKS

There are some related work of this study as follows; Evaluation of Hadoop system consisting of Virtual Machines on Multi-core CPUs by Ishii [5], Studies on Evaluating Performance Efficiency of Distributed File Systems by Sakurai [6], Characterization of Remote Data Access for Hadoop Distributed File System over a long-latency environment by Momose [7], and Consideration on Ad hoc query processing with Adaptive Index in Map Reduce Environment by Okudera [8].

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-311-7

III. Parallel Distributed Processing

A. Hadoop

The Large-scale data are created in various situations by a technological change. The big data such as the GPS of the mobile devices, cameras and action histories of the users with the sensor continue increasing every day. The useful information is provided by analyzing it. Therefore, Hadoop attracts attention as the parallel distributed processing base which can process large-scale data easily.

Hadoop is constructed in HDFS and MapReduce and hBase [9,10]. The MapReduce processing performs important distributed processing. The data are managed by a combination of Key and Value. By the MapRecude processing, three next processing are carried out: Map processing to divide large-scale data into small data, and to extract necessary information, Shuffle processing to bundle up a combination having same Key, and Reduce processing to gather them up, and to output a result.

HDFS is file system to distribute large-scale data, and to manage using plural disks. HDFS improves throughput by being parallel from plural disks and reading data and handles large-scale data efficiently. In addition, it can prevent the loss of data because a value of Replication is set by default to 3, even if it breaks down.

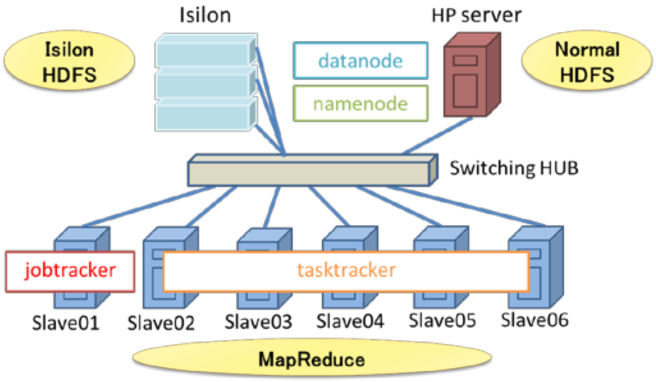


Figure 1. Experiment environment.

B. Isilon

Isilon is scale-out NAS developed by Isilon system. The difference with Isilon and conventional scale-out NAS is the point where NAS controller is not separated.

Isilon system includes CPU, memory and a network device. Furthermore, the system includes software, such as the file system. These devices are called a node and can be expanded from 3 to 144 nodes and the addition of the node is easy. Isilon repositions data automatically and optimizes it if a node is added.

The conventional NAS came to be able to cover it until large-capacity data because the capacity of the disk increased although there is a fault in the scalability in conventional NAS. However, time to load data increases when quantity of data to treat increases because capacity has a limit in the simple substance. Scale-out NAS is designed to solve such a problem and to be able to be expanded to cope with data increasing every day.

IV. Performance Evaluation Using the Benchmark

In this paper, we built a Hadoop environment using Isilon for HDFS, and we performed the comparison in the case with normal HDFS and Isilon for HDFS using a benchmark about writing performance and reading performance.

A. Experiment conditions

The hardware which we used for this experiment is six calculation nodes, Isilon and normal PC (made by Hewlett Packard). Figure 1 shows our experiment environment. We call this normal PC HP server afterward. HP server has the performance that is equal to Isilon. Each calculation node is comprised of CPU (Intel core i5 2500K 3.3GHz), Memory (8GB) and HDD (1TB SATA (7,200rpm)*1). Isilon is comprised of CPU (Nehalem Quad Core), Memory (6GB) and HDD (500GB SATA (7,200rpm)*12). The HP server is comprised of CPU (Xeon E5607 2.26GHz), Memory (6GB) and HDD (1TB SATA (7,200rpm)*8). About the OS, calculation node and the HP server are CentOS6.2, and Isilon is OneFS [11].

One JobTracker and five TaskTracker perform the MapReduce processing. Isilon performs both NameNode and DataNode about the HDFS processing. In comparison, we carried out a similar experiment using HP server.

The number of the Map tasks of each node of Hadoop is two by default and the number of Reduce tasks is one by default. In tour study, we set the number of Map tasks to two and the number of Reduce tasks to two. The number of the replicas of normal HDFS is one by default. The number of the nodes and replicas of Isilon is three because Isilon usually holds three replicas [12].

B. Experiment description

We use three benchmarks (Teragen [13], Grep [9], and Terasort [13]), which are attached to Hadoop. There are one of the Hadoop's widely used benchmarks. Hadoop's distribution contains the input generator, finding keywords and sorting implementations. We performed the comparison about writing and reading performance. In the experiment, we changed parameters of the block size and compared the tendency of the transaction speed. We performed the measurement three times and show the average results. The processing to be carried out in each benchmark is as follow.

a) Teragen: Teragen is the program that only a designated number generates character string of 100 bytes per one record. It is used to measure writing performance.

b) Grep: Grep is a program to count the number of times of appointed character string to develop in input data. It is used to measure reading performance.

c) Terasort: Terasort is a program to output after sorting input data. It is used to measure writing performance and reading performance.

In Grep and Terasort, it is necessary to prepare for data to calculate. Therefore, we generate the random character string data using Teragen. The size of data to treat by this experiment is 20GB, 40GB and 60GB. The parameter of block size in Teragen and Grep is 32MB, 64MB, 256MB, 512MB and 1GB. And the parameter of block size in Terasort is 32MB, 64MB and 256MB.

It is possible to set block size from 4KB to 1GB in Isilon. A value of the defaults of the block size is 64MB. We set a value of 32MB, 64MB, 256MB, 512MB and 1GB in our experiments.

C. *Results*

Figures 2-4 show the results of the experiment by normal HDFS. Figures 5-7 show the results of the experiment by Isilon for HDFS. The vertical axis shows the processing time, and the cross axle shows data size.

As a result, we understand that the processing time became short, so that block size is big by the Teragen of normal HDFS. In the case of data size 60GB, there is the difference of approximately 50 seconds for 32MB, 64MB and 256MB, 512MB, 1GB. As the results of Teragen in Isilon, the difference of the processing time by the difference in block size is not seen. However, there is difference in 100 seconds when we compared the processing time of Isilon with the processing time of normal HDFS in the case of data size 20GB.

As the results of Grep, we understand that there is little processing time if block size is big in normal HDFS as Teragen. In addition, as the results of Teragen, it followed that the differences between 32MB and 64MB spread as data size grew big. The Grep of Isilon resembled processing of normal HDFS in a tendency. The processing time becomes short if a value of the block size is big although 256MB is slightly earlier than 1GB.

Finally, as the Terasort results, we understand that the processing time tended to be fast if block size is big by the processing by normal HDFS. In addition, the difference gradually spread as data size becomes big. In contrast with this, the results of 32MB and 64MB are the earliest block size by the Terasort of Isilon and a big difference matched the result of 256MB.

As the above experimental results, we show different trends in normal HDFS and Isilon. In normal HDFS, processing time is the best if the block sizes is 1GB in read and write processing. And processing time tends to become slow as block size becomes small. On the other hand, processing time is good using Isilon if block size is big in Teragen and Grep as normal HDFS. In contrast to normal HDFS, processing time tends to become fast as block size becomes small.

In our results, the read/write performance of Isilon is equal to or greater than normal HDFS. In addition, the number of replicas of HDFS is one and Isilon is three in the experiment. Therefore, we consider that the difference in performance between the HDFS and Isilon is become greater because the replica number is 2 or 3 using HDFS usually.

V. CONCLUSION

In this paper, we built a Hadoop environment using Isilon for HDFS. And we also performed the comparison in the case with normal HDFS and Isilon for HDFS using a benchmark about writing performance and reading performance. We are going to investigate a tendency when we changed the number of nodes to use and the number of disks.

REFERENCES

[1] “All of scale out NAS "Isilon"”: http://ascii.jp/elem/000/000/730/730401/ [retrieved: October, 2013]

[2] Isilon Scale-out Network Attached Storage (NAS) for Big Data – EMC: http://www.emc.com/domains/isilon/index.htm. [retrieved: October, 2013]

[3] Dhruba Borthaku.: “The hadoop distributed file system: Architecture and design”, 2007.

[4] GHEMAWAT, Sanjay; GOBIOFF, Howard; LEUNG, Shun-Tak. The Google file system. In: ACM SIGOPS Operating Systems Review. ACM, 2003. pp. 29-43.

[5] Asaha Ishii, Yonghwan Kim, Junya Nakamura, Fukuhito Ooshita, Hirotsugu Kakugawa, and Toshimitsu Masuzawa, “Evaluation of Hadoop system consisting of Virtual Machines on Multi-core CPUs”, High Performance Computing (HPC) 2012-HPC-136(20), pp. 1-7, 2012.

[6] Masashi Sakurai, “Studies on Evaluating Performance Efficiency of Distributed File Systems”, 2011.

[7] Asuka Momose and Masato Oguchi, “Characterization of Remote Data Access for Hadoop Distributed File System over a long-latency environment”, The institute of Electronics, Information and Communication Engineers DE lab & PRMU lab, 6: 19-24, 2011.

[8] Shohei Okudera, Daisaku Yokoyama, Miyuki Nakano, and Masaru Kitsuregawa, “Consideration on Ad hoc query processing with Adaptive Index in Map Reduce Environment”, Technical report of IEICE, The Institute of Electronics, Information and Communication Engineers, 2012.

[9] Jeffrey Dean and Sanjay Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Sixth Symposium on Operating System Design and Implementation (OSDI'04), 2004.

[10] Fay Chang, et al., “Bigtable: A Distributed Storage System for Structured Data”, ACM Transactions on Computer Systems (TOCS), vol. 26 Issue 2, 2008.

[11] “Construction of the Hadoop storage environment by EMC Isilon scale out NAS”, 2012.

[12] “High availability and data security of EMC ISILON scale out NAS”, 2012.

[13] Tom White, “Hadoop: The Definitive Guide, 3rd Edition”, O'Reilly Media / Yahoo Press, 2012.

ISBN: 978-1-61208-311-7

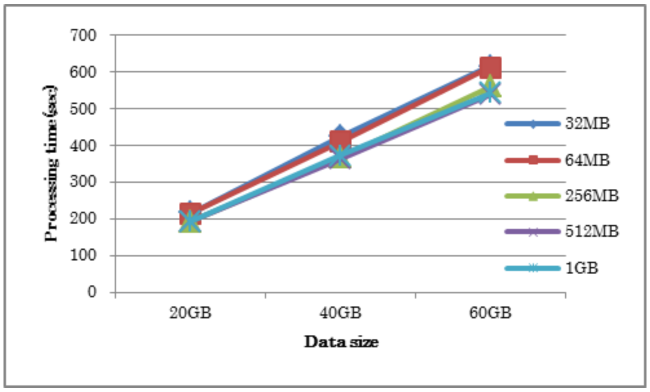


Figure 2. Teragen(HDFS).

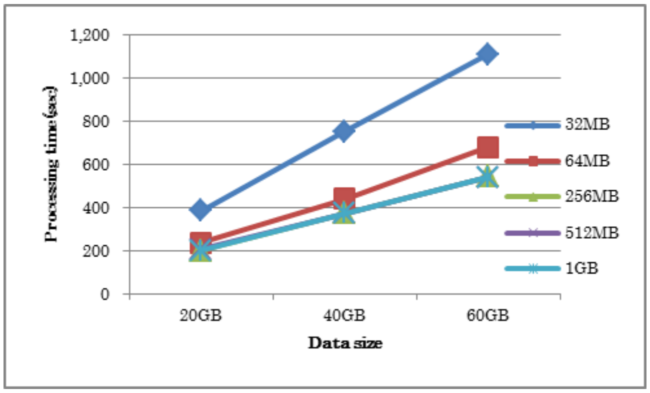


Figure 3. Grep(HDFS).

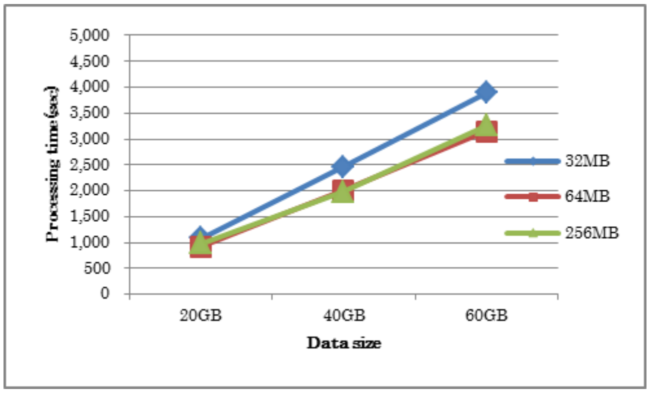


Figure 4. Terasort(HDFS).

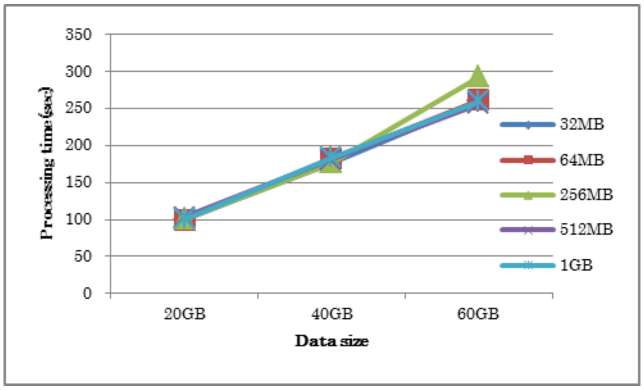


Figure 5. Teragen(Isilon).

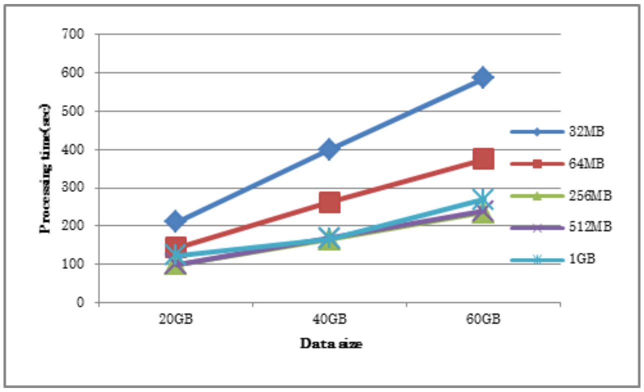


Figure 6. Grep(Isilon).

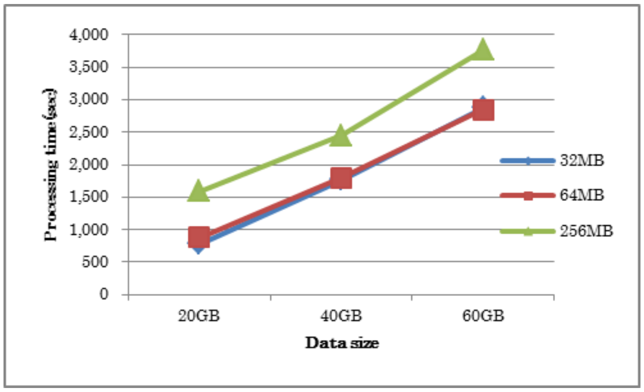


Figure 7. Terasort(Isilon).

Exploring the Ratings Prediction Task in a Group Recommender System that Automatically Detects Groups

Ludovico Boratto and Salvatore Carta
Dip.to di Matematica e Informatica
Università di Cagliari
Via Ospedale 72
09124 Cagliari, Italy
Email: {ludovico.boratto, salvatore}@unica.it

***Abstract*—Recommender systems produce content for users, by suggesting items that users might like. Predicting the ratings is a key task in a recommender system. This is especially true in a system that works with groups, because ratings might be predicted for each user or for the groups. The approach chosen to predict the ratings changes the architecture of the system and what information is used to build the predictions. This paper studies approaches to predict the ratings in a group recommendation scenario that automatically detects groups. Experimental results confirm that the approach to predict the ratings strongly influences the performances of a system and show that building predictions for each user, with respect to building predictions for a group, leads to great improvements in the quality of the recommendations.**

***Keywords*—Group Recommendation; Clustering; Ratings Prediction.**

I. INTRODUCTION

A recommender system suggests items that might be interesting for a user. In order to identify “the useful items for the user, a recommender system must *predict* that an item is worth recommending” [1]. As highlighted in [1], [2], the prediction task is the core recommendation computation.

Group recommendation is designed for contexts in which more than a person is involved in the recommendation process [3]. A scenario in which group recommendation is useful is when the recommendations that can be built are limited.

> *A company decides to print recommendation flyers that present suggested products. Even if the data to produce a flyer with individual recommendations for each customer is available, printing a different flyer for everyone would be technically too hard to accomplish and costs would be too high. A possible solution would be to set a number of different flyers to print, such that the printing process could be affordable in terms of costs and the recipients of the same flyer would be interested by its content.*

With respect to classic group recommendation, these systems add the complexity of defining groups, in order to respect the constraint on the number of recommendations that can be produced. In literature, no system can automatically adapt to such constraints imposed by the system.

According to Jameson and Smyth [3], a group recommender system can use three approaches to predict the ratings: (i) construct group models and predict the ratings for each group using the model, (ii) predict the ratings for each user and merge only the individual recommendations into a group preference, or (iii) aggregate all the predictions built for each user into a group preference. It can be noticed that the ratings prediction task takes a central role also in a group recommender system, since ratings can be predicted for each user or for a group. According to the approach chosen to predict the ratings, the architecture of the system changes and the prediction task takes a different input (i.e., a group model or the individual preferences of each user) and produces a different output (i.e., predictions for a group, predictions for each user or recommendations for each user). This means that the flow of the computation radically changes, in order to allow the system to build the predictions.

This paper explores the ratings prediction task in the previously mentioned scenario, in order to identify the best approach to predict the ratings for a group that has been automatically detected. Three recommender systems have been developed, to produce the predictions according to the previously mentioned approaches. The scientific contributions coming from this paper are the following: (i) the prediction task is explored for the first time in a scenario in which groups are automatically detected; (ii) the three approaches to build the predictions in a group recommender systems are directly compared for the first time in literature and (iii) the trade-off between the number of detected groups and the accuracy of each system is explored, in order to evaluate how the number of groups affects the performances of a system.

The paper is structured as follows: Section II describes the ratings prediction approaches in group recommender systems; Section III presents the three group recommender systems that automatically detect groups, developed to use the three approaches to build the predictions; Section IV illustrates the experiments on the systems; Section V contains conclusions.

II. RATINGS PREDICTION APPROACHES IN GROUP RECOMMENDATION

According to [3], group preferences can be predicted using three approaches: (i) generation of a group model that combines individual preferences, used to build predictions for the group, (ii) merging of recommendations built for each user, or (iii) aggregation of the predictions built for each user.

This section describes each approach in detail.

A. Construction of Group Preference Models

This approach builds a group model using the preferences of the users and predicts a rating for the items not rated by the group using the model. The two performed tasks are:

1) Construct a model M_g for a group g, that represents the preferences of the whole group.
2) For each item i not rated by the group, use M_g to predict a rating p_{gi}.

The architecture of a system that uses this approach is shown in Fig. 1 (the prediction task is highlighted in the figure). In order to build the predictions, the system has to produce a model with the preferences of the group (TASK 1). The task receives as input the ratings for the items evaluated by each user (INPUT 1) and the composition of each group (INPUT 2). Each group model is used to predict the ratings for the group (TASK 2).

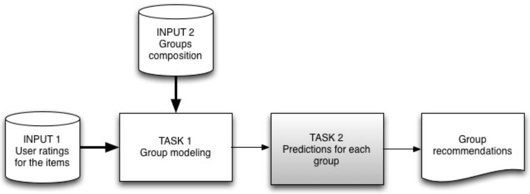


Fig. 1. System that builds predictions using group models

This approach is used by *Let's Browse* [4], *In-Vehicle Multimedia Recommender* [5], *TV4M* [6], *INTRIGUE* [7], and *Travel Decision Forum* [8].

B. Merging of Individual Recommendations

The approach presents to a group a set of items, that is the merging of the items with the highest predicted rating for each member of the group. The approach works as follows.

1) For each member of the group u:
    - For each item i not rated, predict a rating p_{ui}.
    - Select the set C_i of items with the highest predicted ratings p_{ui}.
2) Model the group preferences by producing $\bigcup_i C_i$, the union of the sets of items with the highest predicted ratings.

The architecture of a system that uses this approach is shown in Fig. 2. The system uses the ratings for the items evaluated by each user (INPUT 1) to predict the ratings for each user (TASK 1). The output produced by the task (the top-n predictions, used as recommendations for a user), are given as input to the task that merges the recommendations (TASK 2), along with the composition of the groups (INPUT 2).

This approach is not widely used in literature. The main relevant work that embraces this approach is *PolyLens* [9].

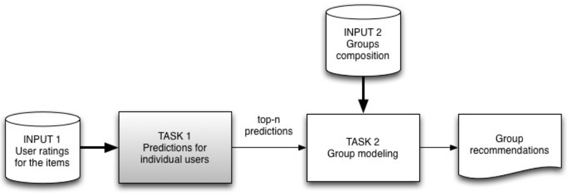


Fig. 2. System that merges the recommendations

C. Aggregation of Individual Predictions

This approach predicts individual preferences for the items not rated by each user, aggregates individual preferences and derives a group preference. The approach works as follows.

- For each item i:
    1) For each member u of the group g that did not rate i, predict a rating p_{ui}.
    2) Calculate an aggregate rating r_{gi} from the ratings of the members of the group, either expressed (r_{ui}) or predicted (p_{ui}).

The architecture of a system that uses this approach is shown in Fig. 3. The ratings for the item evaluated by each user (INPUT 1) are used to predict the ratings for the missing items for each user (TASK 1). The output produced by the task (i.e., all the calculated predictions), is given as input along with the ratings given by the users for the items (INPUT 1) and the composition of each group (INPUT 2) to the task that models the group preferences (TASK 2).

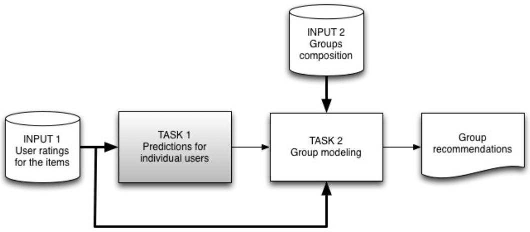


Fig. 3. System that aggregates predictions

This approach is used by *PolyLens* [9] and in [10], [11].

III. PREDICTING RATINGS FOR AUTOMATICALLY DETECTED GROUPS

As mentioned in the Introduction, no group recommendation approach is able to detect groups in order to adapt to constraints on the number of recommendations produced. This section presents three group recommender systems able to automatically detect groups. Each system will implement one of the three approaches to predict the ratings previously described. The tasks that do not predict ratings will be implemented in the same way in all the systems, in order to evaluate how each approach to predict the ratings affects the performances of a group recommender system.

A. System Based on Group Models Construction

ModelBased is a group recommender system that detects groups of similar users, models each group using the preferences of its members and predicts group preferences using the model, according to the approach presented in Section II-A. The tasks performed by the system are the following.

1) *Detection of the groups.* Using the preferences of each user, groups of users with similar preferences are detected with the k-means clustering algorithm [12].
2) *Group modeling.* Once groups have been detected, a group model is built for each group g, using the *Additive Utilitarian* modeling strategy. For each group, a rating is calculated for a subset of items.
3) *Prediction of the Ratings Using a Group Model.* Group ratings are predicted for the items not modeled by the previous task, using the model that contains its preferences with an Item-based approach [13].

Detection of the groups. The set of users has first to be partitioned into a number of groups equal to the number of recommendations. Since in our application scenario groups do not exist, unsupervised classification (*clustering*) is necessary. Users are clustered considering the ratings expressed for the evaluated items. It was recently highlighted in [14] that the k-means clustering algorithm [12] is by far the most used clustering algorithm in recommender systems.

This task detects groups by clustering users with the k-means clustering algorithm. The output of the task is a partitioning of the users in groups (clusters), such that users with similar ratings for the same items are in the same group and can receive the same recommendations.

Group modeling. To create a model that represents the preferences of a group, the *Additive Utilitarian* group modeling strategy [15] is adopted. The strategy sums individual ratings for each item and produces a list of the group ratings (the higher the sum is, the earlier the item appears in the list). The ranked group list of items is exactly the same that would be produced when averaging the individual ratings, so this strategy is also called 'Average strategy'.

The choice to use this strategy to create the model and produce ratings using an average was made for two main reasons: (i) since the considered scenario deals with a limited number of recommendations, the system works with large groups. Therefore, an average, that is a single value that is meant to typify a set of different values, is best way to put together the ratings in this context; (ii) for groups created with the k-means clustering algorithm, creating a group model with an average of the individual values for each item is like re-creating the centroid of the cluster, i.e., a super-user that connects every user of the group.

After a group has been modeled, a rating r_{gi} is a part of the model only if a consistent part of the group has rated item i. In fact, if an item is rated by a small part of the group, the aggregate rating cannot be considered representative of the preferences of the group as a whole. So, a parameter named *coratings*, is set. The parameter expresses the minimum percentage of group members who have to rate an item, in order to include the rating in the model.

Prediction of the Ratings Using a Group Model. In the group models previously created, for a subset of items there is no preference. In order to predict these ratings, an Item-Based Nearest Neighbor Collaborative Filtering algorithm presented in [13], that builds the predictions using the model, is adopted. The choice of using an Item-Based approach is because the algorithm deals with group models. Since groups might be very large, a group model might put together a lot preferences and it would not be significant to make a prediction with a User-based approach that would look for "similar groups". In fact, considering an example with 6000 users and 10 groups, if groups were homogeneous, there would be around 600 users per group. If a User-Based approach was used, when looking for neighbors the algorithm would look for a two similar models, that each contain a synthesis of the preferences of 600 users. This type of similarity would not be accurate enough to make predictions.

The algorithm predicts a rating p_{gi} for each item i that was not evaluated by a group g, considering the rating r_{gj} of each similar item j rated by the group. Equation (1) gives the formula used to predict the ratings:

$$p_{gi} = \frac{\sum_{j \in ratedItems(g)} itemSim(i,j) \cdot r_{gj}}{\sum_{j \in ratedItems(g)} itemSim(i,j)} \tag{1}$$

According to Schafer et al., [13], some authors do not consider all the items in the model (i.e., $ratedItems(g)$), but just the top n correlations. In order to reduce the computational complexity of the algorithm and select the most meaningful correlations, this is the approach used for this task. In order to compute similarity $itemSim(i,j)$ between two items, adjusted-cosine similarity is used. The metric is believed to be the most accurate when calculating similarities between items [13]. It is computed considering all users who rated both item i and item j. Equation (2) gives the formula for the similarity (U_{ij} is the set of users that rated both item i and j and value $\overline{r}_u$ represents the mean of the ratings expressed by user u).

$$itemSim(i,j) = \frac{\sum_{u \subset U_{ij}} (r_{ui} - \overline{r}_u)(r_{uj} - \overline{r}_u)}{\sqrt{\sum_{u \subset U_{ij}} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \subset U_{ij}} (r_{uj} - \overline{r}_u)^2}} \tag{2}$$

B. System that Merges Individual Recommendations

MergeRecommendations is a group recommender system that detects groups of similar users, predicts individual preferences and selects the items with the highest predicted ratings for each user, using the approach presented in Section II-B. Here we describe the tasks performed by the system and how they have been implemented.

1) *Detection of the groups.* Considering the individual preferences, groups of similar users are detected with the k-means clustering algorithm.
2) *Predictions for Individual Users.* Individual predictions are calculated for each user with a User-Based Collaborative Filtering Approach presented in [13].
3) *Generation of the Group Predictions (Group modeling).* Group predictions are built by modeling the top-n items with the highest predicted ratings for each user, by averaging the ratings of the items selected for each user.

ISBN: 978-1-61208-311-7

Detection of the Groups. The first task uses the approach previously presented, i.e., the k-means algorithm.

Prediction of the Missing Ratings. Ratings for the members of a group are predicted with a classic User-Based Nearest Neighbor Collaborative Filtering algorithm, presented in [13]. The algorithm predicts a rating p_{ui} for each item i that was not evaluated by a user u, considering the rating r_{ni} of each similar user n for the item i. A user n similar to u is called a *neighbor* of u. Equation (3) gives the formula used to predict the ratings:

$$p_{ui} = \overline{r}_u + \frac{\sum_{n \subset neighbors(u)} userSim(u,n) \cdot (r_{ni} - \overline{r}_n)}{\sum_{n \subset neighbors(u)} userSim(u,n)} \quad (3)$$

Values $\overline{r}_u$ and $\overline{r}_n$ represent, respectively, the mean of the ratings expressed by user u and user n. Similarity $userSim()$ between two users is calculated using the Pearson's correlation, a coefficient that compares the ratings of all the items rated by both the target user and the neighbor. Pearson' correlation between a user u and a neighbor n is given in Equation (4) (I_{un} is the set of items rated by both u and n).

$$userSim(u,n) = \frac{\sum_{i \subset I_{un}} (r_{ui} - \overline{r}_u)(r_{ni} - \overline{r}_n)}{\sqrt{\sum_{i \subset I_{un}} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \subset I_{un}} (r_{ni} - \overline{r}_n)^2}} \quad (4)$$

The metric ranges between 1.0 (complete similarity) and -1.0 (complete dissimilarity). Negative values do not increase the prediction accuracy [16], so they are discarded by the task.

Generation of the Group Predictions (Group modeling). For each user, the items for which a rating is predicted are ranked in descending order based on the ratings, then the top-n items are selected. Group ratings are predicted by merging the top-n items of each users with a union of the ratings. If an item appears in the list of more members of the same group, the average of the predicted ratings for that item is calculated (*Additive Utilitarian* strategy), in order to derive the preference of that group for the item.

C. System Based on the Aggregation of Individual Predictions

PredictionAggregation is a group recommender system that detects groups of similar users, predicts individual preferences and aggregates the preferences expressed for each item into a group preference, according to the approach presented in Section II-C. Here we describe the tasks performed by the system.

1) *Detection of the groups.* Using individual preferences, groups are detected through the k-means algorithm.
2) *Predictions for Individual Users.* Predictions are built with the previously described User-Based approach.
3) *Aggregation of the Predictions (Group modeling).* Once groups have been detected, a group model is built by aggregating all the predictions of a group.

All the tasks use the same algorithms previously presented, i.e., the k-means clustering algorithm, the User-Based Collaborative Filtering algorithm and the *Additive Utilitarian* modeling strategy. The difference with the *MergeRecommendation* system is on the modeling task, that considers all the predictions and not just the top-n predicted items.

IV. Experimental Framework

This section presents the framework built for the experiments.

A. Experimental Setup

To conduct the experiments, we adopted MovieLens-1M, a dataset widely used in literature.

The clusterings with k-means were created using a testbed program called KMlocal [17], that contained a variant of the k-means algorithm, called *EZ Hybrid*. The k-means algorithm minimizes the *average distortion*, i.e., the mean squared distance from each point to its nearest center. With the dataset used, *EZ Hybrid* is the algorithm that returned the lowest distortion and is the one used to cluster the users.

An analysis has been performed, by comparing the RMSE values obtained by each system considering different numbers of groups to detect. The choice of measuring the performances for different numbers of groups has been made to show how the quality of the systems change as the constraint changes. In each experiment, four different clusterings in 20, 50, 200 and 500 groups were created. Moreover, we compared the results obtained with the four clusterings with the results obtained considering a single group with all the users (i.e., predictions are calculated considering the preferences of all the users), and the results obtained by the system that calculates predictions for each user.

RMSE was chosen as a metric to compare the algorithms because, as the organizers of the Netflix prize highlight [19], it is widely used, allows to evaluate a system through a single number and emphasizes the presence of large errors.

In order to evaluate if two RMSE values returned by two experiments are significantly different, independent-samples two-tailed Student's t-tests have been conducted. In order to make the tests, a 5-fold cross-validation was preformed.

The details of the experiments are described below.

1) *Parameters setting.* For each system, a parametric analysis has been conducted, in order to find the setting that allows to achieve the best performances.
2) *Selection of the best system.* The performances of the systems have been compared, in order to identify the one that allows to predict the most accurate ratings.

B. Dataset and Data Preprocessing

The dataset used, i.e., MovieLens-1M, is composed of 1 million ratings, expressed by 6040 users for 3900 movies. This framework uses only the file `ratings.dat`, that contains the ratings given by users. The file contains four features: *UserID*, that contains user IDs in a range between 1 and 6040, *MovieID*, that contains movie IDs in a range between 0 and 3952, *Rating*, that contains values in a scale between 1 and 5 and *Timestamp*, that contains a timestamp of when a user rated an item. The file was preprocessed for the experimentation, by mapping the feature *UserID* in a new set of IDs between 0 and 6039, to facilitate the computation using data structures. In order to conduct the cross-validation, the dataset was split into five subsets with a random sampling technique (each subset contains 20% of the ratings).

C. Metrics

The quality of the predicted ratings was measured through the Root Mean Squared Error (RMSE). The metric compares each rating r_{ui}, expressed by a user u for an item i in the test set, with the rating p_{gi}, predicted for the item i for the group g in which user u is. The formula is shown below:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(r_{ui} - p_{gi})^2}{n}}$$

where n is the number of ratings available in the test set. In order to compare if two RMSE values returned by two experiments are significantly different, independent-samples two-tailed Student's t-tests have been conducted. These tests allow to reject the null hypothesis that two values are statistically the same. So, a two-tailed test will test if an RMSE value is significantly greater or significantly smaller than another RMSE value. Since each experiment was conducted five times, the means M_i and M_j of the RMSE values obtained by two systems i and j are used to compare the systems and calculate a value t:

$$t = \frac{M_i - M_j}{s_{M_i - M_j}}$$

where

$$s_{M_i - M_j} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

s^2 indicates the variance of the two samples, n_1 and n_2 indicate the number of values considered to build M_1 and M_2 (in our case both are equal to 5, since experiments were repeated five times). In order to determine the $t - value$ that indicates the result of the test, the degrees of freedom for the test have to be determined:

$$\text{d.f.} = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1 - 1) + (s_2^2/n_2)^2/(n_2 - 1)}$$

Given t and $d.f.$, the $t - value$ (i.e., the results of the test), can be obtained in a standard table of significance as

$$t(d.f.) = t - value$$

The $t - value$ derives the probability p that there is no difference between the two means. Along with the result of a t-test, the standard deviation SD of the mean is presented.

D. Experiments

For each system, experiments to set the parameters and find the best configuration are conducted. Then, the performances of the different systems are compared.

1) Setting the Parameters of ModelBased: Here we describe the experiments conducted to set the two parameters of the system (i.e., $coratings$ and n).

***coratings* parameter setting.** The $coratings$ parameter allows to consider in the model only the items rated by a certain part of the group. An experiment to evaluate a suitable value for the parameter is conducted. In this experiment, parameter n is set to 10. Fig. 4 and the underlying table, show that the initial value of $coratings$, i.e., 10%, is the one that allows to achieve better results. This means that the higher is the value of $coratings$, the more ratings are eliminated for the model. So, it is harder to predict the ratings.

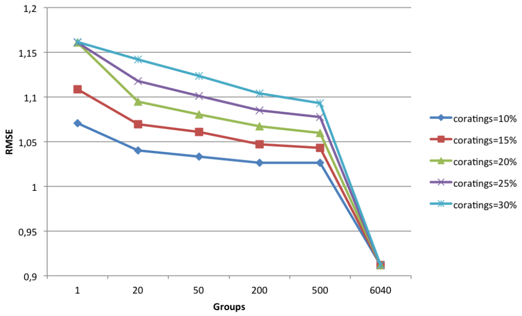


Fig. 4. RMSE for the different values of *coratings*

TABLE I. RMSE FOR THE DIFFERENT VALUES OF *coratings*

	1 group	20 groups	50 groups	200 groups	500 groups	6040 groups
coratings=10%	1.0706	1.0402	1.0335	1.0265	1.0262	0.9120
coratings=15%	1.1086	1.0696	1.0611	1.0471	1.0428	0.9120
coratings=20%	1.1608	1.0948	1.0804	1.0672	1.0597	0.9120
coratings=25%	1.1612	1.1178	1.1011	1.0849	1.0775	0.9120
coratings=30%	1.1617	1.1417	1.1233	1.1039	1.0930	0.9120

Independent-samples t-tests have been conducted, to compare the results for different values of $coratings$ in each clustering. All the tests returned that there is a significant difference in the values obtained with different values of the $coratings$ parameter. The results obtained to compare the results obtained considering 10% and 15% of the group are now presented. Considering 1 group, there is a significant difference in the RMSE values for $coratings = 10\%$ ($M = 1.070556$, $SD = 0.00$) and $coratings = 15\%$ ($M = 1.108634$, $SD = 0.00$); $t(7.85) = 20.26$, $p = 0.0$. For 20 groups, the difference is also significant when comparing the RMSE values for $coratings = 10\%$ ($M = 1.04019$, $SD = 0.00$) and $coratings = 15\%$ ($M = 1.069618$, $SD = 0.00$); $t(9.96) = 9.24$, $p = 0.0$. The test conducted for 50 groups returned a significant difference between $coratings = 10\%$ ($M = 1.033476$, $SD = 0.00$) and $coratings = 15\%$ ($M = 1.06113$, $SD = 0.00$); $t(7.11) = 15.24$, $p = 0.0$. With 200 groups, the obtained results are $coratings = 10\%$ ($M = 1.026542$, $SD = 0.00$) and $coratings = 15\%$ ($M = 1.047102$, $SD = 0.00$); $t(7.88) = 14.60$, $p = 0.0$. For 500 groups, there is a significant difference in the RMSE values for $coratings = 10\%$ ($M = 1.026246$, $SD = 0.00$) and $coratings = 15\%$ ($M = 1.042848$, $SD = 0.00$); $t(7.68) = 13.80$, $p = 0.0$. The results suggest that lowering the $coratings$ value allows to substantially improve the results. Specifically, these results suggest that the less ratings are removed from the model, the better the algorithm predicts the ratings for a group.

Setting parameter *n*. To predict a rating for the group, the items most similar to the one currently predicted are selected. In order to choose the number of neighbors, a parameter n has to be set. Parameter $coratings$ is set to 10%. Fig. 5 and the underlying table, show the performances of the system for different values of n, i.e., considering a different number of similar items. In the results reported in the figure it is hard to see the value that allows to obtain the best results, so Fig. 6 (that focuses on the part between 20 and 500 groups) shows an improvement up to $n = 20$, then results worsen again.

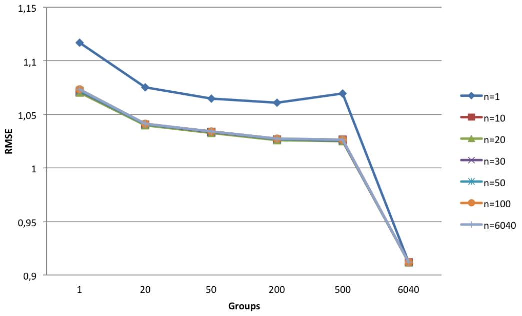


Fig. 5. RMSE for the different values of *n*

TABLE II. RMSE FOR THE DIFFERENT VALUES OF *n*

	1 group	20 groups	50 groups	200 groups	500 groups	6040 groups
n=1	1.1171	1.0752	1.0648	1.0610	1.0694	0.9120
n=10	1.0706	1.0402	1.0335	1.0265	1.0262	0.9120
n=20	1.0705	1.0398	1.0327	1.0257	1.0249	0.9120
n=50	1.0722	1.0410	1.0334	1.0267	1.0256	0.9120
n=100	1.0732	1.0412	1.0341	1.0274	1.0265	0.9120
n=6040	1.0731	1.0413	1.0342	1.0275	1.0266	0.9120

RMSE values are very close; so, it is important to conduct independent-samples t-tests to evaluate the difference between the results. In particular, the tests conducted to compare 20 and 30 groups are reported. For 1 group, there is a difference in the RMSE values for $n = 20$ ($M = 1.070534$, $SD = 0.00$) and $n = 30$ ($M = 1.07217$, $SD = 0.00$); $t(9.92) = 0.87$, $p = 0.41$. Considering 20 groups, there is a difference in the results obtained with $n = 20$ ($M = 1.039798$, $SD = 0.00$) and $n = 30$ ($M = 1.040968$, $SD = 0.00$); $t(7.17) = 0.40$, $p = 0.70$. The test conducted for 50 groups returned a difference between $n = 20$ ($M = 1.03344$, $SD = 0.00$) and $n = 30$ ($M = 1.033898$, $SD = 0.00$); $t(7.36) = 0.49$, $p = 0.63$. With 200 groups, there is a difference between the RMSE values obtained with $n = 20$ ($M = 1.026698$, $SD = 0.00$) and $n = 30$ ($M = 1.026764$, $SD = 0.00$); $t(7.31) = 0.74$, $p = 0.48$. For 500 groups, the test returned a difference between $n = 20$ ($M = 1.02493$, $SD = 0.00$) and $n = 30$ ($M = 1.025626$, $SD = 0.00$); $t(7.94) = 0.67$, $p = 0.52$. The results of the t-tests show that there is not enough confidence to reject the null hipotesys that the values obtained for $n = 20$ and $n = 30$ are different. However, the results obtained with $n = 20$ are always better in terms of RMSE and the t-tests returned that the probability that there is a difference for $n = 20$ ranges between 30% and 59%. So the value of n used is 20.

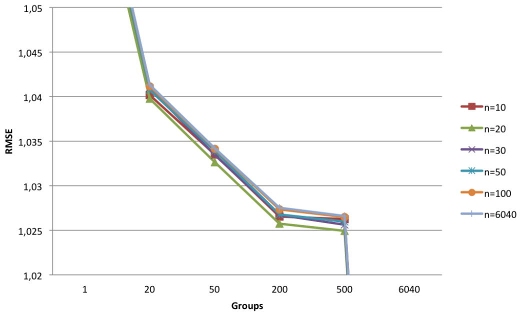


Fig. 6. Detail to study parameter *n*

2) Setting the Parameters of MergeRecommendations: Here, the experiments conducted to set the two parameters used by the system (i.e., $neighbors$ and n) are presented.

Selection of the number of *neighbors*. In order to predict a rating for a user, the users most similar to the one considered are selected. In order to do so, the right number of neighbors has to be selected when computing a prediction. This is done with a parameter called $neighbors$, tested in this set of experiments. Since we have to evaluate the number of neighbors for an algorithm that predicts individual ratings, this evaluation is done out of the group recommendation context. Fig. 7 and the underlying table, show the RMSE values for increasing values of $neighbors$. As highlighted in [18], this is the common way to choose the value. Moreover, our results reflect the trend described by the authors, i.e., for low values of the parameter, great improvements can be noticed. As expected, RMSE takes the form of a convex function (Fig. 8 shows a particular of Fig. 7), that indicates that after a certain value improvement stops. In these experiments, that value is 100.

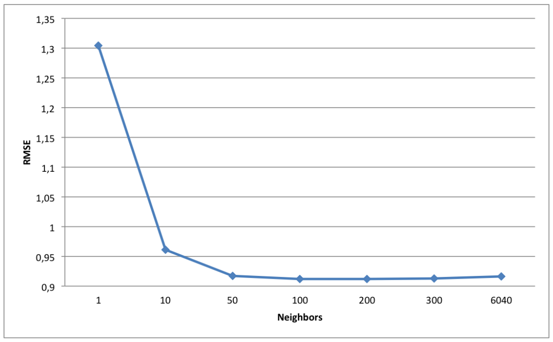


Fig. 7. RMSE for increasing number of *neighbors*

TABLE III. RMSE FOR INCREASING NUMBER OF *neighbors*

	6040 groups
neighbors=1	1.3046
neighbors=10	0.9611
neighbors=50	0.9167
neighbors=100	0.9118
neighbors=200	0.9120
neighbors=300	0.9128
neighbors=6040	0.9160

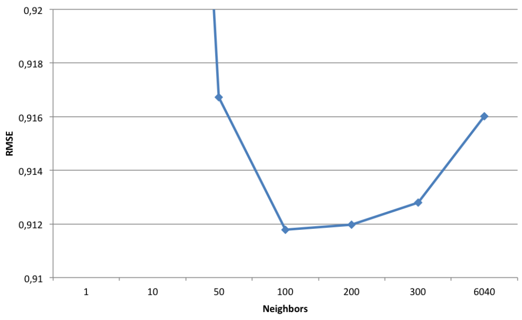


Fig. 8. RMSE takes the form of a convex function.

Independent-samples t-tests, conducted to evaluate the difference between the results obtained between 100 and the

ISBN: 978-1-61208-311-7

other numbers of neighbors, are now presented. There is a significant difference in the RMSE values for 1 neighbor ($M = 1.304622$, $SD = 0.00$) and 100 neighbors ($M = 0.911785$, $SD = 0.00$); $t(7.59) = 450.02$, $p = 0.00$. There is a also a significant difference in the RMSE values for 10 neighbors ($M = 0.961122$, $SD = 0.00$) and 100 neighbors ($M = 0.911785$, $SD = 0.00$); $t(7.41) = 54.44$, $p = 0.00$. A significant difference is also present in the RMSE values for 50 neighbors ($M = 0.916725$, $SD = 0.00$) and 100 neighbors ($M = 0.911785$, $SD = 0.00$); $t(7.97) = 6.02$, $p = 0.00$. The RMSE values present a difference for 100 neighbors ($M = 0.911785$, $SD = 0.00$) and 200 neighbors ($M = 0.911968$, $SD = 0.00$); $t(7.99) = 0.24$, $p = 0.82$. There is a also a difference in the RMSE values for 100 neighbors ($M = 0.911785$, $SD = 0.00$) and 300 neighbors ($M = 0.912803$, $SD = 0.00$); $t(7.97) = 1.06$, $p = 0.33$. There is a significant difference in the RMSE values for 100 neighbors ($M = 0.911785$, $SD = 0.00$) and 6040 neighbors ($M = 0.916022$, $SD = 0.03$); $t(7.99) = 1.27$, $p = 0.24$. For values of $neighbors$ higher than 100, the probability that there is a difference between the values obtained for 100 and 200 neighbors and 100 and 300 neighbors is between 18% and 67%. In particular, there seems to be no difference between choosing 100 and 200 neighbors. Since it is faster to compute predictions considering 100 neighbors instead of 200, $neighbors = 100$ is the value chosen for the algorithm.

Choice of the top-n items. This set of experiments evaluates how big the list of recommendations made for each user (i.e., the top-n) has to be, by testing parameter n. Fig. 9 and the underlying table, show that the choice of the top-5 ratings brings to the best results.

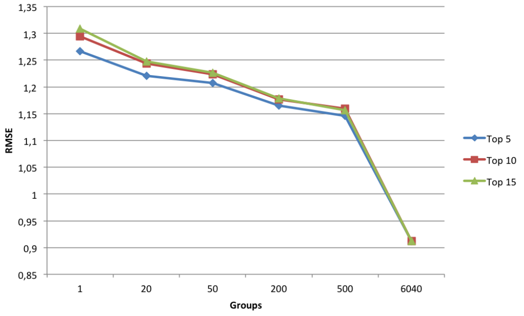


Fig. 9. RMSE values for increasing values of top n ratings

TABLE IV. RMSE FOR THE DIFFERENT VALUES OF THE TOP n RATINGS

	1 group	20 groups	50 groups	200 groups	500 groups	6040 groups
Top 5	1.2667	1.2207	1.2075	1.1653	1.1461	0.9120
Top 10	1.2947	1.2437	1.2235	1.1762	1.1592	0.9120
Top 15	1.3092	1.2473	1.2262	1.1788	1.1562	0.9120

Independent-samples t-tests have been conducted, in order to evaluate if there is a significant difference between the values obtained for the different values of n and different numbers of groups. Such a difference exists and the results of the tests that compare $n = 5$ and $n = 10$ (i.e., the values that obtained the most similar results) are now presented. Considering 1 group, there is a significant difference between $n = 5$ ($M = 1.266718$, $SD = 0.00$) and $n = 10$ ($M = 1.294696$, $SD = 0.00$); $t(7.74) = 3.39$, $p = 0.01$. For 20 groups, there is a difference between $n = 5$ ($M = 1.220748$, $SD = 0.00$) and $n = 10$ ($M = 1.243682$, $SD = 0.00$); $t(7.99) = 1.46$, $p = 0.18$. When 50 groups are considered, there is a difference between $n = 5$ ($M = 1.207548$, $SD = 0.00$) and $n = 10$ ($M = 1.223508$, $SD = 0.00$); $t(7.98) = 0.65$, $p = 0.53$. For 200 groups, there is also a difference between $n = 5$ ($M = 1.16532$, $SD = 0.00$) and $n = 10$ ($M = 1.176224$, $SD = 0.00$); $t(7.99) = 0.54$, $p = 0.60$. With 500 groups, there is a difference between $n = 5$ ($M = 1.146128$, $SD = 0.00$) and $n = 10$ ($M = 1.159176$, $SD = 0.00$); $t(7.45) = 0.65$, $p = 0.54$. Results show that when the number of groups increases, the significance of the difference between the values decreases. Since for $n = 5$ the results are always lower and the highest probability that the values are not different is 40%, the value was chosen for the system.

3) Setting the Parameters of PredictionsAggregation: Since the algorithm used by $PredictionsAggregation$ to predict individual ratings is the same used by $MergeRecommendations$ and it was already tested, no experiments to set the parameters have to be conducted. The system was run with the previously tested value of the $neighbors$ parameter, i.e., $neighbors = 100$ and results are shown in Fig. 10 and the underlying table.

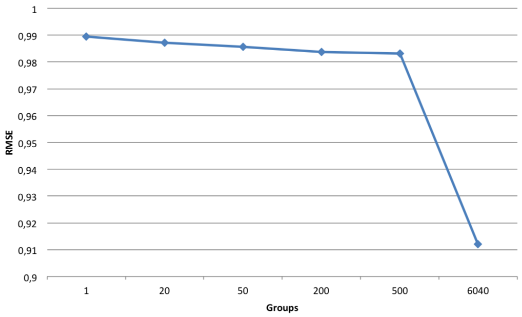


Fig. 10. RMSE values of *PredictionsAggregation*

TABLE V. RMSE VALUES OF *PredictionsAggregation*

	1 group	20 groups	50 groups	200 groups	500 groups	6040 groups
RMSE	0.9895	0.9872	0.9857	0.9837	0.9832	0.9120

4) Selection of the best system: Fig. 11 and the underlying table, report the results obtained by each system with its best configuration. An aspect not previously deepened in the previous experiments, is that for all the systems, as the number of groups grows, the quality of the results improves. So as the number of groups increases, the RMSE values get lower. This means that the systems can have better performances when more recommendations can be produced.
The results obtained by three approaches show how the prediction tasks affects the quality of group recommendation. $MergeRecommendations$, the system that merges individual recommendations achieves the worst results. This is the sign that with automatically detected groups, if the preferences of a user are expressed just with a small subset of items (in this case five), a group recommendation algorithm is not able to properly satisfy users. The approach based on a group model (i.e., $ModelBased$) lays in the middle of

the figure. So, building predictions using a group model that uses an average to calculate predictions does not allow to capture the individual preferences and predict significant group ratings. At the bottom of the figure, with the best results, there is the system that merges individual preferences (i.e., $PredictionsAggregation$). This means that predicting the ratings for each user and considering all the predictions in the group models leads to great improvements in the quality of the results. Independent-samples t-tests confirm that there is a significant difference between the RMSE values of $PredictionsAggregation$ and the ones obtained by the other systems. Results of the tests are not presented to facilitate the reading of the paper.

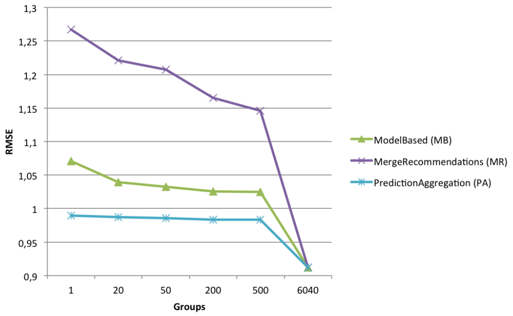


Fig. 11. RMSE obtained by the each system

TABLE VI. RMSE OBTAINED BY THE EACH SYSTEM

	1 group	20 groups	50 groups	200 groups	500 groups	6040 groups
MB	1.0705	1.0398	1.0327	1.0257	1.0249	0.9120
MR	1.2667	1.2207	1.2075	1.1653	1.1461	0.9120
PA	0.9895	0.9872	0.9857	0.9837	0.9832	0.9120

V. CONCLUSIONS AND FUTURE WORK

This paper explored the ratings prediction task in group recommendation scenario in which groups are automatically detected. The experiments conducted allowed to achieve important contributions, now recapped.

- Exploring the ratings prediction task in this scenario allowed to understand that the use of all the predictions built for each user allows to achieve better results, with respect to the approaches that build predictions for a group or use the individual recommendations.
- Results improve as the number of groups grows. Analyzing the performances for different numbers of groups allowed to explore the trade-off between the number of recommendations built and the accuracy of the system.

Future work will focus on improving the cohesion of the groups. This will be done by adding more information to the input of the clustering algorithm, in order to detect groups with more homogeneous preferences and to produce more accurate group recommendations.

ACKNOWLEDGMENT

This work is partially funded by Regione Sardegna under project CGM (Coarse Grain Recommendation) through Pacchetto Integrato di Agevolazione (PIA) 2008 “Industria Artigianato e Servizi”.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in *Recommender Systems Handbook*. Berlin: Springer, 2011, pp. 1–35.

[2] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[3] A. Jameson and B. Smyth, “Recommendation to groups,” in *The Adaptive Web, Methods and Strategies of Web Personalization*. Springer, 2007, pp. 596–627.

[4] H. Lieberman, N. W. V. Dyke, and A. S. Vivacqua, “Let’s browse: A collaborative web browsing agent,” in *IUI*, 1999, pp. 65–68.

[5] Y. Zhiwen, Z. Xingshe, and Z. Daqing, “An adaptive in-vehicle multimedia recommender for group users,” in *Proceedings of the 61st Semiannual Vehicular Technology Conference*, vol. 5, 2005, pp. 2800–2804. [Online].

[6] Z. Yu, X. Zhou, Y. Hao, and J. Gu, “Tv program recommendation for multiple viewers based on user profile merging,” *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63–82, Mar. 2006. [Online].

[7] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso, “Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices,” *Applied Artificial Intelligence*, vol. 17, no. 8-9, pp. 687–714, 2003.

[8] A. Jameson, “More than the sum of its members: challenges for group recommender systems,” in *Proceedings of the working conference on Advanced visual interfaces*. ACM Press, 2004, pp. 48–54.

[9] M. O’Connor, D. Cosley, J. A. Konstan, and J. Riedl, “Polylens: A recommender system for groups of user,” in *Proceedings of the Seventh European Conference on Computer Supported Cooperative Work*. Kluwer, 2001, pp. 199–218.

[10] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu, “Group recommendation: Semantics and efficiency,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 754–765, 2009.

[11] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, “Group recommending: A methodological approach based on bayesian networks,” in *Proceedings of the 23rd International Conference on Data Engineering Workshops, ICDE 2007*. IEEE Computer Society, 2007, pp. 835–844.

[12] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. University of California Press, 1967, pp. 281–297.

[13] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The Adaptive Web, Methods and Strategies of Web Personalization*. Springer, 2007, pp. 291–324.

[14] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, “Data mining methods for recommender systems,” in *Recommender Systems Handbook*. Springer, 2011, pp. 39–71.

[15] J. Masthoff, “Group recommender systems: Combining individual models,” in *Recommender Systems Handbook*. Springer, 2011, pp. 677–702.

[16] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, “An algorithmic framework for performing collaborative filtering,” in *Research and Development in Information Retrieval*. American Association of Computing Machinery, 8/1999 1999. [Online].

[17] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 881–892, July 2002. [Online].

[18] C. Desrosiers and G. Karypis, “A comprehensive survey of neighborhood-based recommendation methods,” in *Recommender Systems Handbook*. Berlin: Springer, 2011, pp. 107–144.

[19] Netflix Prize, “Frequently asked questions.” [Online]. Available: http://www.netflixprize.com/faq

When Diversity Is Needed... But Not Expected!

Sylvain Castagnos*, Armelle Brun* and Anne Boyer*
*KIWI team - LORIA
Campus Scientifique, B.P. 239
54506 Vandœuvre - France
Email: {sylvain.castagnos, armelle.brun, anne.boyer}@loria.fr

***Abstract*—Recent studies have highlighted the correlation between users' satisfaction and diversity within recommenders, especially the fact that diversity increases users' confidence when choosing an item. Understanding the reasons of this positive impact on recommenders is now becoming crucial. Based on this assumption, we designed a user study that focuses on the utility of this new dimension, as well as its perceived qualities. This study has been conducted on 250 users and it compared 5 recommendation approaches, based on collaborative filtering, content-based filtering and popularity, along with various degrees of diversity. Results show that, when recommendations are made explicit, diversity may reduce users' acceptance rate. However, it helps increasing users' satisfaction. Moreover, this study highlights the need to build users' preference models that are diverse enough, so as to generate good recommendations.**

Keywords*—*Recommender systems, diversity, user modeling, user study.

I. Introduction

Recommender systems aim at helping users during their information search, by suggesting items that fit their needs and preferences. Recommender systems, that have emerged two decades ago, have been much studied by academic researchers, and are now an indispensable part of most of web services. However, a paradox is remaining in recommender systems: most of recommender systems aim at maximizing the precision of the recommendations, but do not consider human factors, which have an important role in decision processes. For example, in 2009 the Netflix Prize [1] has been won by BellKor's Pragmatic Chaos team, after a three-year long competition. The mean quadratic error (RMSE) has been improved by two hundredth [2], [3]. However, the corresponding algorithm has never been used, as it has become obsolete due to the emergence of new interaction modes and new user behaviors [4].

During the same period, works focusing on users' acceptance and adoption of recommender systems have shown that a difference of 10% of the RMSE cannot be perceived by users [5], [6]. However, these studies highlight the influence of human factors on users' satisfaction. These factors can be users' confidence in the recommender, the explanations provided by the recommender or the need in diversity of recommendations. The work conducted in this paper focuses on this last factor: the diversity of the recommendations. The experiments conducted in 2010 by Castagnos *et al.* [5] aimed at identifying the steps of a user's decision process, when facing a recommender system. In that study, diversity, which has not been anticipated, has appeared as an important factor in decision processes, but the experiments conducted did not allow to quantify the importance of this factor.

The contributions of our work are the following: we conduct a new user study, that aims at better understanding the role and the impact of diversity in recommender systems; several recommender systems algorithms (Collaborative Filtering (CF), Content-Based Filtering (CBF) and Popularity-based filtering (POP)) are implemented, as well as two new hybrid algorithms (combining CF and CBF), that allow to tune the degree of diversity of the recommendations. Analyzing these last two algorithms relies on a five-level inter-group model (one group for each algorithm) and a two-level intra-user model (implicitly and explicitly provided recommendations) in the movie domain. The training dataset is made up of a complete description of more than 500 movies and more than 3000 users. The experiment has been conducted during a one-week period, where data about 250 users has been collected. These users have been randomly split in groups of 50 users.

This study confirms the positive influence of the diversity on users' satisfaction. It also surprisingly highlights the importance of building preference models with enough diversity between items, especially for the cold-start phase, by encouraging users to rate different items. In addition, this study shows that despite having a positive impact on users' satisfaction, diversity has to be used carefully, as too much diversity may result in users who do not understand the coherence of the recommendations provided.

This paper is organized as follows: Section II is an overview of the state of the art of diversity in recommender systems, from conception and evaluation points of view. Section III is dedicated to the presentation of our study and Section IV presents and discusses the results. The last section concludes this paper.

II. Related work

Diversity is an emerging research topic in recommender systems. Diversity is well known for playing an important role in the improving the interaction between users and information retrieval systems [7], [8]. However, the question about knowing why and how to improve the diversity remains open. Two main approaches are adopted in the literature. The first one analyzes the impact of diversity on users' behavior. The second one integrates diversity in machine learning algorithms from recommender systems. Both following subsections present these approaches.

A. Role and impact of diversity

Diversity in recommender systems has been defined by Smyth and McClave [9] as the opposite dimension to similarity. We choose to refine this definition by defining diversity as the measure that quantifies the dissimilarity within a set of items. Thus, the task of introducing diversity in a recommender system consists in finding the best set of items that are highly similar to users' known preferences by taking care to not freeze recommendations (if novelty is never introduced in recommendations) as well as taking care to recommend sets of items not too similar. The first case is referred to as intrinsic diversity, which avoids redundancy between the items to be recommended [10]. The second case is referred to as extrinsic diversity, which aims at alleviating the uncertainty due to data ambiguity or sparsity in user preference models, by recommending a large set of items [11]. In both cases, mechanisms used to introduce diversity rest on the same metrics (see section II-B). Note that a new classification of diversity has been recently proposed by Adomavicius and Kwon [12]. It distinguishes individual diversity and aggregated diversity, depending on if we are interested in generating recommendations to individuals, or to groups of users. Here, we focus on individual diversity.

The seminal work focusing on the role of diversity in recommender systems has been conducted on conversational recommender systems [7]. It has been the first to show that diversity improves the efficiency of recommendations. Works presented by Zhang and Hurley [13], or Lathia [14], even talk of user frustration when no diversity is provided. McGinty and Smyth [7] have also put forward the issues related to this dimension. For example, diversity does not have to be integrated in each recommendation step.

To thoroughly understand this last point, we focus on the two steps of an item selection process in information access systems [15]. In the first step, the user uses the system's interface to identify the pertinent criteria for his/her current search. The identification of these criteria is made, most of the time, by trial and errors, the recommendation cycles. The second phase aims at comparing all the possible solutions related to these criteria. Users unconsciously use a trial and error approach, by choosing an item, having a look to the related recommendations, then making a backward step. Once a first starting point is found, he/she continues his/her exploration of the set of items by using the recommendations (as well as the recommendations related to the recommendations chosen, etc.).

Given this behavioral model, we can admit that, as presented by McGinty and Smyth [7], diversity does not positively impact each recommendation cycle, especially in those where the user aims at increasing his confidence in the system. This conclusion is confirmed by Castagnos [5], which measured the evolution of diversity need through time.

Many discussions about the role of diversity have emerged these last years. McNee *et al.* [16] studied the limitations of precision measures used in recommender systems: less accurate recommendations may be more pertinent from the users point of view. They also focused on the difference between the diversity proposed to regular users and to new users. Several works also highlight that diversity is intrinsically present in collaborative filtering-based recommendations, through serendipity [17], [18]. In parallel, some works focused on the best way to present the recommendation, so as diversity is perceived by users [19], [20].

B. *Integrating diversity in recommender systems*

The design of a recommender systems can divided into thee parts: (1) implicit or explicit collection of user traces, left by users when interacting with the system (preferences, tastes, usage, context); (2) building user models, based on these traces; (3) exploiting these models and machine learning algorithms to determine the adequate set of recommendations to be proposed to the active user.

Recommender systems are generally split into two families [21]: collaborative and content filtering. Until recently, the approaches dedicated to the improvement of the diversity were developed in the frame of content filtering [22], [23], [24]. These mechanisms can be used directly on the metric, or/and on the clustering/ranking algorithm used to generate recommendations. These approaches aim at increasing the diversity at the level of the attributes of the items.

In [9], diversity is represented by several metrics, that rely on the similarity between items: the more the items are similar, the lower is the diversity between them. Similarity between two items is defined as the weighted sum of the similarities on the attributes (see (1)).

$$Similarity(i_1,i_2) = \frac{\sum_{j=1..n} w_j \ * \ sim_{attribute=j}(i_1,i_2)}{\sum_{j=1..n} w_j} \qquad (1)$$

Starting from this similarity metric, Smyth and McClave [9] has introduced two new diversity measures. The first one, called $Diversity$, computes the average dissimilarity within a class C, made up of m items. The second one is a relative diversity ($RelDiversity$), that computes the added value in terms of diversity of an item on a class of items C (see (2)).

$$RelDiversity(i,C) = \begin{cases} 0 \text{ if C = \{\},} \\ \frac{\sum_{j=1..m}(1-Similarity(i,c_j))}{m} \text{ otherwise.} \end{cases} \qquad (2)$$

These metrics have then been used in content-based filtering to reorder the recommendation list, according to a diversity criterion. Two main approaches have been proposed: clustering-based [25] and selection-based approaches [22]. In clustering-based approaches, the aim is to build an optimal class of items, compared to a diversity criterion (that corresponds to the maximal diversity). The selection-based methods integrate the diversity in the recommender systems, without decreasing precision. Bradley and Smyth have been the first to propose a greedy-based selection algorithm, to find the most similar items to a user query, which are also diverse by pairs [22]. This algorithm selects the K most similar items to a target item t (see (1)). The recommendation list is filled iteratively, by choosing at each step the best quality item (see (3)), until getting the $top - N$ recommendations ($K \ < \ N$).

$$Quality(i,t,C) \ = \ Similarity(i,t) \ * \ RelDiversity(i,C) \qquad (3)$$

The $top-N$ reordering algorithms, as in [22], are known for their tradeoff between speed and accuracy (including precision and diversity). Radlinski *et al.* propose 3 methods that rely on query reformulation, in order to increase diversity in the $top-N$ list [11]. Zhang and Hurley suggest to maximize the diversity while not decreasing the similarity; they view this task as a binary optimization problem [13].

In addition to these content-based algorithms, works have focused on a way to integrate diversity in collaborative filtering. Ziegler and McNee have proposed a generic formalism based on an intra-list similarity (ILS) and a $top-N$ selection, which can be used in several algorithms, such as collaborative filtering [26]. Said *et al.* [27] have studied a new way to integrate diversity in collaborative filtering, by adapting the clustering algorithms. These 3 last works are purely based on collaborative filtering, thus use similarity measures based on votes, not on attributes.

C. Discussion

The main goal of this paper is to understand thoroughly the impact and the utility of diversity during the interaction between users and recommender systems. The state of the art presented in section II-A has illustrated how complex this dimension is, without having a complete view of all its dimensions. Indeed, the studies conducted to this date have measured the impact of diversity on satisfaction *a posteriori*, by using questionnaires [6]. Even if many studies have measured the impact of diversity on users' satisfaction with content-based filtering [7], [13], [14] on one side or collaborative filtering [28], [29] on the other side, no user study has been conducted to understand the role of diversity by comparing these two families of algorithms. [17] has addressed the diversity dimension thanks to the serendipity of such algorithms, but the degree of diversity is not controlled, and not always guaranteed. In this paper, we thus propose to conduct a user study, that focuses on diversity an that allows to: compare different families of algorithms (collaborative filtering, content-based filtering, popularity-based filtering); study users' behavior: from the collection of traces to the choice of items to be recommended, and check if diversity only plays a role during the recommendation phase as suggested by the literature, or if it also impacts the process of user modeling; study users' perception of diversity and differences in users' behavior, when recommendations are implicitly or explicitly presented.

Section II-B has shown that no hybrid algorithms (collaborative and content-based) that allow an equilibrium between accuracy and diversity of recommendations exist. To cope with this lack, we took inspiration from [22] and [26] to design two new algorithms, that combine collaborative and content-based filtering, to get the desired level of diversity. We propose to compare five algorithms.

III. Experiment Setup

A. Support

We conducted our experiment in the domain of cinema for several reasons. First, it is quite easy to collect a large dataset related to movies, so as to observe users' behaviors in a realistic context. Second, movies have a great number of attributes and are rated very often by users, unlike some other types of items. At last, cinema is a popular domain users are familiar with. This maximizes the chances that users know enough items in the proposed lists.

For the needs of our experiment, we built a website [30] and paid attention to users' cognitive load by spreading the experiment on several pages.

We started by collecting as much data as we could about the content of more than 500 movies, which includes titles, summaries, pictures, trailers, average ratings of press and spectators (and the corresponding number of ratings), movie genres, actors and actresses, directors, writers, release dates, languages, runtimes, and the fact that they belong to a saga or not. This information allow users to recognize movies, and is used by our recommender system as a training set to implement content-based filtering algorithms.

Collaborative filtering requires individual ratings from a large number of users. For this reason, we collected ratings from 3,158 users for the training data. In order to do so, we first gathered all the ratings of Allociné's real users [31] for the 509 selected movies. Then, we cleaned the database so as each user provides at least 20 movie ratings, and each movies is rated by at least 20 users. These thresholds represent the minimum number of ratings estimated by [32] to reach a good recommendation precision and quality level with collaborative filtering algorithms. We had to build this training set by our own, instead of relying on MovieLens or NetFlix corpuses, so as to guarantee that it is always possible to compute similarities between movies whatever are the attributes used. The size of our corpus is quite comparable to MovieLens. Moreover, and contrary to MovieLens, we have provided a good distribution of movies in term of popularity. We selected movies by paying attention to the fact that they must have more than 200 ratings on IMDb [33], and we manually checked with a sample group of 20 users that all the selected movies are known by most of them. Likewise, we randomly selected movies among those matching our criteria, while ensuring a good distribution on the rating scale from 1 to 5 (and in particular a good representativeness among the top-250 and the bottom-100 of IMDb movies). The average rating from the 3,158 users of the training set is 3.66, with a standard deviation of 1.37. To summarize, the whole set of information on movies and ratings represents our training dataset. The latter has been created thank to APIs of IMDb and Allociné. Characteristics of this dataset are made explicit in Tab I.

TABLE I: FEATURES FROM THE TRAINING DATA

Type	Movies	Actors	Directors	Writers	Genres	Countries	Sagas	Ratings
Number	509	903	310	351	23	17	98	173,120

B. Algorithms

We used 3 algorithms from the state-of-the-art, called **POP**, **CBF** and **CF**. We also propose 2 new hybrid algorithms called **CFRD** and **CFFD**. The choice and the implementation of these algorithms have been motivated by a need of personalization in real time. We consequently had to choose algorithms that are known to be fast and precise, and adapted them to our architecture.

All the pieces of information provided by the volunteers of our user study constitute the test set. Recommendations are computed from the active user's profile and the training set. None of the data from the test set is included in the training set (i.e., our 250 users are different from those in the training set). In this way, recommendations are computed in the same conditions for all the 250 volunteers of this study.

POP is our baseline and recommends items randomly chosen among most popular items.

CBF. This algorithm recommends items in function of their similarities with items liked by the active user. In this case, we voluntarily focus on preference similarity, rather than recommendation diversity, to verify if users perceive a difference in comparison with other algorithms.

So as to compute the similarity between two movies (see (1)), we optimize weighting coefficients and similarity measures per attribute on our training set. The weighting coefficients on the different attributes are: $w_{date} = 0.5$; $w_{director} = 1$; $w_{actor} = 1$; $w_{genre} = 1.5$; $w_{language} = 0.25$; $w_{popularity} = 0.5$; $w_{saga} = 1$; $w_{scenarist} = 0.25$. Thus, as an example, the fact that two movies have the same director has two times more impact in the similarity computation than the fact that the released dates are closed from each other.

Similarity measures per attribute are defined as: $sim_{actor}(i_1, i_2) = \frac{\cap_{actors}}{\cup_{actors}}$; $sim_{genre}(i_1, i_2) = \frac{\cap_{genres}}{\cup_{genres}}$. The similarity for the release date is equal to 1 if the gap between the release dates is less than 5 years, 0 otherwise. The similarity for the popularity is equal to 1 if the two movies belong to the same popularity class (when the difference between the average ratings of these two movies is below a fixed threshold, and when the numbers of ratings for each of these movies are quite comparable). At last, similarities for the director, language, saga and writer are equal to 1 if the two movies have the same value for this attribute, 0 otherwise.

CF. We used an item-based collaborative filtering algorithm, as proposed in [34]. This algorithm transforms the user-item rating matrix in an item-item similarity matrix. Then, it applies a formula to predict the rating of an item i that has not been rated by the active user yet. This rating is the mean of the ratings already provided by the active user, weighted by the similarities between the item i and each of the items contained in the preference model of the active user. Our implementation relies on the Pearson correlation coefficient. At each iteration, we select the 10 items that got the highest predicted notes.

At last, we conceived two new algorithms called CFRD and CFFD, variants of the CF algorithm with a content-based hybridation. Our objective was to make the diversity level vary within the recommendation set. These two alternatives allow us to study the possible differences of users' perception when confronted with different diversity levels.

CFRD. (Collaborative Filtering with Relative Diversity). This algorithm first applies CF algorithm to compute the top-50. The first element of top-50 is included in the recommendation set. Then, items are added one by one, by selecting at each iteration the item from the top-50 that maximizes the relative diversity, in comparison with items already in the recommendation set (eq. (2)). We continue until we reach the expected number of recommendations.

Let us notice that this algorithm is quite similar to the algorithm proposed by [26], which is re-used in several user studies [28], [29]. In these papers, they build the top-10 recommendations by re-ranking the top-50 items of a CF algorithm according to a diversity metric. However, in our case, we used a different CF algorithm and a more complete diversity metric. In [29], authors explain that their diversification algorithm only reduces the similarity between movies in terms of genre and recognize that this may not fit the definition of similarity as the users of the system judge it.

But, more importantly, [26] use a diversification factor to find a compromise between the ranking of the CF algorithm and the diversity-based ranking. In other words, the higher movies are in the CF ranking, the more they have chances to appear in the top-10 list of the diversification algorithm. In our case, we consider that all of the movies in the top-50 of the CF algorithm are relevant, and have the same level of importance (except for the first one). Thus, we only re-rank the top-10 according to our diversity metric. In this way, we can more easily measure the impact of our diversity-driven approach, since diversity has more weight in the re-ranking phase.

CFFD. (Collaborative Filtering with Fixed Diversity). It is quite similar to CFRD, except that only a fixed percentage (x%) of recommendations has to come from the CF algorithm. In other words, instead of initializing the recommendation set with the first element of the top-50 (CFRD), we select the n first items of the top-50 (with n = the expected number of recommendations * x%). In our implementation, this threshold has been fixed to 60%.

C. Procedure

Our experiment is expected to last from 15 to 20 minutes per participant. After a short homepage that introduces the context of our study, each volunteer is invited to complete the 4-step procedure described below.

Step 1. A first questionnaire allows us to collect demographic data (first name, last name, email, gender, age, nationality, profession), and users' habits related to cinema (frequencies of visits in theaters, movie genres that they like, with whom they go to theaters and how they choose a movie, and if they read websites, magazines or books related to movies). Those habits reveal the users' expertise level in the domain of cinema. These questions are only uses for statistics on participants, and eventually to discard users whose answers might be irrelevant. At the end of step 1, each user is registered and the system assigns him/her one of the 5 recommendation algorithms. As a consequence, the participants are automatically spread into 5 groups. Each group has the same number of users.

Step 2. The system asks each user to rate a set of 100 films from 1 (I hate) to 5 (I like), under the pretext of filling his/her preference model. These 100 movies are displayed 10 by 10 (on 10 different pages), to avoid fatigue and cognitive overload. By default, movies are displayed in a synthetic way with minimal information such as the title, movie cover, director, genres, main actors and release date. However, participants can get more details from the interface. What users do not know is that only the 3 first pages of ratings (30 movies) are common to every participant to initialize profiles. In pages 4 to 10, the list of movies to be rated is provided by the recommendation algorithm that has been assigned to the active user. In this way, movies are likely to interest users, but they are not aware of the fact that these lists are built in accordance with their preferences. To avoid any bias, movies are displayed in a random order on each page. Of course, given the size of our training set (509 movies) there is a risk that the quality of recommendations decreases due to the lack of interesting but not rated items. However, this risk is low since they only rate 20% of the database. Moreover, this phenomenon impacts all the algorithms in the same way.

Step 3. The system proposes a one-week TV program (one movie for each day of the week). The TV program is made of five TV channels, one for each recommendation algorithms (POP, CBF, CF, CFRD, CFFD). In this phase, we explicitly told users that these channels were made of recommendations from different algorithms. The goal was to measure if there are differences of confidence level toward these algorithms, if users can distinguish the recommended lists, and if the lack of diversity can pose a problem over a week. Users have to order channels from 1 to 5 according to their preferences.

Step 4. A post-questionnaire allows users to make explicit and quantify the performance of algorithms during step 3. In particular, we asked users to evaluate the recommendation relevance, the diversity levels during steps 2 and 3, and their confidence level within their ranking of step 3. We used a Likert scale with 7 modalities.

D. Hypotheses

Before conducting this experiment, we enumerated the following hypotheses:

H1. Users perceive diversity. Results from the post-questionnaire should reflect this tendancy, in particular for groups assigned to CFRD and CFFD.

H2. Diversity improves users' satisfaction. Ratings collected in Step 2 should be higher for groups assigned to CFRD and CFFD, than for the other groups.

H3. Content-based algorithms increase the level of confidence of users, on the contrary of those based on diversity. Recommendations from the CBF algorithm should meet with more success in Step 3.

E. Participants

We collected data over one week by contacting 250 volunteers through social networks. As a reminder, these 250 users were completely different from the 3,158 users of the training set. Moreover, none of them were part of a course on recommender systems, so as to avoid any bias.

We split them into 5 groups of 50 persons (G1 to G5). These 250 volunteers were 114 women and 136 men; 205 of them were French, the 45 others being from different countries over the world (Canada, Syria, Belgium, Roumania, Ivory Coast, Tunisia, Great Britain, Mexico, China, Lebanon, Algeria and Switzerland). There were 152 students, 62 senior executives, 25 employees, 5 retired persons, 4 self-employed people, 1 worker and 1 artisan. 4 of them were minors, 146 were between 18 and 24 years old, 65 were between 25 and 39 years old, 31 were between 40 and 59 years old, and 4 persons were more than 60 years old. Everybody, except one person, claimed going to theaters at least occasionally. They all were interested about movies. In order to motivate participants to diligently rate movies, they were told that there would be a lottery, that would reward 20 participants with a DVD in accordance with their preferences expressed within the frame of this study.

IV. Results

A. Measure of performance of algorithms

Before analyzing users' data, we measured the diversity level provided by each of the 5 algorithms in Step 2 (see Figure 1). As we used the average Intra-List Similarity measure from page 4 to page 10, the lower the similarity is, the more diverse the algorithm is. Let us remind that the 3 first pages were manually selected to initialize users' profiles. Thus, recommendations started at page 4. As expected, algorithms based on collaborative filtering (CF, CFRD, CFFD) provide much more diversity that CBF. Our diversity-based algorithms (CFRD and CFFD) are more diverse than the classical CF algorithm.

It is also not surprising that the POP algorithm provides a high level of diversity, since movies are randomly selected among the most popular. However, the POP algorithm does not provide any personalization since it does not rely on the active user' preferences. Thus, there is an important risk that users have a low confidence in these recommendations and/or do not find them relevant. The POP algorithm is only used as a baseline in our experiment.

At last, let us notice that the ILS measure decreases over time for the CBF algorithm, while it remains quite stable for the other algorithms. This is due to the small size of our movie corpus. Indeed, the CBF algorithm first recommends the movies that are the most similar as regards attributes with those that have been liked by the active user. Once it has recommended all the highly similar movies (movies that are at the same time from the same saga, with the same director, the same actors, the same popularity, and so on), it necessarily increases diversity by proposing movies that only have a few attributes in common.

B. Validation of hypotheses

To validate our hypotheses, we analyzed results from the post-questionnaire (step 4). First, we converted answers into numerical values (from "Strongly disagree = 1" to "Strongly agree = 7"). Second, we computed answers' means for each of the group from G1 to G5 (see Table II).

Validation of H1. Groups G3 to G5, whose members used algorithms based on collaborative filtering in Step 2 (CF, CFRD, CFFD), found recommendations from the 5 algorithms more diverse in Step 3 than the other groups (see column "Diversity" in Tab II). We used a Student t-test to confirm the statistical significance of this result (p=0.05 between G2 and G3, p=0.07 between G2 and G4). Moreover, only 36 users from group G2 (CBF) found that the list of movies to be rated in Step 2 were diverse, against 45 to 47 users among a total of

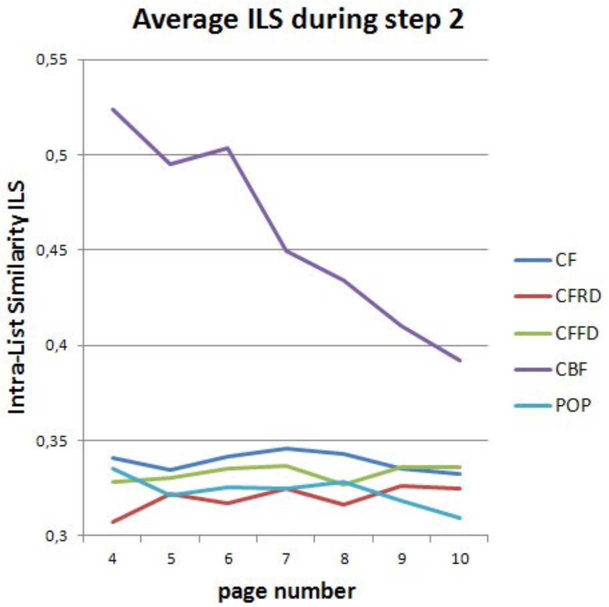


Fig. 1: Intra-List Similarities for each page in Step 2.

TABLE II: RESULTS FROM THE POST-QUESTIONNAIRE, AND MEANS OF RATINGS IN STEP 2

Group number (algo. step 2)	Step 3 (all algorithms together)			Mean in ratings
	Diversity	Relevance	confidence	
G1 (POP)	4.64	3.94	4.98	3.49
G2 (CBF)	4.44	3.26	5.34	3.55
G3 (CF)	**5**	4.04	5.32	3.79
G4 (CFRD)	**4.96**	4.1	**5.38**	3.61
G5 (CFFD)	4.88	**4.45**	5.30	3.60

50 users for the other groups. Users are consequently capable of perceiving diversity within the recommendation set, even in the cases recommendations are made implicit (Step 2), which validates our hypothesis H1.

Validation of H2. The average ratings of collaborative filtering (CF) and diversity-based filtering (CFRD, CFFD) in Step 2 (column on the right in Tab II) are higher than those of CBF. This seems to confirm that diversity-driven algorithms (CF, CFRD, CFFD) improve users' satisfaction in comparison with other algorithms when recommendations are made in an implicit way. Nevertheless, the difference of satisfaction between these 3 variants of collaborative filtering (CF, CFRD, CFFD) remains marginal, and more particularly between CFRD and CFFD. Thus, we hypothesize that the degree of diversity does not have any impact on users' satisfaction, while a minimal threshold is reached. The latter will have to be clarified through another study where the degree of diversity will vary more finely, and on a greater number of recommendations.

Validation of H3. If diversity seems to improve satisfaction during the phase of implicit recommendation (Step 2), results are much more contrasted in Step 3 where users have been warned that the list of movies are recommended according to their explicit preferences. As shown in Tab III, we computed the number of times that each algorithm has been ranked first in Step 3, that is to say perceived by the user as the best TV channel. All groups together, we notice that CBF algorithm got the highest number of votes. This confirms hypothesis H3 according to which content-based filtering arises a higher level of user confidence (see the last line of Tab III). The comments provided by volunteers at the end of the study provides a piece of explanation: thanks to similarities of attributes, it is much easier for users to understand the link between preferences made explicit and recommendations from CBF, in comparison with other algorithms. As a consequence, each user can easily imagine an implicit explanation for a given recommendation (for example, the active user has highly rated the movie "The Matrix", which probably explains why the system recommends him/her the movie "The Matrix Reloaded").

TABLE III: NUMBER OF VOTES FOR EACH TV PROGRAM IN STEP 3

Group Number	Algorithm chosen at Step 3				
	POP	CBF	CF	CFRD	CFFD
G1 (POP)	14	**22**	7	3	4
G2 (CBF)	9	**29**	7	5	**0**
G3 (CF)	7	**17**	16	6	4
G4 (CFRD)	9	**15**	6	12	7
G5 (CFFD)	**14**	10	8	5	12
confidence (all users together)	4.98	**5.34**	5.32	**5.34**	5.32

On the other hand, according to the column entitled "Relevance" in Tab II, groups G4 and G5 – assigned to our diversity-driven algorithms in Step 2 (CFRD and CFFD) – found recommendations in Step 3 more relevant (all algorithms together) with more than one point of difference in comparison with group G2 assigned to content-based filtering. This result is statistically significant with a 99% level of confidence ($p = 0.004$ between G2 and G4, and $p = 4.27e - 05$ between G2 and G5). Providing a more diverse set of items during Step 2 (CFRD) has also improved the overall degree of confidence of users within recommendations, even if the CBF algorithm got the highest number of votes in Step 3. As a consequence, whatever the recommendation algorithm used, the system has to make sure that the active user's preference model contains items diverse enough to provide better recommendations. This conclusion constitutes an unexpected influence of diversity, which will lead us to further investigate items that have to be rated during the cold-start phase.

V. Conclusion and perspectives

This work constitutes an explorative study of the role and impact of diversity within recommender systems. It highlighted the necessity to build preference models containing items various enough to ensure a good level of relevance and confidence of recommendations. Moreover, we proved that diversity is perceived by users and improve users' satisfaction. Nevertheless, diversity in the recommendation set can require additional explanations to users who may not see the link between their preferences made explicit and the items recommended by the system. In summary, diversity is a complex dimension, which is good for users, if it is used at the right time and in the appropriate manner. Following these conclusions, a perspective will consist in studying means to guarantee an adequate level of diversity during the cold-start phase.

REFERENCES

[1] Netflix prize, http://www.netflixprize.com/, 2009.

[2] Y. Koren, R. M. Bell, and C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.

[3] J. Sill, G. Takacs, L. Mackey, and D. Lin, Feature-weighted linear stacking, Cornell University, Netflix Prize Report, 2009.

[4] P. Sawers, Remember netflix's $1m algorithm contest? well, here's why it didn't use the winning entry, http://thenextweb.com/media/2012/04/13/remember-netflixs-1m-algorithm-contest-well-heres-why-it-didnt-use-the-winning-entry/, 2012.

[5] S. Castagnos, N. Jones, and P. Pu, Eye-tracking product recommenders' usage, in Proceedings of RecSys'10, Barcelona, pp. 29–36, 2010.

[6] N. Jones, User perceived qualities and acceptance of recommender systems, PhD Thesis, Ecole Polytechnique Fédérale De Lausanne, 2010.

[7] L. McGinty and B. Smyth, On the role of diversity in conversational recommender systems, in ICCBR'03, pp. 276–290, 2003.

[8] S. Castagnos, N. Jones, and P. Pu, Recommenders' influence on buyers' decision process, in In proc. of RecSys'09, New York, pp. 361–364, 2009.

[9] B. Smyth and P. McClave, Similarity vs. diversity, in Proceedings of the 4th International Conference on Case-Based Reasoning, Vancouver, pp. 347–361, 2001.

[10] Charles L.A. Clarke et al., Novelty and diversity in information retrieval evaluation, Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 659-666, 2008.

[11] F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims, Redundancy, diversity and interdependent document relevance, SIGIR Forum, vol. 43, no. 2, pp. 46–52, 2009.

[12] G. Adomavicius and Y. Kwon, Improving aggregate recommendation diversity using ranking-based techniques, IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 5, pp. 896–911, 2012.

[13] M. Zhang and N. Hurley, Avoiding monotony: Improving the diversity of recommendation lists, in Proceedings of RecSys'08, Lausanne, pp. 123–130, 2008.

[14] N. K. Lathia, Evaluating collaborative filtering over time, PhD Thesis, University College London, 2010.

[15] G. Häubl and K. Murray, Preference construction and persistence in digital marketplaces: The role of electronic recommendation agents, Journal of Consumer Psychology, vol. 13, no. 1, pp. 75–91, 2003.

[16] S. M. McNee, J. Riedl, and J. A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in CHI '06: CHI '06 extended abstracts on Human factors in computing systems. Montréal: ACM, pp. 1097–1101, 2006.

[17] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems, vol. 22, pp. 5–53, 2004.

[18] Ana Beln Barragáns-Martínez et al., A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, Journal of Information Sciences, Elsevier, vol. 180, pp. 4290-4311, 2010.

[19] C. Yu, L. V. Lakshmanan, and S. Amer-Yahia, Recommendation diversification using explanations, in Proceedings of the 2009 IEEE International Conference on Data Engineering (ICDE'09), pp. 1299–1302, 2009.

[20] M. Ge, F. Gedikli, and D. Jannach, Placing high-diversity items in top-n recommendation lists, in Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP'11), Barcelona, Spain, pp. 65–68, 2011.

[21] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Recommender Systems Handbook. Springer, 2011.

[22] K. Bradley and B. Smyth, Improving recommendation diversity, in Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, pp. 85–94, 2001.

[23] D. McSherry, Diversity-conscious retrieval, in Proceedings of EC-CBR'02, London, pp. 219–233, 2002.

[24] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, Diversifying search results, in Proceedings of WSDM'09, Barcelona, pp. 5–14, 2009.

[25] Shengxian Wan et al., ICTNET at Web Track 2011 Diversity Track, Text REtrieval Conference (TREC'11), 2011.

[26] C.-N. Ziegler, S. McNee, J. Konstan, and G. Lausen, Improving recommendation lists through topic diversification, in Proceedings of the 14th international conference on World Wide Web (WWW'05), pp. 22–32, 2005.

[27] A. Said, B. Kille, B. J. Jain, and S. Albayrak, Increasing diversity through furthest neighbor-based recommendation, in Proceedings of the WSDM'12 Workshop on Diversity in Document Retrieval, Seattle, USA, 2012.

[28] M. Willemsen, B. Knijnenburg, M. Graus, L. Velter-Bremmers, and K. Fu, Using latent features diversification to reduce choice difficulty in recommendation lists, in RecSys'11 Workshop on Human Decision Making in Recommender Systems, Chicago, IL, pp. 14–20, 2011.

[29] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell, Explaining the user experience of recommender systems, User Modeling and User-Adapted Interaction, vol. 22, no. 4-5, 2012.

[30] S. Castagnos, Website dedicated to a diversity-oriented experiment within recommender systems, http://www.movit.tv/tut5/index.php, 2013.

[31] Allociné website, http://www.allocine.fr/, 2013.

[32] V. Schickel and B. Faltings, Using an ontological a-priori score to infer user's preferences, in Workshop on Recommender Systems, in Conjunction with the 17th European Conference on Artificial Intelligence (ECAI 2006), pp. 102–106, August 2006.

[33] Imdb website, http://www.imdb.com/, 2013.

[34] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, Item-based collaborative filtering recommendation algorithms, in *World Wide Web*, pp. 285–295, 2001.

Is the Polarity of Content Producers Strongly Influenced by the Results of the Event?

Liliana Ibeth Barbosa Santillán
University of Guadalajara
Information Technology
Guadalajara, México
Email: ibarbosa@cucea.udg.mx

Inmaculada Álvarez de Mon y Rego
Universida Politécnica de Madrid
Lingüistica aplicada a la ciencia y la tecnología
Madrid, Spain
Email: ialvarez@euitt.upm.es

***Abstract*—This paper presents an approach to compare two types of data, subjective data (Polarity of Pan American Games 2011 event by country) and objective data (the number of medals won by each participating country), based on the Pearson correlation. When dealing with events described by people, knowledge acquisition is difficult because their structure is heterogeneous and subjective. A first step towards knowing the polarity of the information provided by people consists in automatically classifying the posts into clusters according to their polarity. The authors carried out a set of experiments using a corpus that consists of 5600 posts extracted from 168 Internet resources related to a specific event: the 2011 Pan American games. The approach is based on four components: a crawler, a filter, a synthesizer and a polarity analyzer. The PanAmerican approach automatically classifies the polarity of the event into clusters with the following results: 588 positive, 336 neutral, and 76 negative. Our work found out that the polarity of the content produced was strongly influenced by the results of the event with a correlation of .74. Thus, it is possible to conclude that the polarity of content is strongly affected by the results of the event. Finally, the accuracy of the PanAmerican approach is: .87, .90, and .80 according to the precision of the three classes of polarity evaluated.**

Keywords*—*Polarity; Subjective; Objective Corpus Analysis.

I. Introduction

Content producers are currently emerging from the social web where the majority of the population is young people with very specific needs in terms of communication. The Web has facilitated social networking phenomena through both structured and unstructured data. The analysis of this content may have considerable influence on important decisions that affect society.

The amount of Internet resources that exist in the Web dealing with a specific event, such as newspapers, chat rooms, social networking, Internet commerce, product reviews and blogs, have heterogeneous content that proliferates in an uncontrolled fashion.

However, there are difficulties in measuring the polarity of the content generated by a person. Some of them are: a) identifying noise polarity and fake reviews and b) dealing with slang and inaccurate use of language.

In addition, one of the most important and complex tasks for the entrepreneurs, officials and the organizers of an event is to know as precisely as possible how citizens perceive it. In this sense, the diversity of opinions and assessments related to events that involve different countries vary greatly. A possible solution is to obtain metrics for measuring the polarity of the content expressed by producers in their writings.

The motivation of this research is to develop a first approach on how to assess the appreciation of an event through the opinions of citizens grouped by their origin country for an event of Pan-American scale.

This study focuses on answering the following research question: Is the polarity of content producers strongly influenced by the results of the event?

The hypothesis that the authors propose is:

H1 The polarity of content producers is strongly influenced by the results of the event.

This study aims to provide benefit in the form of classify the social point of view of polarity of the 2011 Pan American Games.

This project could be of benefit for both governments and citizens.

Compared with previous work, the major contributions of this paper are the following:

- Focusing on unstructured opinions in order to contrast subjective polarity and objective data related to the same event.

This work draws on at least 5,600 opinions of citizens of the Pan American countries, helping to understand the impact of the event among the citizens of the 42 nations participating in 36 sports. Some of the advantages of our analysis are that it facilitates knowing the polarity of the citizens through fresh opinions articulated in Internet resources.

This paper is structured as follows. Section II briefly discusses the related work. Section III gives an overview of the design, describing the proposed PanAmerican approach. Sections IV and V discuss the analysis and experiments. Finally, Section VI contains the conclusions of our research work.

II. Related Work

The related work takes into account two topics: 2.1) polarity, and 2.2) systems related to olympics.

ISBN: 978-1-61208-311-7

A. Polarity

According to Cambria [1], Opinion Mining ”mainly concerns polarity detection”, whereas sentiment analysis, as defined by Pang [2], is ”the specific application of classifying reviews as to their polarity (either positive or negative)”. Opinion mining and sentiment analysis are used in this research as synonyms in order to deal with the literature related to both topics.

In recent years, opinion mining has been studied by many researchers. The authors have focused on three aspects: a) question answering, b) recommendation systems, and c) sentiment-relevant lexicons.

- **Question answering:** Earlier work showed that disambiguating instances of subjectivity clues is useful for sentence-level attitude-type classification. Somasundaran et al. [3] developed automatic classifiers to recognize when a sentence is expressing one of the two main types of attitude. Stoyanov [4] developed a corpus of opinion questions and answers; his research compared and contrasted the properties of facts and opinions in question answering. Vlad et al. [5] defined qualitative dimensions for evaluating answers and showed how ignored terms in the process of entity definition can help users to discover underlying information.
- **Recommendation systems:** Efforts in this area were carried out by Nitin et al. [6] who proposed a collaborative exploration system helping users to explore movie reviews from various viewpoints. Reputation is a topic of collective interest; Morinaga [7] demonstrated that it is possible to help users to discover important knowledge regarding the reputations of products of interest through the following tasks: characteristic word extraction, co-occurring word extraction, sentence extraction, and correspondence analysis. Ungar et al. [8] used clustering methods for collaborative filtering.
- **Sentiment-relevant lexicons:** Previous research has focused on the creation of lexicons in English such as that of: Higashinaka [9] who used a set of dialogues to build her own lexicon. Lexicons are also available as linguistic resources on the Internet, some examples being: SentiWordnet [10], NTU Sentiment Dictionary [11]. Pak [12] build automatically sentiment relevant lexicon from Internet Resources, and the OpinionFinder system for subjectivity analysis [13], among others.

B. Systems

The research of Gruzd et al. [14] measures if happiness is contagious online in 2010 winter olympics and they determined that were more positive messages than negative in twitters. It also influenced the level of retweet from positive versus negative messages. SentiStrength [15] splits the tweets into positive and negative conversations and filters them through a programme, which systematically converts them into a lightshow. It was used for measure the Olympic London Eye.

As opposed to these works, we aim to model multiple users' location posts and learn polarity from numerous opinions by different individuals on the Pan American Games 2011.

III. The PanAmerican approach Proposed

The PanAmerican approach aims at performing the classification of polarity in a set of Internet resources focused on an event. The approach is based on four components: a crawler (A), a filter (B), a synthesizer (C), and a polarity analyzer (D). The main function of the crawler component is to search and find data from internet resources related to the event of interest. After locating the data, the filter component processes the data in order to remove noise. The filter component only debugs internet resources that are associated with the event. At this point, the corpus consists of numerous posts containing large amounts of data from many countries and in many languages. The synthesizer component represents the amount of data into clusters with similar expressions using unsupervised learning. Finally, the Polarity analyzer component classifies each cluster into positive, neutral or negative. Each of the components in the PanAmerican approach is described in greater detail below:

A. Crawler

The crawler is a component that obtains internet resources related to one event and stores them in a repository for later use.

The main challenges faced by this component are the size of the internet resources that continue to grow in a highly dynamic way and the fact that some of them appear and disappear in a very brief period of time. The depth is the link levels that a seed can have; if a seed has another link embedded in the body of the web this will conduct a search on this link, and so on up to the number of levels configured on the system. It is important to note that an unlimited number of levels can take months of processing in multiple threads. The solution is to seek and obtain Internet resources based on a depth of four links [16].

In addition, the crawler component stores items in a knowledge base as follows: the URL, the contents of the internet resource in natural language, a bag of words and the position in the sentence of each one of these.

An important challenge is the quality of the internet resources because over 40 percent of the data collected is not semantically related to the event under study. One of the reasons is that many people manipulate their content specifically with keywords, titles, and descriptions in order for their Internet resource to rank high in search results during the event and for that reason the filter component is essential.

1) Data: The quality of the corpus is measured by the degree of compliance of the posts that meet the purpose for which the corpus is compiled. Thus it was necessary to take special care in the selection of posts attempting to maintain homogeneity. Therefore, it was necessary to establish the following criteria that govern the selection and inclusion of posts.

Quantity: It was decided to include 5600 posts of different dimensions.

Quality of text: Given that the selection was automatic, special care was taken in that the texts were written in the correct language, without spelling mistakes, in clear writing.

Published in Pan American Games 2011: Due to the nature of the project, we only included published posts.

Type of opinion: the opinion must have been carried out with the results of the event.

Text form: The texts must be written in the form of general impressions.

Style: The texts must be comprehensive, describing the opinion from beginning to end, discarding free or incomplete texts introduced in unfinished or abandoned posts.

Additional information: Each sample must be marked with a series of additional data, which gives extra information and allows for identification. These marks are the: web page from which it has been extracted, country or area where the opinion has been realized, language, and date of the opinion.

B. Filter

The filter is a component that processes Internet resources to remove unwanted data related to an event. The filter uses an anti-noise function to minimize 37.5 percent of the noise in the corpus content. The filter works by analyzing the most frequent words in the post titles and descriptions that are related to the name of the event and then classifies each post as noise or not noise. Finally, the filter builds a knowledge base populated with titles, descriptions and posts related to the event. The output from the filter component is still a large volume of data because there are one hundred sixty-eight Internet resources that are producing dozens of posts daily in several languages. As a result, the knowledge base grows tremendously and is full of instances that were saved sequentially; thus, the next step is a preprocessing of all the resulting data, grouping and classifying them according to common themes using the synthesizer component.

C. Synthesizer

The main function of the synthesizer component is to take all the posts that the filter has classified as not noise and without previous knowledge about them identify groups of similar expressions using a Bayes classifier [17].

Therefore, the next step is to group all the posts that express similar content to represent the data in a lower dimension space. One reason to use unsupervised learning [18] is that people observe the same event in several ways; however, their perceptions of one event have clear differences based on their nationality. We used a Bayes classifier, which has shown good results in previous work. The results are stored in two plain text files: one with a set of clusters, and the other one with the six most representative patterns for each cluster of posts. The synthesizer component creates a new population of posts represented in a lower dimension space by clustering.

D. Polarity Analyzer

The polarity analyzer component is in charge of the polarity analysis of clusters based on a Multilingual Lexical Ontology (MLO) (see section 1) and classifies each cluster into positive, negative or neutral. It uses K-means cluster.

For each cluster, we obtained the six most representative patterns: the component performs a semantic analysis of patterns based on the MLO ontology.

Therefore, subjectivity is calculated with an additional operation as follows: positive (more than zero), negative (less than zero), and neutral (equal to zero).

Finally, each cluster is classified into positive, negative or neutral.

1) Multilingual Lexical Ontology (MLO): The two main characteristics of the MLO ontology are that it is language-independent and provides multilingual population in any language. However, their instances are in the four languages, these being Spanish, English, Portuguese and French because they are the languages most used by people in the 2011 Pan American Games.

Definition: the MLO Ontology is a conceptual description based on a lexicon of the subjective words in Natural Language as shown in (1). The MLO Ontology consists of four disjoint sets C, R, A, and τ where C means concept identifiers (2), R means relation identifiers (3 and 4), A means attribute identifiers (5), and τ means data types (6).

$$MLO := (C, \leq c, R, \gamma_R, \leq_R, A, \gamma_A, \tau) \tag{1}$$

The set C of concepts is:

$$C :\doteq \left\{ \begin{array}{l} Adjectives, NegativeAdjectives, \\ PositiveAdjectives, Adverbs, NegativeAdverbs \\ , PositiveAdverbs, Articles, Authors, \\ DomainResources, Nouns, NegativeNouns, \\ PositiveNouns, Paragraphs, Posts, \\ Predicates, Prepositions, Sentences, \\ Subjects, Titles, InternetResources, \\ Verbs, NegativeVerbs, PositiveVerbs \end{array} \right. \tag{2}$$

The set R of relations is:

$$R :\doteq \left\{ \begin{array}{l} author\_of, post\_of, paragraph\_of, sentence\_of, \\ adverb\_in, articles\_inprepositions\_in, nouns\_in, \\ adjectives\_in, verbs\_in, subject\_of, predicate\_of \end{array} \right. \tag{3}$$

where the relation hierarchy defines that DomainResources has the relation author_of that belongs to Authors. InternetResources has the relation post_of that belongs to Posts, following the same logic the rest of the relations are defined, as shown in equation (4).

$$\begin{array}{l} \gamma R(author\_of) = (Authors, DomainResources) \\ \gamma R(post\_of) = (Posts, InternetResources) \\ \gamma R(paragraph\_of) = (Paragraphs, Posts) \\ \gamma R(sentence\_of) = (Sentences, Paragraphs) \\ \gamma R(adverbs\_in) = (Adverbs, Sentences) \\ \gamma R(articles\_in) = (Articles, Sentences) \\ \gamma R(prepositions\_in) = (Prepositions, Sentences) \\ \gamma R(nouns\_in) = (Nouns, Sentences) \\ \gamma R(adjectives\_in) = (Adjectives, Sentences) \\ \gamma R(verbs\_in) = (Verbs, Sentences) \\ \gamma R(subject\_in) = (Subjects, Sentences) \\ \gamma R(predicate\_in) = (Predicates, Sentences) \end{array} \tag{4}$$

The set A of attribute identifiers is:

$$A :\doteq \left\{ \begin{array}{l} blog, author, title, post, paragraph, sentence, \\ subject, predicate, article, noun, nounP, nounN, \\ verb, verbN, verbP, adjective, adjectiveP, \\ adjectiveN, preposition, adverb, adverbP, \\ adverbN \end{array} \right. \tag{5}$$

The set τ of datatypes contains only one element a string, as shown in (6).

$$\tau := (string) \tag{6}$$

The first axiom defines the concept NegativeAdverbs as equivalent to saying that there is a negativeAdverb, which stands in a adverb_in relation with the corresponding sentence, following the same logic the rest of the axioms are defined as shown in (7).

$$\begin{array}{l}
\forall x(NegativeAdverbs(x) \longleftrightarrow \exists y \wedge adverb\_in(x,y) \wedge Sentences(y)) \\
\forall x(PositiveAdverbs(x) \longleftrightarrow \exists y \wedge adverb\_in(x,y) \wedge Sentences(y)) \\
\forall x(NegativeVerbs(x) \longleftrightarrow \exists y \wedge verbs\_in(x,y) \wedge Sentences(y)) \\
\forall x(PositiveVerbs(x) \longleftrightarrow \exists y \wedge verbs\_in(x,y) \wedge Sentences(y)) \\
\forall x(NegativeNouns(x) \longleftrightarrow \exists y \wedge nouns\_in(x,y) \wedge Sentences(y)) \\
\forall x(PositiveNouns(x) \longleftrightarrow \exists y \wedge nouns\_in(x,y) \wedge Sentences(y)) \\
\forall x(NegativeAdjectives(x) \longleftrightarrow \exists y \wedge adjectives\_in(x,y) \wedge Sentences(y)) \\
\forall x(PositiveAdjectives(x) \longleftrightarrow \exists y \wedge adjectives\_in(x,y) \wedge Sentences(y))
\end{array} \tag{7}$$

To summarize, the PanAmerican approach proposed is shown in Fig. 1, where the input is the Official Web of Pan American Games 2011 and the output is the polarity value of each country involved.

```
procedure CRAWLER($SubsetWeb$)
    for $i \leftarrow 1, SizeSubsetWeb$ do
        $InternetResources(i) \leftarrow DownloadURL((Get(URL(i)));$
    end for
end procedure
procedure FILTER($InternetResources$,$Term$)
    for $i \leftarrow 1, NumberofInternetResources$ do
        if $SyntacticFilter(InternetResource(i))$ then
            if $NoiseFilter(InternetResource(i))$ then
                $noise(InternetResource(i)) \leftarrow InternetResource((i));$
            else
                if $SemanticFilter(InternetResource(i))$ then
                    $Titles(InternetResource(i)) \leftarrow Split(Titles(InternetResource(i)));$
                    $Descriptions(InternetResource(i)) \leftarrow Split(Descriptions(InternetResource(i)));$
                    $Posts(InternetResource(i)) \leftarrow Split(Posts(InternetResource(i)));$
                end if
            end if
        end if
    end for
end procedure
procedure SYNTESIZER($Posts$)
    for $i \leftarrow 1, NumberofPosts$ do
        $ClusterMultilingual(i) \leftarrow ClusterMultilingual(Posts(i));$
        $Patterns(i) \leftarrow Patterns(Posts(i));$
    end for
end procedure
procedure POLARITY($Web$)
    $InternetResources \leftarrow Crawler(Web);$
    $Posts \leftarrow Filter(InternetResources, "PanAmericangames2011");$
    $Clusters \leftarrow Syntesizer(Posts);$
    for $k \leftarrow 1, NumberofClusters$ do
        $PolarityValue(k) \leftarrow PolarityValue(Cluster(k), MLO);$
        $Sum(k) \leftarrow OpinionValue(k);$
    end for
end procedure
```

Fig. 1: The PanAmerican approach

IV. Experimental Details and Performance Results

The following section includes a detailed description of how the experiment was conducted. The first part describes the objectives of the experiment and the second part focuses on the results obtained after the experiment was conducted.

The experimental setup had four objectives: 1) to obtain a subset of Internet resources related to the 2011 Pan American Games, 2) to delete noise in the Internet resources obtained, 3) to classify sets of posts that are close in meaning and group them into clusters, and 4) to assess the polarity of each cluster.

The analysis was carried out in two ways: 1) the crawler component was run in order to obtain Internet resources over a period of three months, and 2) the type of PanAmerican was carried out for 168 Internet resources.

The first task was to obtain posts in Internet resources using the crawler component. However, the output is a set of Internet resources that grows rapidly with data irrelevant to the analysis. For that reason it was necessary to adjust the crawler component to the event. The focus was placed on Internet resources related syntactically to the term ”Pan American games 2011” in order to reject those Internet resources, which were not related to this event. We deal with two problems: a) the seed [19] had 300 related Internet resources so the crawler component was restricted to a search of four levels deep and b) identifying the source country based on the meaning of its posts is a major task. Thus, we assumed that, depending on the Internet resources, the top level domain of the post was used for the country of the author.

The second task was to filter out Internet resources not related semantically to the event and as a result classified as noise by the filter component. For example, some of the following Internet resources had a Pan American 2011 term but not all contained data related to the event; there are noise traders who use the same term but with a different meaning:

```
{http://www.emailbrain.com/134087/rss,
http://feeds2.feedburner.com/noc-aho\,
http://www.argentina.ar/rss/rss_prensa_es.xml,
http://www.dushi-curacao.info/1-dushi-curacao.html,
http://www.bahamasolympiccommittee.org/_rss/news,
http://www.amandala.com.bz/inc/rss.php?id=11806,
http://bmxbolivia.org/?feed=rss2,
http://www.olympic.ca/fr/,
http://co.elpais.feedsportal.com/c/33807/f/607321/index.rss,
http://juegospanamericanos.ain.cu/feed/,
http://www.colimdo.org/rss.aspx,
http://ministryofhealth.gov.ky/feed/rss.xml,
http://www.elcaribe.com.do/rss,
http://www.avn.info.ve/rss/6 , etc. }
```

As a result, we obtained a corpus of 3500 posts extracted from one hundred sixty-eight different Internet resources, sampled from a comprehensive range of 2011 Pan American Games Internet resources with 147 MB of text.

The third task was to find group similarities so as to represent in a lower dimension space the PanAmerican analysis. The PanAmerican approach synthesizes the corpus into clusters with similar data. As an example, we show three posts that are grouped to their similar themes:

```
Commonwealth Youth Games Team Selected,
Team Bahamas Deports Mexico with 3 Medals,
BOC Ammounces Guadalajara 2011 Pan Am Games Team
```

The PanAmerican approach processes each one of the clusters and extracts the six most relevant patterns for the task of PanAmerican; these patterns are also assessed through a parser that performs semantic matching between each pattern and the MLO ontology in order to carried out the tagging of polarity.

The MLO ontology measures are shown in Table I where the numbers of local instances are the same for all the four languages involved. For example, PositiveVerbs in Spanish amount to one hundred instances and this number is the same for each of the other languages: English, Portuguese and French.

TABLE I: MULTILINGUAL LEXICAL ONTOLOGY (MLO) MEASURES

Type	Measures	
	$Local$	$Inferred$
$NumberofTriples$	11616	1
$NegativeAdjectives$	1092	0
$PositiveAdjectives$	1468	0
$NegativeAdverbs$	52	0
$PositiveAdverbs$	100	0
$Internetresources$	286	0
$NegativeNouns$	880	0
$PositiveNouns$	800	0
$Titles$	1000	0
$DomainInternetResources$	1	0
$NegativeVerbs$	472	0
$PositiveVerbs$	400	0
$owl : Class$	22	0
$owl : DatatypeProperty$	22	0
$owl : NamedIndividual$	5400	0
$owl : ObjectProperty$	13	0
$owl : Ontology$	1	0

Each cluster is classified using a well-known formula of the sum of the value of the patterns polarity that is shown in (8).

$$f(n) = \begin{cases} n > 0 & \text{if } n \text{ is Positive (P)} \\ n < 0 & \text{if } n \text{ is Negative (N)} \\ n = 0 & \text{if } n \text{ is Neutral (Z)} \end{cases} \tag{8}$$

A partial result is shown in Table II where Cluster 2 (C2) contains posts, which are linked to people in wheelchairs, despite the fact that the first post is not explicitly linked to that term. However, Tito Bautista was a participant with a wheelchair, so it is correctly specified. The C2 PanAmerican result is positive (P) polarity.

The C6 language is French and most of the posts are negative; therefore, the PanAmerican analyzer component value is negative (N) polarity.

In C5, the first two posts are in Portuguese and the third post is in Spanish; all of these are linked to the Brasil term. The first two posts are neutral polarity and the third is negative polarity and as a result the cluster is negative polarity.

TABLE II: A PARTIAL VIEW OF CLUSTERS COMPONENT OUTPUT

Clusters	Cluster of Posts
C1 (P)	México llega a la Villa Miranda de México, una favorita de Guadalajara Respalda el Presidente Calderón propuesta de EGM para buscar los Juegos Olímpicos para Jalisco
C2 (P)	Perseverancia y coraje, palabras que definen a Tito Bautista Seleccionados mexicanos liderearon el Circuito Nacional de Tenis en Silla de Ruedas El Tenis en Silla de Ruedas entrará en acción
C3 (Z)	Registration [closed] Sport Management Course Start of CAC Games 2010 Pan American Games 2011
C4 (P)	Team Bahamas Departs Mexico with 3 Medals DIF Jalisco y COPAG hacen mancuerna Commonwealth Youth Games Team Selected
C5 (N)	Lettre de démission du Ministre de la Justice Affaire Bélizaire : Rapport de la Commission Spéciale dú2019Enqueate (Partie 1) Brasil equipo a vencer en Voleibol Sentados
C6 (N)	Mission afghane: départ devancé de lúAustralie? NY: Un homme aurait voulu faire sauter des sites Tripoli veut juger Seif al-Islam en Libye

A. Performance Results

In the first place, the crawler component identified four Internet resources for each participating country in the 2011 Pan American Games and therefore obtained 168 Internet resources containing 5600 posts as shown in Fig. 2

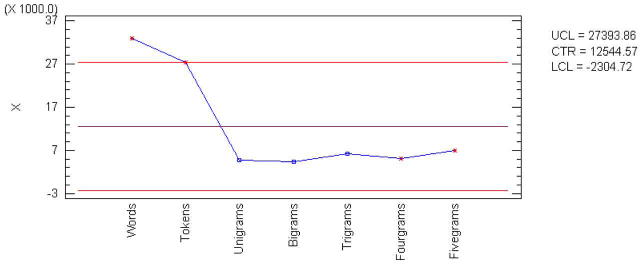


Fig. 2: Identification of the 5600 posts structure based on number of words, tokens, unigrams, bigrams, trigrams, fougrams, and fivegrams.

At this point, the filter component was applied in order to delete posts, which contained noise (2100) and posts semantically related to the event (3500) were identified. Next, 1000 clusters were obtained using the synthesizer component. Finally, 588 positive clusters, 336 neutral clusters and 76 negative clusters were tagged using the polarity analyzer component.

To measure the accuracy of the cluster classification task we used well-known formulae in the area of information retrieval as shown in equations 9 through 12, where precision and recall were evaluated for each polarity (P, Z, N).

Precision was calculated by dividing the True Positives (TP) between the sum of True Positives and False Positives (FP) as shown in (11). Recall is the division between True Positives and the sum of True Positives and False Positives as shown in (9).

$$Recall \equiv TPRate = TP/(TP + FN)) \qquad (9)$$

$$FPRate = FP/(FP + TN)) \qquad (10)$$

$$Precision = TP/(TP + FP)) \qquad (11)$$

$$F - Measure = 2TP/(2TP + FP + FN)) \qquad (12)$$

The results for precision for each polarity -Positive, Negative and Neutral (P, N, Z) respectively- are shown in Table III. The highest accuracy is in the negative polarity with a value of .9.

TABLE III: DETAILED ACCURANCY OF POLARITY COMPONENT

TP Rate	FP Rate	Precision	Recall	F-Measure	Polarity
0.997	0.204	0.875	0.997	0.932	P
0.118	0.001	0.9	0.118	0.209	N
0.762	0.096	0.8	0.762	0.78	Z

The clusters that were correctly classified amount to 85.1% and the faulty were 14.9% . The sample comprised 1000 clusters derived from 3500 posts that were filtered from 168 Internet resources from 42 countries and 4 languages. The absolute and relative errors are also shown in Table IV.

TABLE IV: STRATIFIED CROSS-VALIDATION OF COMPONENT SYNTHESIZER

Correctly Classified Clusters	85.1 %
Incorrectly Classified Clusters	14.9 %
Kappa statistic	0.7007
Mean absolute error	0.1881
Root mean squared error	0.2929
Relative absolute error	52.6328 %
Root relative squared error	69.3158 %
Total Number of Clusters	1000

The PanAmerican results and the medals won for each country are shown graphically in Fig. 3a and, as it can be seen, the positive clusters dominate. Fig. 3b shows the polarity results for each country.

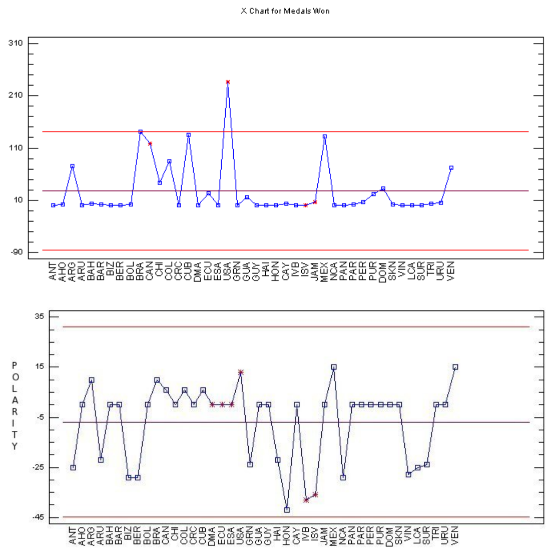


Fig. 3: Medals won in Pan American 2011 Games and Polarity Results of PanAmerican Approach for each country.

where the Id for each country is in Table V.

TABLE V: ID FOR EACH COUNTRY

Name	Id	Name	Id
Antigua and Barbuda	(ANT)	Guyana	(GUY)
Netherlands Antilles	(AHO)	Haiti	(HAI)
Argentina	(ARG)	Honduras	(HON)
Aruba	(ARU)	Cayman Islands	(CAY)
Bahamas	(BAH)	Virgin Islands (GB)	(IVB)
Barbados	(BAR)	Virgin Islands(US)	(ISV)
Belize	(BER)	Jamaica	(JAM)
Bermudas	(ANT)	Mexico	(MEX)
Bolivia	(BOL)	Nicaragua	(NCA)
Brazil	(BRA)	Panama	(PAN)
Canada	(CAN)	Paraguay	(PAR)
Chile	(CHI)	Peru	(PER)
Colombia	(COL)	Puerto Rico	(PUR)
CostaRica	(CRC)	Dominican Republic	(DOM)
Cuba	(CUB)	Saint Kitts Nevis	(SKN)
Dominica	(DMA)	Saint Vincent and the Grenadines	(VIN)
Ecuador	(ECU)	Saint Lucia	(LCA)
El Salvador	(ESA)	Suriname	(SUR)
United States of America	(USA)	Trinidad and Tobago	(TRI)
Grenada	(GRN)	Uruguay	(URU)
Guatemala	(GUA)	Venezuela	(VEN)

In addition, the research hypothesis claimed that the assessments of content producers would be influenced strongly by the results of an event regardless of their nationality. From our results it can be seen that the appraisal in some countries was positive because of the high number of medals won, as in the case of the United States, which took 236 medals. This is in contrast with Honduras, which did not win in any field, and thus, the overall assessment was strongly influenced and negatively evaluated, such as is shown in Fig. 3b. Fig. 4 shows that the correlation coefficient is equal to 0.74, indicating a strong relationship between the medals won and the global polarity by each country.

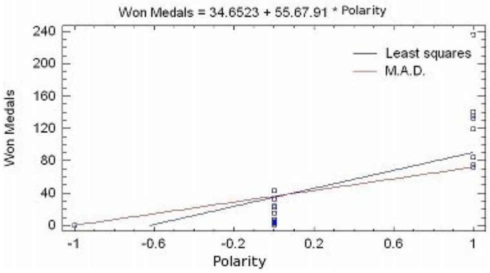


Fig. 4: Shows that the correlation coefficient is equal to 0.74, indicating a strong relationship between the medals won in Pan American 2011 Games and the global polarity of the PanAmerican Approach by each country.

Fig. 5 shows the polarity of six Internet resources - Resource 1 (R1), ..., , Resource 6 (R6)- for the last thirty-three countries. For example in Fig. 5(19) the polarity of Jamaica (JAM) is positive for the first three resources, negative for the following two, and postive for the last one.

V. Conclusion and Future Work

This paper has presented an approach to analyse a subset of Internet resources focused on a specific event, the PanAmerican Games, and based on four components: a crawler, a filter, a synthesizer and a polarity analyzer.

The PanAmerican approach has the following advantages: it allows analysis of a set of real subjective expressions used by people and their polarity classification as positive, neutral, or negative. This approach reduces ambiguity and 37.5 percent of noise in the subjective elements and classifies only those, which are not identified as noise component. Also, the PanAmerican approach found out that the polarity of content producers would be influenced strongly by the results of an event with a correlation of .74. Thus, it is possible to conclude that the polarity of content producers is strongly influenced by the results of the event.

In this case, the experiments reported are of a limited scale and serve mostly to demonstrate that the PanAmerican approach is feasible. In addition, there is the potential to scale up to use with a sizeable dataset.

Furthermore, if we included all the posts of Internet resources then the PanAmerican analysis would get an accuracy less than .5. However, we developed a approach based on an polarity classification with the precision of between .8 and .9, where the precision for positive and neutral polarity are acceptable and the recall is also good. In contrast with the negative polarity where precision is higher but the recall is very low.

To conclude, one of the benefits of the results of our research is the MLO presented because it is suitable for integration with other systems.

Finally, further research should be carried out to explore geographical differences in jargon and language. For example, we aim to identify certain evaluative words that are used only in some local geographical areas.

Acknowledgements

We are grateful to the Sciences Research Council (CONACYT) for funding this research project and Multilingüismo en ontologías y linked data (BabelData), TIN2010-17550, funded by the Ministry of Science and Technology, 2011-2013.

References

[1] E. Cambria and A. Hussain, “Sentic computing: Techniques, tools, and applications,” *SpringerBriefs in Cognitive Computation*, vol. 2, pp. 1–135, 2012.

[2] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Found Trends Information Retrieval*, vol. 2, pp. 1–135, January 2008.

[3] S. Somasundaran, T. Wilson, J. Wiebe, and V. Stoyanov, “Qa with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news,” *In International Conference on Weblogs and Social*, pp. 1–8, 2007.

[4] V. Stoyanov, C. Cardie, and J. Wiebe, “Multi-perspective question answering using the opqa corpus,” *in Proceedings of HTL-EMNLP 2005*, pp. 923–930, 2005.

[5] L. V. Lita, A. H. Schlaikjer, W. Hong, and E. Nyberg, “Qualitative dimensions in question answering: Extending the definitional qa task,” *In AAAI-2005*, 2005.

[6] N. Chiluka, N. Andrade, and J. Pouwelse, “A link prediction approach to recommendations in large-scale user-generated content systems,” *Advances in Information Retrieval, The 33rd European Conference on Information Retrieval (ECIR 2011)*, 2011.

[7] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, “Mining product reputations on the web,” *ACM Press*, pp. 341–349, 2002.

[8] L. Ungar and D. Foster, “Clustering methods for collaborative filtering,” *AAAI Press, Menlo Park California*, 1998.

[9] R. Higashinaka, M. Walker, and R. Prasad, “Learning to generate naturalistic utterances using reviews in spoken dialogue systems,” *ACM Transactions on Speech and Language Processing (TSLP)*, 2007.

[10] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC-06*, pp. 417–422, 2006.

[11] http://nlg18.csie.ntu.edu.tw:8080/opinion/userform.jsp (Accessed: September 2013).

[12] A. Pak, “Automatic, adaptive,and applicative sentiment analysis,” *Thèse de l’École Doctorale d’Informatique de l’Université Paris-Sud*, June 2012.

[13] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan, “Opinionfinder: a system for subjectivity analysis,” *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pp. 34–35, 2005.

[14] A. A. Gruzd, S. Doiron, and P. Mai, “Is happiness contagious online? a case of twitter and the 2010 winter olympics.” *IEEE Computer Society*, pp. 1–9, 2011.

[15] M. Thelwall, “Heart and soul: Sentiment strength detection in the social web with sentistrength,” *To appear in Holyst, J. (Ed). Cyberemotions*, pp. 1–14, 2013.

[16] J. E. Campos-Quirarte, L. I. Barbosa-Santillan, and A. Castro-Munguia, “A focused crawler in order to get semantic web resources (csr),” *Thirteenth Mexican International Conference on Computer Science (ENC’13), Morelia (Mexico)*, pp. 1–6, October 2013.

[17] P. Domingos and M. Pazzani, “On the optimality of the simple bayesian classifier under zero-one loss,” *Machine Learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[18] Y.-S. Kim, W. Street, and F. Menczer, “Feature selection in unsupervised learning via evolutionary search,” *Proc. 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[19] http : //www.guadalajara2011.org.mx/es/rss (Accessed: September 2013).

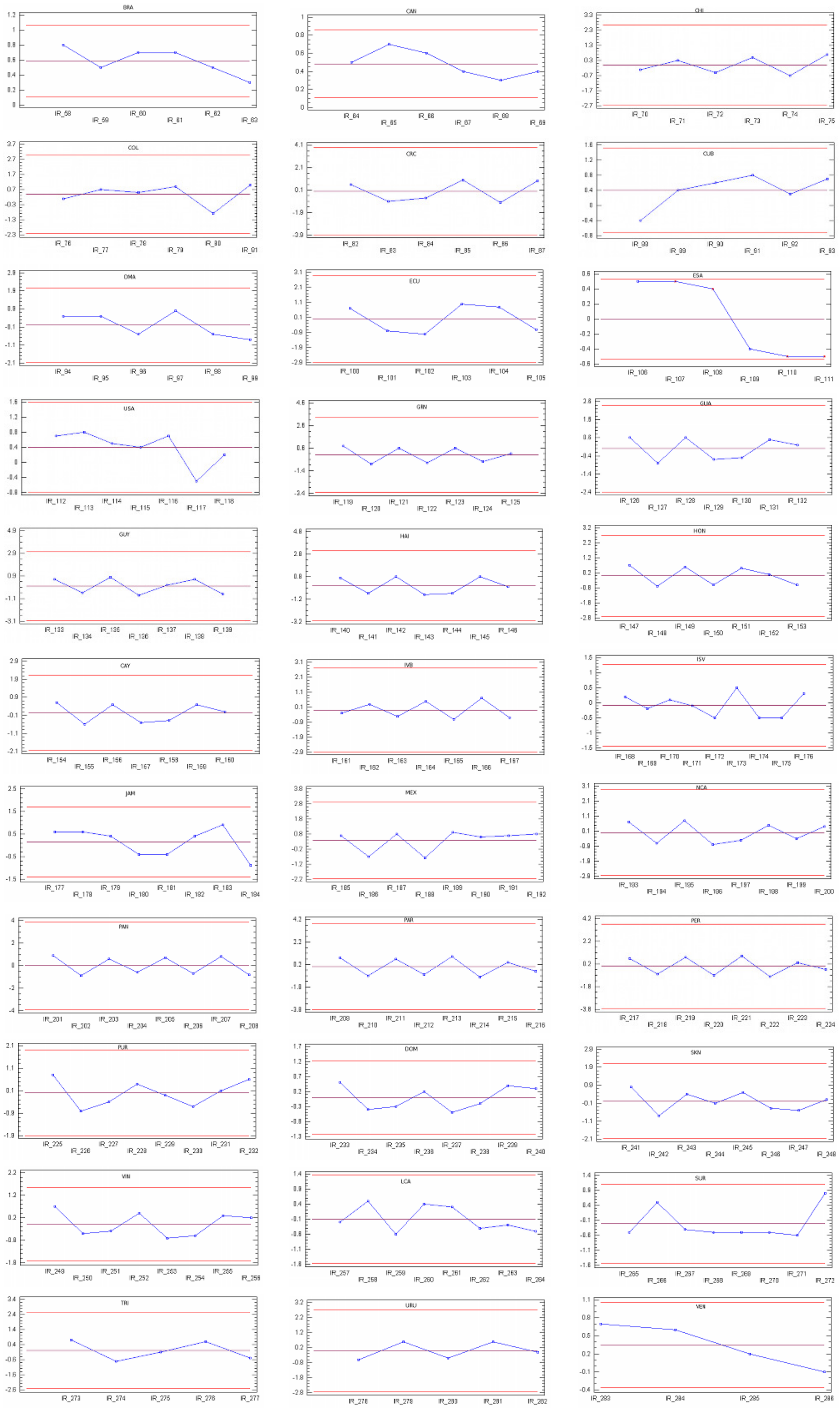


Fig. 5: Polarity of six Internet resources -Internet Resource 1 (IR1), ..., , Internet Resource 6 (IR6)- for the last thirty-three countries. For example in Fig. 5(19) the polarity of Jamaica (JAM) is positive for the first three resources, negative for the following two, and postive for the last one.

Producing Friend Recommendations in a Social Bookmarking System by Mining Users Content

Matteo Manca, Ludovico Boratto, Salvatore Carta
Dip.to di Matematica e Informatica
Università di Cagliari
Via Ospedale 72
09124 Cagliari, Italy
Email: {matteo.manca,ludovico.boratto,salvatore}@unica.it

***Abstract*—Social Bookmarking Systems (and Social Media Systems in general) are experiencing a quick growth in the number of active users. This expansion led to the well-known "social interaction overload" problem, that means that each user has too many potential people to interact with. In order to address this problem, user recommender systems are widely proposed in the social media literature to recommend friends or people to follow. Currently, there are no approaches able to produce friend recommendations in the Social Bookmarking Systems domain. In this paper we propose a friend recommendation algorithm for a Social Bookmarking System, based on low computational effort heuristics that allow real time applications. Experimental results show that, when users tag in the same way and are also interested in the same content, they can be recommended as friends. The proposed algorithm produces better results, with respect to policies that use only tags and do not consider content.**

Keywords*—*Social Bookmarking; Friend Recommendation; Tagging System.

I. Introduction

With the explosion of the Web 2.0, we observed a rapid growth of Social Bookmarking Systems and, in general, of all the forms of Social Media Systems. Social Bookmarking Systems allow users to use keywords (*tags*) to describe web pages that are of interest for them, helping to organize and share the resources with other users in the network [14]. The most widely-known example of Social Bookmarking Systems is Delicious.

In this domain, where users are connected and interact with each other, the growth of the population and the large amount of content led to scarcity of attention and to the well-known "social interaction overload" problem [7], [8]. These two problems are strongly related, since each user has too many potential users and items to interact with and this does not allow to focus on users or items that might be interesting for her/him. As a solution, the recommender systems research area recently put a lot of attention in the Social Media Systems domain, by developing a new class of systems named *social recommender systems* [6]. One of the most important areas social recommender systems focus on is *user recommendation*. User recommendation in the social domain aims at suggesting *friends* (i.e., recommendations are built for pairs of users that are likely to be interested in each other's content) or *people to follow* (i.e., recommendations are built for a user, to suggest users that might be interesting for her/him) [7]. These systems can be classified into three categories:

1) Systems based on the exploration of social graphs, that analyze the set of people related to the considered user, in order to produce recommendations. These systems recommend either the closest users in the graph, like friends of friends and followees of followees (the most famous example of this type of systems is Facebook [17]), or perform a random walk on the graph, in order to recommend the users that have the highest probability to be crossed (the main reference for this type of systems is Twitter [2]).
2) Systems based on the analysis of the interactions of the users with the content of the system (tags, likes, shares, posts on news, bookmarks, pictures, etc.). In order to exploit the interests, these systems usually apply complex algorithms. For example, some approaches build a user profile using TF-IDF (Term Frequency - Inverse Document Frequency) vectors [20] that, in order to be built, need to analyze each content the user interacts with [11]. Recommendations are produced by identifying users with similar profiles.
3) Hybrid systems, that consider both the social graph and the interactions of the users with the content (an example is represented by [9]). The use of different sources of data to produce the recommendations increases the complexity of these systems.

As highlighted in the previous classification, social recommender systems that recommend users are often based on approaches that filter content, make classifications and explore graphs. These systems certainly achieve a high accuracy but most of them are so complex that it would be hard to apply them to a real world scenario that, as previously said, grows quickly and involves huge amounts of data. The application of a complex algorithm to a real world scenario would involve difficulties in capturing the evolution of the users interests when building the recommendations.

Since user recommendation in a social domain aims at suggesting friends or people to follow, it is important to notice that the recommendation of a friend involves mutual interests and that the list of recommended *friends* might be different from the list of recommended *people to follow*. In fact, given two users u_i and u_j, u_j might be interesting for u_i, but not vice versa. This means that u_j would be recommended to u_i as a user to follow, but not as a friend.

So, the design of these two types of systems is different, since they involve different notions of users similarity. To the best of our knowledge, there are no approaches in literature that build friend recommendations in a Social Bookmarking System.

This paper presents a friend recommendation algorithm in a social bookmarking system that, by mining the content of the target user, recommends users that have similar interests. The algorithm has the capability to make a selective use of the available information and does not consider the social graph, in order to use as less information as possible. For this reason, it lends itself well to real time evaluations. The algorithm has been compared with two other reference algorithms, in order to evaluate the performances in terms of accuracy and to infer which aspects are more beneficial to produce recommendations in this domain; another aspect that we explored is the trade-off between the accuracy of the algorithm and the number of users involved in the recommendation process.

Our work brings relevant scientific contributions to the social recommender systems research area, now described in detail:

- This is the first algorithm able to recommend friends in a Social Bookmarking System.
- This algorithm is able to exploit the interests of a user in a selective way and produce recommendations using a simple approach, that can be applied in real time.
- The proposed algorithm has been tested, in order to evaluate how the considered information should be exploited (i.e., what information should be used and which weight should the considered interests have in the recommendation algorithm).

The proposed algorithm, both for its simplicity and because it is the first developed in this application domain, puts the basis on a research area previously not explored in the rapidly growing domain of social bookmarking systems.

The rest of the paper is organized as follows: Section II presents a formalization of a social bookmarking system; Section III describes the details of the recommender algorithm presented in this paper; Section IV illustrates the conducted experiments; Section V presents related work and Section VI contains comments, conclusions and future work.

II. Social Bookmarking Systems

Starting from the definition of a Social Tagging System given by Zhou et al. [16], we can state that a Social Bookmarking System consists of a set of users U, a set of bookmarks B, a set of tags T and a set of links between users L. Let $S = \{U, B, T, L\}$ be a Social Bookmarking System where:

- $U = \{u_i\}_{i=1}^{n}$ is a set of n users;
- $B = \{b_i\}_{i=1}^{w}$ is a set of w bookmarks;
- $T = \{t_i\}_{i=1}^{k}$ is a set of k tags;
- $L = \{l_i\}_{i=1}^{m}$ is a set of m links between pairs of users; these links may be bi-directional (i.e., a friendship) or uni-directional (i.e., one user follows the other).

Starting from the definition given above, we can define:

- $UB \subseteq B \times U = \{b_i | b_i \in B$ is a bookmark tagged by user $u \in U\}$ is the set of bookmarks used by u;
- $UT \subseteq T \times U = \{t_i | t_i$ is a tag used by user $u \in U$, $t_i \in T\}$ is the set of tags used by u;
- $BT \subseteq T \times U \times B = \{t_i | t_i$ is a tag used by user $u \in U$ to annotate the bookmark $b \in B$, $t_i \in T\}$;

The algorithm presented in this paper aims at finding previously unknown bi-directional links $l_k \in L(u_i, u_j)$, in order to recommend a friendship between user u_i and u_j.

III. Recommending Friends in a Social Bookmarking System

A. Motivation

The motivation of our algorithm is twofold. As mentioned in the Introduction, to the best of our knowledge there are no studies that propose an approach to recommend friends in the Social Bookmarking Systems domain. Secondly, a relevant aspect of a recommender system that operates in the social domain is the need to capture the user interests using lightweight algorithms; in fact too complex approaches may require too much time to infer the users interests. Therefore, when recommendations are produced, the estimated interests of a user may not consider the current ones that, in the meanwhile, may have been updated. So, motivated by the thesis proposed in [16] that the tagging activity of the users reflects their interests and by the intuition that users with similar interests use similar tags and the same bookmarks, we developed an algorithm that, given a Social Bookmarking System S, makes a selective use of the available information about interests to produce accurate friend recommendations. To be more precise, our algorithm computes user similarities with low computational cost metrics based on the set of bookmarks B and on the set of tags T.

B. Algorithm

Our algorithm is based on two similarity metrics, computed considering the tags and the bookmarks used by a user. Given a target user $u_t \in U$, the algorithm recommends the users with high tag-based and bookmark-based similarities. The algorithm works in three steps:

1) *Tag-based similarity computation.* The first similarity calculated among a target user u_t and the other users, is based on the tags used by each user. Given the number of times each tag was used by a user, Pearson's correlation is used to derive the similarity.
2) *Bookmark-based similarity computation.* The second type of similarity is the percentage of common bookmarks among u_t and the other users.
3) *Recommendations selection.* This step recommends to u_t the users with both a tag-based and a bookmark-based similarity higher than a threshold value.

In the following, we will give a detailed description of each step.

1) Tag-based Similarity Computation: Considering the previously given definition of $S = \{U, B, T, L\}$, we represent each user u with a vector $\overrightarrow{v_u} = \{v_{u1}, v_{u2}, ..., v_{uk}\}$, where each element v_{ui} is the relative frequency of each tag $t_i \in T$ used by $u \in U$ and is computed as follows:

$$v_{ui} = \frac{f_{ui}}{\#UT(u)} \tag{1}$$

Value f_{ui} represents the frequency of a tag $t_i \in T$ for user u. Given that each user is represented by a vector based on tag frequencies and that [16] states that users' interests are reflected in their tagging activities, our algorithm computes the first user similarity with the Pearson's correlation coefficient [19], to infer users with similar interests. We chose to use this metric because, as proved by Breese et al. [15], it is the most effective technique for the similarity assessment among users.

Let $\{u, m\}$ be a pair of users represented respectively by vectors $\overrightarrow{v_u}$ and $\overrightarrow{v_m}$. Our algorithm computes the tag-based user similarity tb as defined in (2):

$$tb(u,m) = \frac{\sum_{i \subset T_{um}}(v_{ui} - \overline{v}_u)(v_{mi} - \overline{v}_m)}{\sqrt{\sum_{i \subset T_{um}}(v_{ui} - \overline{v}_u)^2}\sqrt{\sum_{i \subset T_{um}}(v_{mi} - \overline{v}_m)^2}} \tag{2}$$

where T_{um} represents the set of tags used by both users u and m and values $\overline{v}_u$ and $\overline{v}_m$ represent, respectively, the mean of the frequencies of user u and user m. The metric compares the frequencies of all the tags used by the considered users. The similarity values range from 1.0, that indicates complete similarity, to -1.0, that indicates complete dissimilarity. Herlocker et al. [13] demonstrated that negative similarities are not significant to evaluate the correlation among users, so in our algorithm we consider only positive values.

2) Bookmark-based similarity computation: To increment the system knowledge on user interests, our algorithm combines the tag-based similarity presented above with another metric based on bookmarks. The metric calculates the percentage of common bookmarks between two users u and m.

Let us consider $UB(u)$, i.e., the set of bookmarks used by a user $u \in U$. We define $D(u,m) = UB(u) \cap UB(m) = \{b_i | b_i \in UB(u) \land b_i \in UB(m)\}$ as the sets of bookmarks used by both user u and user m. Given a pair of users $\{u, m\}$, we compute the bookmark-based user similarity *bb*, by considering the common bookmarks among the users, as follows:

$$bb(u,m) = \frac{\#D(u,m)}{\#UB(u)} \tag{3}$$

where $\#D(u,m)$ and $\#UB(u)$ represent, respectively, the cardinality of the sets $D(u,m)$ and $UB(u)$. We can notice that, since the *bb* metric is calculated as a percentage, the similarity is based on the number of bookmarks used by the user that we are comparing (i.e., $\#UB(u)$). This means that, differently from previously computed metric, similarity $bb(u,m)$ can be (and often it is) different from $bb(m,u)$. Our algorithm considers both values.

3) Recommendations selection: Once the tag-based and the bookmark-based user similarities are computed for each pair of users, our algorithm chooses a set of users to recommend to the target user by selecting:

- the ones that have a tag-based user similarity higher than a threshold value α (i.e., $tb > \alpha$);
- the ones that have a bookmark-based user similarity (at least one of the two computed) higher than a threshold value β (i.e., *bb* $> \beta$).

So, given a target user u_t, the candidate set $CS(u_t)$ of users to recommend is selected as follows:

$$CS(u_t) = \{u_i \in U \,|tb(u_t, u_i) > \alpha \,\&\&(bb(u_t, u_i) > \beta) \,\|\, (bb(u_i, u_t) > \beta)\} \tag{4}$$

IV. Experimental Framework

This section presents the framework used to perform the experiments. The dataset used and the data preprocessing are first described. Then, the metrics used for the evaluation are presented. The last part of the section presents the experimental setup and the obtained results.

A. Dataset and pre-processing

Experiments were conducted on a Delicious dataset distributed for the HetRec 2011 workshop [12]. It contains 1867 users, 69226 URLs, 7668 bi-directional user relations, 53388 tags, 437593 tag assignments (i.e., tuples [user, tag, URL]), 104799 bookmarks (i.e., distinct pairs [user, URL]).

We pre-processed the dataset, in order to remove all the users that were considered as "inactive", i.e., the ones that used less than 5 tags and less then 5 URLs.

B. Metrics

Given a set of recommendations $R = \{\cup CS(u_t), \forall u_t \in U\}$ and a set of correct recommendations $C \subseteq R$ (i.e., the pairs of recommended users that also appear in the dataset as a bi-directional user relations), recommendation *accuracy* is defined as the ratio of correct recommendations among all recommendations and it is computed as showed in (5).

$$accuracy = \frac{\#C}{\#R} \tag{5}$$

The other aspect considered in the evaluation is the *user coverage*, that represents the percentage of users involved in the recommendations, i.e., for how many users the algorithm is able to produce recommendations, given a specific threshold value. The metric can be computed as follows:

$$userCoverage = \frac{\#R}{\#U} \tag{6}$$

C. Experiments

Strategy. We performed two different experiments. The first aims to make an *evaluation of the recommendations*, by exploring the accuracy of the algorithm with different thresholds, while the other aims to make an *evaluation of the user coverage*, by exploring the trade-off between accuracy and user coverage.

In order to evaluate the recommendations we implemented a state-of-the-art policy [16], that we used as reference algorithm. Zhou et al. [16] developed a tag-based user recommendation framework and demonstrated that tags are the most effective source of information to produce recommendations. We compare the performances of our algorithm with respect to that of the reference one, that uses only tags i.e., with $bb = 0$), in terms of accuracy. Supported by the thesis that the use of only one source of data leads to better performances, we designed a second reference algorithm that considers only the bookmark-based similarity (i.e., with $tb = 0$).

In order to explore the trade-off between the accuracy analyzed in the previous experiments and the user coverage, we evaluate how the number of involved users (i.e., the user coverage) changes with respect to the tag-based user similarity tb and the bookamrk-based user similarity bb.

During the analysis of the accuracy, we evaluated all the values of parameters α and β between 0 and 1, using a 0.1 interval. When analyzing the user coverage, we also considered the values of β between 0 and 0.1, with a 0.01 interval, in order to evaluate in more the detail the user coverage studied considering bb (results will help motivating our choice to extend this analysis).

The experimental setup and the results are now described.

Evaluation of the recommendations. Given a target user u_t, the algorithm built a candidate set, $CS(u_t)$, of users to recommend. For each recommended user $u_i \in CS(u_t)$, we analyze the bi-directional user relations in the dataset to check if there is a link between the target user u_t and the recommended user u_i (i.e., if the users are friends). Let $R = \{\cup CS(u_t), u_t \in U\}$ be the set of all recommendations. We consider as *correct recommendations* the set $C \subseteq R$ of all the recommendations for which there is a correspondence in the relations of the dataset. This experiment analyzes the amount of correct recommendations in terms of *accuracy*. Given different values of α and β, the accuracy of the algorithm is calculated, in order to analyze how the performances of the algorithm vary as the similarities between users grow. The obtained results are illustrated in Fig. 1 and Fig. 2.

Fig. 1 shows how the accuracy values change with respect to the bookmark-based user similarity bb. The figure contains a line for each possible value α of the tag-based user similarity tb. We can observe that the accuracy values grow proportionally to the bb values. This means that the more similar are the users (both in terms of tag-based similarity and of bookmark-based similarity), the better the algorithm performs. However, for bb values higher than 0.5 no user respects the constraints, so we cannot make any recommendation; this is the reason why there are no accuracy values for bookmark-based user similarities higher than 0.5 ($bb > 0.5$). Fig. 2 shows the same results from the tag-based user similarity point of view. The figure illustrates the accuracy values with respect to the tag-based user similarity tb; here each line presents the results for a given value β of the bookmark-based user similarity bb. As results show, also from this perspective, the accuracy values grow proportionally to the tb values. The red lines in Fig. 1 and Fig. 2 show the results of the reference algorithms, where $tb = 0$ and $bb = 0$. In both cases, the two metrics combined improve the quality of the recommendations with respect to the cases where only one is used.

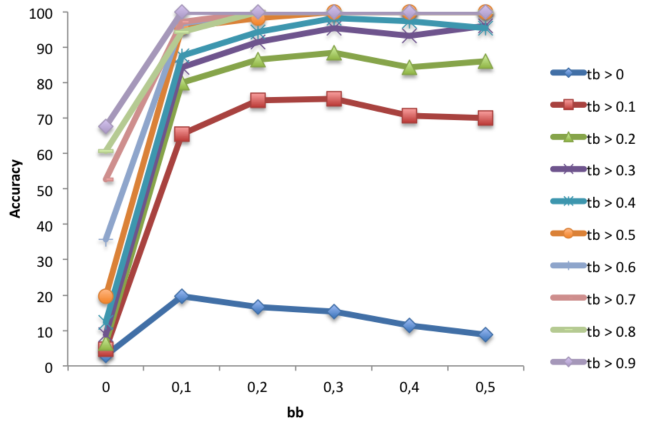


Fig. 1. Accuracy of the algorithm with respect to bookmark-based user similarity bb, for each value of the tb user similarity

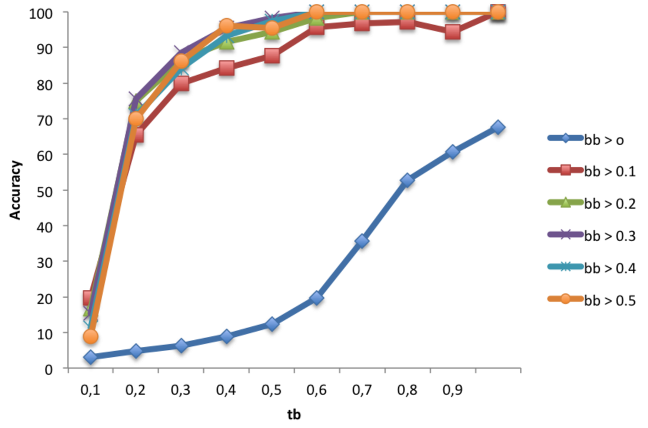


Fig. 2. Accuracy of the algorithm with respect to tag-based user similarity tb, for each value of the bb user similarity

Evaluation of the user coverage. In this experiment, we study how the *user coverage* of the algorithm (i.e., the percentage of users involved in the recommendations) changes with respect to the tag-based user similarity tb and the bookmark-based user similarity bb. As Fig. 1 shows, when the behavior of the user coverage with respect to the bookmark-based user similarity bb is analyzed, each value of bb is combined with several values of tag-based user similarity tb. In this experiment we are interested only in the tb value that leads to the maximum values of user coverage. In the same way, we evaluate the user coverage with respect to the tag-based user similarity tb values, considering only the bookmark-based user

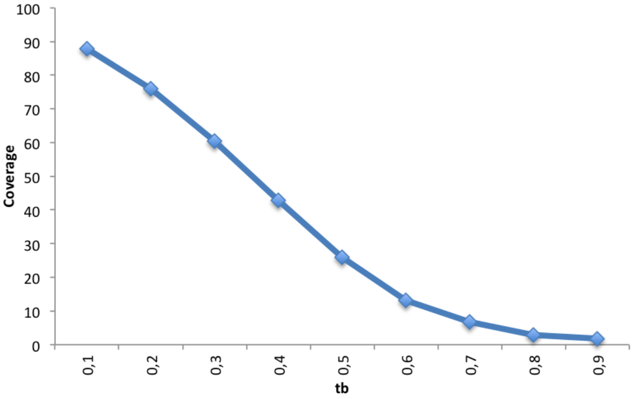


Fig. 3. User coverage of the algorithm with respect to tag-based user similarity tb

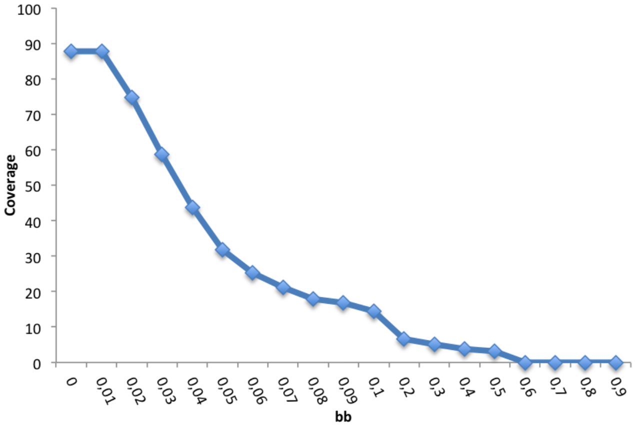


Fig. 4. User coverage of the algorithm with respect to bookmark-based user similarity bb

similarity bb that leads to the maximum user coverage values. Experiments are repeated with different values of α and β.

The results are presented in Fig. 3 and Fig. 4.

Fig. 3 shows the user coverage values with respect to the values of the tag-based user similarity tb; as previously mentioned, in this figure we do not have a line for each value of bookmark-based user similarity bb but we represent just a line that corresponds to $bb = 0.01$, which is the case that leads to the maximum values of user coverage. The same consideration can be made for Fig. 4, that represents the trend of the user coverage with respect to the bookmark-based user similarity bb; also in this case, we do not represent a line for each possible value of the tag-based user similarity tb, but just the values that correspond to $tb = 0.1$ (i.e., the value that allows to reach the maximum user coverage). As expected, high values of the thresholds α and β (that indicate a high similarity among users) correspond to low user coverage values. Effectively, in both Fig. 3 and Fig. 4 we can observe that we have user coverage values lower than 50% (that on a scale which ranges from 0 to 100 can be considered low values) for values of tb higher than 0.3 and for values of bb higher than 0.03. In Fig. 4 we can also observe that for values of the bookmark-based similarity bb higher than 0.5 the user coverage is 0 and that a consistent variation of the user coverage is between 0 and 0.1 (this is why we chose to extend our analysis by considering also those values).

V. Related Work

In the last years, Social Bookmarking Systems have been studied from different points of view. This section presents related work on user recommendation in this research area.

In [2], Gupta et al. present Twitter's user recommendation service based on shared interests, common connections, and other related factors. The proposed system builds a graph in which the vertices represent users and the directed edges represent the "follow" relationship; this graph is processed with an open source in-memory graph processing engine called Cassovary. Finally, recommendations are built by means of a user recommendation algorithm for directed graphs, based on SALSA (Stochastic Approach for Link-Structure Analysis). Our algorithm differs because we make friend recommendations and, furthermore, our algorithm uses just a restricted set of available information, without considering the social graph.

In [11], Chen et al. describe a people recommender system in an enterprise social network domain. They compare four algorithms, two based on social relationship information and two based on content similarity and demonstrate that the algorithms that use social information are stronger to find known contacts, while algorithms based on content similarities are better to discover new friends. We cannot compare with this approach, since it is applied to a delimited enterprise social network domain.

Guy et al. [10] describe a people recommender system for the IBM Fringe social network. The system uses enterprise information like org chart relationships, paper and patent co-authorship and project co-membership, which are specific of this social network, so it is hard to compare to them.

Hannon et al. [9] describe a followee recommender system for Twitter based on tweets and relationships of their Twitter social graphs. By using this information, they build user profiles and demonstrate how these profiles can be used to produce recommendations. In our work, we do not use any social connection information and furthermore we recommend friendship relationship and not users to follow.

In [3], a recommender system based on collocation (i.e., the position of the user) is presented. It uses short-range technologies of mobile phones, to infer the collocation and other correlated information that are the base for the recommendations. In our domain we do not have such a type of information, so we cannot compare with this algorithm.

Zhou et al. [16] propose a framework for users' interest modeling and interest-based user recommendation (meant as people to follow and not as a friend), tested on the Yahoo! Delicious dataset. Recommendations are produced by analyzing the network and fans properties. Differently from this framework, our algorithm produces friend recommendations.

In [1], a study about what cues in a user's profile, behavior, and network might be most effective in recommending people, is presented. As previously highlighted, we are interested in producing recommendations only based on users' content.

Liben-Nowell and Kleinberg [5] studied the user recommendation problem as a link prediction problem. They develop several approaches, based on metrics that analyze the proximity of nodes in a social network, to infer the probability of new connections among users. Experiments show that the network topology is a good tool to predict future interactions. We aim at using more basic information and not graphs or network topologies.

In [18], Arru et al. propose a user recommender system for Twitter, based on signal processing techniques. The considered approach defines a pattern-based similarity function among users and makes use of a time dimension in the representation of the users profile. Our algorithm is different because we aim at suggesting friends while on Twitter there is no notion of "friend" but it works with "people to follow".

VI. CONCLUSIONS

This paper presented a friend recommendation algorithm in the Social Bookmarking System domain as a means to link users with similar interests. The goal was to infer users' interests from content, making a selective use of the available information and without using complex algorithms, hard to apply to a real world scenario. As results show, our algorithm produces accurate recommendations by using the tags and the bookmarks used by users. We also explored the trade-off between recommendation accuracy and user coverage and observed that high values of similarity lead to low values of coverage. A comparison with a state-of-the-art policy, that considers only the tags, shows that the combined use of tags and bookmarks leads to improvements with respect to this one.

Future work will focus on evaluating the accuracy of the recommendations by using different metrics, like Precision and Recall, that allow both to measure the amount of correct recommendations and to evaluate the proposed algorithm from new perspectives.

ACKNOWLEDGMENT

This work is partially funded by Regione Sardegna under project CGM (Coarse Grain Recommendation) through Pacchetto Integrato di Agevolazione (PIA) 2008 "Industria Artigianato e Servizi".

REFERENCES

[1] M. J. Brzozowski and D. M. Romero, "Who should i follow? recommending people in directed social networks," in *ICWSM*, 2011.

[2] J. L. Pankaj Gupta, Ashish Goel, A. Sharma, D. Wang, and R. Zadeh, "Wtf: The who to follow service at twitter," in *Proceedings of www2013 Conference*, 2013.

[3] D. Quercia and L. Capra, "Friendsensing: recommending friends using mobile phones," in *Proceedings of the third ACM conference on Recommender systems*, ser. RecSys '09. New York, NY, USA: ACM, 2009, pp. 273-276.

[4] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi, "Short and tweet: experiments on recommending content from information streams," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1185–1194.

[5] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, ser. CIKM '03. New York, NY, USA: ACM, 2003, pp. 556–559.

[6] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Berlin: Springer, 2011, pp. 1–35.

[7] I. Guy, L. Chen, and M. X. Zhou, "Introduction to the special section on social recommender systems," *ACM TIST*, vol. 4, no. 1, p. 7, 2013.

[8] H. A. Simon, "Designing organizations for an information rich world," in *Computers, communications, and the public interest*, M. Greenberger, Ed. Baltimore: Johns Hopkins Press, 1971, pp. 37–72.

[9] J. Hannon, M. Bennett, and B. Smyth, "Recommending twitter users to follow using content and collaborative filtering approaches," in *Proceedings of the fourth ACM conference on Recommender systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 199–206.

[10] I. Guy, I. Ronen, and E. Wilcox, "Do you know?: recommending people to invite into your social network," in *Proceedings of the 14th international conference on Intelligent user interfaces*, ser. IUI '09. New York, NY, USA: ACM, 2009, pp. 77–86.

[11] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 201–210.

[12] I. Cantador, P. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011)," in *Proceedings of the fifth ACM conference on Recommender systems*, ser. RecSys '11. New York, NY, USA: ACM, 2011, pp. 387–388.

[13] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Research and Development in Information Retrieval*, American Association of Computing Machinery. American Association of Computing Machinery, 8/1999 1999.

[14] U. Farooq, T. G. Kannampallil, Y. Song, C. H. Ganoe, J. M. Carroll, and L. Giles, "Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics," in *Proceedings of the 2007 international ACM conference on Supporting group work*, ser. GROUP '07. New York, NY, USA: ACM, 2007, pp. 351–360.

[15] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, ser. UAI'98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[16] T. C. Zhou, H. Ma, M. R. Lyu, and I. King, "Userrec: A user recommendation framework in social tagging systems," in *AAAI*, M. Fox and D. Poole, Eds. AAAI Press, 2010.

[17] F. Ratiu, "Facebook:people you may know," May 2008. [Online]. Available: https://blog.facebook.com/blog.php?post=15610312130

[18] G. Arru, D. F. Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti, "Signal-based user recommendation on twitter," in *WWW (Companion Volume)*, L. Carr, A. H. F. Laender, B. F. Lóscio, I. King, M. Fontoura, D. Vrandecic, L. Aroyo, J. P. M. de Oliveira, F. Lima, and E. Wilde, Eds. International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 941–944.

[19] K. Pearson, "Mathematical contributions to the theory of evolution. iii. regression, heredity and panmixia," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Math. or Phys. Character (1896-1934)*, vol. 187, pp. 253–318, Jan. 1896.

[20] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

Social Network-based Entity Extraction for People Ontology

Tian Tian
Department of Computer Science
Manhattan College
tina.tian@manhattan.edu

Soon Ae Chun
College of Staten Island
City University of New York
soon.chun@csi.cuny.edu

***Abstract*—When users want to search people, search engines face two basic challenges. One challenge is due to the fact that there are many people (entities) with the same name, i.e. a homonym problem. The other is an entity linking issue, where several words are linked to the same person. The homonyms create the search results with a long list of hits with mingled information of the different person with the same name. The end users need to sift through the documents that fit their needs. To improve the ambiguous search arising from homonyms, we previously implemented an Ontology-Supported Web Search System (OSWS) that utilizes an ontology to disambiguate the search term and that provides search results in different possible categories that a search term may belong to. For a prototype of the OSWS system, we developed an ontology by mining person names and retrieving data from resources such as DBpedia. However, DBpedia is incomplete and often outdated. In this paper, we extend our approach to using social networks for building a People Ontology (PO). Specifically, personal profile attributes and their values of famous people are extracted from public social networks pages, cleaned and mapped to the ontology, resulting in a significant increase of the domain coverage achieved by the People Ontology to support the Ontology-Supported Web Search System.**

Keywords-social networks; ontology; mining from social networks; semantic Web search

I. INTRODUCTION

Users' information needs in the digital era can be fulfilled by keyword-based search engines. However, the major search engines do not disambiguate homonymous search terms, especially the person names that may refer to several different people. The search results thus contain information of different people of the same name. To address the homonymous names, we developed a domain-specific ontology, namely People Ontology, where people with the same name are categorized into different classes based on their properties. This category information and other properties can be used for disambiguation. The utility of the People Ontology is shown in the Ontology-Supported Web Search (OSWS) System [1] [2] we have developed. The search system uses the People Ontology to disambiguate the people search by providing users with separate search results of each homonymous person separated with different categories.

Our approach to develop a domain-specific People Ontology for the Ontology-Supported Web Search system (OSWS) involves (1) retrieving person names by Google search suggestions and (2) extracting category and attribute information from DBpedia [3]. Google search completion feature suggests a set of potential names which is used to generate a candidate list of person concepts for the People Ontology [4]. We classified these names suggested by Google search into three different famous people categories, namely, A-List, B-List and C-List. We used the working definition of the famous people as the person whose full name is suggested with a minimal substring of the name. Thus, the smaller the substring of the name is, the more famous entity it may refer to. The A-List contains more famous people than B-List since the full name is suggested with fewer substrings (e.g. first name only) than the B-List candidates where the search suggestion requires more than the first name string. Similarly, C-List contains names with the least famous people, according to our working definition. This working definition is based on the assumption that the famous (or infamous) people are more likely to be used as the search keywords, that influences the Google search suggestion.

In order to establish the unique entity for the person in these candidate lists, we used DBpedia for any additional attributes and person categories for the People Ontology. The resulting ontology in [4] contains 3,241 people instances and over 60,000 relationships emanating from them.

DBpedia is a huge public resource that can be used for developing ontology. However, it was found that many concepts in the candidate lists, especially not so famous people, did not exist in DBpedia. Furthermore, DBpedia is a slow-changing data resource. To overcome these shortcomings to improve PO, additional sources of information were needed. In this paper, we use social networks such as Facebook or Twitter profiles, to gather additional, "fast-changing" information that can further disambiguate the homonymous people concepts.

In this paper, we present how we used a social network as a secondary resource to extract knowledge in the domain of famous people. Choosing Facebook [5] as a sample social network is motivated by the fact that it has become the largest social networking site in recent years [6]. Millions of users have integrated Facebook into their daily practices [6].

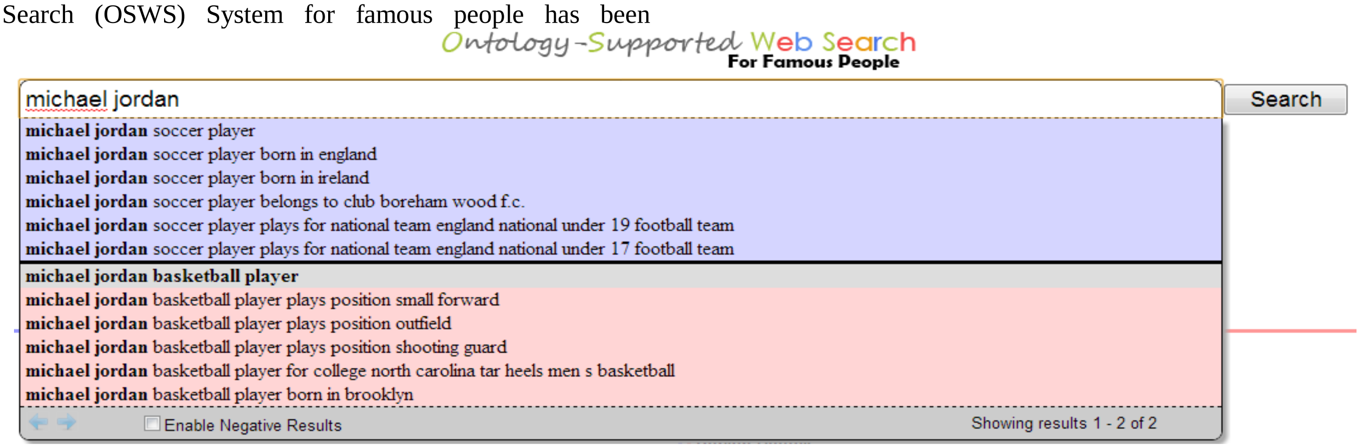
One can create public pages in Facebook. Public pages are for organizations and celebrities to broadcast information about them in an official, public manner [7]. More and more famous people are joining Facebook to publicize their profiles and news.

ISBN: 978-1-61208-311-7

The rest of the paper is organized as follows. Section II briefly explains the Ontology-Supported Web Search System to illustrate the application of People Ontology. In Section III, we describe the process of mining information from a social network and report the resulting enrichment of the People Ontology (PO). Section IV presents the related work. Section V concludes the paper and proposes future directions.

II. ONTOLOGY-SUPPORTED WEB SEARCH SYSTEM

In previous research, an Ontology-Supported Web Search (OSWS) System for famous people has been developed to improve the Web search process for homonymous terms. The system visually categorizes homonyms and provides search suggestions to the users [1] [2]. The OSWS System uses an ontology to derive the disambiguated search terms and suggested search completions based on the knowledge in the ontology (Fig. 1). The ontology was built by using search suggestions retrieved from Google, together with information extracted from DBpedia.



michael jordan

michael jordan soccer player
michael jordan soccer player born in england
michael jordan soccer player born in ireland
michael jordan soccer player belongs to club boreham wood f.c.
michael jordan soccer player plays for national team england national under 19 football team
michael jordan soccer player plays for national team england national under 17 football team
michael jordan basketball player
michael jordan basketball player plays position small forward
michael jordan basketball player plays position outfield
michael jordan basketball player plays position shooting guard
michael jordan basketball player for college north carolina tar heels men s basketball
michael jordan basketball player born in brooklyn

Showing results 1 - 2 of 2

Figure 1. Interface of the Ontology Supported Web Search System.

The candidate list of famous people was mined from Google's suggested completions [4]. We then passed the concepts in the list to DBpedia and extracted the related knowledge. Various methods have been applied to clean the extracted DBpedia information [4]. Despite its massive multi-domain coverage, many concepts in the candidate list were not found in DBpedia. Furthermore, new DBpedia releases appear only every couple of months [3], but some people become famous overnight. Therefore, we used social networks as the secondary, fast-changing resource to create a new famous People Ontology.

Using Facebook as an example of the "social network to ontology approach," concepts were checked against Facebook's Graph Search and the ones belonging to the "people categories" were selected as targets. A threshold was used to identify who qualifies as "famous" person. We then extracted relevant information regarding the famous persons. After data cleaning, the mined knowledge was integrated into the People Ontology.

III. SOCIAL NETWORK MINING FOR DEVELOPING THE PEOPLE ONTOLOGY

A. *Social Networks*

Turning to social networks as sources of information about famous people is a natural choice, as social networks utilize people as the primary topic of representation. Thus, we can view a social network as

- a set of concept nodes where each node represents a person or an organization;
- a set of semantic relationships between those nodes, expressing how different nodes relate to each other;
- one or more categorization relationships assigning person or organization concepts to different classes;
- a set of attributes of each concept node that characterizes and distinguishes different person/organization concepts from each other.

The described structure of a social network is remarkably similar to the structure of an ontology, as (some flavors of) ontologies are also based on concepts that are interconnected by IS-A relationships and semantic relationships and have additional attributes describing the concepts.

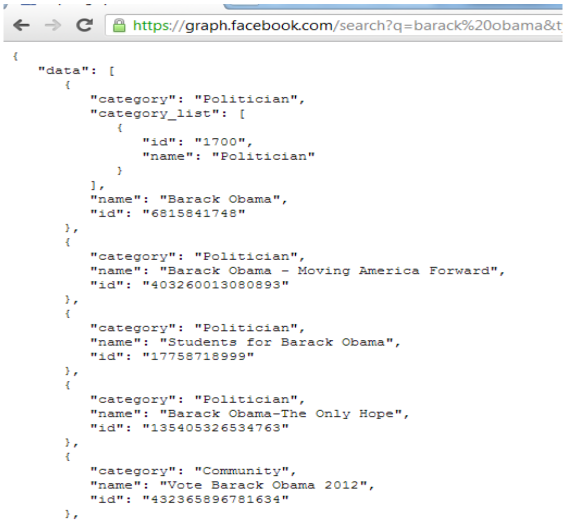
Examples of attributes in Facebook include "id," "name," "picture," "website," "birthday," "description," "likes," etc. "Likes" is an especially useful attribute, which represents the number of people that like a specific page.

Facebook users are linked to exactly one "category." The category information is mandatory to fill when the user creates a Facebook public page. There are 24 possible categories that a Facebook person page may belong to, including "actor/director," "artist," "athlete," "politician," "writer" etc. Mining of social network pages is possible because users can access category and attribute information by program.

B. *Identifying People in a Social Network*

One can search people in social networks by using their APIs or sending Web queries. Facebook provides such searches through http requests. The url below returns the first 10 Facebook pages with "Barack Obama" in the name:

https://graph.facebook.com/search?q=barack%20obama &type=page&limit=10.



```
{
   "data": [
      {
         "category": "Politician",
         "category_list": [
            {
               "id": "1700",
               "name": "Politician"
            }
         ],
         "name": "Barack Obama",
         "id": "6815841748"
      },
      {
         "category": "Politician",
         "name": "Barack Obama - Moving America Forward",
         "id": "403260013080893"
      },
      {
         "category": "Politician",
         "name": "Students for Barack Obama",
         "id": "17758718999"
      },
      {
         "category": "Politician",
         "name": "Barack Obama-The Only Hope",
         "id": "135405326534763"
      },
      {
         "category": "Community",
         "name": "Vote Barack Obama 2012",
         "id": "432365896781634"
      },
```

Figure 2. Top five Facebook ressults of query “Barack Obama.”

As Fig. 2 shows, several public pages have been created for Barack Obama. One can choose to consider all of them as valuable sources of information, or one can decide to use only the “authorized” page.

In previous research [4], we discussed how to create a small list of a few thousand very famous people, a larger list of famous people and a much larger list of somewhat famous people. We called these lists the A-List, B-List and C-List. The name lists were retrieved from the Google’s suggested completions. The A-List contains the most famous people, as they are the suggestions returned by giving a first name [4]. The B-List was retained by entering a first name with a letter. The C-List includes the least famous people by giving a first name with two letters to the search engine.

We began with a sublist of 2,564 names in the A-List that do not exist in DBpedia. We name it the “reduced-A-List.” The first 10 Facebook results were collected. The one page with the largest “likes” was chosen as the selected page, as a page with more attention tends to be more authoritative.

We checked the category information of the selected pages and identified 954 of them as people. However, some had very few fans. We found it necessary to define a threshold to determine which people are important enough to be considered famous, or which page(s) of a celebrity is (are) popular enough so that this person should be stored in the PO. Statistics show that median Facebook page has 218 fans [8]. In this study, 218 was used as the minimum number of “likes” for a Facebook page to be selected for analyzing and storing its namesake in the PO. Note that the number of Facebook fans is not the only measure in evaluating a person’s popularity. The names were first selected through Google’s search completions. Facebook then was used to validate if the names refer to people.

626 people were found in the “reduced-A-List” with over 218 “likes”. The pseudocode below shows the procedure of identifying famous people from Facebook given a name list.

```
PEOPLE_IDENTIFICATION(list){
  FOR each name in list{
    Search name in Facebook Graph Search
    Save the top 10 pages returned
    max_likes = -infinity
    FOR each returned page{
      IF (page.likes > max_likes){
        page_with_max_likes = page
        max_likes = page.likes
      }// End of IF
     }// End of inner FOR
     IF  (page_with_max_likes.category!=PERSON  OR max_likes < THRESHOLD)
       Remove page_with_max_likes
  }// End of outer FOR
}
```

C. *Mining Knowledge from a Social Network*

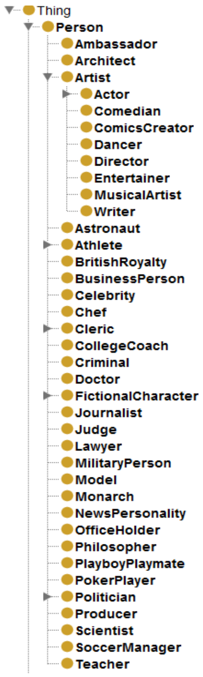
Most social network sites require users to establish their profiles when creating their accounts. Such a user profile may contain valuable information regarding the person. This section presents the process of extracting useful profile attributes from social networks. We save the mined attributes in a database and map them to the People Ontology.

In total, 33 different kinds of Facebook attributes were returned among the selected people. However, some attributes are Facebook-centric and have no use in suggesting search completions in the OSWS system. Thus, considering the usefulness of the attributes and after manually checking the quality and trustworthiness of the returned values of the attributes, a number of person attributes were chosen to be transferred into the PO. They include “name,” “category,” “likes,” “birthday,” “location,” “hometown,” “affiliation” of athlete, and “genre” and “record_label” of musician.

In a few cases, “location” stores irrelevant or even “wrong” information. Some examples include location data like “in the kitchen,” “in the world” and “home.” There are two ways to solve this problem. One is to query the returned location in a search engine and check if the first page of hits contains any url from mapping services. This method works fine for the “reduced-A-List,” but would cause delays when dealing with the much larger B-List [4]. In this study, yet another solution was applied. We used the “A-List” as the training data to extract stop words appeared in attribute “location.” Location data in the “B-List” was then automatically filtered with the stop word list.

“Category” is the most important attribute in disambiguating homonymous names. Most Facebook categories can be mapped directly to the PO. However, a few of them need further processing in order to provide better precision. They include “athlete,” “actor/director” and “public figure.”

An athlete could be a sportsman of different kind. A search suggestion of “Michael Jordan basketball player” is more informative than suggestion “Michael Jordan athlete.” By analyzing the additional Facebook attributes, such as “bio,” “description” and “personal_info,” and checking for matches with the 22 subdomains of athlete in the ontology, it was possible to specify 51 people playing specific sports among the 112 athletes found.



- Thing
  - Person
    - Ambassador
    - Architect
    - Artist
      - Actor
      - Comedian
      - ComicsCreator
      - Dancer
      - Director
      - Entertainer
      - MusicalArtist
      - Writer
    - Astronaut
    - Athlete
    - BritishRoyalty
    - BusinessPerson
    - Celebrity
    - Chef
    - Cleric
    - CollegeCoach
    - Criminal
    - Doctor
    - FictionalCharacter
    - Journalist
    - Judge
    - Lawyer
    - MilitaryPerson
    - Model
    - Monarch
    - NewsPersonality
    - OfficeHolder
    - Philosopher
    - PlayboyPlaymate
    - PokerPlayer
    - Politician
    - Producer
    - Scientist
    - SoccerManager
    - Teacher

Figure 3. Partial view of the expanded “Person” hierarchy in Protégé.

Facebook combines actors and directors in one category: “actor/director.” To provide better specification, we parsed other descriptive attributes to determine if the person belongs to the “actor” or the “director” category.

“Public figure” is one of the biggest categories found in the “reduced-A-List.” However, they do not provide much valuable information in disambiguating homonymous names. In order to assign those people with more concrete categories, we checked their Facebook attributes with a list generated with all Facebook person categories, classes in the PO and their synonyms in WordNet [9]. We used a Synonym API [10] to collect synonyms. The API is based on REST calls, which return well-formatted XML results, providing synonyms based on the WordNet database.

For the category-specific attributes, “affiliation” of athlete carries information about the team the athlete is playing for. A list of stop words was built to remove noise from the data. “Record_label” and “genre” were processed with the same filtering method. “Record_label” provides information about the company that manages the musician. “Genre” describes the type of music the musician plays. Fig. 3 shows a partial view of the final Person hierarchy in Protégé format.

D. Mapping Social Network Profiles to the People Ontology

The previous section described how the relevant social network attributes were selected and cleaned. In this section, we explain how we mapped and integrated the social network profiles to the PO.

Many Facebook categories exist in the People Ontology, such as “artist,” “athlete,” “journalist,” etc. Therefore, these categories can be directly mapped to the PO. For categories that do not exist in the People Ontology, we expanded the ontology by adding the new classifications in the hierarchy.

The other Facebook attributes were also mapped to the People Ontology, as seen in Table 1. The first column shows the Facebook attribute and the second column represents the corresponding attribute in the PO. The third column in the table shows the type of property (data type or object) used in the ontology. Attributes “name,” “likes” and “birthday” were stored as data properties. The remaining attributes were mapped to the PO as objects, thus, it was necessary make sure that no repetition of objects occurs in the ontology. An object property was only added if it did not already exist in the People Ontology.

TABLE I. FACEBOOK ATTRIBUTES TO FAMOUS PEOPLE ONTOLOGY MAPPING

Facebook Attribute	Ontology Mapping	Property Type
name	name	datatype
birthday	dateofBirth	datatype
likes	facebookLikes	datatype
location	currentPlace	object
current_location	currentPlace	object
hometown	placeofBirth	object
affiliation	playsForTeam	object
genre	musicalGenre	object
record_label	recordLabel	object

The processing of the previous OSWS Ontology used the number of relationships and attributes to determine the popularity of a famous person [4]. The Facebook number of “likes” provides the same measurement, but at a different scale, meaning the numbers cannot be combined. Therefore, a separate data type property “facebookLikes” was created in the PO.

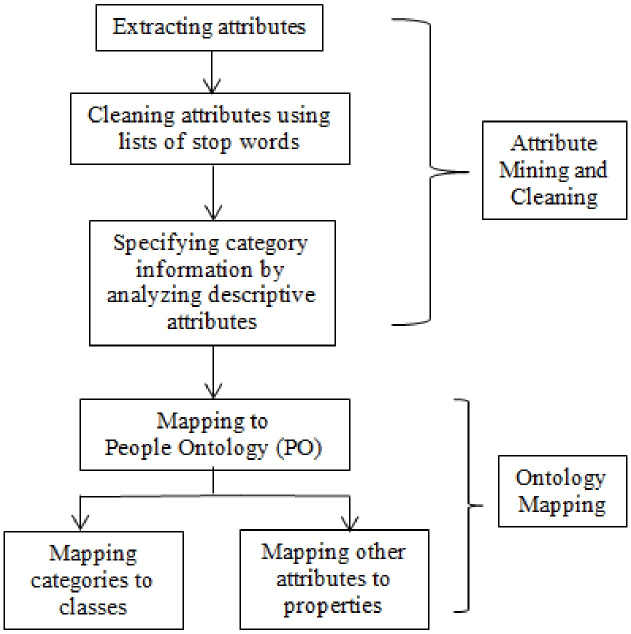

Extracting attributes
Cleaning attributes using lists of stop words
Specifying category information by analyzing descriptive attributes
Attribute Mining and Cleaning
Mapping to People Ontology (PO)
Mapping categories to classes
Mapping other attributes to properties
Ontology Mapping

Figure 4. Algorithm flow of mining and processing the A-List data .

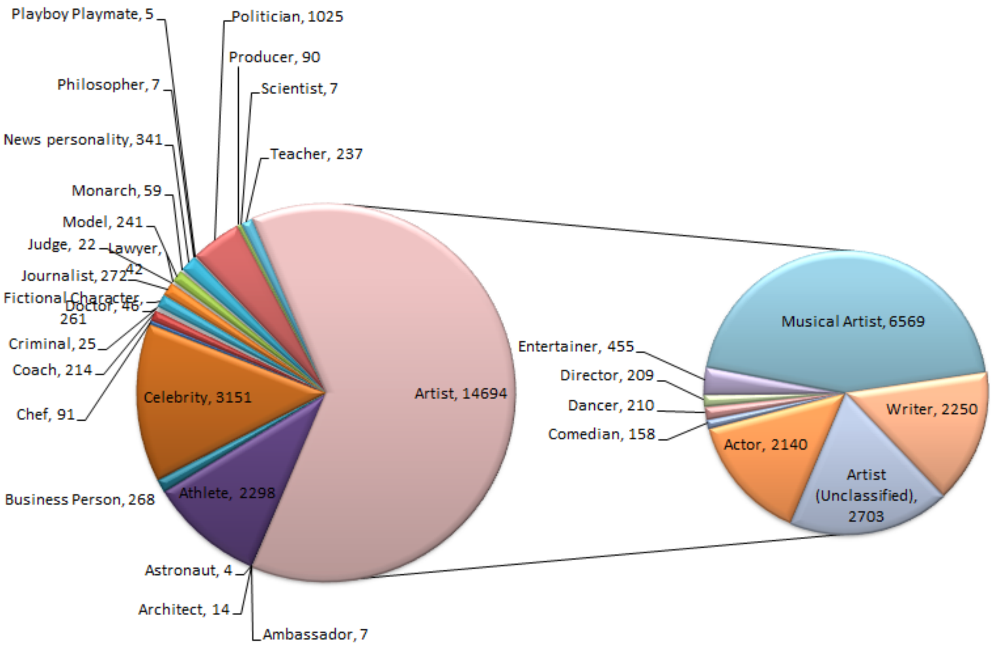


Figure 5. Distribution of the newly added B-List famous people among categories in the Famous People Ontology.

In total, 622 additional people who did not exist in the system were added to the PO by mining Facebook. Among the 626 names we analyzed in the A-List, 4 are musical band names instead of individuals. Since our ontology is about famous persons, we removed information representing groups of people from the results.

The new entities include 279 artists, 112 athletes, 113 celebrities and many other famous people from the remaining categories. For example, Sam Adams the singer is newly added to the ontology. His homonyms include Sam Adam the politician. Fig. 4 shows the algorithm flow of mining and processing the A-List data.

Using the A-List as the training set, we applied the same method of data extraction and data cleaning onto the B-List. Processing the B-List data was a complete automatic procedure. Among the 155,403 names in the B-List, only 35,626 were found in DBpedia. To expand the PO, the rest 119,777 names were stored in the "reduced-B-List" and queried against Facebook. In total, 40,360 people from the "reduced-B-List" were found in Facebook. Among them, we were able to identify 23,421 famous people with more than 218 likes on their Facebook pages.

Artist is the largest category among all with 14,694 people, including musical artists, actors, dancers, writers, etc. Fig. 5 shows the distribution of the famous people in the B-List among different categories. Each colored area is marked with the name of the category and the number of people that were found in this category.

This part of work was developed using the Facebook Graph API in Java. Unfortunately, Facebook allows only a limited number of Graph API calls per minute. Thus, a timer was set in the programs to send out one API call every two seconds. This slowed down our work considerably.

E. *Other Social Networks*

Twitter has become one of fastest growing online social networking services [11]. It has gained worldwide popularity with over 500 million users, including many public figures.

Twitter stores the following attributes for users: "user_id," "screen_name," "name," "profile_image_url," "location," "url," "description," "created_at," "followers_count," "friends_count," "statuses_count," "time_zone," and "last_update." In addition, Twitter stores information about every single tweet that is not statically associated with the user, e.g., "geo_lat" and "geo_long."

Twitter users are invited to follow people from certain categories. For example, when logging in, Twitter suggests people from "music," "sports," and "entertainment." Currently the following categories are supported: Music, sports, entertainment, twitter, funny, fashion, family, technology, food&drink, news, art&design, books, business, science, health, travel, government, staff picks, charity, nascar, pga, mtv movie awards, mlb, faith and religion, NBA, television, CMT awards, billboard music awards, US election 2012, NHL.

Twitter data can be mined by using the "phirehose" library [12] or with one of the built-in Twitter APIs [13]. Twitter's API provides the GET users search, which searches for users similar to Find People button on Twitter's official site [14]. The GET search returns a json object with all associated properties of the person, which includes useful information, such as name, location, id, etc.

It is common to have more than one Twitter user returned from a search request. The first returned result, in most cases, is the official Twitter account of the famous person. Another valuable attribute returned by the GET search is the account’s “verified” value. The property identifies if the returned Twitter profile is a verified account. Verification has been used to establish authenticity of identities on Twitter, including highly sought users in music, acting, fashion, government, politics, religion, journalism, media, advertising, business, etc. [15].

Twitter’s GET search also returns several attributes associated with the user’s account, including “name,” “location,” “description,” “followers_count,” etc. Compared to Facebook, Twitter has fewer profile attributes that can contribute to our People Ontology. Mapping can be built using string matching techniques.

Twitter limits its GET search to 60 calls per hour [14], which will be a major obstacle in this study.

IV. Related Work

Previous research has been reported on extracting data from social networks. Thelwall et al. have mined MySpace comments to detect the emotions [16]. Chu et al. have mined Facebook live feeds regarding social networking forensics [17]. Xu et al. investigated retrieving user opinions in social network services [18]. SONAR is an API for gathering and sharing social network information [19]. POLYPHONET was built as a social network extraction system [20].

Shibaki et al. have constructed a person ontology from Wikipedia by extracting person categories and the IS-A relationships among them [21]. However, the ontology does not contain other relationships or attributes other than the parent-child relations.

Mika [22] presents an approach to construct an ontology (folksonomy), based on the sub-community of an actor who interact with other actors, the semantic annotations (tags) the community use to describe documents, using tripartite graph. The concepts and ontology emerge from the associative relations within each sub-community and its interacting actors. He argues that the incorporation of the social context into the ontology models captures the idea that ontologies are inseparable from the context of the community in which they are created and used. It also highlights the emerging nature of ontologies, as opposed to the slow growing knowledge base such as WordNet. It is an algorithmic approach to construct folksonomy. Similar approach is proposed by Himanshu et al. [24] to construct ontology from social network using semantic tags used in a sub-community. However, the folksonomy constructions is a general approach to identify the concepts and disambiguation of concepts using social network, but not necessarily focus on a person ontology.

Finin et al. [23] have proposed the use of FOAF ontology (i.e., FOAF documents) to identify person, link and fuse distributed personal information using RDF, and develop a social network based on foaf:knows relations. It suggests the potential use cases of the person ontology represented in FOAF, but it does not address how to construct the ontology from the Web site.

In Information Retrieval community, the entity disambiguation is approached based on textual occurrences of names and its context. Bhattacharya and Getoor [25] use mutual relations between authors for entity resolution. In the context of citations we may conclude that "R. Srikant" and "Ramakrishnan Srikant" are the same author, since both are coauthors of another author. They consider the mutual relations between authors, paper titles, paper categories, and conference venues. Hassel et al. [26] uses the attribute information such as affiliation, topics of interests, etc. contained in an ontology derived from the DBLP [27], while Pilz [28] exploits the category information from Wikipedia for disambiguation. However, the focus is not to build an ontology of entities but utilizing them to disambiguate the names in a text.

We expand our previous approaches on exploiting social networks and DBpedia to construct and enrich a people ontology with more relationships [1] [2] [29]. To our knowledge, little research has been done in constructing ontologies from the social networking sites.

V. Conclusions and Future Work

This paper presented the process of mining a social network as a secondary resource to enrich the People Ontology, since the primary source of DBpedia had missing information of candidate people we extracted from Google Search suggestion. Using the social network mining approach we presented, we were able to classify 954 names in the A-List whose information was lacking in DBpedia.. A series of data extraction and data cleaning steps were performed to mine the Facebook public pages of the selected people. The standardized data was then mapped to the PO.

Using the same automated method, we were able to mine and map more than 23,000 people in the B-List that were located in Facebook to the People Ontology. Our approach shows a potential to develop ontology that can be scalable. The People Ontology can be utilized in semantic disambiguation of entities, and the linking of separate references. Our prototype semantic search system shows one utility of the People Ontology.

Currently Facebook is used for static information gathering. We plan to develop a mechanism in the future to automatically detect updates in the pages. In addition, we plan to investigate the friendship relations between people in the social network to further enrich the ontology.

In the future, we plan to extend the People Ontology by mining knowledge from other social networks, such as Twitter, LinkedIn and MySpace.

Acknowledgment

Christopher Ochs has contributed to the implementation of the OSWS system.

REFERENCES

[1] T. Tian, J. Geller, and S. A. Chun, "Improving web search results for homonyms by suggesting completions from an ontology," 2nd ICWE Workshop on Semantic Web Information Management (SWIM). Lecture Notes in Computer Science (LNCS), 2010, issue 6385, pp. 175-186, Springer.

[2] T. Tian, J. Geller, and S. A. Chun, "Enhancing interface for ontology-supported homonym search," CAiSe'11 Workshop on Semantic Web Search (SSW). Lecture Notes in Business Information Processing (LNBIP), 2011, issue 83, pp. 544-553, Springer Verlag, Berlin.

[3] DBpedia, http://dbpedia.org/About, retrieved 07/08/2013.

[4] C. Ochs, T. Tian, J. Geller, and S. A. Chun, "Google knows who is famous today: Building an ontology from search engine knowledge and DBpedia," 5th IEEE International Conference on Semantic Computing (ICSC), Palo Alto, CA, 2011, pp. 320-327.

[5] Facebook, www.facebook.com, retrieved 07/08/2013.

[6] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," Journal of Computer-Mediated Communication, 2007, vol. 13, issue 1, pp. 210-230.

[7] Facebook Graph API, http://developers.facebook.com/docs/reference/api/, retrieved 07/08/2013.

[8] T. McCorkindale, "Can you see the writing on my wall? a content analysis of the fortune 50's Facebook social networking sites," Public Relations Journal, 2010, vol. 4, no. 3, pp. 1-10.

[9] WordNet, http://wordnet.princeton.edu/, retrieved 07/08/2013.

[10] Stands4 API, http://www.abbreviations.com/api.asp, retrieved 07/08/2013.

[11] Twitter, www.twitter.com, retrieved 07/08/2013.

[12] The "Phirehose" Library, https://github.com/fennb/phirehose, retrieved 07/08/2013.

[13] Twitter REST API, https://dev.twitter.com/docs/api, retrieved 07/08/2013.

[14] Twitter GET User/Search, https://dev.twitter.com/docs/api/1/get/users/search, retrieved 07/08/2013.

[15] Twitter Account Verification, https://support.twitter.com/groups/31-twitter-basics/topics/111-features/articles/119135-about-verified-accounts, retrieved 07/08/2013.

[16] M. Thelwall, D. Wilkinson, and S. Uppal, "Data mining emotion in social network communication: Gender differences in MySpace," Journal of the American Society for Information Science and Technology, 2010, vol. 61, issue 1, pp. 190-199.

[17] H. Chu, D. Deng, and J. H. Park, "Live data mining concerning social networking forensics based on a Facebook session through aggregation of social data," IEEE Journal of Selected Areas in Communications, 2011, vol. 29, issue 7, pp. 1368-1376.

[18] K. Xu, S. S. Liao, Y. Song, and L. Liu, "Mining user opinions in social network webs," The Fourth China Summer Workshop on Information Management, Wuhan, China, 2010, pp. 39-49.

[19] I. Guy, M. Jacovi, E. Shahar, N. Meshulam, and V. Soroka, "Harvesting with sonar - the value of aggregating social network information," CHI, Florence, Italy, 2008, pp. 1017-1026.

[20] Y. Matsuo, J. Mori, and M. Hamasaki, "POLYPHONET: An advanced social network extraction system from the web," International World Wide Web Conference (WWW), Edinburgh, Scotland, 2006, pp. 262–278.

[21] Y. Shibaki, M. Nagata, and K. Yamamoto, " Constructing large-scale person ontology from Wikipedia," 2nd Workshop on Collaboratively Constructed Semantic Resources, Beijing, China, 2010, pp. 1-9.

[22] P. Mika, "Ontologies are us: a unified model of social networks and semantic," Web Semantics: Science, Services and Agents on the World Wide Web, 2007, vol. 5, issue 1, pp. 5-15.

[23] T. Finin, L. Ding, L, Zhou, and A. Joshi, "Social networking on the semantic web," Learning Organization, 2005, vol. 12, issue 5, pp. 418-435.

[24] M. Hamasaki, Y. Matsuo, T. Nisimura, and H. Takeda, "Ontology extraction using social network," International Workshop on Semantic Web for Collaborative Knowledge Acquisition, Harderabad, India, 2007.

[25] I. Bhattacharya and L. Getoor, "Relational clustering for multitype entity resolution," Fourth International Workshop on MultiRelational Data Mining, 2005, pp. 3-12.

[26] J. Hassel, B. Aleman-Meza, and I. B. Arpinar, "Ontology-driven automatic entity disambiguation in unstructured text," Lecture Notes in Computer Science (LNCS), 2006, pp. 44-57.

[27] DBLP, http://www.informatik.uni-trier.de/~ley/db/, retrieved 10/22/2013.

[28] A. Pilz, "Entity disambiguation using link based relations extracted from Wikipedia," 26th International Conference on Machine Learning, Haifa, Israel, 2010.

[29] S. A. Chun, T. Tian, and J. Geller, "Enhancing the famous people ontology by mining a social network," 2nd International Workshop on Semantic Search over the Web, Istanbul, Turkey, 2012.

A Ranking Algorithm for the Detection of Composite Concepts Based on Multiple Taxonomies

Daniel Kimmig
Institute for Applied Computer Science
Karlsruhe Institute of Technology
Germany
daniel.kimmig@kit.edu

Steffen Scholz
Institute for Applied Computer Science
Karlsruhe Institute of Technology
Germany
steffen.scholz@kit.edu

Andreas Schmidt
Department of Computer Science and
Business Information Systems
Karlsruhe University of Applied Sciences
Germany
andreas.schmidt@hs-karlsruhe.de

***Abstract*—A full-text search is typically not appropriate for concept mining. For that reason, we use taxonomies to describe the concepts we are looking for. A typical input for our search consists of two or more taxonomies, describing the concept we are looking for. In this paper, we present a similarity measure between the input taxonomies and the searched documents. The algorithm is based on the idea of word n-tuples, where each word in a result tuple comes from another taxonomy. Because of the vast number of available documents, our similarity function must be fast to allow a quick ranking of the retrieved documents. We also provide an optimized implementation for our algorithm, which allows a fast ranking of the searched documents.**

***Keywords*–ranking algorithm; taxonomy based search; similarity function; performance measure**

I. Introduction

In previous work [1], [2] we developed a searching strategy for (composite) concepts in document sets. The strategy was based on the idea of formulating concepts as taxonomies. So for example to look for documents containing information about "energy", we can use the taxonomy in Figure 1. The search process is then performed by looking for every word or phrase (and also defined synonyms - not shown in the Figure) in the taxonomy tree in the documents. The result for such a search is then a quantified taxonomy tree for each document, containing the number of occurrences of the words, phrases and synonyms. Additionally, the counts are cumulated toward the root of the tree. Figure 2 shows such a quantified result tree. Below each word in the taxonomy you find the number of occurrences inside the document. For all non-leaf nodes (*renewable fuel*, *coal*, *oil*, *fossil fuel*, and *energy*) you find additionally the accumulated number of occurrences for this sub tree, i.e., for the sub tree *renewable fuel*: $3\,(solar\ energy) + 4\,(wind\,power) + 2(geothermal) + 5(renewable fuel) = 14$. A more intuitive representation form could represent the weight of a node by different font sizes or colors or by different line widths of the edges in the taxonomy. The ranking of different documents can than be performed by simply taking the weight at the root of a taxonomy tree and an additional normalization step (i.e., division by the number of words in a text).

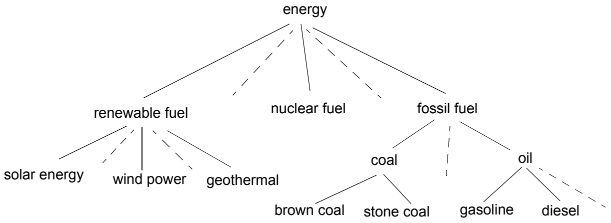


Figure 1. Energy taxonomy

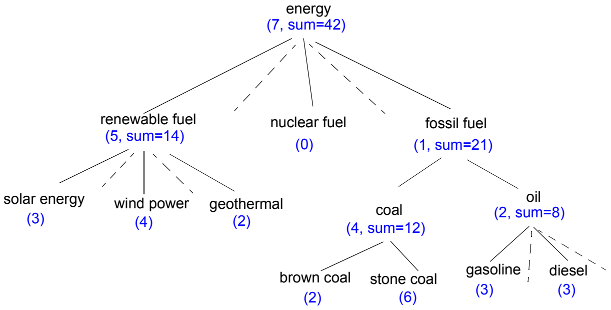


Figure 2. Quantified result taxonomy

Typically, we do not only search for one concept (like *energy*), but for a combination of concepts forming a more sophisticated concept like, i.e., *"used materials in the automotive industry"*. So instead of only looking for the concept of *material*, one is required to consider the application of different materials in *automotive manufacturing*. Relevant terms and phrases in the context of *vehicle manufacturing* (right side) and *material* (left side) are shown in Figure 3, representing a *isa* and a *is-part-of* taxonomy.

Looking for a single concept in a document is technically speaking a query which looks for the different words from the given taxonomy with an adjacent construction of the quantified result taxonomy, based on the words found in the document. In contrary, when we look for multiple concepts in a document, we have to search the documents for tuples, from which one word is from taxonomy A and the other word is from taxonomy B (in the case of two taxonomies).

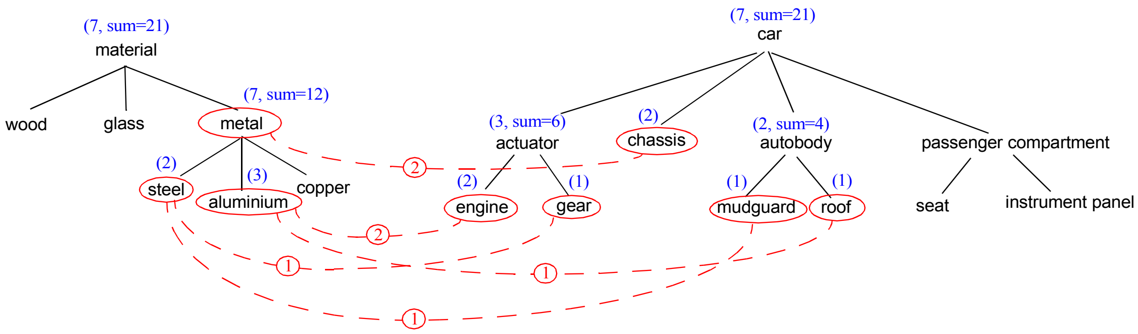


Figure 3. Relationship between taxonomies

In the case of three, four or more concepts to search, we have to find the corresponding n-tuples. This is illustrated in Figure 3. Here, the two taxonomies *car* and *material* are shown and the tuples which can be found in a concrete document are shown by connected ellipses (in red). The value along a connection link indicates how often a tuple combination was found in a document (i.e., the tuple *metal*, *chassis* appears two times in the document). As in the case of a single taxonomy these values are propagated toward the roots of the two trees (in blue). Mind, that in contrast to the case with one taxonomy, not the number of occurrences of the words/short phrases is counted, but only those words of the taxonomies which occur in a tuple. But this is only a simplified case, because it does not consider the distance of the words from the different taxonomies inside a found tuple. Consider the situation in Figure 4. In both situations the same number of tuples were found. In the first case (a), the average distance between words in the result tuple is much higher than in case (b). If the words from the different taxonomies appear near to each other, the probability is high, that the desired concept is described (i.e., an aluminum chassis). As a consequence, we have not only to consider the number of tuples found, but also the distance of the words in the found tuples for the ranking function.

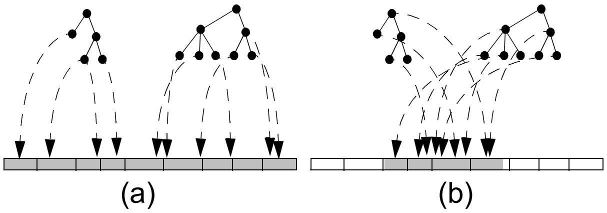


Figure 4. Interrelationships between concepts

While the implementation of a ranking function for the “one taxonomy” case is straightforward using an `OR` search in a conventional full text search engine like Lucene [3] or Sphinx [4] as basis and implementing the tree aggregation part on top, this is not possible for the multi-taxonomy case. The remainder of the paper is structured as follows: In Section II we review related work, which has already been done in this field. Afterwards, in Section III the concept of our fast ranking-algorithm is explained. Section IV shows the runtime behaviour of our algorithm, compared to the naive approach. We finish our paper in Section V with a scientific outlook for furher research.

II. Related Work

In the work of Cummins and O’Riordan [5], different proximity measures between pairwise terms were defined. Some of these measures are based on the distance between the occurences of the terms. Other measures take into account the term frequencies (tf) [6] of the related terms. In our work, we also use a proximity measure based on the distance of the involved terms, but in contrast to the previously mentioned work, we have to consider multiple taxonomies instead of multiple terms. Tao and Zhai [7] also mention the distance of the search terms in a document as an important factor for proximity measure. The authors add different metrics as complementary scoring components to different existing retrieval models to slightly adjust the final ranking. The results show that adding metrics, based on the distance of the terms, improve the overall retrieval accuracy, compared to the more coarse span-based measures. Again, this research focuses on single terms instead of taxonomies as in our work.

III. Algorithm

The main goal of our approach is to identify tuples of words and or phrases, which originate from different taxonomies and rank these based on the distance of each involved word/phrase. The following Figure 5 illustrates our concept. In the example, two taxonomies as well as an exemplary text serve as input to the ranking algorithm. Words from the taxonomies which appear within the exemplary text are highlighted in the respective colors (blue for T_1, red for T_2). Below the exemplary text, an array-like data structure is shown, which keeps track of the position of the word in the text as well as the origin of the taxonomy. Our ranking algorithm will iterate over this list of hits to identify

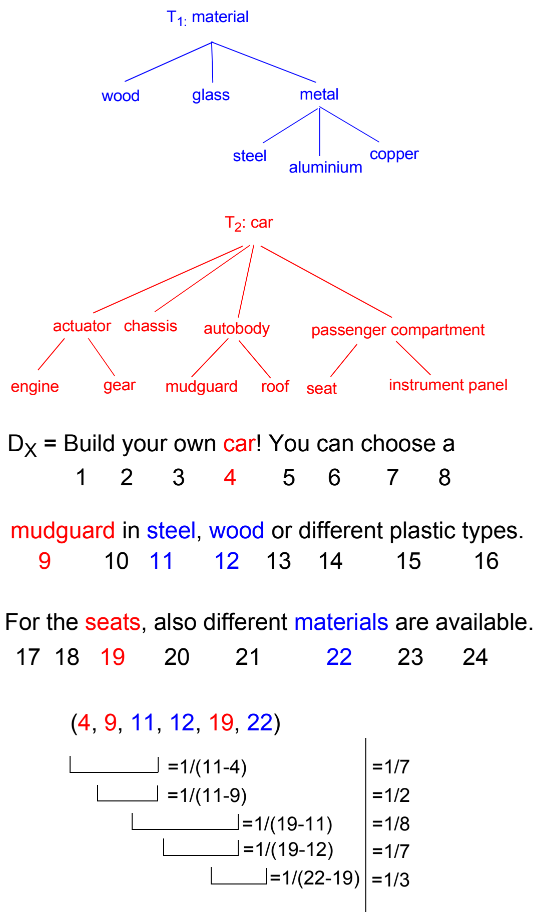


Figure 5. Ranking example using tuples from two taxonomies

tuples. A tuple is complete if words from all participating taxonomies are found. The first tuple is (car/steel) at position four and eleven. The distance value of a tuple plays a major role in the ranking formula. It is defined as the difference between the highest and lowest position value of the words of the tuple. In our case the distance value of the tuple (car/steel) is seven. This distance value embraces the fact that we are looking for cases in which words from all taxonomies appear very close to each other, as this indicates that the text is actually about a composite concept (see Figure 4). After all tuples have been identified as well as their distance is available, the ranking formula is calculated, which is shown in the following equation:

$$rank(T_1, T_2, D_x) = (\tfrac{1}{7} + \tfrac{1}{2} + \tfrac{1}{8} + \tfrac{1}{7} + \tfrac{1}{3}) * \tfrac{1}{24} = 0.052$$

The resulting ranking value is the sum of the inverse of each distance value divided by the length of the text. The inverse of the distance value is chosen to reduce the impact of higher distance values. Lower distance values indicate a high probability of the occurrence of a composite concept, which means the ranking should increase.

IV. Runtime Behavior

A naive approach to determine the ranking value is to iterate over the hit list while trying to find a closed tuple for each element. This makes it necessary to find complementary words from all other taxonomies for every element of the hit list, which requires three nested loops. This can become very costly depending on the size of the hit list as well as the amount of specified taxonomies as they determine when a closed tuple is achieved.

One potential optimization is to avoid the nested loops in some cases. This can be achieved by maintaining a dictionary of subsequent words from other taxonomies while calculating the ranking value across the entire hit list. This dictionary is used as a lookup table to complete tuples without searching for subsequent words for each element of the list. This dictionary is filled initially whenever the search for a closed tuple starts. The dictionary hereby serves as a memory of previous iterations in order to reduce the workload of subsequent stages of the processing.

Another idea is to stop the calculation once it is first encountered that no more tuple can be closed. Depending on the frequency distribution of words from each taxonomy, this can also reduce the required processing compared to the naive approach. However, this is very dependent on the dataset and might not be triggered at all in some cases, e. g., when a word from an infrequent taxonomy appears at the end of the hit list. We also tried to introduce a threshold value to limit the search space, in which tuples can be found. However, this technique actually decreased the performance of our algorithm, as words from infrequent taxonomies were not physically located within the area between the current index and the threshold value. This led to a lot of missing values in our lookup dictionary, which meant that the algorithm started to behave similar to the naive approach. We therefore decided to remove the search threshold and only keep the optimization approaches described before.

In the following, we will introduce performance metrics to illustrate the runtime behavior of our algorithm depending on the size of the hit list in Figure 6 as well as the amount of taxonomies under consideration in Figure 7. To run the benchmarks, we used a machine with a single 2,66 GHz Quad-Core Intel Xeon and 24 GB of main memory. The implementation is done in Java based on the 1.7.0u25 Oracle JDK in "-server" mode. We separate the benchmark in a warm-up and run phase of each 10.000 runs to reduce the impact of startup and JIT compilation effects. The Figures 6, 7 display the average runtime in msec. The first benchmark, is about the size of the hit list. A hit list is a data structure very similar to the example given in Figure 5. It is an ascending list of hits, which store the position of the word and the taxonomy it belongs to.

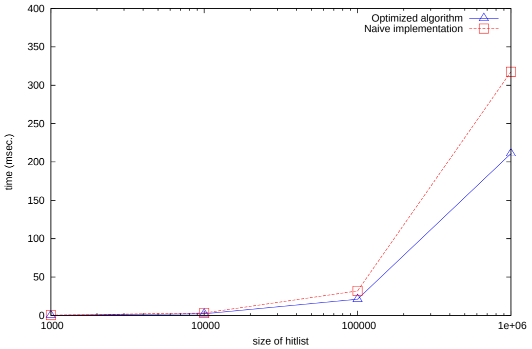


Figure 6. Runtime behavior depending on hit-list size

In the first benchmark, we used three taxonomies and manually generated a synthetic hit list. The frequency of words from the three different taxonomies is distributed by 70%, 25% and 5%. This is based on our observation from previous work [1], [2] in which we learned that hits are not distributed evenly among taxonomies, but rather follow Zipf's law. Based on this test setup, we generated hit lists of increasing sizes and calculated the ranking value using the naive and optimized algorithms. It can be concluded, that as the hit list grows in size, our optimizations have a bigger effect.

The following benchmark compares the algorithms based on various amounts of taxonomies, which are required to find a closed tuple. The frequencies are distributed in a similar fashion as the previous benchmark, which means that one taxonomy dominates the hit list, while others are rare in order to make the synthetic hit lists comply with our experiences from previous work. While the amount of taxonomies vary, each hit list has a size of 10.000 items.

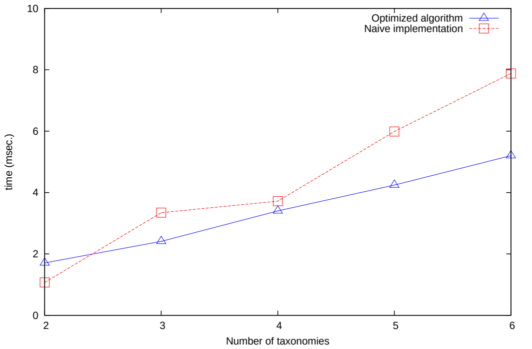


Figure 7. Runtime behavior depending on considered taxonomies

In the case of a small amount of taxonomies, the cost of maintaining the dictionary outweighs its benefits. However, once more taxonomies are considered, the optimizations become more effective. Although the gains do not meet our expectations, overall it is clear that our optimizations improve the required processing time in both test scenarios.

V. Conclusion and Outlook

We presented an algorithm for the ranking of documents based on taxonomy based queries. The algorithm calculates a similarity measure between the used taxonomies and a document which than can be used to rank the searched documents according to the used taxonomies. This measure is based on the occurrences of tuples containing words or phrases from all the different taxonomies and also the distance of the words found in the text.

Actually, we consider all the words with the same relevance. A more elaborate similarity function can give every word a different weight, so for example based on the term frequency and the inverse document frequency (weight: $w_{term,doc} = tf_{term,doc} * idf_{term}$) [6].

In our current implementation, the weight of every tuple is defined by the inverse of the distance of the words found in a tuple. Some (but not all) search results suggest that this measure probably favors tuples with words occurring consecutive inside a document too much. Optionally a measure which decreases the weight only logarithmically based on the distance could be more appropriate (i.e., $1/log(pos_{max} - pos_{min})$). But, to clarify this point, we need more input from our domain experts, when evaluating our ranked results. Another point for the future is to integrate our taxonomic based search inside the Lucene code base.

References

[1] A. Schmidt, D. Kimmig, and M. Dickerhof, "Search and graphical visualization of concepts in document collections using taxonomies," 46th Hawaii International Conference on System Sciences, 2013, pp. 1429–1434.

[2] A. Schmidt, D. Kimmig, and R. Senger, Poster: "Extraction and visualisation of semantic concepts from document-sets using taxonomies," First International Conference on Data Analytics (DATA ANALYTICS), 2012.

[3] E. Hatcher and O. Gospodnetic, Lucene in Action (In Action series). Greenwich, CT, USA: Manning Publications Co., 2004.

[4] A. Aksyonoff, Introduction to Search with Sphinx: From installation to relevance tuning. O'Reilly Media, 2011.

[5] R. Cummins and C. O'Riordan, "Learning in a pairwise term-term proximity framework for information retrieval," in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, 2009, pp. 251–258.

[6] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, 1972, pp. 11–21.

[7] T. Tao and C. Zhai, "An exploration of proximity measures in information retrieval," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 295–302.

Truss Decomposition for Extracting Communities in Bipartite Graph

Yanting Li
Graduate School of Computer Science
and Systems Engineering
Kyushu Institute of Technology
Iizuka, Japan
Email: k791502g@ai.kyutech.ac.jp

Tetsuji Kuboyama
Computer Centre
Gakushuin University
Tokyo, Japan
Email: ori-immm2013@tk.cc.gakushuin.ac.jp

Hiroshi Sakamoto
Graduate School of Computer Science
and Systems Engineering
Kyushu Institute of Technology
Iizuka, Japan
Email:hiroshi@ai.kyutech.ac.jp

***Abstract*—We propose a novel method for extracting communities, i.e., dense subgraphs, embedded into a bipartite graph. Our method is based on a technique for graph decomposition. Decomposing a large graph into cohesive subgraphs plays an important role in identifying community structures in social network analysis. Among a lot of definitions of cohesive subgraphs, the *k*-truss formed by triangles is one of the simplest cohesive subgraphs with a good trade-off between computational efficiency and clique approximation. This decomposition is, however, not applicable to bipartite graphs because bipartite graphs contain no triangles. In this paper, a *quasi*-truss decomposition algorithm for bipartite graphs is proposed based on the truss decomposition algorithm for general graphs. The proposed method can be used for analyzing the international business, such as the relationship between clients and sales volume in a certain period, and also analyze the social networking, such as users-topics relations in the twitter community.**

***Keywords*—bipartite graph, triangle, truss decomposition, dense subgraph, community discovery.**

I. Introduction

Communities are interpreted as dense subgraphs in a given graph G. The problem of identifying communities has attracted much attention recently due to the increased interest in studying various graphs with complicated structures. It helps to analyze graph structures, and mining useful information from graphs. Various techniques of data mining have been proposed for approaching graph analysis problems from different aspects. Therefore, we focus on this framework of community discovery, and apply it to an attractive domain of data, such as social networks.

In this research, we consider the problem of extracting communities in a bipartite graph using the notion of *truss*, which is a dense structure in a graph. Originally, the *truss* is defined as a dense subgraph composed of triangles, i.e., cliques with three nodes, in a graph [1], and the *truss decomposition* algorithm for extracting hierarchical dense subgraphs based on truss structures is proposed [2].

On the other hand, a bipartite graph is a common structure for modeling relations between two classes of objects, and is found in many real-world data sets such as user-item relations in an online shop. The truss decomposition is not applicable to bipartite graphs since any bipartite graphs include no triangles. To expand the notion of *truss* to the class of bipartite graphs, we introduce a new notion called *quasi-truss*. We also develop an efficient algorithm for bipartite graph decomposition, and examine the scalability of it with real-world bipartite data.

Organization: Section II introduces some related works about community extraction and bipartite graph analysis. Section III introduces basic notions used in this paper. In Section IV, we propose the *quasi*-truss decomposition algorithm. The experiments verify the efficiency of this algorithm for graph analysis in Section V. Finally, Section VI concludes the paper.

II. Related Work

An interesting substructure in a graph is called *community*, which is a subgraph densely connected by edges among nodes. According to the definition by Flake et al. [4], a community is a set of nodes in which each member has at least as many edges connecting to members as it does to non-members. This definition is unambiguous, and for any set of nodes, we can determine whether it is a community or not.

In [5], [6], a community of a graph $G = (V, E)$ is defined as a subgraph containing at least one *clique*, i.e., a subset $V' \subseteq V$ such that the subgraph in G induced by V' is a complete graph. Generally, the clique is extracted as a set of the nodes with high degrees. For this reason, the nodes with relatively lower degrees are liable to be ignored, and are not so much effective for uniformly sparse graphs. Moreover, the problem of finding maximal cliques is computationally hard. Thus, in the last decade, several efficient algorithms to find *quasi*-cliques, instead of exact cliques, have been proposed.

The *quasi*-clique is a relaxation notion of clique, for example, on the density [7] or the degree [8], [9]. However, the problem of finding these *quasi*-cliques remains NP-hard. Moreover, it may be difficult to capture the entire structure of communities in a graph since these subgraphs may substantially overlap, or be completely be separated.

To address these difficulties, a definition of dense subgraph called k-core has been proposed. It is defined as a maximal connected subgraph among all of its nodes with higher degree than k in G. Besides, the truss decomposition algorithm has been proposed: given a graph G, the k-truss of G is the largest subgraph of G in which any edge is contained in at least (k - 2) triangles within the subgraph [10]. The problem of truss decomposition is to find all k-trusses where $k \geq 3$.

While the problem of finding the densest subgraph is NP-hard, there is an efficient polynomial algorithm for the k-truss detection. From the point of view of the clique approximation, the k-truss is better than k-*core* [11], [12], which is a well-known subgraph for community discovery. For the problem of finding all k-trusses in a graph, i.e., truss decomposition problem, an efficient in-memory algorithm [1] and two I/O-efficient algorithms [2] have been presented to handle massive networks, and the efficiency of truss decomposition has been proved.

Many interesting relations are represented by bipartite graphs such as user-item relations in an online shop. Recently, we have proposed an algorithm for enumerating triangles in a bipartite graph [3]. In this paper, we improve it, and propose a new *quasi*-truss decomposition algorithm. Our algorithm is based on the following fundamental algorithms for bipartite graphs.

One is for testing bipartiteness to examine whether a graph is a bipartite or not [13]. The main idea of testing bipartiteness algorithm is to assign every node with a certain color in order to distinguish the color of its parent in a preorder traverse. This provides a two-colored spanning tree which consists of the edges connecting nodes to their parents. However, some nodes may be not colored properly. In the case of depth-first search, one of the two endpoints of every non-tree edge is another endpoint's ancestor. These pairs of nodes have different colors when non-tree edges are found. An odd-cycle can be formed by the path from ancestor to descendant within the incorrect colored edges together. With such an evidence, the graph is not bipartite. Every edge should be colored properly if the algorithm is terminated without detecting any odd-cycle of this type. It returns a bipartite graph with color.

Another one is the matching algorithm on bipartite graph. Matching in a graph $G = (V, E)$ is a subset of E such that no two edges share a common node. A node is matched if it is an endpoint of one of the edges in the matching. Matching problem is easier to solve by using bipartite graph than non-bipartite graph in many cases, such as the popular Hopcroft-Karp algorithm [14] for maximum cardinality matching which working correctly only with bipartite graphs.

III. Basic Notion

A triangle is one of the fundamental structures of graph that represents the smallest non-trivial clique. Indeed, the triangle plays an important role in graph analysis, especially in the computation of clustering coefficient, the triangular connectivity and transitivity ratio in massive networks. Three nodes in a triangle are fully connected by three edges formed by nodes $\{v_1, v_2, v_3\}$ that either directed edge or undirected edge, denoted as follows:

$$T_{123} = \{(v_1, v_2), (v_2, v_3), (v_1, v_3)\}$$

The notion of truss is defined by such triangles embedded in a graph. For the threshold k, the k-truss is a type of cohesive subgraphs that represents the largest subgraph of G such that every edge is contained in at least $(k-2)$ triangles within the subgraph. This value is called the support of an edge $e = (u, v) \in E_G$, denoted by $sup(e)$. The support of an edge e in G is the number of triangles in G that contain e. Thus, the k-truss of G where $k \geq 2$, denoted as T_k so that $\forall e \in E_{T_k}$, $sup(e, T_k) \geq (k-2)$. The task of truss decomposition in G is to find all trusses in G where $2 \leq k \leq k_{max}$. The k_{max} denotes the maximum truss number of any edge in G. The truss number of an edge e in G is defined as $max\{k : e \in E_{T_k}\}$, denoted by $\phi(e)$. From the definition of truss number, another definition k-class that denoted by Φ_k, defined as $\{e : e \in E_G, \phi(e) = k\}$. Relatively, the k-truss can be obtained from the set of edges $E_{T_k} = \cup_{i \geq k} \Phi_i$.

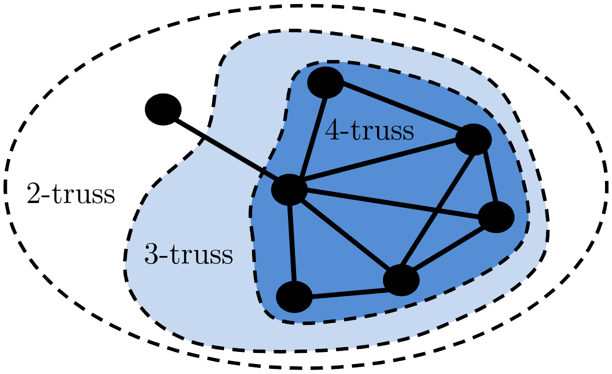


Fig. 1. Illustration of the 2-, 3-, and 4-truss decomposition

Fig. 1 illustrates the k-truss decomposition of a given graph G. The edges are contained in different number of triangles in G. The 2-class Φ_2 is the set of edges e with $sup(e) = 0$. The 3-class Φ_3 is the set of edges with $sup(e) = 1$, i.e., for $e = (x, y)$, there exist at least one node z such that $(x, z), (y, z) \in E_G$. The 4-class is analogous.

From the k-classes, k-trusses of G can be obtained as follows. The 2-truss T_2 is simply G itself. The 3-truss T_3 is the subgraph formed by the edge set $\Phi_3 \cup \Phi_4 \cup \Phi_5$, etc. It can be verified that each edge of T_k is contained in at least $k-2$ triangles for $2 \leq k \leq 5$. The k-trusses represent the hierarchical structures of G at different level of granularity.

On the other hand, there are many relations represented by bipartite graphs, which are equivalent to transaction data. However, as shown in Fig. 2, bipartite graph contains no triangle due to the definition: the node set is divided into two disjoint subsets V_1, V_2 such that no edge (u, v) $(u \in V_1, v \in V_2)$ is defined. Thus, we propose an extended version of the truss decomposition for bipartite graphs in the following section. Now, we prepare some important notions in our algorithm.

Given a bipartite graph $G = (V_1 \cup V_2, E)$, the algorithm transforms G to $G' = (V_1 \cup V_2, E \cup E')$ such that $E' = \{(u, v) \mid u, v \in V_1, u \neq v$, and $x \in V_2$ is adjacent to $u, v\}$.

We call $e' \in E'$ the *special edge*. For two distinct adjacent edges $e_1, e_2 \in E$ in G, there exists a triangle with a special edge $e' \in E'$ in G'. With more triangles sharing a unique special edge e' in G', a dense subgraph in G is expected to be identified. We introduce a novel notion of dense subgraph in bipartite graphs, the *quasi*-truss. The *quasi*-truss of G' is defined as the largest subgraph in G' containing exactly one special edge e'. Consequently, we obtain the substructure of G by removing all $e' \in E'$ from the *quasi*-truss.

In the following section, we design an algorithm to extract such components from a large network data.

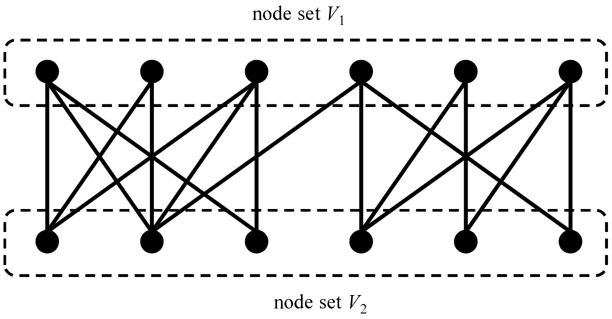


Fig. 2. The structure of a bipartite graph

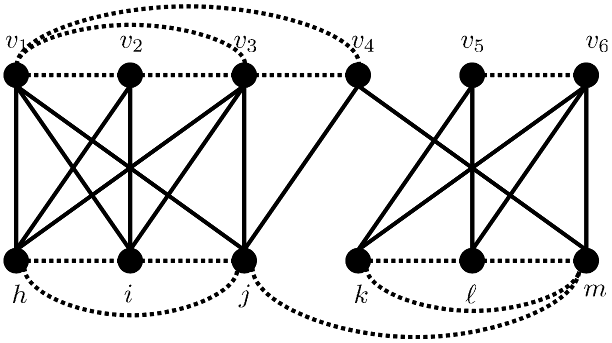


Fig. 3. Generate edges in a bipartite graph

IV. Quasi-Truss Decomposition

A. Quasi-truss

Conceptually, the definition of *quasi*-truss is similar with k-truss. In a given bipartite graph G, clearly, G contains no triangle. Then we define special edges $e' \in E'$ among nodes included in V_1 or V_2 exclusively. Initially, every node in both two node sets of the bipartite graph G will be visited. Determine whether any two adjacent nodes in the same node set sharing one common neighbor node in another node set or not. Then, the connectivity occurs between these two nodes, and connected by a special edge, denoted as e' if these two nodes have one common neighbor node in another node set of G. More formally, a special edge $e' = (x,y)$ is defined for $x,y \in V_1$ if there exists $z \in V_2$ such that $(x,z),(y,z) \in E$. After generating all special edges e', the structure of original bipartite graph is transformed to $G' = (V_1 \cup V_2, E \cup E')$.

In this definition, an edge $e' = (v_i, v_k)$ is generated in G so that $v_i, v_k \in V_1$ have a common neighbor node $v_j \in V_2$. The edge e' is denoted as e'_{ik} where $i,k \in V_1$ and $e'_{ik} \notin E$. The special edge e' is essential to form a triangle in bipartite graph G. The number of common neighbor nodes of edge e' is simply equal to the number of triangles which contain the edge e'. The common neighbor nodes of e' belong to the node set which does not contain two endpoints of e'. All of the triangles T belong to the bipartite graph G.

According to the definition, the Q-truss is the trusses of reconstructed G' where $Q > 0$. Here, the Q indicates the hierarchy of *quasi*-truss in order to distinguish from k-truss. Thus, the maximum *quasi*-truss of G can be defined as the special edge e' contained in maximal number of triangles in G.

When $Q = 1$, the *quasi*-truss is simply G itself since one edge e' is contained in one triangle exactly. We suppose that e'_{ik} has only one neighbor node v_i in another node set. Then, three nodes form a triangle $T_{ijk} = \{(v_i,v_j),(v_j,v_k),(v_i,v_k)\}$ which is a 1-truss subgraph for e'_{ik} contained in one triangle exactly. This differs from the definition of k-truss decomposition algorithm.

Fig. 3 illustrates the generation of edge e' in a given bipartite graph G. There are two types of edges: the original existed edges $e \in E$ which is in solid line, and the generated edge $e' \in E'$ which are illustrated by the dotted line. For instance, two nodes v_2 and v_3 that are in the same node set have a common neighbor node i, then, v_2 and v_3 will be connected by one generated edge e' so that a triangle $T_{23i} = \{(2,3),(3,i),(2,i)\}$ is formed. At the same time, the edge e'_{23} also contained in another triangle in G, the triangle $T_{23h} = \{(1,2),(2,h),(1,h)\}$ for node v_h also their common neighbor node. Thus, the four nodes v_2, v_3, v_h and v_i can be considered as 2-trusses for both two generated edges e'_{23} and e'_{hi} are contained in two triangles. In another situation, the nodes v_1 and v_2 have only one common neighbor node h that the generated edge e'_{12} is contained in only one triangle. The subgraph that contains three nodes v_1, v_2 and v_h can be considered as 1-truss. The nodes in the same node set such as the nodes v_4 and v_5 do not connected by any edge e' as they do not have any common neighbor node in another node set.

B. Decomposition algorithm

Quasi-truss decomposition algorithm is summarized in algorithm 1.

We employ the hash table to store and sort the special edge e' in this improved *quasi-truss decomposition algorithm*. Initialize the hash table of E', denoted as $hash[E']$, and the triangle set, denoted as T. The graph traverse begins from node v. A special edge e'_{jk} is generated to connect any two nodes v_j and v_k directly connect to v. Then, $T = \{v, v_j, v_k\}$ can be formed in G after this process. An e' is contained in at least one triangle where $Q = 1$. All of $e' \in E'$ is stored in the $hash[E']$, and sorted hierarchically. A common neighbor node of an e' represents a vertex of a triangle. For any two nodes in the same node set connected by an e', the number of their common neighbor nodes is equivalent to the number of triangles contains the e'. The $hash[E']$ only stores each unique edge e' instead of storing all triangles in an array [3]. The memory usage can be significantly reduced. Moreover, pointer is adopted to point to the common neighbor nodes of each e'. To extract the maximum *quasi*-truss represents the largest community, it was essential that counted the total number of common neighbor nodes of each e' in $hash[E']$, and output the e' with maximal number of common neighbor nodes. Next, the e' in $hash[E']$ is removed iteratively based on the number of its common neighbor nodes. For example, an e' will be removed from $hash[E']$ if the e' has less than five common neighbor nodes, or in other words, an e' is contained in less than five triangles where $Q = 5$. Finally, enumerate all triangles which satisfy the parameter. These enumerated triangles represent the dense subgraphs of G in different hierarchy.

Algorithm 1 *Quasi-Truss Decomposition Algorithm*

```
• $q$ := queue for graph traverse
• $Q$ := input threshold of hierarchy of trusses
• $E'$ := the edge set contains all special edge $e'$
• $hash[E']$ := the hash table of $E'$
• $T$ := a set of triangles in $G$
• $num_{(v)}$ := number of nodes belong to an $e'$
Require: $G = (V_1 \cup V_2,\ E)$, $Q = 1,2,3,4....m$
Ensure: $T$ within $Q$ hierarchy
init $hash[E'] = \phi$, $T = \phi$;
for all v $\in (V_1 \cup V_2)$ do
    $v$.mark = 0
    q.enqueue($v_0$);
    while not q.empty() do
        $v$ = q.dequeue()
        $v$.mark = 1;
        if $v \in e(v, v_j) \cap e(v, v_k)$ then
            $e' = (v_j,\ v_k)$ generated;
            $hash[E'] = hash[E'] \cup hash[e'_{jk}]$
        end if
        $T = T \cup \{v, v_j, v_k\}$
    end while
    for $Q$ = 1 to $m$ do
        for all $e' \in hash[E']$ do
            if $num_{(v)} \in e' < Q$ then
                remove $e'$ from $hash[E']$
            end if
        end for
    end for
end for
output $T$ contains $e'$ within $Q$ hierarchy
```

V. Experiment and Evaluation

We observed the performance of the proposed method via a succession of experiments in this section. The experimental results evaluated the effectiveness of the *quasi*-truss algorithm. All of the experiments were done on a machine with the Inter i7 2.3GHz CPU, 8GB RAM, and the version 4.1.2 of C compiler in Mac OS 10.8.3.

A. Data characteristics

Five real-world graph datasets with different sizes were used in these experiments. Table I indicates the features of the datasets. $|V_1|$ and $|V_2|$ indicated the number of nodes of each node-set in these given bipartite graphs. $|E|$ showed the number of edges in each dataset.

TABLE I. Features of datasets

File name	$\|V_1\|$	$\|V_2\|$	$\|E\|$	size(kb)
cmuDiff	3,000	5,932	263,325	32.1
cmuSame	3,000	7,666	185,680	46.6
cmuSim	3,000	10,083	288,989	260
HetRec	9,372	6,257	26,232	259
MovieLens	3,706	6,040	1,000,209	40.1

The three datasets *cmuDiff*, *cmuSame* and *cmuSim* were chosen from 20 newsgroups datasets, which were also refered in [16]. They were collections of newsgroup documents. Each of them corresponded to a certain topic, and recorded the relationship between keywords and news documents. Both *HetRec* and *MovieLens* were two datasets released from the framework of Information Heterogeneity and Fusion in Recommender Systems. The *HetRec* recorded the relationship between online users and artists/musics from *Last.fm online music system* in 2011. The *MovieLens* dataset contained anonymous ratings by MovieLens users toward a certain number of movies in 2000.

Matrix blocking proposed in [16] was a community detection technique based on the connectivity occurence among all nodes in G. Oppositely, the proposed algorithm in this paper was designed to decompose a bipartite graph, and identify the subgraphs within different hierarchy. Therefore, in these experiments, we mainly observed three aspects for evaluating the proposed algorithm. First, we stated the total number of triangles which included all special edges e'. Second, we observed the time cost for enumerating all triangles in each bipartite graph. Finally, the largest community structure represented by the maximum *quasi*-truss was extracted from each bipartite graph.

B. Experimental results

Table II shows the experimental results by using the five datasets. The #T indicates the total number of triangles formed in each given bipartite graph. The next column shows the time cost for triangles' forming. The results of Q_{max} clearly indicates the maximal *quasi*-truss in each bipartite graph. The maximal *quasi*-truss represents the largest communities in each given bipartite graph.

TABLE II. Statistics of experimental results

File name	#T	size(kb)	time(in sec.)	Q_{max}
cmuDiff	61,638	874	0.374	238
cmuSame	174,363	2,458	0.78	390
cmuSim	1,838,827	27,471	7.207	112
HetRac	945,043	15,020	6.739	351
MovieLens	63,271	891	0.453	223

The running time increased linearly with the increasing number of triangles. Meanwhile, the number of edges $e' \in E'$ also increased. But it was worth to notice that the number of edges $e' \in E'$ did not equal to the total number of triangles for an edge e' was contained in more than one triangles.

In the third column of Table II, we observed the maximal *quasi*-truss for each given dataset. According to the definition of *quasi*-truss, the subgraph with Q_{max} represented the largest community in which a unique edge *e'* was contained in maximal number of triangles. Thus, the subgraph with Q_{max} was the densest subgraph, since it represented the core of a given graph. In this experiment, "Q_{max}"-truss were extracted from the given G by adopting a technique which was similar to the *Top-Down approach of truss decomposition* concluded in [2]. Thought observing the experiment results, the maximum *quasi*-truss of G also increased with the total number of triangles. Meanwhile, the value of "Q_{max}" was difficult to estimate. However, the result of dataset *cmuSim* was an exception although this dataset contained the maximal number of triangles compared with results of other datasets. The value of Q_{max} of *cmuSim* dataset was the smallest. This result illustrated that the connectivity among all nodes in both V_1 and V_2 of G had a significant impact on the density of subgraphs.

Another reasonable evaluation strategy was to observe the density of subgraphs. Bipartite graph had a special structure differs from ordinary graphic structures. Therefore, it was necessary to observe the extracted dense subgraphs separately. Table III concluded the features of each maximum *quasi*-truss.

TABLE III. Features of maximum quasi-truss

File name	node	edge	size(kb)	$\#T$
cmuDiff	240	477	2.95	238
cmuSame	392	781	5.27	390
cmuSim	114	225	1.54	112
HetRac	353	703	3.88	351
MovieLens	225	447	2.58	223

The node and edge indicated the number of nodes and the number of edges in each dense subgraph respectively. The size was the total amount of subgraphs defined as $(|V_1|\cup|V_2|, |e|\cup|e'|)$. The $\#T$ indicated the number of triangles anchored in each dense subgraphs. In the case of bipartite graph, a subgraph had the density one if and only if it was a biclique according to the concluded definition in [16]. The definition of density for bipartite graph in [16] cannot be adopted directly to estimate the density of *quasi*-trusses in a bipartite graph for it was based on triangular structure. Thus, we simply addressed the amount ratio that compaired the size of dense subgraphs with their matrix graphs containing all triangles. Moreover, we also compaired the number of triangles of the dense subgraps with their matrix graphs containing all triangles in order to observe the ratio of number of triangles. Then, analyzed the relationship between the amount ratio and the ratio of the number of triangles based on the statistic results shown as Fig. 4.

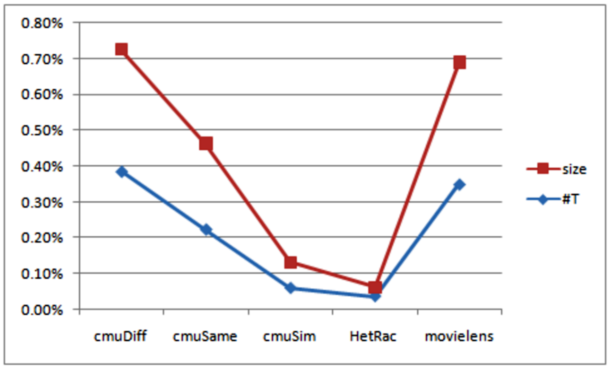


Fig. 4. Relationhsip between size and the number of triangles

Each point in the red graph illustrated the percentage of the amount of the largest subgraphs in each bipartite graph. Each point in the blue graph illustrated the percentage of the number of triangles of the largest subgraphs in each bipartite graph. From Fig. 4, the amount of the dense subgraphs increased linearly with the number of triangles anchored in the dense subgraphs. These statistical results also proved the previous experimental results in Table II.

VI. Conclusion

We implemented the algorithm for *quasi*-truss, which was a novel notion of dense subgraph in a bipartite graph introduced in [3]. This notion was an expanded version of k-truss decomposition [2]. An effective algorithm was also introduced for *quasi*-truss decomposition in a bipartite graph. We verified the scalability of our algorithm by experiments on real-world datasets. The results showed a significant effectiveness on decomposing a bipartite graph based on triangle structure.

We plan to research the theoritical proof for the density evaluation of bipartite graph by adopting the *quasi*-truss decomposition algorithm, and time complexity for dense subgraph extraction as the future perspective. Furthermore, as one of triangle's properties, the research of clustering coefficient of a bipartite graph is available to analyze the connectivity situation.

References

[1] J. Cohen, Truss: cohesive subgraphs for social network analysis, 2008.

[2] J. Wang and J. Cheng, Truss decomposition in massive networks, VLDB2012, 2012, pp. 812-823.

[3] Y. T. Li, Kuboyama, and H. Sakamoto, Mining twitter data: discover quasi-truss from bipartite graph, 5th International Conference on Intelligent Decision Technologies, to appear

[4] G. W. Flake, S. Lawrence, and C. L. Giles, Efficient identification of web communities, KDD2000, 2000, pp. 150-160.

[5] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, SODA1998, 1998, pp. 668-677.

[6] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, Extracting large-scale knowledge bases from the web, VLDB1999, 1999, pp. 639-650.

[7] J. Abello, M. G. C Resende, and S. Sudarsky, Massive quasi-clique detection, LATIN2002, 2002, pp. 598-612.

[8] H. Matsuda, T. Ishihara, and A. Hashimoto, Classifying molecular sequences using linkage graph with their pairwise similarities, Theor. Compt. Sci., 1999, 210(2)305-325.

[9] J. Pei, D. Jiang, and A. Zhang, On mining cross-graph quasi-cliques, 2005, SIGKDD.

[10] J. Cohen, Graph twiddling in a mapreduce world, Computing in Science and Engineering, 2009, 11(4)29-41.

[11] S. B. Seidman, Network structure and minimum degree, Social Networks, 1983, 5(3)269-287.

[12] V. Batagelj, M. Zaversnik, An $O(n)$ algorithm for cores decomposition of networks, advances in data analysis and classification, 2011, Vol. 5, No. 2, pp. 129-145

[13] N. Alon and M. Krivelevich, Testing k-colorability, SIAM J. Discrete Math., 2002, 15(2)211-227.

[14] J. E. Hopcroft and R. M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput., 1973, 2(4)225-231.

[15] J. Cheng, Y. Ke, A. W. C. Fu, J. X. Yu, and L. Zhu, Finding maximal cliques in massive networks, ACM Transactions on Database Systems, 2011, 36(4) Article No. 21.

[16] J. Chen and Y. Saad, Dense subgraph extraction with application to community detection, Knowledge and Data Engineering, IEEE Transaction, 2012, Volume:24, Issue:7.

[17] H. Y. Zha, X. F. He, C. Ding, M. Gu, and H. Simon, Bipartite graph partitioning and data clustering, 2001, Proceedings of ACM CIKM 2001.

[18] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment, 2005, Vol. 2005, p. P09008.

Analysis of Medical Publications with Latent Semantic Analysis Method

José Román Herrera-Morales, Liliana Ibeth Barbosa-Santillán
Information System Department
University of Guadalajara, México
Email: {rherrera, ibarbosa}@cucea.udg.mx

***Abstract*—This article presents a review of the Latent Semantic Analysis (LSA) method used to extract knowledge from large sets of text documents, describing its origins, main applications, basic operation and dimensionality optimization. To evaluate its performance and usefulness in identifying semantic relatedness a series of experiments were conducted with various collections of texts, varying number of files that were part of each corpus and using different indexes. It was shown that LSA can serve as a mechanism for grouping and classifying documents that are related to the themes, in particular in obedience, to the search expressions according to their semantic relevance. It was also evident, however, that the computational performance of LSA will deteriorate as more files are added to generate indexes, since index and search response times increased significantly.**

***Keywords*— Latent Semantic Analysis; Semantic Relatedness; Semantic Relevance; Text Processing.**

I. Introduction

This article is an analysis of Latent Semantic Analysis (LSA) [1], one of the most frequently used methods in search engines, text comparators, and recommender systems, since it allows the extraction of meaning and non-obvious relationships of terms in large sets of text documents. The idea is to describe the utility of LSA when applied to collections of texts from the medical field, to find documents that are the most relevant or similar in terms of content (semantic relatedness) according to the search terms. LSA represents an alternative to the need for human experts to analyse and digest information and is very important to apply it to the area of Health Sciences, one of the areas in which a large amount of scientific content is generated every day.

The structure of this document is as follows: the next section is a review of the concepts of LSA and the relationship between the application of matrix decomposition techniques of linear algebra such as the Singular Values Decomposition (SVD). Section III describes the experimentation phase, outlining the collection of medical documents, the implementation of the method in a LSA prototype to perform several tests and the integration of test scenarios to carry out the semantic relatedness test. Section IV describes the major results obtained, from the time of indexing and responding to queries, to the relevance in similarities of the meaning in documents. Finally, conclusions and comments about the results obtained are included in Section V.

II. Latent Semantic Analysis

LSA is a computational model of human knowledge representation that approximates the ability to make judgments of semantic relationship, which is based on a very simple premise, namely, that the similarity in the meaning of two words can be induced by how they are used in texts [1]. By means of this principle, words and text are created in a specific domains. LSA examines the frequency in a set of texts and then uses semantic relatedness in order to build the matrix decomposition. In a nutshell, LSA is a knowledge representation model, which is based on the patterns of word usage in a range of documents. This set of documents is commonly called a corpus and the mapping between documents and terms is called Latent Semantic Space [2].

The following subsections address issues related to LSA that include: its origin, the basic operation of LSA and its relationship with SVD, the importance of optimizing the dimensionality of the matrices, and finally, the main areas of application of LSA.

A. Origin and first applications

LSA was released under patent #4,839,853 of the U.S. Patent Office issued on June 13, 1989 to Bell Labs researchers Deerwester, Dumais, Furnas, Harshman, Landauer, Lochbaum and Streeter, and was originally used as a mechanism to support tasks of Information Retrieval [3] [4]. It was mentioned as Latent Semantic Indexing (LSI) in order to use techniques of dimension reduction for improving the indexing process of textual content [5]. Subsequently, Landauer and Dumais, who were interested in human learning and how people learn new vocabulary from the texts that they read [6], proposed the LSA as a new theory regarding acquisition, induction and knowledge representation to reflect the similarity of words and passages of text, making use of the analysis of a large corpus of natural text. They observed that, by inducing global knowledge indirectly from a co-occurrence data locally on a large body of characteristic texts, the LSA can acquire knowledge of the entire English vocabulary in a manner comparable to the way a child learns. After reading texts, children can learn new words every day and after several readings can apparently understand the meaning of many other words that they did not know before. This feature of human language learning has been a topic of debate and research interest. In ancient times, it was known as Platon's problem, i.e., how people can know much more than they have been exposed to. Platon suggested that people had all this knowledge within themselves and only needed small patterns or guides to be able to produce it. LSA means analogous hidden meanings can be extracted when information processing of a collection of texts is performed.

B. Basic operation

The LSA method consists of a series of basic steps for extracting meaning from a collection of documents. First, it generates a term-document matrix, where each row represents the words in the whole collection of texts and the columns represent the documents. In this first step those words whose occurrence in the collection of documents is too frequent or too infrequent should be eliminated, as should words that do not add any value, so-called stop-words. Second, an algorithm to calculate the weights for each of the cell-matrix document terms is applied, this for emphasis of the words according to a certain domain. Finally, the SVD process is applied. As a result of this process, three partial orthogonal matrices are produced: the term-matrix (commonly called matrix U or Left Singular Values), the document-matrix (commonly called matrix V, or Right Singular Values) and the diagonal matrix S, whose main diagonal contains singular values and other positions zeros [7]. Fig. 1 shows the original matrix A (term-document matrix) and the resulting SVD matrices.

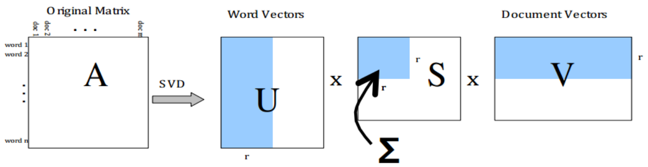


Fig. 1: SVD process applied to matrix A that produces three orthogonal partial matrices as a result (source: Kireyev & Landauer, 2011) [8].

C. Optimal dimension reduction

Reducing the number of dimensions by applying minimal SVD decomposition significantly reduces the noise and the amount of data, memory and processing time required to obtain results with LSA. This process is called optimization dimensionality [9] and involves finding the K-th dimension (the columns that represent collections of documents) for the best K-dimensional approximation of the original matrix. Thus, the document collection is represented by a K-dimensional vector space derived by SVD. In many cases, the value of K is much smaller than the number of terms that are present in the matrix of term-document, but for application related simulation language learning, it was found that the optimum value of K is in a range of 300 +/- 50 [6], [9], [10] and validated with a formal study applied similarity in meaning tests for text samples from the Groliers Academic American Encyclopaedia described by Landauer and Dumais [6]. Fig. 2 shows the original graph of this study where it can be seen that there are sufficient values close to 300 in the number of dimensions to be considered, since it is this range which gives the best similarity in meaning.

D. Areas of application

LSA can greatly improve the extraction and representation of knowledge in the domain of human learning to represent objects and contexts present but can also be applied in situations with a large volume of data, such as Data Mining. Wolfe and Goldman [1] found LSA useful in a processing and text analysis, such as quality assessment and summary trials, finding differences or similarities between texts verifying internal coherence, and in identifying the original source of students' work, among many others. All these applications have had very good results, and the reliability of LSA has been so good that it is comparable to human experts.

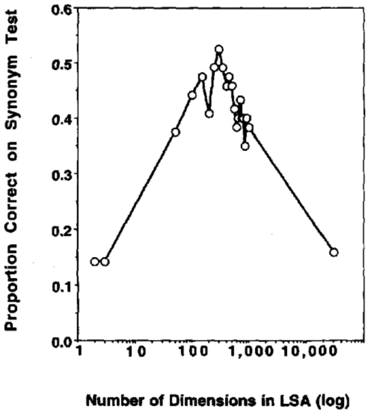


Fig. 2: The effect of K-dimensions retained in LSA-SVD simulations of meaning similarities. K-dimensions is in log scale (taken from Landauer & Dumais, 1997).

The following section describes in detail the experiments that have been conducted to evaluate the operation, performance and utility of the LSA method in identifying semantic concordance using like a corpus a collection of texts containing abstracts of articles in health sciences field. The medical field is one of the fields of research that is growing more rapidly and all new medical information is being published everyday [11]. Hence, the importance in working to generate mechanisms that allow this scientific community to have better access to this large amount of resources.

III. Experiments

This stage of experimentation, where the semantic relatedness tests were carried out, was divided into three main phases: (1) Getting the text collection, in order to obtain a raw material that represents items in the health sciences field, (2) implementation of the LSA method, a tool coded in C# language, and (3) the definition of test scenarios that included a series of searches in several corpora with different characteristics in terms of the number of files and the value of K, to optimize the LSA process.

A. Text collection

The first step before testing LSA was to generate multiple text files to serve as the corpus or data source. This information can be obtained with the OAI-PMH Service from PubMed Central (PMC-OAI) [12] that provides access to the metadata for all items in its collection. The Open Archives Initiative Protocol Of Metadata Harvesting (OAI-PMH) [13] is a standard protocol for the collection of metadata records designed to be shared openly and freely, and it is promoted by the Open Archive Initiative and it is based on the exchange of XML messages on a transport service such as HTTP.

Once an excellent source of information in the medical field has been identified and there is a reliable way to obtain it, a .Net TCP client application is used to make requests to the OAI-PMH server in PubMed Central in order to download the metadata records. Given the characteristics of PMC-OAI service, the resulting records were delivered in Dublin Core simplified format and provided more than 300,000 metadata records through the OAI-PMH harvester client. For each record retrieved, the OAI-PMH harvester client generated a text file in a local directory, and each file contained the following information fields: title, authors, abstract, date of publication, journal, publisher and the ID assigned by the PMC-OAI service.

These files were metadata items that were filtered and identified as research articles and were discharged in chronological order by publishing date, the most recent first, i.e., April 2013 up to July 2008. Although the PMC portal states that it had 2.7 million articles, it stopped downloading them because for experimentation conducted in this article was considered as the limit 1000 files due to the considerable time that indexing is required for this amount. This behavior is described in more detail in the results section.

B. Implementation of LSA method

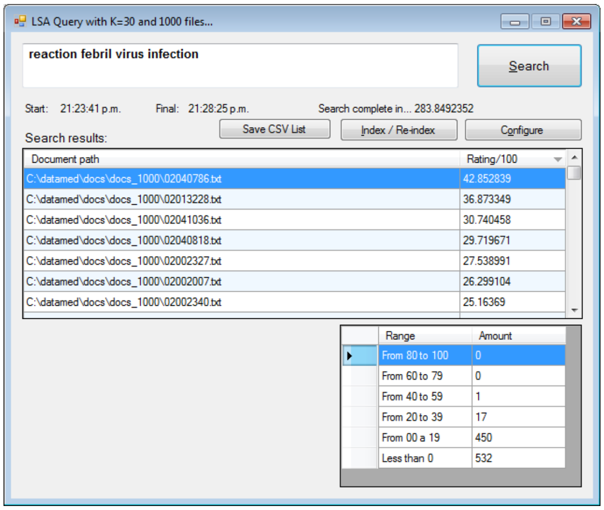
Various options for implementing SVD were reviewed, such as Bluebit - Online Matrix Calculator that allows online calculations of small matrices and other tools much more complete as the ”R”; which contains a specialized package for LSA. But, familiarity with .NET platform and the ability to adapt and customize the code, as well as to select a local folder with n number of files as a data source, were the main reasons for the implementation in C# by Anup Shinde [14] should be selected. This used the open-source libraries DotNetMatrix [15] for all tasks concerning the matrix algebra including SVD decomposition.

The main features of the prototype in C# are: set configuration options, maintenance of indexes and use of queries to verify the consistency of a search expression in the corpus. In the settings section, the user can define a local folder where it takes the collection of documents to the index, and set the value of K to be used for dimensionality reduction of the resulting matrices of SVD. For queries, the results are provided in two ways: first, as an individual list of documents ranked according to their percentage of semantic relatedness with the search expression, and the second, as a view grouped into ranges of percentage of relevance to know the quantity of documents that fall into each category. Additionally, options were enabled to store the full and reduced SVD matrices, as well as functionality of exporting to CSV format. In Fig. 3, a screenshot of the GUI of this prototype is shown.

The workflow of this implementation can be analysed by dividing it into two main groups: first, the LSI index generation for test collection of documents, and second, the search process in the document collection. This last part includes how to present the results in the GUI, so that they can be interpreted in a simpler way.

C. Definition of test scenarios

Before the experiment started, several indexes were generated with different numbers of files so that semantic matching tests could be performed under different test scenarios. Different indexing times were counted, considering first small numbers of files, starting with 10, 100, 150, 200,250, 300, 400, 500, 600, 700, 800, 900 and 1000 files. For each of these quantities two indexes were generated, one considering the value of K as 10% of the files and one with K= 50%. Files that were considered in each set were selected indexed over 300,000 files retrieved with OAI-PMH harvester and ordered in ascending order according to their name in the local folder. The largest index always includes in its entirety all of the previous index files.


LSA Query with K=30 and 1000 files...

reaction febril virus infection

Start: 21:23:41 p.m.   Final: 21:28:25 p.m.   Search complete in... 283.8492352

Search results:

Document path	Rating/100
C:\datamed\docs\docs_1000\02040786.txt	42.852839
C:\datamed\docs\docs_1000\02013228.txt	36.873349
C:\datamed\docs\docs_1000\02041036.txt	30.740458
C:\datamed\docs\docs_1000\02040818.txt	29.719671
C:\datamed\docs\docs_1000\02002327.txt	27.538991
C:\datamed\docs\docs_1000\02002007.txt	26.299104
C:\datamed\docs\docs_1000\02002340.txt	25.16369

Range	Amount
From 80 to 100	0
From 60 to 79	0
From 40 to 59	1
From 20 to 39	17
From 00 a 19	450
Less than 0	532

Fig. 3: The GUI of implementation of LSA method in C#

Several special cases were presented, when K= 50% represented more than 300 files (amount that exceeds the recommended optimal value). These cases generated new indexes with K constant values such as 250 and 300, when applied to the corpus translated into 700, 800, 900 and 1000 files. In this way the maximum dimensionality considered was 250 and 300.

To provide consistency in terms of the evaluation and comparison of the results that were obtained, a list of queries was generated and applied in the same way to each of the test events with the several LSI indexes. The list of queries, formed by sequences of non-sorted terms, is as follows:

Query 1: reaction febril
Query 2: reaction febril virus infection
Query 3: tissue epidermis skin carcinogen
Query 4: cancer tumor carcinoma
Query 5: cancer tumor carcinogenesis

IV. Results

The results are described in terms of three main groups, and are referring to: (a) the indexing times, (b) the average response times for queries, and (c) the semantic relatedness tests.

A. Indexing time

Fig. 4 shows that there is no significant difference in time indexing for indexes with fewer than 600 files, but more than 600 means an increase in time indexing when considering a K

ISBN: 978-1-61208-311-7

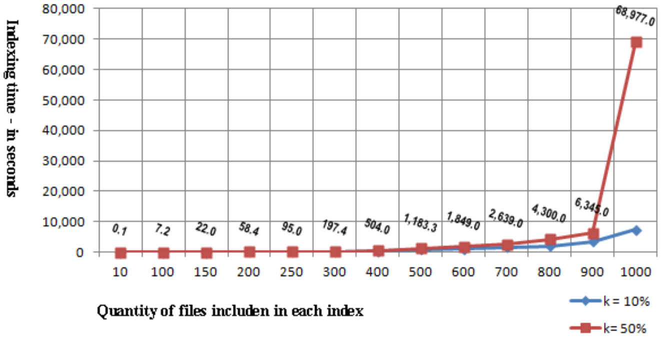


Fig. 4: Statistics for LSI time with K-values in different indexes

with a value of 50% of the number of files. A very important fact is that for 1000 files, considering K= 50% (500 columns for reduced SVD), indexing time increased very considerably; in fact, when there were 900 files, 6345 seconds (105 minutes) increased to 68977 seconds (1150 minutes; or 19 hours or almost a full day indexing); this means that from one index to another it grew in more than 10 times the necessary time to be able to index all the documents. By contrast, with K= 10% indexed time observed normal growth.

Additional indexes with 600, 700, 800, 900 and 1000 files were performed, considering the value of K as constant values, 250 and 300, values considered optimum [6], [9], [10]; this was done to reduce the dimensionality of the resulting matrices on SVD.

Fig. 5 includes indexing times; when using K= 300, as seen, for indexes many files with a greater double the recommended value of K and has a slight increase in the case of 1000 files also begin to increase but not as disproportionately as in the case where K= 50%. In the latter index, for 1000 files it took 7392 seconds for K= 10% (K= 100), 22605 seconds for K= 300 and the aforementioned 68977 seconds for K= 50% (K= 500).

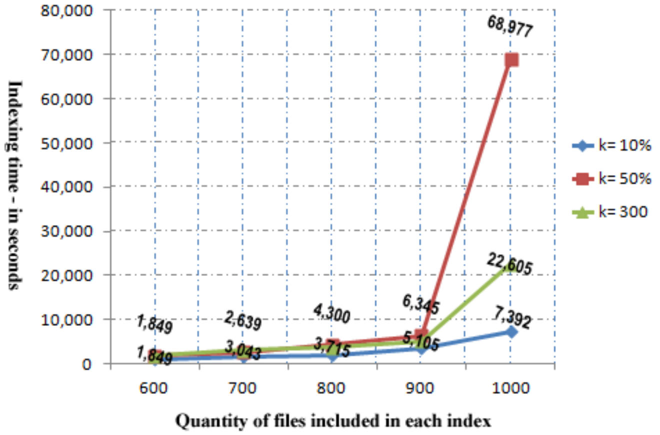


Fig. 5: Comparison of reduced indexing time with K= 300

Considering the times obtained in the previous 900 indexed files, the rate of increase over the previous indexes increased approximately 2x for K= 100, 4x for K= 300 and 10x for K= 500. When it became clear that, when the number of files to be indexed is close to or greater than 1000, the indexing time increases significantly, the decision was taken to set this value as the file limit for these LSA semantic relevance tests, so that all queries that are described in the next section treat 1000 as the maximum number of files for the larger index.

B. Average response times for queries

After the indexing process, the verification queries came where each repeated one defined and executed only 12 different indexes (Fig. 6), in which the number of files and the respective value of K varied.

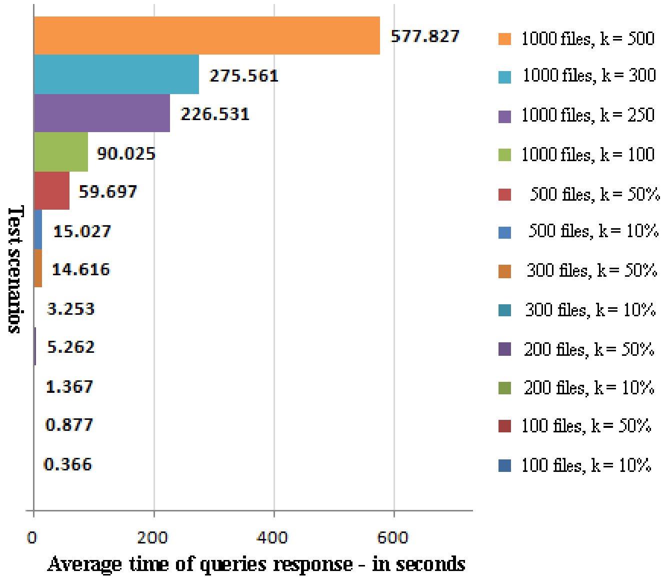


Fig. 6: Average response time for queries with different indexes varied.

Fig. 6 shows a graph with the respective average response times and the remarkable time it takes to answer a query in the case where K= 500 to 1000 files; it takes 577 seconds (nearly 10 minutes) to deliver the results on screen.

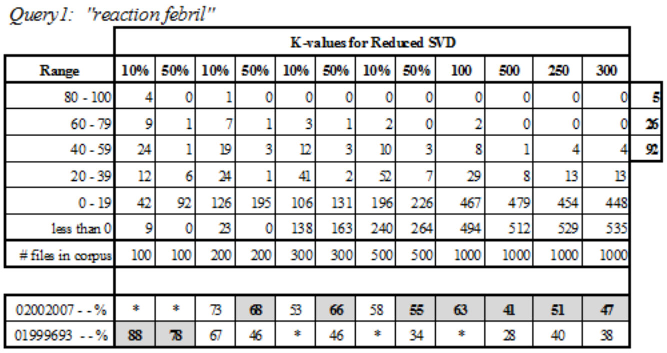
C. Semantic relevance tests

For each of the queries, response times were recorded and the GUI of the prototype in C# displayed the most relevant files according to the semantic coherence of its content. Furthermore, a clustering result was generated when files were included in six ranges relevant percentage according to the query made. The ranges were 80 to 100%, 60 to 79%, 40 to 59%, 20 to 39%, 0 to 19% and less than 0%. To find out how many files corresponded to each of the ranges of relevance, the percentage obtained was recorded at each event; this in order to have a quantitative way to measure the semantic relatedness of each query.

For a better analysis of the results the data are presented in tables. In these tables, the columns represent each of the events in which searching indexes have been used with different values of K; in the first columns, the values of K are expressed in % of files of the corpus, while in the last columns there are a certain number of files (100, 500, 250 and 300 files). On the other hand, the first rows represent the amount of files that falls in each range of percentages of relevance, and the last two rows show the percentages of two of the files more relevant for each query. When an ”*” appears in the cell it means that the examined file does not appear in the Top Ten. When the cell value is displayed in bold and shaded it means that it occupied the first place.

Table I shows then concentration of these results in relation to search expression ”febrile reaction”. It show that very few files, accumulated events, fell within the range of 40 to 59% of significance (92 files), while 26 files were in the range 60 to 79%, and only five in the range of 80 to 100%. View the last semi-right column. These figures indicate that it has a small number of files whose content is related to febrile reactions. One file in particular identified as ”02002007”, in more than half of the search events, reached first position in the ranking of relevant files. This is shown in the last row of Table I.

TABLE I: CONCENTRATED DATA RESULTING FROM QUERY 1



Query1: "reaction febril"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	4	0	1	0	0	0	0	0	0	0	0	0	5
60 - 79	9	1	7	1	3	1	2	0	2	0	0	0	26
40 - 59	24	1	19	3	12	3	10	3	8	1	4	4	92
20 - 39	12	6	24	1	41	2	52	7	29	8	13	13	
0 - 19	42	92	126	195	106	131	196	226	467	479	454	448	
less than 0	9	0	23	0	138	163	240	264	494	512	529	535	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
02002007 - - %	*	*	73	**68**	53	**66**	58	**55**	**63**	**41**	**51**	**47**	
01999693 - - %	**88**	**78**	67	46	*	46	*	34	*	28	40	38	

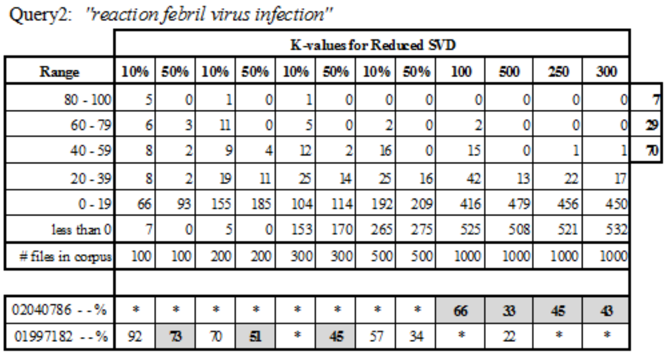
Query2 ”reaction febril virus infection” was very similar to query1, two more words being added for a more precise search in medical articles for febrile reactions, but in this case caused by viral infection. Table II shows the concentrated results.

The results of Table II evidence how a very reduced group of files was in the first three ranks (last semi-column to the right) which shows the specialization of their contents in accordance with the search expression. For this query2, the file ”02002007” did not achieve the top position in any of the tests, and instead the file ”01997182” reaches the first places only in the first events. A different file was the most similar in terms of content, it was the file ”02040786” which in the last four events (last columns that represent corpus with 1000 files) was located in the first position of relevance with 66%, 33%, 45% and 43%, respectively.

The effect of specialization has been evidenced in the query3 ”tissue epidermis skin carcinogen” when more precise terms were added to the search expression. Table III shows that the file ”02001792” always was ranked in the first place, also in the last columns with the corpus of 1000 files, shown stability in results, because the average relevance was 61% +/- 3% (with K=100, 250 and 300); and except in the case where K=1000 the relevance fell to 47%. Here is evidence that a greater amount of items in the index does not help to improve its effectiveness, but on the contrary this distorts the result.

Table V shows data very similar to the previous query results (Table IV). The difference between query4 and query5 was only the third word (”carcinoma” and ”carcinogenesis”) which a human expert would interpret them as equivalent. In this case, the results show that both queries yield nearly identical results to the four first events according to the corpus considering more files, (events that are further to the right, the results are more specialized and even grow in a few percentage points of semantic relatedness in case the file ”01997145” which rises from 41% to 50% with K=250, and 39% to 46% with K=300, in both cases with the index with 1000 files.)

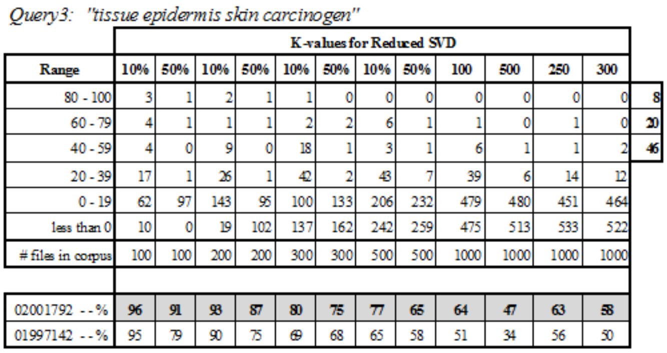
TABLE II: CONCENTRATED DATA RESULTING FROM QUERY 2



Query2: "reaction febril virus infection"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	5	0	1	0	1	0	0	0	0	0	0	0	7
60 - 79	6	3	11	0	5	0	2	0	2	0	0	0	29
40 - 59	8	2	9	4	12	2	16	0	15	0	1	1	70
20 - 39	8	2	19	11	25	14	25	16	42	13	22	17	
0 - 19	66	93	155	185	104	114	192	209	416	479	456	450	
less than 0	7	0	5	0	153	170	265	275	525	508	521	532	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
02040786 - - %	*	*	*	*	*	*	*	*	**66**	**33**	**45**	**43**	
01997182 - - %	92	**73**	70	**51**	*	**45**	57	34	*	22	*	*	

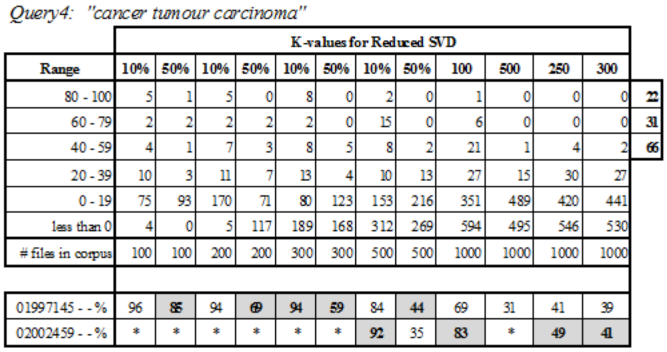
TABLE III: CONCENTRATED DATA RESULTING FROM QUERY 3



Query3: "tissue epidermis skin carcinogen"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	3	1	2	1	1	0	0	0	0	0	0	0	8
60 - 79	4	1	1	1	2	2	6	1	1	0	1	0	20
40 - 59	4	0	9	0	18	1	3	1	6	1	1	2	46
20 - 39	17	1	26	1	42	2	43	7	39	6	14	12	
0 - 19	62	97	143	95	100	133	206	232	479	480	451	464	
less than 0	10	0	19	102	137	162	242	259	475	513	533	522	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
02001792 - - %	**96**	**91**	**93**	**87**	**80**	**75**	**77**	**65**	**64**	**47**	**63**	**58**	
01997142 - - %	95	79	90	75	69	68	65	58	51	34	56	50	

TABLE IV: CONCENTRATED DATA RESULTING FROM QUERY 4

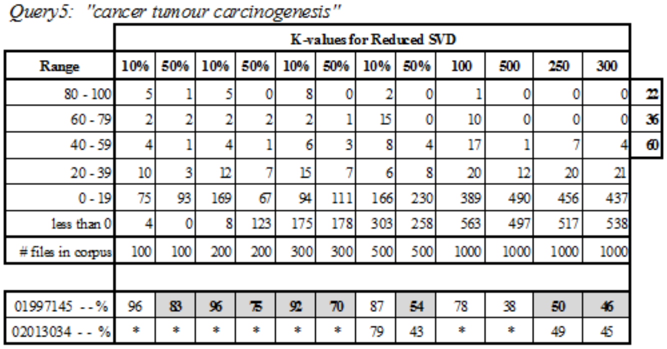


Query4: "cancer tumour carcinoma"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	5	1	5	0	8	0	2	0	1	0	0	0	22
60 - 79	2	2	2	2	2	0	15	0	6	0	0	0	31
40 - 59	4	1	7	3	8	5	8	2	21	1	4	2	66
20 - 39	10	3	11	7	13	4	10	13	27	15	30	27	
0 - 19	75	93	170	71	80	123	153	216	351	489	420	441	
less than 0	4	0	5	117	189	168	312	269	594	495	546	530	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
01997145 - - %	96	**85**	94	**69**	**94**	**59**	84	**44**	69	31	41	39	
02002459 - - %	*	*	*	*	*	*	**92**	35	**83**	*	**49**	**41**	

Another interesting situation that can be noted from Table V is that file ”02002459” practically does not appear in the first place of relevance; this is because the third word used in query5, ”carcinogenesis”, was a term even more technical in health sciences domain and therefore in its place was the file ”2013034” that in the last two events was second in importance

TABLE V: CONCENTRATED DATA RESULTING FROM QUERY 5



Query5: "cancer tumour carcinogenesis"

	K-values for Reduced SVD												
Range	10%	50%	10%	50%	10%	50%	10%	50%	100	500	250	300	
80 - 100	5	1	5	0	8	0	2	0	1	0	0	0	22
60 - 79	2	2	2	2	2	1	15	0	10	0	0	0	36
40 - 59	4	1	4	1	6	3	8	4	17	1	7	4	60
20 - 39	10	3	12	7	15	7	6	8	20	12	20	21	
0 - 19	75	93	169	67	94	111	166	230	389	490	456	437	
less than 0	4	0	8	123	175	178	303	258	563	497	517	538	
# files in corpus	100	100	200	200	300	300	500	500	1000	1000	1000	1000	
01997145 - - %	96	83	96	75	92	70	87	54	78	38	50	46	
02013034 - - %	*	*	*	*	*	*	79	43	*	*	49	45	

and only by a few tenths of a percentage point missed first place.

A relevant fact that is presented in all scenarios and event searches is the demonstration of the optimal value of K, taking into consideration the corpus of 1000 files. Comparing the results of semantic relevance in different queries, we found that the similarity values of the file contents in first place were always more consistent with values of K= 250 or K= 300, while the value of K= 100 where most ranged up to differences of more than 10 percentage points with respect to the others. In the case of K=500 relevant results are generally within the average range, but we must not forget that for this value of K the indexing time is up to six times greater and the response time of other three times slower.

V. CONCLUSIONS

In the light of the results, several points should be made in relation to the operation and computational performance of the LSA method. It has been very interesting to have proven that the proper value of K is 250 and 300 as optimal solution.

Although LSA obtains semantic relatedness based on statistical techniques of frequency of terms, it has been possible to demonstrate that when search expressions include more terms, they are identified with greater precision and a more precise classification of documents dealing with the same topic, whereas documents that are not related are clearly separated from the rest of the group (Table I and Table II). Also, it was possible to verify that if it had a sufficient number of files that belong to the application domain, in this case the health science area, LSA can establish semantic relatedness to identify those words or terms that are equivalent for the same context (Table IV and Table V).

Finally, one of the major issues of the LSA method has been described as related to computational performance, the times of the indexing process and the corresponding time of execution of queries. While most files were added to the corpus, the indexing time was increasing considerably. Particularly, when the indexing of 1000 files with K = 500 took several hours to complete, and the search response times went from seconds to minutes (Fig. 5); in these cases the use of LSA is no longer convenient. Fortunately, the computer technology continues to evolve and it is probably that with greater computing power and applied techniques of clustering it will be possible to solve this type of problems.

ACKNOWLEDGMENT

The authors would like to thank to SEP-PROMEP México and University of Guadalajara for the financial support to make this research.

REFERENCES

[1] M. B. Wolfe and S. R. Goldman, ”Use of Latent Semantic Analysis for predicting psychological phenomena: Two issues and proposed solutions”, Behavior Research Methods, Instruments, & Computers, vol. 35(1), 2003, pp. 22–31.

[2] K. Christidis, G. Mentzas, and D. Apostolou, ”Using latent topics to enhance search and recommendation in enterprise social software”, Expert Systems with Applications, vol. 39(10), 2012, pp. 9297–9307.

[3] S. C. Deerwester, S. T. Dumais, G. W. Furnas, R. A. Harshman, T. K. Landauer, K. E. Lochbaum, and L. A. Streeter, ”Computer information retrieval using Latent Semantic structure”, June 13 1989. US Patent 4,839,853.

[4] Y. Tonta and H. R. Darvish, ”Diffusion of Latent Semantic Analysis as a research tool: A social network analysis approach”, Journal of Informetrics, vol. 4(2), 2010, pp. 166–174.

[5] S. T. Dumais, ”LSA and information retrieval: Getting back to basics”, Handbook of Latent Semantic Analysis, 2007, pp. 293–321.

[6] T. K. Landauer and S. T. Dumais, ”A solution to plato’s problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge”, Psychological review, vol. 104(2), 1997, pp. 211–240.

[7] M. Ahat, S. B. Amor, M. Bui, S. Jhean-Larose, and G. Denhire, ”Document classification with LSA and pretopology”, Stud. Inform. Univ., vol. 8(1), 2010, pp. 125–144.

[8] K. Kireyev and T. K. Landauer, ”Word maturity: Computational modeling of word knowledge”, Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, vol. 1, 2011, pp. 299–308.

[9] T. K. Landauer and S. T. Dumais, ”Latent Semantic Analysis”, Scholarpedia, vol. 3(11), 2008, pp. 4356.

[10] S. T. Dumais, ”Improving the retrieval of information from external sources”, Behavior Research Methods, Instruments, & Computers, vol. 23(2), 1991, pp. 229–236.

[11] M. Gillam, C. Feied, J. Handler, E. Moody, B. Shneiderman, C. Plaisant, M. Smith, and J. Dickason, ”The Healthcare Singularity and the Age of Semantic Medicine”. In The Fourth Paradigm Data-Intensive Scientific Discovery, Microsoft Research, 2009, pp. 57–64.

[12] PubMED Central. The Pubmed Central OAI-PMH Service (PMC-OAI). [Online] Available: http://www.ncbi.nlm.nih.gov/pmc/tools/oai/ (Last visit: 2013, July 12).

[13] OAI-PMH. Open Archives Initiative Protocol for Metadata Harvesting. [Online] Available: http://www.openarchives.org/pmh/ (Last visit: 2013, July 12)

[14] A. Shinde. Anup Shinde’s web page, with LSI sample coded in C#. Available: http://www.anupshinde.com/latent-semantic-indexing/ (Last visit: 2013, July 12)

[15] P. Selormey. DotNetMatrix libraries. A basic linear algebra package for .Net, Available: http://www.codeproject.com/Articles/5835/DotNetMatrix-Simple-Matrix-Library-for-NET/ (Last visit: 2013, July 12)

A General Metadata Schema Operation using Formula Expression

Toshio Kodama
School of Engineering, University of Tokyo
and Maeda Corp.
Tokyo, Japan
kodama@ken-mgt.t.u-tokyo.ac.jp
kodama.ts@jcity.maeda.co.jp

Yoichi Seki
Software Consultant
Tokyo, Japan
gamataki51@hotmail.com

***Abstract*—In data management, there is a situation where equivalent objects are managed in different management spaces. This often brings about a lack of data consistency, which can often decrease the efficiency of management work. We call it the data overlapping problem. We consider the attaching function by an equivalent relation in the Incrementally Modular Abstraction Hierarchy to be quite effective to solve the problem. In this paper, we propose a metadata centralized space, a data centralized space, and their interconversion maps using Formula Expression. We then apply them to parts ledger management, where part data oftentimes becomes unexpectedly overlapped in metadata schema-centered management. These help users to arrange dynamic worlds from a data-centric viewpoint and prevent data overlap. In other words, if you utilize these functions in data management, you can reconstruct data spaces from different viewpoints.**

Keywords-metadata schema; topological space; formula expression; attaching function.

I. Introduction

In recent data management, situations where data and their dependencies change dynamically and constantly have been increasing in business environments. When data are managed after designing metadata schemas, data overlap occurs, which brings about a lack of data inconsistency. For example, when customer ledgers are designed and managed in different departments within a company, data on the same customer may not be recognized as the same in the system. As a result, the more the number of customer ledgers increases, the more complexity of the system increases.

To avoid this, certain functions are needed: 1. As with data, metadata schemas should also work flexibly; and 2. A data model should support the mechanism which guarantees an equivalence relation. But, in data management using conventional data models [2][3][5], unlike data, metadata schemas are not generally dealt with. Instead, they have to be defined in advance in the system design, and an equivalent relation is not modeled. A more powerful mathematical and fundamental background and a finite automaton to implement it are needed to model dynamic worlds accurately. Then, we propose the Incrementally Modular Abstraction Hierarchy (IMAH) [1] as the most appropriate model. The IMAH consists of the following seven mathematical space levels:

1. A homotopy level
2. A set level
3. A topology level, and a graph theoretical level as a special case
4. An adjunction space level
5. A cellular structured space level
6. A representation model level
7. A projection level

In modeling cyberworlds in cyberspaces, we define general properties of cyberworlds at the higher level and add more specific properties step by step, while moving down IMAH. The properties defined at the homotopy level are invariants of continuous changes of functions. The properties that do not change by continuous modifications in time and space are expressed at this level. At the set theoretical level, the elements of a cyberspace are defined, and a collection of elements constitutes a set with logical calculations. When we define a function in a cyberspace, we need domains that guarantee continuity such that the neighbors are mapped to a nearby place. Therefore, a topology is introduced into a cyberspace through the concept of neighborhood. Cyberworlds are dynamic. Sometimes cyberspaces are attached together, an exclusive union of two cyberspaces where attached areas of two cyberspaces are equivalent. It may happen that an attached space is obtained. These attached spaces can be regarded as a set of equivalent spaces called a quotient space that is another invariant. At the cellular structured level, an inductive dimension is introduced into each cyberspace. At the presentation level, each space is represented in a form which may be imagined before designing cyberworlds. At the view level, the cyberworlds are projected onto view screens.

In IMAH, elements as data are defined at the set level while information corresponding to a metadata schema is defined at the topological space level for the first time.

Next, we propose Formula Expression [9][11] as a finite automaton, which is explained in Section II. Since it expresses symmetry and recursiveness of information with minimum restrictions, it can be considered that general versatility in modeling is higher than with any other data model. In this paper, we focus on a generalization of metadata schema operation to prevent data overlap. In Section III, we first design a metadata centralized space, a data centralized space with Formula Expression, and their interconversion maps using the quotient map and the attaching map [9]. Next, we implement them in Section IV.

We demonstrate them in a simple example of parts ledger management to show their effectiveness in Section V. We reference related work in Section VI, and we conclude in Section VII.

II. The definition of Formula Expression

Formula Expression is a finite automaton defined as follows:

Formula Expression in the alphabet is the result of finite times application of the following (1)-(7).

(1) a ($\in\Sigma$) is Formula Expression
(2) unit element ε is Formula Expression
(3) zero element φ is Formula Expression
(4) when r and s are Formula Expression, addition of $r+s$ is also Formula Expression
(5) when r and s are Formula Expression, multiplication of $r\times s$ is also Formula Expression
(6) when r is Formula Expression, (r) is also Formula Expression
(7) when r is Formula Expression, $\{r\}$ is also Formula Expression

Combination is stronger in (5) than in (4). If there is no confusion, ×, (), {} can be abbreviated. + means disjoint union and is expressed as Σ specifically and × is also expressed as Π.

III. The design of topological spaces and interconversion maps

A. The space design

We design a formula for two topological spaces with a metadata schema by Formula Expression as follows:

1. metadata centralized spaces:

$$\Sigma\ \textit{metadata id}\times(\Sigma\ \textit{data id})$$

where each metadata id is uniquely identified.

2. data centralized spaces:

$$\Sigma\ (\Sigma\ \textit{metadata id})\times\textit{data id}$$

where each data id is uniquely identified.

B. The design of interconversion maps

Next, we design the two interconversion maps f and g between the above spaces using the quotient map and the attaching map [6].

f: Σ *metadata schema id*×(Σ *data id*)
→ Σ (Σ *metadata schema id*)×*data id*
g: Σ (Σ *metadata schema id*)×*data id*
→ Σ *metadata schema id*×(Σ *data id*)

f is onto mapping from a disjoint union of metadata centralized spaces to disjoint union of data centralized spaces attaching equivalent data identifiers, and g is also onto mapping from a disjoint union of data centralized spaces to disjoint union of metadata centralized spaces attaching equivalent metadata identifiers. These designs make the general operation of a metadata schema with data possible. The simple example of map f is shown below.

f (metadata 1×(data 1+data 2+data 3)+metadata 2×(data 1+data 3+data 4)+metadata 3×(data 1+data 2+data 4))
=(metadata 1+metadata 3)×data 1+(metadata 1)×data 2+(metadata 1+metadata 3)×data 3+(metadata 2+metadata 3)×data 4

IV. Implementation

This system is a JAVA application using JDK6. Below is the coding for the interconversion map f. Pseudo-code is used for simplicity. The focus is the recursive process (line 7) that is done if a coming numerical calculation is of the type ().

```
Function f (the argument p)
1   term = null; factor = p;
2   while (factor is not null){
3   term = getTerm(factor);
4   while (term is not null & term includes p){
5           factor = getFactor(term)
6           if(factor is of the type ()){
7                   factor = Function f (the contents);
                }
8           newFactor = newFactor×factor;
    }
9   newTerm = newTerm + term;
10  newFormula = newFormula + newTerm;
    }
11  return newFormula;
```

V. A Case Study: parts ledger management

A. Outline

In this section, we take up an example of *parts ledger* management, which is done in most manufacturing companies.

Parts ledgers management with consistency is generally considered to be difficult due to its complexity. The major reasons are: 1. Parts ledgers are managed in different places with different metadata schemas within a company; 2. Parts ledgers often change dynamically during mergers in companies or departmental integration within a company; and 3. Parts codes, which identify each part, are oftentimes different for the same part, because the codes are named differently by suppliers and there are also many inconsistencies in the way data is entered, since parts information is managed in different departments. For these reasons, important information for management, such as information about changes in the total price of a product due to changes in the unit price of a part cannot be outputted promptly by the management system. To avoid this, we arrange parts ledger data using the above design with Formula Expression.

In this case study, we assume that company A and company B have merged, and that their parts ledgers data need to be managed in an integrated way. To do so, we first create a formula for metadata centralized spaces of parts ledgers, and then convert it to a formula for data centralized spaces by the interconversion map *f*. Example data are shown in Figure 1, which is simplified as much as possible without losing generality.

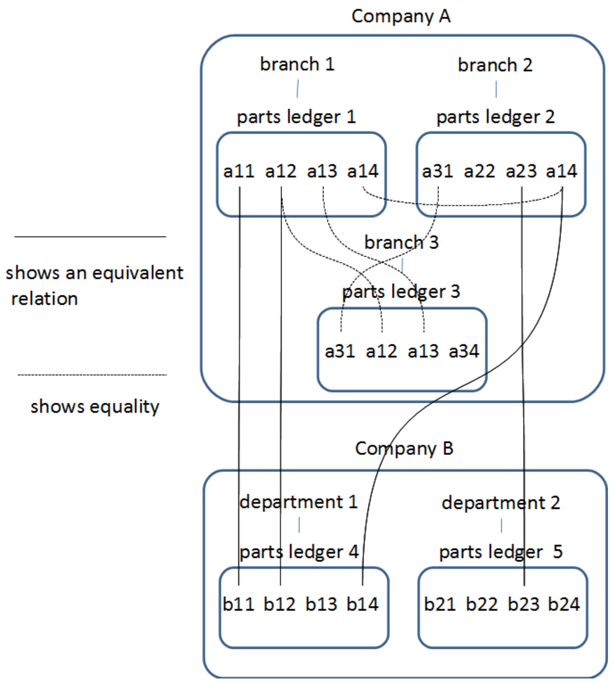


Figure 1. An example of metadata centralized spaces in parts ledger management

B. *Input according to the space design*

We first create a formula for parts ledgers in company A and B according to the metadata centralized space (III.A) as follows:

Formua1:

company A×(branch 1×parts ledger 1×(a11+a12+a13+a14)+branch 2×parts ledger 2×(a31+a22+a23+a14)+branch 3×parts ledger 3×(a31+a12+a13+a34))+company B×(department 1×parts ledger 4×(b11+b12+b13+b14)+department 2×parts ledger 5×(b21+b22+b23+b24))+company B×(branch 1×parts ledger4×(b11+b12+b13+b14)+branch 2×parts ledger 5×(b21+b22+b23+b24))

Here, identifiers of *company A and B, branch 1~3, department 1~2 and parts ledger 1~5* express *metadata id*, and *a11~a34* and *b11~b24* express *parts id*.

C. *Data conversion by the interconversion maps*

Next, you convert *Formula 1* to data centralized spaces thorough map *f* and also you attach the image recognizing equivalent relations of *a11 ~ b11, a12 ~ b12, a14 ~ b14* and *a23 ~ b23 as seen in Figure 1.* The result is the formula below:

Formula 2:

{company A×branch 1×parts ledger 1+company B×department 1×parts ledger 4}×{a11+b11}
+{(company A×branch 1×parts ledger 1+company A×branch 3×parts ledger 3)+company B×department 1×parts ledger 4}×{a12+b12}
+(company A×branch 1×parts ledger 1+company A×branch 3×parts ledger 3)×a13
+{(company A×branch 1×parts ledger 1+company A×branch 2×parts ledger 2)+company B×department 1×parts ledger 4}×{a14+b14}
+(company A×branch 2×parts ledger 2)×a22
+{company A×branch 2×parts ledger 2+company B×department 2×parts ledger 5}×{a23+b23}
+(company A×branch 2×parts ledger 2+company A×branch 3×parts ledger 3)×a31
+(company A×branch 3×parts ledger 3)×a34
+(company B×department 1×parts ledger 4)×b13
+(company B×department 2×parts ledger 5)×b21
+(company B×department 2×parts ledger 5)×b22
+(company B×department 2×parts ledger 5)×b24

In the outputted formula, you can know that there is no overlap of parts data, consequently, which ledgers a specified part belongs to accurately. See Figure 2.

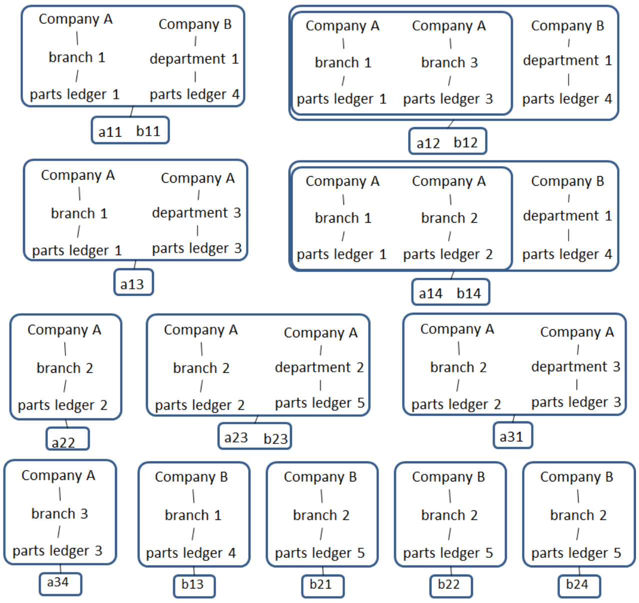


Figure 2. An example of data centralized spaces after the interconversion map in parts ledger management

D. *Considerations*

As we see in this example, using the above design with Formula Expression, we can say that (1) in data input, you only have to create a formula of spaces, instead of a metadata schema design or data input programs in advance, (2) in data output, you can see metadata schemas from a specified part's data, instead of developing output programs of metadata schemas, and (3) you only have to attach equivalent factors, instead of the design of unified notation. This means that the parts- centered spaces, which include no overlap of parts data, are constructed from the parts ledgers spaces, which include some overlap. In other words, the parts ledger data are arranged from a parts-centric view. The novelty of this function in the system is that data spaces can be reconstructed generally from other points which differ from the initial metadata schema design. Consequently, data overlap can be prevented using the function.

VI. Related work

One of the distinctive features of our research is the attaching function by equivalent relations, which can eliminate data overlap and return it back to the previous state [9]. Such a function, based on the adjunction space level which extends the topological space level, has never before been seen in other research [1]. Another feature is the application of the concept of topological process, which deals with a subset as an element, and that the cellular space extends the topological space, as seen in Section 2. Relational OWL as a method of data and schema representation is useful when representing the schema and data of a database [2][5], but it is limited to representation of an object that has attributes. Our method can represent both objects: one that has attributes as a cellular space and one that does not have them as a set or a topological space. Many works applying other models to XML schema have been done. The motives of most of them are similar to ours. The approach in [8] aims at minimizing document revalidation in an XML schema evolution, based in part on the graph theory. The X-Entity model [9] is an extension of the Entity Relationship (ER) model and converts XML schema to a schema of the ER model. In the approach of [6], the conceptual and logical levels are represented using a standard UML class and the XML represents the physical level. XUML [10] is a conceptual model for XML schema, based on the UML2 standard. This application research concerning XML schema is needed because there are differences in the expression capability of the data model between XML and other models. On the other hand, objects and their relations in XML schema and the above models can be expressed consistently by CDS, which is based on the cellular model. That is because the tree structure, on which the XML model is based, and the graph structure [3][4][7], on which the UML and ER models are based, are special cases of a topological structure mathematically. Entity in the models can be expressed as the formula for a cellular space in CDS. Moreover, the relation between subsets cannot in general be expressed by XML.

VII. Conclusions

In this paper, we designed the metadata schema centralized spaces, the data centralized spaces, and their interconversion maps. And we successfully applied them to parts ledger management, preventing data overlap. We conclude that the attaching function using Formula Expression is effective to model dynamic changing information worlds.

References

[1] T. L. Kunii and H. Kunii, “A Cellular Model for Information Systems on the Web - Integrating Local and Global Information”, In Proc. of DANTE'99, IEEE Computer Society Press, 1999, pp. 19-24.

[2] C. Laborda and S. Conrad, “Bringing Relational Data into the Semantic Web using SPARQL and Relational OWL”, In Proc. of 22nd International Conference On Data Engineering workshop 2006, IEEE Computer Society Press, 2006, pp. 55.

[3] Z. H. Liu, H. J. Chang, and B. Sthanikam, “Efficient Support of XQuery Update Facility in XML Enabled RDBMS”, In Proc. of 2012 IEEE 28th International Conference on Data Engineering (ICDE), IEEE Computer Society Press, 2012, pp. 1394-1404.

[4] J. Zhang, B. Lang and Y. Duan, “An XML Data Placement Strategy for Distributed XML Storage and Parallel Query”, In Proc. Of 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), IEEE Computer Society Press, 2011, pp. 433-439.

[5] H. Zhang, Z. Wang, Z. Gao and W. Li, “Design and Implementation of Mapping Rules from OWL to Relational Database”, In Proc. of 2009 WRI World Congress on Computer Science and Information Engineering, IEEE Computer Society Press, 2009, pp. 71-75.

[6] V. Mascardi, A. Locoro, and P. Rosso, “Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation”, IEEE Transactions on knowledge and data engineering, IEEE Computer Society Press, no.5, 2010, pp. 609-623.

[7] F. A. Currim, S. A. Currim, C. E. Dyreson, R. T. Snodgrass, S. W. Thomas, and R. Zhang, “Adding Temporal Constraints to XML Schema”, IEEE Transactions on knowledge and data engineering, IEEE Computer Society Press, vol. 24, no. 8, 2012, pp. 1361-1377.

[8] P. Kilpeläinen and R. Tuhkanen, “Towards Efficient Implementation of XML Schema Content Models,”, In Proc. of 2004 ACM Symposium on Document Engineering, ACM Press, 2004, pp. 239-241.

[9] T. Kodama, T. L. Kunii and Y. Seki, “A New Method for Developing Business Applications: The Cellular Data System”, In Proc of CW'06, IEEE Computer Society Press, 2006, pp. 65-74.

[10] K. Ohmori and T. L. Kunii, “Designing and modeling cyberworlds using the incrementally modular abstraction hierarchy based on homotopy theory”, The Visual Computer: International Journal of Computer Graphics, vol. 26, no.5, Springer-Verlag, 2010, pp. 297-309.

[11] T. Kodama, T. L. Kunii, and Y. Seki, “An Example of a Charge Calculation System using the Numerical Value and Exponential Calculation of the Cellular Data System”, Proceedings of the 2011 International Conference on Cyberworlds 2011 (CW2011, Oct.4-6, 2011, Banff, Alberta, Canada), IEEE Computer Society Press, 2011, pp. 31-37.

REEF: Resolving Length Bias in Frequent Sequence Mining

Ariella Richardson
Industrial Engineering
Jerusalem College of Technology
Jerusalem, Israel
Email: richards@jct.ac.il

Gal A. Kaminka and Sarit Kraus
Computer Science
Bar Ilan University
Ramat Gan, Israel
Email: galk,sarit@cs.biu.ac.il

***Abstract*—Classic support based approaches efficiently address frequent sequence mining. However, support based mining has been shown to suffer from a bias towards short sequences. In this paper, we propose a method to resolve this bias when mining the most frequent sequences. In order to resolve the length bias we define *norm-frequency*, based on the statistical z-score of support, and use it to replace support based frequency. Our approach mines the subsequences that are frequent relative to other subsequences of the same length. Unfortunately, naive use of *norm-frequency* hinders mining scalability. Using *norm-frequency* breaks the anti-monotonic property of support, an important part in being able to prune large sets of candidate sequences. We describe a bound that enables pruning to provide scalability. Experimental results on textual and computer user input data establish that we manage to overcome the short sequence bias successfully, and to illustrate the production of meaningful sequences with our mining algorithm.**

***Keywords*—Frequent Sequence Mining; Data Mining; Z-score;**

I. Introduction

The frequent sequence mining problem was first introduced by Agrawal and Srikant [1] and by Mannila et al. [2]. There are many possible applications for frequent sequential patterns, such as DNA sequence mining [3], text mining [4] anomaly detection [5] classification [6], and Web mining [7].

Frequent sequential pattern generation is traditionally based on selecting those patterns that appear in a large enough fraction of input-sequences from the database. This measure is known as *support*. In support based mining a threshold termed *minsup* is set. All sequences with a *support* higher than *minsup* are considered frequent.

Support based mining is known to suffer from a bias towards short patterns [8]: Short patterns are inherently more frequent than long patterns. This bias creates a problem, since short patterns are not necessarily the most interesting patterns. Often, short patterns are simply random occurrences of frequent items. The common solution of lowering the *minsup* results in obtaining longer patterns, but generates a large number of useless short sequences as well [9]. Using confidence measures lowers the number of output sequences but still results in short sequences.

Thus, removing the short sequence bias is a key issue in finding meaningful patterns. One possible way to find valuable patterns is to add weights to important items in the data. Yun [10] provides an algorithm for frequent sequence mining using weights. The drawback of this technique is that for many data sets there is no knowledge of what weights to apply. Seno and Karypis [11] propose eliminating the length bias by extracting all patterns with a support that decreases as a function of the pattern length. This solution is based on the assumption that a short pattern must have a very high support to be interesting, and a long pattern may be interesting even with a lower support. Although this is a fair assumption in many scenarios, it is challenging to find a measure that can be used for frequent pattern mining without making an assumption on the relationship between frequency and length. Searching for closed or maximal patterns [12]–[14] is another way to approach this bias. However, mining closed or maximal patterns may not be the best approach to solve the short sequence bias. Using closed and maximal sequences ignores shorter partial sequences that may be of interest. Other approaches include comparing the frequency of a sequence to its subsequences [15], and testing for self sufficient sequences [16]. We propose an algorithm that mines sequences of all lengths without a bias towards long or short sequences. Horman and Kaminka [8] proposed using a normalized support measure for solving the bias. However, their solution is not scalable. Furthermore they cannot handle subsequences that are not continuous or have multiple attributes. We allow holes in the sequence, for example: if the original sequence is ABCD, Horman and Kaminka can find the subsequences AB, ABC, ABCD, BC etc, but cannot mine ACD or ABD, whereas our proposed method can.

In this paper, we present an algorithm for **RE**solving l**E**ngth bias in **F**requent sequence mining (REEF). REEF is an algorithm for mining frequent sequences that normalizes the support of each candidate sequence with a length adjusted z-score. The use of the z-score in REEF eliminates statistical biases towards finding shorter patterns, and contributes to finding meaningful patterns as we will illustrate. However, it challenges the scalability of the approach: z-score normalization lacks the anti-monotonic property used in support based measures, and thus supposedly forces explicit enumeration of every sequence in the database. This renders useless any support based pruning of candidate sequences, the basis for scalable sequence mining algorithms, such as SPADE [17].

In order to provide a means for pruning candidate sequences, we introduce a bound on the z-score of future sequence expansions. The z-score bound enables pruning in the mining process to provide scalability while ensuring closure. Details on how the bound is calculated will be described later in the paper. We use this bound with an enhanced SPADE-like algorithm to efficiently search for sequences with high z-score values, without enumerating all sequences. A previous preliminary study [18] indicates that this bound assists the speedup substantially. We use three text corpora and computer user input to demonstrate how REEF overcomes the bias towards short sequences. We also show that the percentage of real words among the sequences mined by REEF is higher than those mined with SPADE.

The structure of the paper is as follows: Section II provides background and notation and introduces Norm-Frequent Sequence Mining Problem. In Section III the algorithm used for the Norm-Frequent Sequence Mining is described in detail. Experimental evaluation is provided in Section IV, and finally Section V concludes our paper.

II. Norm-Frequent Sequence Mining

Norm-Frequent Sequence Mining solves the short sequence bias present in traditional *Frequent* Sequence Mining. We begin by introducing the notation and the traditional *Frequent* Sequence Mining problem in Section II-A. We then define the *Norm-Frequent* Sequence Mining problem in Section II-B. We explain why the scalability is hindered by the naive implementation of normalized support and how this is resolved in Section II-C. Section II-C addresses scalability by introducing a bound that enables pruning in the candidate generation process. Finally in Section III we bring all parts together to compose the REEF algorithm.

A. Notation and Frequent Sequence Mining

We use the following notation in discussing Norm Frequent Sequence Mining.

event Let $I = \{I_1, I_2, ..., I_m\}$ be the set of all *items*. An *event* (also called an *itemset*) is a non-empty unordered set of *items* denoted as $e = \{i_1, ..., i_n\}$ where $i_j \in I$ is an item. Without loss of generality we assume they are sorted lexicographically. For example, $e = \{ABC\}$ is an event with items A B and C.

sequence A *sequence* is an ordered list of *events*, with a temporal ordering. The sequence $s = e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_q$ is composed of q events. If event e_i occurs before event e_j, we denote it as $e_i < e_j$. e_i and e_j do not have to be consecutive events and no two *events* can occur at the same time. For example, in the sequence s=$\{ABC\} \rightarrow \{AE\}$ we may say that $\{ABC\} < \{AE\}$ since $\{ABC\}$ occurs before $\{AE\}$.

sequence size and length The *size* of a sequence is the number of events in a sequence, $size(\{ABC\} \rightarrow \{ABD\}) = 2$. The *length* of a sequence is the number of items in a sequence including repeating items. A sequence with length l is called an *l-sequence*. $length(\{ABC\} \rightarrow \{ABD\}) = 6$.

subsequence and contain A sequence s_i is a *subsequence* of the sequence s_j, denoted $s_i \preceq s_j$, if $\forall e_k, e_l \in s_i, \exists e_m, e_n \in s_j$ such that $e_k \subseteq e_m$ and $e_l \subseteq e_n$ and if $e_k < e_l$ then $e_m < e_n$. We say that s_j *contains* s_i if $s_i \preceq s_j$. E.g., $\{AB\} \rightarrow \{DF\} \preceq \{ABC\} \rightarrow \{BF\} \rightarrow \{DEF\}$.

database The database D used for sequence mining is composed of a collection of sequences.

support The *support* of a sequence s in database D is the proportion of sequences in D that *contain* s. This is denoted $supp(s, D)$.

This notation allows the description of multivariate sequence problems. The data is sequential in that it is composed of ordered events. The ordering is kept within the subsequences as well. The multivariate property is achieved by events being composed of several items. The notation enables discussion of mining sequences with gaps both in events and in items, as long as the ordering is conserved. The mined sequences are sometimes called patterns.

In traditional support based mining, a user specified minimum support called *minsup* is used to define frequency. A *frequent* sequence is defined as a sequence with a support higher than *minsup*, formally defined as follows:

Definition 1 (Frequent): Given a database D, a sequence s and a minimum support *minsup*. s is *frequent* if $supp(s, D) \geq minsup$.

The problem of frequent sequence mining is described as searching for all the *frequent* sequences in a given database. The formal definition is:

Definition 2 (Frequent Sequence Mining): Given a database D, and a minimum support *minsup*, find all the *frequent* sequences.

In many support based algorithms such as SPADE [17], the mining is performed by generating candidate sequences and evaluating whether they are frequent. In order to obtain a scalable algorithm a pruning is used in the generation process. The pruning is based on the anti-monotonic property of support. This property ensures that support does not grow when expanding a sequence, e.g., $supp(\{AB\} \rightarrow \{C\}) \geq supp(\{AB\} \rightarrow \{CD\})$. This promises that candidate sequences that are *not frequent* will never generate *frequent* sequences, and therefore can be pruned. *Frequent* sequence mining seems to be a solved problem with a scalable algorithm. However, it suffers from a bias towards mining short subsequences. We provide an algorithm that enables mining subsequences of all lengths.

B. Norm-Frequent Sequence Mining using Z-Score

In this section, we define the problem of *Norm-Frequent* Sequence Mining. We use the statistical z-score for normalization. The z-score for a sequence of length l is defined as follows:

Definition 3 (Z-score): Given a database D and a sequence s. Let $l = len(s)$ be the length of the sequence s. Let μ_l and σ_l be the average support and standard deviation of support for sequences of length l in D. The *z-score* of s denoted $\zeta(s)$ is given by $\zeta(s) = \frac{supp(s) - \mu_l}{\sigma_l}$.

We use the z-score because it normalizes the support measure relative to the sequence length. Traditional mining, where support is used to define frequency, mines sequences that appear often relative to **all** other sequences. This results in short sequences since short sequences always appear more often than long ones. Using the z-score normalization of support for mining finds sequences that are frequent relative to other **sequences of the same length**. This provides an even chance for sequences of all lengths to be found frequent.

Based on the definition of z-score for a sequence we define a sequence as being *Norm-Frequent* if the z-score of the

seq 1:	$\{AB\} \rightarrow \{A\}$
seq 2:	$\{AB\} \rightarrow \{B\}$
seq 3:	$\{BC\} \rightarrow \{A\}$
seq 4:	$\{AB\} \rightarrow \{A\}$
seq 5:	$\{BC\} \rightarrow \{B\}$
seq 6:	$\{AC\} \rightarrow \{B\}$
seq 7:	$\{AB\} \rightarrow \{A\}$
seq 8:	$\{AC\} \rightarrow \{C\}$
seq 9:	$\{BC\} \rightarrow \{C\}$
seq 10:	$\{AC\} \rightarrow \{A\}$

Figure 1: Example database

sequence is among the top z-score values for sequences in the database. The formal definition follows:

Definition 4 (Norm-Frequent): Given a database D, a sequence s of length l and an integer k. Let Z be the set of the k highest z-score values for sequences in D, s is *norm-frequent* if $\zeta(s) \in Z$. In other words, we perform top-K mining of the most norm-frequent sequences.

We introduce the problem of *Norm-Frequent* Sequence Mining. This new problem is defined as searching for all the *norm-frequent* sequences in a given database. The formal definition follows and will be addressed in this paper.

Definition 5 (Norm-Frequent Sequence Mining): Given a database D and integer k, find all the *norm-frequent* sequences.

In Figure. 1, we provide a small example. The sequences $\{AB\}$, $\{A\} \rightarrow \{A\}$ and $\{B\} \rightarrow \{A\}$, of length 2, all have a support of 0.4 and are the most frequent patterns using support to define frequency. Notice that there are several sequences with this support, and no single sequence stands out. Consider the sequence $\{AB\} \rightarrow \{A\}$ of length 3. This sequence only has a support of 0.3. However, all other sequences of length 3 have a support no higher than 0.1. Although there are several sequences of length 2 with a higher support than $\{AB\} \rightarrow \{A\}$, this sequence is clearly interesting when compared to other sequences of the same length. This example provides motivation for why support may not be a sufficient measure to use. The norm-frequency measure we defined is aimed at finding this type of sequence.

Unfortunately, the z-score normalization test hinders the anti-monotonic property: we **cannot** determine that $\zeta(\{AB\} \rightarrow \{C\}) \geq \zeta(\{AB\} \rightarrow \{CD\})$.
Therefore, pruning becomes difficult; we cannot be sure that the z-score of a candidate sequence with length l will not improve in extensions of length $l{+}1$ or in general $l{+}n$ for some positive n. Therefore, we cannot prune based on z-score and ensure finding all *norm-frequent* sequences. This is a problem since without pruning our search space becomes unscalable.

Another problem with performing *Norm-Frequent* Sequence Mining is that the values for μ_l and σ_l must be obtained for sequences of all lengths prior to the mining process. This imposes multiple passes over the database and hinders scalability.

These important scalability issues are addressed and solved in Section II-C resulting in a scalable frequent sequence mining algorithm that overcomes the short sequence bias.

C. Scaling Up

As we explained in Section II-B, pruning methods such as those described in SPADE [17] cannot be used with *norm-frequent* mining. We propose an innovative solution that solves the scalability problem caused by the inability to prune.

Our solution is to calculate a bound on the z-score of sequences that can be expanded from a given sequence. This bound on the z-score of future expansions of candidate sequences is used for pruning. We define the bound and then explain how it is used. Z-score was defined in definition 3. The bound on z-score is defined in definition 6.

Definition 6 (Z-score-Bound): Given a database D and a sequence s. Let $\mu_{l'}$ and $\sigma_{l'}$ be the average support and standard deviation of support for sequences of length l' in D. The *z-score-bound* of s, for length l' denoted $\zeta^B(s, l')$ is given by $\zeta^B(s, l') = \frac{supp(s) - \mu_{l'}}{\sigma_{l'}}$.

We know that support is anti-monotonic, therefore as the sequence length grows support can only get smaller. Given a candidate sequence s of length l with a support of $supp(s)$ we know that for all sequences s' generated from s with length $l' > l$ the maximal support is $supp(s)$. We can calculate the bound on z-score, $\zeta^B(s, l')$, for all possible extensions of a candidate sequence. Notice that for all sequences s' that are extensions of s, $\zeta(s') \leq \zeta^B(s, l')$. The ability to calculate this bound on possible candidate extensions is the basis for the pruning.

In order to mine *frequent* or *norm-frequent* sequences, candidate sequences are generated and evaluated. In traditional *frequent* sequence mining there is only one evaluation performed on each sequence. If the sequence is found to be *frequent* it is both saved in the list of *frequent* sequences and expanded to generate future candidates, if it is not *frequent* it can be pruned (not saved and not used for generating candidates). For *norm-frequent* mining we perform two evaluations for each sequence. The first is to decide whether the proposed sequence is *norm-frequent*. The second is to determine if it should be expanded to generate more candidate sequences for evaluation. There are two tasks since z-score is not anti-monotonic and a sequence that is not *norm-frequent* may be used to generate *norm-frequent* sequences. This second task is where the bound is used for pruning. The bound on future expansions of the sequences is calculated for all possible lengths. If the bound on the z-score for all possible lengths is lower than the top n z-scores then no possible expansion can ever be *norm-frequent* and the sequence can be safely pruned from the generation process. If for one or more lengths the bound is high enough to be *norm-frequent* we must generate candidates from the sequence and evaluate them in order to determine if they are *norm-frequent* or not. This process guarantees that all *norm-frequent* sequences will be generated.

Using the bound enables pruning of sequences that are guaranteed not to generate *norm-frequent* candidates. The pruning enabled by using the bound resolves the first scalability issue of sequence pruning in the generation process. The second scalability problem of calculating μ_l and σ_l is resolved by calculating the values for μ_l and σ_l on a small sample of the data in a preprocessing stage described below.

III. REEF Algorithm

In this section, we combine all the components we have described in the previous sections and describe the implementation of REEF. The REEF algorithm is composed of several phases. The input to REEF is a database of sequences and an integer *'k'* determining how many Z-scores will be used to find *norm-frequent* sequences. The output of REEF is a set of *norm-frequent* sequences. Initially a sampling phase is performed to obtain input for the later phases. Next we perform the candidate generation phase. First norm-frequent 1-sequences and 2-sequences are generated. Once 2-sequences have been generated, an iterative process of generating candidate sequences is performed. The generated sequences are evaluated, and if found to be *norm-frequent* are placed in the output list of *norm-frequent* sequences. These sequences are also examined in the pruning process of REEF in order to determine if they should be expanded or not.

Sampling Phase - The sampling phase is performed as a preprocessing of the data in order to gather statistics of the average and standard deviation of support for sequences of all possible lengths. This stage uses SPADE [17] with a *minsup* of 0 to enumerate all possible sequences in the sampled data and calculate their support. For each length the support average and standard deviation are calculated. These values are distorted and corrected values are calculated using the technique described in [18]. These corrected values provide the average support μ_l and standard deviation of support σ_l that are used in z-score calculation and the bound calculation.

Candidate Generation Phase - The candidate generation phase is based on SPADE along with important modifications. As in SPADE we first find all 1-sequence and 2-sequence candidates. The next stage of the candidate generation phase involves enumerating candidates and evaluating their frequency.

We make two modifications to SPADE. The first is moving from setting a *minsup* to setting the $'k'$ value. $'k'$ determines the number of z-score values that norm-frequent sequences may have. Note that there may be several sequences with the same z-score value. The reason for this modification is that z-score values are meaningful for comparison within the same database but vary between databases. Therefore, setting the $'k'$ value is of more significance than setting a min-z-score threshold.

The second and major change we make is swapping *frequency* evaluation with *norm-frequency* evaluation. In other words, for each sequence s replace the test of is $supp(s, D) > minsup$ with the test of is $\zeta(s) \in Z$ where Z is the set of the $'k'$ highest z-score values for sequences in D. This replacement of the frequency test with the norm-frequency test is the essence of REEF and our main contribution.

The improved version of sequence enumeration including the pruning is presented in Figure. 2 and replaces the enumeration made in SPADE. The joining of *l*-sequences to generate *l+1*-sequences ($A_i \bigvee A_j$ found in line 6) is performed as in SPADE [17].

```
1:  for all x is a prefix in S do
2:      $T_x = \emptyset$
3:  $F_R = \{$k empty sequences$\}$
4:  for all items $A_i \in S$ do
5:      for all items $A_j \in S$, with $j \geq i$ do
6:          $R = A_i \bigvee A_j$ (join $A_i$ with $A_j$)
7:          for all $r \in R$ do
8:              if $\zeta(r) > \zeta($a seq $s$ in $F_R)$ then
9:                  $F_R = F_R \bigcup r \backslash s$ //replace $s$ with $r$
10:             for all $l' = l+1$ to input sequence length do
11:                 if $\zeta^B(r, l') > \zeta($a seq $s$ in $F_R)$ then
12:                     if $A_i$ appears before $A_j$ then
13:                         $T_i = T_i \bigcup r$
14:                     else
15:                         $T_j = T_j \bigcup r$
16:     enumerate-Frequent-Seq-Z-score($T_i$)
17:     $T_i = \emptyset$
```

Figure 2: Enumerate-Frequent-Seq-Z-score(S). Where S is the set of input sequences we are mining for frequent subsequences, A set of *norm-frequent* subsequences is returned, F_R is a list of sequences with the top $'k'$ z-scores

.

Pruning Phase using Bound - Obviously REEF cannot enumerate all possible sequences for norm-frequency evaluation. Furthermore as we discussed in Section II-B the z-score measure is not anti-monotonic and cannot be used for pruning while ensuring that norm-frequent candidates are not lost. In Section II-C we introduced the bound on z-score that is used for pruning.

The pruning in REEF calculates $\zeta^B(s, l')$ for all possible lengths $l' > l$ of sequences than could be generated from s. The key to this process that there is no need to actually generate the extensions s' that can be generated from s. It is enough to know the $supp(s)$, μ_l and σ_l for all $l' > l$. If for any length $l' > l$ we find that $\zeta^B(s, l') \in Z$ (in the list of 'k' z-scores) we keep this sequence for candidate generation, if not then we prune it. Using the bound for pruning reduces the search space while ensuring closure or in other words ensuring all frequent sequences are found. The pruning is performed as part of the enumeration described in algorithm Figure. 2. This pruning is the key to providing a **scalable** *norm-frequent* algorithm.

IV. Evaluation

In this section, we present an evaluation of REEF on a corpora of literature of various types. Section IV-A will show that *norm-frequent* mining overcomes the short sequence bias present in *frequent* mining algorithms. In Section IV-B we will provide evidence that the sequences mined with REEF are more meaningful than sequences mined with SPADE.

TEXT is a corpus of literature of various types. We treat the words as sequences with letters as single item events. We removed all formatting and punctuation from text (apart from space characters) resulting in a long sequence of letters. Mining this sequential data for frequent sequences produces sequences of letters that may or may not be real words. The reason we chose to mine text in this fashion is to show how interesting the frequent sequences are in comparison to norm-frequent sequences by testing how many real words are discovered. In other words, we use real words from the text as ground truth against which to evaluate the algorithms. We use three sets of textual data, one is from Lewis Carroll's "Alice's Adventures in Wonderland" [19], another is Shakespeare's "A Midsummer Night's Dream" [20] and the third is a Linux installation guide [21]. Evaluation is performed on segments of the corpus. Each test is performed on five segments.

User **P**attern **D**etection (UPD), is a data set composed of real world data used for evaluation. UPD logs keyboard and mouse activity of users on a computer as sequences, for a detailed description see [18]. Sequences mined from the UPD data can be used to model specific users and applied to security systems as in [22], [23] and [18]. The experiments are run on 11 user sessions.

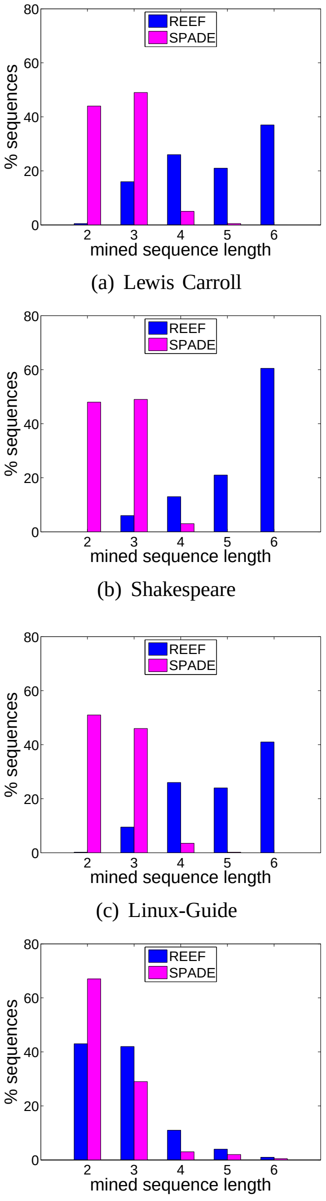
The input is composed of long sequences. In order to use REEF these sequences are cut into smaller sequences using a sliding window thus creating manageable sequences for mining. The size of the sliding window is termed *input sequence length* in our results. We use a setting of *minsup=1%* and *'k'=50* throughout all experiments and a sample rate of 10% for the preprocessing sampling component. Further details on implementation, running times etc. can be found in [24].

A. Resolving Length Bias in Frequent Sequence Mining

In this section, we establish how REEF successfully overcomes the short sequence bias that is present in the frequent sequence mining techniques. We performed *frequent* sequence mining with SPADE and *norm-frequent* sequence mining with REEF. We compared the lengths of the mined sequences for both algorithms. The results are displayed in Figure. 3. Results are shown for all three TEXT data sets and for the UPD set. The x-axis shows the lengths of the mined sequences. The y-axis displays the percentage of sequences found with the corresponding length. For each possible length we counted the percentage of mined sequences with this length.

The text results on all three text corpora show how SPADE mines mainly short sequences, while REEF manages to mine a broader range of sequence lengths as displayed in Figure. 3(a),(b),(c). REEF results are much closer to known relation between word length to frequency [25] than the SPADE output. In the next section we count how many of these sequences are real words to illustrate superiority of REEF.

For the UPD data REEF again overcomes the short sequence bias and provides output sequences of all lengths in a more normal distribution than with SPADE. This can be seen in in Figure. 3(d). We must point out that in contrast to the TEXT corpora, there is no known ground truth as to what the length of frequent sequences should be in this domain, and what their distributions are. Thus, there is no way to confirm whether we have found the correct distribution of the frequent sequences. However, we do show that we are not restricted to mining short sequences alone.



(a) Lewis Carroll

(b) Shakespeare

(c) Linux-Guide

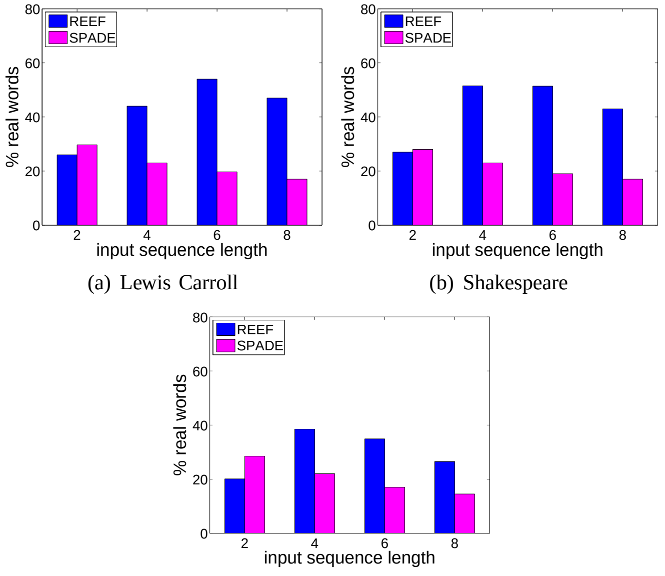
(d) UPD

Figure 3: Removal of length bias.

B. Mining Meaningful Sequences with REEF

The text domain was chosen specifically in order to illustrate the quality of the output sequences. We wanted a domain where the meaning of interesting sequences was clear. TEXT is obviously a good domain for this purpose since words are clearly more interesting than arbitrary sequences of letters. We hope to find more real words when mining text than nonsense words. Our evaluation is performed on three sets of text as described above. Results appear in Figure. 4. We compare results on *frequent* sequence mining using SPADE with *norm-frequent* sequence mining using REEF. The x-axis shows different input sequence lengths (window sizes). For each input sequence length we calculated the percentage of real words that were found in the mined sequences. This is displayed on the y-axis. For example the top 15 mined sequences in Shakespeare using REEF: {*e he,or,e and,her,n th,though,he,s and,her,thee,this,thou,you,love,will*}and using SPADE: {*rth,mh,lr,sf,tin,op,w,fa,ct,ome,ra,yi,em,tes,t l*} Using REEF yields many more meaningful words than using SPADE.

For all text sets REEF clearly outdoes SPADE by far. REEF



(a) Lewis Carroll

(b) Shakespeare

(c) Linux Guide

Figure 4: Percentage of real words found among sequences.

manages to find substantially more words than SPADE for all input lengths. The short input-sequence sizes of 2 does not produce high percentages of real words for REEF or SPADE. Using longer input sequence lengths exhibits the strength of REEF in comparison to SPADE. For input lengths of 4,6 and 8 REEF manages to find a much higher percentage of words than SPADE. Clearly for text REEF performs much better mining than SPADE and the sequences mined are more meaningful.

V. Conclusion and Future Work

We developed an algorithm for frequent sequence mining named REEF that overcomes the short sequence bias present in many mining algorithms. We did this by defining *norm-frequency* and using it to replace support based frequency used in algorithms such as SPADE. In order to ensure scalability of REEF we introduced a bound for pruning in the mining process.

Our experimental results show without doubt that the bias is indeed eliminated. REEF succeeds in finding frequent sequences of various lengths and is not limited to finding short sequences. We illustrated that REEF produces a more variant distribution of output pattern lengths. We also clearly showed on textual data how REEF mines more real words than SPADE. This seems to indicate that when mining sequences are not textual, we can expect to mine meaningful sequences as well. In the future we hope to improve the bound used for mining. Thus providing an algorithm that is more efficient while still producing the high quality sequences we found in REEF.

References

[1] R. Agrawal and R. Srikant, “Mining sequential patterns,” in Proceedings of the Eleventh International Conference on Data Engineering, 1995, pp. 3–14.

[2] H. Mannila, H. Toivonen, and A. I. Verkamo, “Discovering frequent episodes in sequences (extended abstract),” in 1st Conference on Knowledge Discovery and Data Mining, 1995, pp. 210–215.

[3] F. Elloumi and M. Nason, “Searchpattool: a new method for mining the most specific frequent patterns for binding sites with application to prokaryotic dna sequences.” BMC Bioinformatics, vol. 8, 2007, pp. 1–18.

[4] N. Zhong, Y. Li, and S.-T. Wu, “Effective pattern discovery for text mining,” IEEE Transactions on Knowledge and Data Engineering, vol. 24, 2012, pp. 30–44.

[5] W. Fan, M. Miller, S. J. Stolfo, W. Lee, and P. K. Chan, “Using artificial anomalies to detect unknown and known network intrusions,” Knowledge and Information Systems, vol. 6, 2004, pp. 507–527.

[6] C.-H. Lee and V. S. Tseng, “PTCR-Miner: Progressive temporal class rule mining for multivariate temporal data classification,” in IEEE International Conference on Data Mining Workshops, 2010, pp. 25–32.

[7] P. Senkul and S. Salin, “Improving pattern quality in web usage mining by using semantic information,” Knowledge and Information Systems, 2011, pp. 527–541.

[8] Y. Horman and G. A. Kaminka, “Removing biases in unsupervised learning of sequential patterns,” Intelligent Data Analysis, vol. 11, 2007 , pp. 457–480.

[9] C. Luo and S. M. Chung, “A scalable algorithm for mining maximal frequent sequences using sampling,” in ICTAI ’04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence. Washington, DC, USA: IEEE Computer Society, 2004, pp. 156–165.

[10] U. Yun, “An efficient mining of weighted frequent patterns with length decreasing support constraints,” Knowledge-Based Systems, vol. 21, 2008 , pp. 741–752.

[11] M. Seno and G. Karypis, “SLPMiner: An algorithm for finding frequent sequential patterns using length-decreasing support constraint,” in ICDM ’02: Proceedings of the 2002 IEEE International Conference on Data Mining. Washington, DC, USA: IEEE Computer Society, 2002, p. 418.

[12] P. Tzvetkov, X. Yan, and J. Han, “Tsp: Mining top-k closed sequential patterns,” Knowledge and Information Systems, vol. 7, 2005, pp. 438–457.

[13] C. Luo and S. Chung, “A scalable algorithm for mining maximal frequent sequences using a sample,” Knowledge and Information Systems, vol. 15, 2008, pp. 149–179.

[14] N. Tatti and B. Cule, “Mining closed strict episodes,” in ICDM ’10: Proceedings of the 2010 Tenth IEEE International Conference on Data Mining, 2010, pp. 34–66.

[15] N. Tatti, “Maximum entropy based significance of itemsets,” Knowledge and Information Systems, vol. 17, 2008, pp. 57–77.

[16] G. I. Webb, “Self-sufficient itemsets: An approach to screening potentially interesting associations between items,” ACM Trans. Knowl. Discov. Data, vol. 4, Jan 2010 , pp. 1–20.

[17] M. J. Zaki, “SPADE: An efficient algorithm for mining frequent sequences,” Machine Learning Journal, vol. 42, 2001, pp. 31–60.

[18] A. Richardson, G. Kaminka, and S. Kraus, “CUBS: Multivariate sequence classification using bounded z-score with sampling,” in IEEE International Conference on Data Mining Workshops, 2010, pp. 72–79.

[19] L. Carroll, “Alice’s Adventures in Wonderland,” Project Gutenberg.

[20] W. Shakespeare, “A Midsummer Night’s Dream,” Project Gutenberg.

[21] J. Goerzen and O. Othman, “Debian gnu/linux : Guide to installation and usage,” Project Gutenberg.

[22] A. E. Ahmed, “A new biometric technology based on mouse dynamics,” IEEE Transactions on Dependable and Secure Computing, vol. 4, 2007, pp. 165–179.

[23] R. Janakiraman and T. Sim, “Keystroke dynamics in a general setting,” in Advances in Biometrics, ser. Lecture Notes in Computer Science, vol. 4642/2007. Springer Berlin/ Heidelberg, Aug 2007, pp. 584–593.

[24] A. Richardson, “Mining and classification of multivariate sequential data,” Ph.D. dissertation, Bar Ilan University, 2011.

[25] B. Sigurd, M. Eeg-Olofsson, and J. V. Weijer, “Word length, sentence length and frequency zipf revisited,” Studia Linguistica, vol. 58, 2004 , pp. 37–52.

Semantic Tools for Forensics: Approaches in Forensic Text Analysis

Michael Spranger and Dirk Labudde
University of Applied Sciences Mittweida
Mittweida, Germany
Email: {*name.surname*}@hs-mittweida.de

***Abstract*—The analysis of digital media and particularly texts acquired in the context of police securing/seizure is currently a very time-consuming, error-prone and largely manual process. Nevertheless, such analysis are often crucial for finding evidential information in criminal proceedings in general as well as fulfilling any judicial investigation mandate. Therefore, an integrated computational solution for supporting the analysis and subsequent evaluation process is currently developed by the authors. In this work, we present an approach for categorizing texts with adjustable precision combining rule-based decision formula and machine learning techniques. Furthermore, we introduce a text processing pipeline for deep analysis of forensic texts as well as an approach for the identification of criminological roles.**

***Keywords*—forensic; ontology; German; text processing**

I. Introduction

The analysis of texts that are subject of legal considerations with the goal of obtaining criminalistic evidence is a branch of general linguistics [1]. Such texts are retrieved by persons involved in the criminal proceedings from a variety of sources, e.g., secured or confiscated storage devices, computers and social networks. Forensic texts, as considered in this work, relate to textual data that may contain evidential information. In contrast to the texts usually considered in scientific work focussing text processing tasks this kind of texts are neither clearly defined nor thematically unified. Additionally, such texts may vary in quality with respect to their grammar, wording and spelling which strongly depends on the author's language skills and the target audience. Rather, textual data of different type and origin need to be meaningfully linked to answer a specific criminological question reasonably and above all accurately. Furthermore, forensic linguistics cover beside other research topics, utterance and word meaning or authorship analysis and proof [2].

The results of these analyses are used to solve other more complex problems in the criminal investigations, like

- recognition and separation of texts with a case-related criminalistic relevance
- recognition of relations in these texts in order to reveal whole relationship networks and planned activities
- identification and/or tracking of fragmented texts
- identification or tracking of hidden semantics

In the considered context, the term *hidden semantics* is synonymous with one kind of linguistic steganography. In this work only the first two points are in the focus. However, this kind of deep analysis takes a long time, especially if the amount and heterogeneity of data, the fast changeover of communication forms and communication technologies is taken into account. In order to solve this problem, computer linguistic methods and technologies can be applied. These are originated in the crossover of linguistics and computer sciences [3]. The complexity of the evaluation makes it difficult to develop one single tool covering all fields of application. In order to address this problem, a domain framework is currently under development (see [4] for further discussions).

As a consequence of the analysis of the secured data from a historical case of business crime and the exploration of the special needs of criminologists discussed in Section II we present in this work a pipeline for categorizing texts with adjustable precision using an approach which is combined of rule-based decision formula and machine learning techniques. Especially that leaves the opportunity to the criminologist to decide whether the specificity (precision) is more important or the sensitivity (recall), although a high sensitivity may be of greater practical importance. Thus, a high sensitivity is principally necessary to find all incriminating or even exculpatory documents but the results need to be filtered manually since they may be interspersed with irrelevant documents, whereas a high specificity is sometimes more appropriate to get a quick overview about the corpus. Furthermore, we outline a text processing pipeline for deep analysis of forensic texts based on these insights and a rule-based approach for identifying special roles of named entities. Currently, the text categorization module is evaluated in practice whereas the deep analysis pipeline including the role identification is under implementation.

In the next Section the peculiarities of the considered kind of texts is shown at a glance. Subsequently, a pipeline for analysing forensic texts deeply as well as a first approach for detecting forensic roles is outlined before a practicable method for categorizing such texts is introduced and discussed.

II. Assessment of Requirements

This work focusses textual data secured by police officers as part of the evidence process. Hence, for the purposes of this work historical data in a case of business crime is provided by the prosecutorial. A first manual assessment of these data enables to determine, whether:

- the data material is of considerable heterogeneity related to its structure and domain

ISBN: 978-1-61208-311-7

- important information may be situated in non-text based data (e.g photocopies of invoices)
- there are totally irrelevant texts that may hide relevant information through their abundance (e.g forms, templates)
- information may have been deliberately obscured in order to protect them from discovery
- some texts can be characterized by strong syntactic weaknesses
- some texts may be fragmented by erasing/reconstruction

These specific characteristics distinguish the examined corpus from other corpora commonly used and evaluated in research.

Further, a survey made by the authors, which was conducted by affiliated criminologists has revealed that finding and separating relevant documents seized in the database is the most time consuming and difficult part during the evaluation.

III. Approaches in Forensic Text Analysis

In this section, several strategies for handling forensic texts respecting the insights from the needs assessment (section II) are introduced. Since the most aspects of this work are currently under implementation no final results will be presented yet. Thus, these aspects are outlined subsequently.

A. *Pipeline for Deep Analysis*

The deep analysis of forensic texts has to respect their characteristics described in the previous section. It includes particularly tasks in Information/Event Extraction to instantiate a criminological ontology as the central element in the solution developed under this work. In particular, the work of Wimalasuriya and Dou [5], Embley [6] and Maedche [7], shows that the use of ontologies is suitable for assisting the extraction of semantic units as well as their visualization and structures such processes very well. We have divided the whole process in three sub-processes:

1) creation of both the criminological ontology and the analysis corpus
2) basic textual processing and detection of secondary contexts
3) instantiation of the ontology and iteratively refinement

In order to define the extraction tasks as well as to introduce case-based knowledge the first of all is the creation of the criminological ontology in its specialized form as Topic Map we have developed in an earlier work [4]. This step may be supported by using existing ontologies created in similar previous cases. Subsequently, the analysis corpus needs to be created, especially for separating the textual data from other files and extracting the raw texts from the documents including optical character recognition in cases of digital images like photocopies. This data is stored in a database together with extracted meta-data and added to an index for quick access. In the second step some state-of-the-art textual processing steps like Part-of-Speech-tagging, language recognition and some special operations for structured texts may be performed. Especially, we detect event-narrative documents. This task has been introduced by Huang and Riloff [8] for exploring secondary contexts. They define these as sentences that are not explicitly part of the main event description. Nevertheless, these secondary contexts could yield information related to the event of interest that could provide important evidence or lead to the booty, further victims or accomplices. The final step within the main process is constituted by the actual extraction process. Here, the actual event sentences that are suitable to instantiate at least one part of the ontology are recognized and, if needed, extracted together with the information from secondary contexts. Then, we try to refine the instantiated model iteratively by identifying forensic roles as described in III-B. Figure 1 illustrates the whole process schematically.

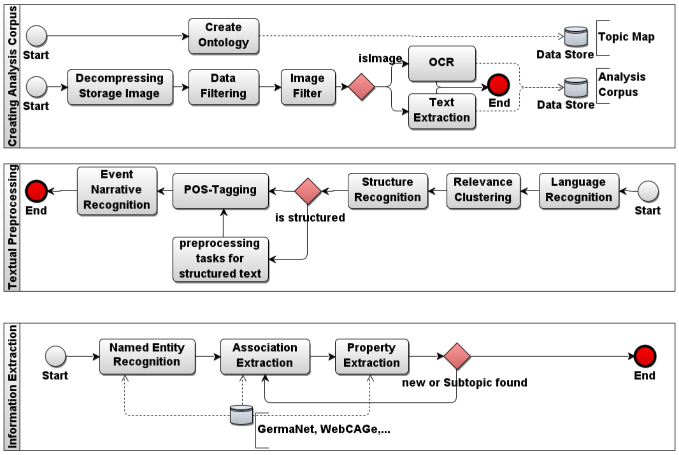


Fig. 1. The tool-pipeline for deep analysis. We have divided the whole process in three sub-processes: 1) creating analysis corpus 2) textual preprocessing 3) information extraction

B. *Identification of Forensic Roles*

The recognition of named entities is a well-researched part of Text Mining and a regular task in every Information/Event Extraction solution as well as in our pipeline mentioned in III-A. The general task is to identify all instances $i \in I$ of each concept $c \in C$ taking into account their hypernymy and hyponymy relationships. This task can be solved practically by using Gazeteer-based solutions via supervised learning methods [9], [10] up to the usage of semi-/unsupervised learning approaches [11]. However, no existing solution we applied has been proven itself to be able to assign forensic roles. The assignment of such a role is often dependent on more than one document as well as the contribution of case-based knowledge by the criminologist. Therefore, our framework is based on an ontology acting as an extraction and visualization template that is able to provide such knowledge. The ontology model we used is based on the Topic Map standard. In our previous work [4] we stated that each topic can contain a set of facets. These facets are used beside others to model rules that an inference machine can use to reason the appropriate role of an entity within a post-process. In this way the level of detail within the computational recognition of entities is able to be increased. Figure 2 shows a detail of a fictional forensic Topic Map that may have been created by a criminologist. Here, a accomplice is described as a person that satisfies one or two of the following rules:

- the person has common interest in the deed exactly when he has instantiated an association possess with the topic booty
- the person has shared worked exactly when their related instance in the Topic Map has an instantiated association drive to an instance of the topic getaway-car

The number of rules that have to be satisfied depends on rule weights which act as indicators for rule importance. The concrete instance defines the same facets with binary values depending on the matching behaviour of each rule.

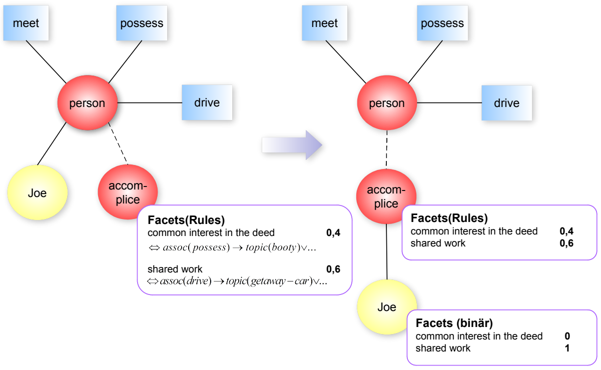

meet
possess
person
drive
Joe
accom-plice
Facets(Rules)
common interest in the deed 0,4
⇔ assoc(possess) → topic(booty)∨...
shared work 0,6
⇔assoc(drive)→topic(getaway−car)∨...
meet
possess
person
drive
accom-plice
Facets(Rules)
common interest in the deed 0,4
shared work 0,6
Joe
Facets (binär)
common interest in the deed 0
shared work 1

Fig. 2. Gradually refining of named entities. The entity *Joe* as instance (yellow circle) of the abstract topic (red circle) *person* can gradually assigned to their concrete manifestation *accomplice* which is a subtopic by iterative comparison of its facets lodged as rules.

C. Categorization of Forensic Texts

As discussed in Section II, filtering and categorization is the most important task in evaluation of forensic texts and a regular Information Retrieval task. Categorization as a specialization of classification aims to place a document in one small set of categories using machine learning techniques. More formal, given a set of documents $D = \{d_1, ..., d_m\}$ and further a set of categories $C = \{c_1, ..., c_n\}$ the task can be described as an surjective mapping $f : C \to D$. Ikonomakis et al. [12] have given an overview about supervised machine learning methods for solving this problem. However, they observed that the performance is significantly depending on a corpus of high quality and sufficient size. Riloff and Lehnert [13] introduced an approach for high-precision text classification. The augmented relevancy signature algorithm they introduced reached up to 100% precision with over 60% recall on the MUC-4 corpus. Nevertheless, in the focussed domain these results are not always sufficient especially since they do not relate to the properties of forensic texts. It has to be emphasized, that each false-negative (a not identified, case-relevant document) could provide crucial evidences. This highlights the necessity for a method which yields 100% recall with justifiable precision. Beebe and Clark [14] has introduced an approach to handle the information overload resulting from the recall-precision trade-off problem. They considered a similar problem and suggest to cluster the results thematically. However, designing and training a suitable classifier is a challenging problem. Since the knowledge of the criminologist (general and case-based) is available related to a concrete judicial investigation order, rules can improve the performance in some cases. This leads to a combined approach. Since the categories has modelled as a taxonomy tree we can extend this model so that we are able to assign a set of rules (e.g., regular expressions applied on the documents body) to each category. These rules are combined by disjunction within the categories itself and by conjunction between different categories in cases of one continuous chain of parent-child relationships (figure 3). Each of these rules has to define the target that it should applied on (e.g., file name or content), a rule type that helps to select the corresponding rule solver and the rule itself. In this way, we are able to select a certain number of seeds that ensure high precision which is required to start an appropriate bootstrapping machine learning algorithm to classify the remaining documents (figure 4). Notice, the performance can be influenced by rephrasing the corresponding rules, since the performance of a bootstrapping algorithm significantly depends on the seed elements chosen, more precise their representativeness. Thus, strictly formulated rules may result in high precision but low recall, whereas applying more weak rules will increase the recall. First measures of performance using probability-based classifiers, like Naive Bayes, as well as similarity-based classifiers, like k-NN or TF-IDF shows that the performance reaches up to

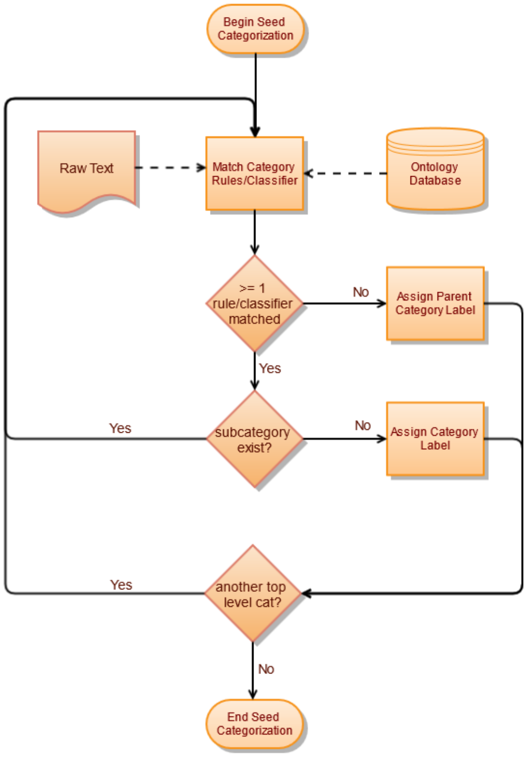


Fig. 3. *Acquisition of seed documents:* The raw text under consideration is checked against a set of category rules recursively. Starting at a top-level category, at least one category rule/classifier has to match until the match of each subcategory, drawn from recursion, has failed. In this way only the label of the most specific category starting at each existing top-level category is assigned.

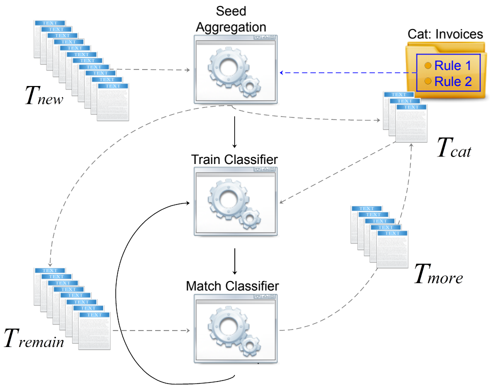


Fig. 4. Bootstrapping Algorithm for classifying forensic texts. From the texts T_{new} a set of seed documents for each category is acquired using the rules annotated in the taxonomy. This set T_{cat} is used to train one initial weak binary classifier per category. Subsequently, this classifier is used to classify the remaining texts T_{remain} and store the new labelled documents T_{more} to T_{cat}. Finally, the classifier is going to be improved iteratively using T_{cat} until no document is left or no further improvement is possible.

100% precision and recall applied on the corpus provided by the prosecutorial as mentioned in Section II depending on the employed algorithm and the concrete category. This result could be a consequence of classifier over-fitting caused by the underlying homogeneous corpus. We have observed that in the in the corpus we used the documents are characterized by great similarity. Therefore, a more appropriate corpus is created currently.
One of the biggest advantages of this combined approach lays in the adjustable precision depending on an intelligent combination of rules and machine learning algorithms.

IV. Conclusion

In this work, we have outlined some kernel processes for information extraction in the environment of the criminal proceedings. These processes are suitable to deal with very heterogeneous data concerning their domain as well as their quality. In the task of deep exploration of the raw data there was great emphasis on the discovery of all relevant information using secondary contexts to avoid misunderstandings and lacks in the evidence. In the identification of forensic roles we have described a new approach in refining ontology instances by deriving and applying semantic roles logic-based. A corresponding module using Prolog is currently under development. In the task of classification of forensic texts we have to respect that each misclassified file could lead to a lack of evidence. Therefore, it must be ensured that at best no type II errors occur during the categorization. At the same time the taxonomy definition has to remain flexible. Because of a lack of training data supervised learning is not applicable. Therefore, a bootstrapping approach is chosen, combined with a rule-based search for seed files we have earned very good preliminary results at 100% accuracy in selected domains. However, this unexpected result could be due to an over-fitting to the used corpus. For this reason we currently create a new extended Corpus with the support of the prosecutorial.

Acknowledgment

The authors would like to thank the members of the criminal investigation department and the prosecutorial Chemnitz (Germany). We acknowledge funding by ”Europäischer Sozialfonds” (ESF), the Free State of Saxony and the University of Applied Sciences Mittweida.

References

[1] H. Kniffka, Working in Language and Law. A German perspective. Palgrave, 2007.

[2] E. Fobbe, Forensische Linguistik - Eine Einführung. (Forensic Linguistics - An Introduction) Narr Frankcke Attempto Verlag, 2011.

[3] K.-U. Carstensen, C. Ebert, C. Ebert, S. Jekat, R. Klabunde, and H. Langer, Computerlinguistik und Sprachtechnologie - Eine Einführung (Computational Linguistics and Language Technology - An Introduction), 3rd ed. Spektrum Akademischer Verlag, 2010.

[4] M. Spranger, S. Schildbach, F. Heinke, S. Grunert, and D. Labudde, “Semantic tools for forensics: A highly adaptable framework,” in Proc. 2nd. International Conference on Advances in Information Management and Mining, IARIA. ThinkMind Library, 2012, pp. 27 – 31.

[5] D. C. Wimalasuriya and D. Dou, “Ontology-based information extraction: An introduction and a survey of current approaches,” Journal of Information Science, vol. 36, no. 3, 2010, pp. 306–323.

[6] D. W. Embley, “Toward semantic understanding: an approach based on information extraction ontologies,” in Proceedings of the 15th Australasian database conference - Volume 27, ser. ADC ’04. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 3–12.

[7] A. Maedche, G. Neumann, and S. Staab, “Bootstrapping an ontology-based information extraction system,” Studies In Fuzziness And Soft Computing, vol. 111, 2003, pp. 345–362.

[8] R. Huang and E. Riloff, “Peeling back the layers: detecting event role fillers in secondary contexts,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, ser. HLT ’11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1137–1147.

[9] A. Mikheev, M. Moens, and C. Grover, “Named entity recognition without gazetteers,” in Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, ser. EACL ’99. Stroudsburg, PA, USA: Association for Computational Linguistics, 1999, pp. 1–8.

[10] G. Zhou and J. Su, “Named entity recognition using an hmm-based chunk tagger,” in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ser. ACL ’02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 473–480.

[11] Z. Kozareva, “Bootstrapping named entity recognition with automatically generated gazetteer lists,” in Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, ser. EACL ’06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 15–21.

[12] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, “Text classification using machine learning techniques,” WSEAS Transaction on Computers, vol. 4, no. 8, 2005, pp. 966–974.

[13] E. Riloff and W. Lehnert, “Information extraction as a basis for high-precision text classification,” Transactions on Information Systems, vol. 12, no. 3, 1994, pp. 296–333.

[14] N. L. Beebe and J. G. Clark, “Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results,” Digital Investigation, vol. 4, 2007, pp. 49–54.

Logo-DM – A Speech Therapy Optimization Data Mining System

Mirela Danubianu, Adina Luminita Barala
Faculty of Electrical Engineering and Computer Science
Stefan cel Mare University of Suceava
Suceava, Romania
e-mail: mdanub@eed.usv.ro; adina@eed.usv.ro

Abstract— **This paper presents Logo-DM, a prototype for a data mining system dedicated to help the optimization of the personalized speech therapy. It uses data collected by TERAPERS system that was implemented at the "Stefan cel Mare" University of Suceava to assist speech therapists in the treatment of children suffering from dyslalia. Over these data, some data mining methods has been applied. The patterns obtained are useful for specialists for an efficient current activity. These can also provide knowledge that serves to improve the support offered by TERAPERS by raising the quality of its embedded expert system.**

Keywords-computer-based speech therapy; data mining; classification; association rules.

I. INTRODUCTION

Speech impairment, one of the most common issues in childhood, might be the source of adult's integration problems in the community. This is one of the reasons why a special attention was paid to speech therapy. A speech disorder can be corrected, if it is discovered and properly treated in due time. However, therapy is a complex process, which must be adapted to each child.

Since 1990-2000, computer-assisted speech therapy became a frequent practice. In this context, many Computer-Based Speech Therapy (CBST) tools or systems were developed.

For example, IBM has developed Speechviewer III system [1]. While users perform several speech actions, Speechviewer III creates an interactive visual model of speech. Another project is the ICATIANI device, developed by TLATOA Speech Processing Group, CENTIA Universidad de las Américas, Puebla Cholula, Pue, México [2]. It uses sounds and graphics in order to ensure the practice of Spanish Mexican pronunciation. The third example, Articulation Tutor (ARTUR) [3] provides an integrated speech therapy system. It contains two main components: an intuitive graphical interface named *Wizard-of-Oz* and a virtual speech tutor named *Artur*. Using audio (user's utterance) and video (facial data) information, the system can recognize and reproduce mispronunciations. After that, ARTUR suggests the correct pronunciation (audio data) and the correct speech elements' position (virtual articulator model).

The use of these systems has allowed researchers and practitioners to collect a considerable volume of data, related to children' particularities, therapeutically paths, and results. But, contrary to expectations, a large amount of data does not automatically lead to a significant increase of the volume and quality of information, because traditional data processing tools are not applicable.

For these reasons, modern methods that aim to discover new and potentially useful patterns from large volumes of data were implemented. This process is called Knowledge Discovery in Databases (KDD) [4]. Its central step is data mining that involves the application of algorithms, which with acceptable performance, provide a particular enumeration of patterns from data.

In 2008, at Research Center in Computer Science from "Stefan cel Mare" University of Suceava the TERAPERS system was implemented.

This is a CBST that aims to assist the personalized therapy of dyslalia – an articulation disorder found to a significant percentage of children from age of 3-4 years. This is the first CBST developed for Romanian language. During its exploitation, data about few hundred cases were collected. This was the starting point for the idea to try the optimization of personalized speech therapy by data mining techniques.

Our paper's purpose is to show an overview of the Logo-DM system – a dedicate data mining system, that aims to optimize the personalized therapy of Romanian children suffering from dyslalia.

This system is designed so that useful patterns can be easily discovered by speech therapists. They may use Logo-DM to analyze datasets obtained by integrating data collected in all speech therapy offices that use TERAPERS.

In Section II, some basic concepts related to the Knowledge Discovery in Databases and the position occupied by data mining stage within this process are presented. Section III refers to speech disorders and their implications on the individual's development. It highlights also the complexity of speech therapy. Section IV makes a brief description of the Logo-DM system. Finally, Section V contains some conclusion and future work.

II. KNOWLEDGE DISCOVERY IN DATABASES PROCESS AND DATA MINING

Knowledge Discovery in Databases concept was developed as a result of the emergence of very large volumes of data, whose analysis was not possible by using traditional database techniques. It aims to identify "valid, novel, potentially useful, and understandable patterns in data" [4], and is a complex, interactive and iterative process.

In time, many models for this process have been proposed. No matter if they originated from academia [4], [5], [6] or industry [7], [8] they consist of a succession of steps, that start from the understanding the domain and the data that is represented, continues with the preparation of data for data mining algorithms and their effective implementation in order to detect existing patterns, and ends with the interpretation of these patterns.

We used for our system design and implementation CRISP-DM model, presented in Fig. 1.

It starts with a business analysis for determining the KDD goals and continues with a data understanding stage that aims to collect and describe data and to verify data quality.

In order to give data set the proper format for a certain data mining algorithm, a data preparation step is necessary. During this step data are filtered and the relevant features are selected. Data type transformation, discretization or sampling is performed also.

The central point of KDD process is the data mining stage. Data mining performs analysis of large volumes of data using specific algorithms. These are designed to offer good performances of calculation on large amounts of data, and produce a particular enumeration of patterns from such data. Using patterns or rules with a specific meaning, data mining may facilitate the discovery, from apparently unrelated data, of relationships that are likely to anticipate future problems or might solve the problems under study. It involves the choice of the appropriate data mining task, and, taking into account specific conditions, the choice and the implementation of the proper data mining algorithm.

For the next stage, the mined models are evaluated against the goals defined in the first stage. The last stage of the process uses the knowledge discovered in order to simply generate a report or to deploy a repeatable data mining process.

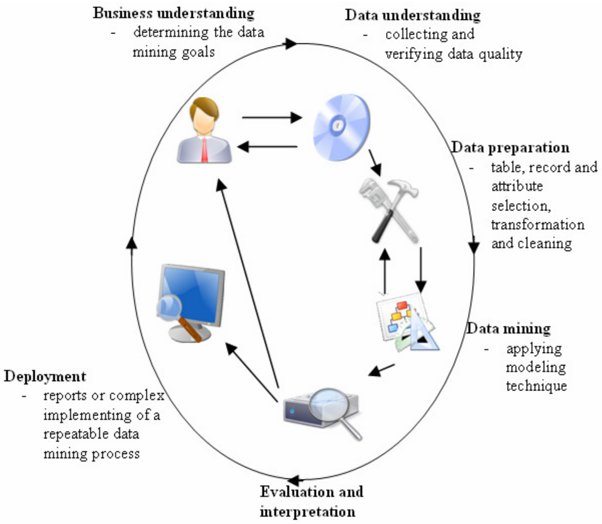

Business understanding
- determining the data mining goals
Data understanding
- collecting and verifying data quality
Data preparation
- table, record and attribute selection, transformation and cleaning
Data mining
- applying modeling technique
Evaluation and interpretation
Deployment
- reports or complex implementing of a repeatable data mining process

Figure 1. CRISP-DM model for KDD

So, data mining aims to discover, from apparently unrelated data, relationships that can anticipate future problems or might solve the problems under study. Using appropriate methods, data mining is able to provide answers for two wide categories of problems: prediction and description.

Although, many researchers believe that prediction is the main use of the patterns discovered by data mining, it should be noted that often it is preceded by the description. For example, prior to predict the state of a patient at certain moments, it is necessary to make a description of the profiles encountered and to find the best association between these profiles and different therapy schemes.

Both problems require the use of appropriate techniques.

Classification aims to find a model, which places data items in one of several predefined classes, based on information from a set of predictive variables.

Association rules explore relationships or affinities between different data that seem to have no dependencies.

III. Speech disorders and speech therapy

Speech and language impairments address problems in communication and related areas, such as oral motor function [9]. A speech disorder is a problem with fluency, voice, and/or how a person says speech sounds.

Classification of speech into normal and disorder is a complex task. Statistics points out that only 5% to 10% of the population has a completely normal manner of speaking, all others suffer from one disorder or another.

The most common speech disorders are: stuttering, cluttering, voice disorders, dysartria, and speech sound disorders.

Dyslalia is defined as the articulation disorder that consists of difficulties in the way sounds are formed and strung together. The most encountered problems are characterized by omitting, distorting a sound or substituting one sound for another.

Dyslalia has the greatest frequency among handicaps of language for psychological normal subjects as well as for those with deficiencies of intellect and sensory. Thus, the opinion of Sheridan (1946) is that at the age of eight years dyslalia are in proportion of 15% for girls and in proportion of 16% for boys. In this context, a lot of attention is paid to its prevention and treatment.

In order to obtain the desired results, speech disorder therapy should begin as soon as possible. More studies have demonstrated that if children are enrolled in therapy early in their development, this means younger than 5 years, their outcomes are better than those who begin therapy later.

Differential diagnosis decides upon the therapy for correcting language, as psycho diagnosis allows an adequate therapeutic program, and the elaboration of a prognosis regarding the evolution of the child, along with the therapeutic process. The therapy has to be adapted to each language therapist, to each particular case, to the child’s learning rhythm and style, as well as to the level of the impairment. The key issues in dyslalia therapy are shown in Fig. 2.

Modern speech therapy was deeply influenced by the use of Information and Communication Technology (ICT).

On the one hand, the use of computers and other communication tools facilitated communication among persons with speech disorders. On the other hand, computers were used in speech therapy, contributing to the acquisition of written and verbal language, helped by various computer-based programs and software.

Analyzing how it is possible to use the computer to support therapy, experts have concluded that it can contribute to the diagnosis of speech disorder, produces audiovisual feedback during the treatment, monitors and assesses the therapeutic progress and provides various types of practical exercises for children with speech disorders.

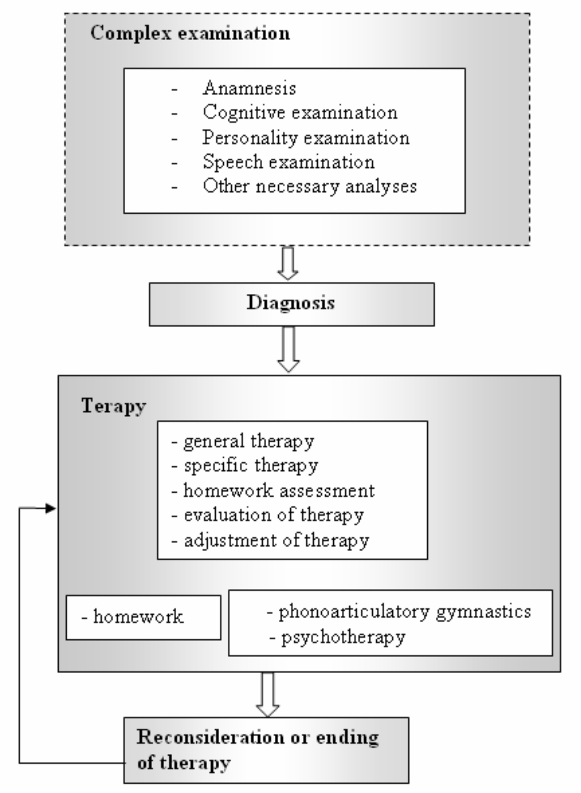


Figure 2. Key issues in dyslalia therapy

Additionally, the use of a CBST allows to collect and store a considerable amount of data about patients, diagnoses and treatment schemes.

The Center for Computer Research in the University "Stefan cel Mare" of Suceava has implemented the TERAPERS project [10]. TERAPERS is a system able to assist teachers in speech therapy of dislalya and to track how the patients respond to various personalized therapy programs. It contains two main components, as shown in Fig. 3: intelligent systems deployed on computers located in the speech therapists' offices and more mobile devices used by patients in order to solve the independent homework.

This system is currently used by the therapists from Regional Speech Therapy Center of Suceava.

During the operating period, data about 300 children were collected and stored in the TERAPERS' database, which includes about 60 tables and several hundreds of features.

Anamnesis data collected may provide information relative to various causes that may negatively influence the normal development of the language. It contains historical data and data provided by the cognitive and personality examination.

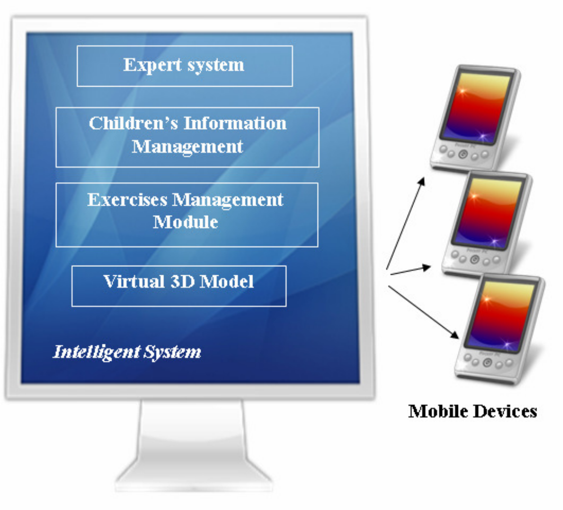


Figure 3. TERAPERS' Architecture [11]

In order to design personalized therapy programs, is useful to know: how many sessions per week are necessary for each child, exercises that are suitable for each phase of therapy, and how can be changed the original program in order to be adapted to the patient evolution. In addition, the report downloaded from the mobile device collects data on the efforts of child self-employment. These data refer to the exercises done, the number of repetitious for each of these exercises and the results obtained. The tracking of child's progress materializes data that indicate the moment of assessing the child and his status at that time.

All these data can hide useful patterns, which are very useful in personalizing speech therapy, and that could be detected using appropriate data mining techniques.

Clustering may group people with speech disorders on the basis of similarity of different features and allows creating some patients' profiles. This is a way to help the therapists to understand who they patients are.

Classification places people with different speech impairments in predefined classes. Based on the information contained in many predictor variables, such as personal or familial anamnesis data or related to lifestyle, it can be used to join the patients with different segments and to track the size and structure of various groups.

The goal of association rules mining is to identify combinations of items that often occur together. In the personalized speech therapy area, its task is to determine why a specific therapy program has been successful on a segment of patients with speech disorders and on the other was ineffective.

Those mentioned above, were the basis of the initiative to develop a data mining system, dedicated to support efforts to better personalize the speech therapy.

IV. LOGO-DM - A DATA MINING SYSTEM FOR SPEECH THERAPY OPTIMIZATION

A. System Objectives

The sustainable development, in which special attention is given to all aspects of health care and the need to respond to the high efficiency requirements have led to the need for handling information, such as [12]: "what is the predicted final state for a child or what will be his/her state at the end of various stages of therapy, which are the best exercises for each case, and how patients can focus on their effort to effectively solve these exercises, or how the family receptivity - that is an important factor in the success of the therapy - is associated with other aspects of family and personal anamnesis". For all of these, the answer may be obtained by applying data mining techniques on data collected by TERAPERS.

It is also interesting to try to enrich the knowledge base of expert system embedded in TERAPERS, with knowledge discovered in data mining process. In order to achieve these goals, we have proposed the development of Logo-DM system.

Essentially, its objectives aim to perform an analysis of available data collected from children assisted by TERAPERS system and to prepare them in order to assure a proper quality for data mining algorithms, to try to select only those features that contribute to the model building by removing those that are irrelevant or redundant, to choose the most appropriate methods and algorithms for data mining, to find models that can help to solve problems raised in speech disorders therapy, and to validate these models on new cases.

It is worth mentioning that the patterns, represented as rules, provided by Logo-DM, could, after some post-processing operations, be used to enrich the knowledge base of the embedded expert system in TERAPERS.

Although, market claims many systems that allow data mining implementation, such as Weka and RapidMiner, their use implies IT skills. Our system is designed so that patterns can be easily discovered by speech therapists. They process a real dataset obtained by integrating data collected by all speech therapists that use TERAPERS.

B. General Architecture

The proposed architecture for Logo-DM system is presented in Fig. 4.

The graphical user interface allows the successive operations required by the knowledge discovery process.

The preprocessing module prepares data for data mining algorithms and performs data transformation and feature selection for patterns building. These operations can be made both in centralized, distributed, or parallel ways.

In order to achieve the proposed goals, the data mining kernel performs classification and association rules mining. Finally, the extracted models are evaluated by experts. If they meet the requirements of novelty and utility, they are considered knowledge.

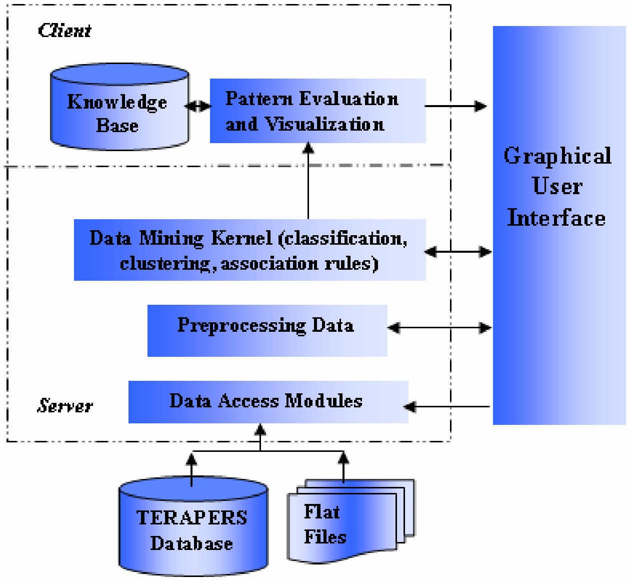


Figure 4. Logo_DM Architecture

C. Sistem implementation

As previously mentioned, the graphical interface connects the system with the speech therapist, which is able to control the KDD process. The access can be achieved through menu options or, for the most common tasks, via shortcuts, as presented in Fig. 5. It should be noted that the following figures contain Romanian texts, because they are screenshots from Logo-DM, which is implemented in Romanian.

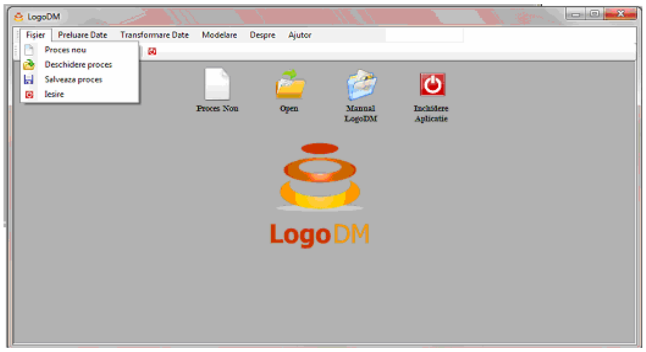
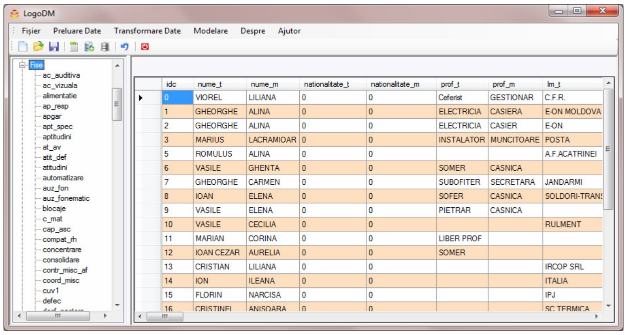


Figure 5. Logo-DM- Main page

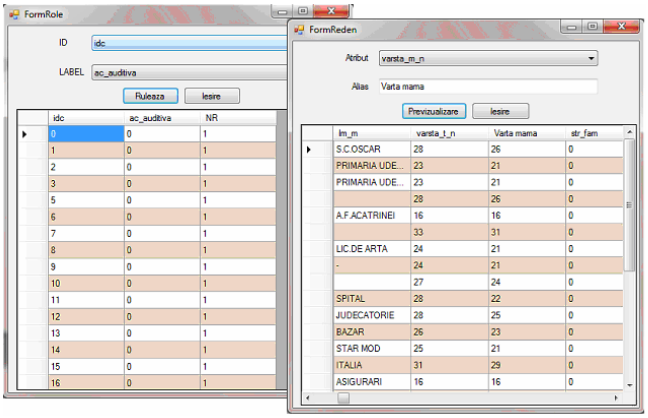
The main source of data for Logo-DM is the database from TERAPERS, which is implemented in Access. In these conditions, modules for data acquisition from Access were implemented, as shown in Fig. 6. If other data sources are required, data can be retrieved from Excel, if they were previously converted in this format.



idc	nume_t	nume_m	nationalitate_t	nationalitate_m	prof_t	prof_m	lm_t
0	VIOREL	LILIANA	0	0	Ceferist	GESTIONAR	C.F.R.
1	GHEORGHE	ALINA	0	0	ELECTRICIA	CASIERA	E-ON MOLDOVA
2	GHEORGHE	ALINA	0	0	ELECTRICIA	CASIER	E-ON
3	MARIUS	LACRAMIOAR	0	0	INSTALATOR	MUNCITOARE	POSTA
5	ROMULUS	ALINA	0	0			A.F.ACATRINEI
6	VASILE	GHENTA	0	0	SOMER	CASNICA	
7	GHEORGHE	CARMEN	0	0	SUBOFITER	SECRETARA	JANDARMI
8	IOAN	ELENA	0	0	SOFER	CASNICA	SOLDORI-TRANS
9	VASILE	ELENA	0	0	PIETRAR	CASNICA	
10	VASILE	CECILIA	0	0			RULMENT
11	MARIAN	CORINA	0	0	LIBER PROF		
12	IOAN CEZAR	AURELIA	0	0	SOMER		
13	CRISTIAN	LILIANA	0	0			IRCOP SRL
14	ION	ILEANA	0	0			ITALIA
15	FLORIN	NARCISA	0	0			IPJ
16	CRISTINEL	ANISOARA	0	0			SC TERMICA

Figure 6. Logo-DM – Data aquisition

In order to implement the data transformation, we have considered operation, such as: data type conversion, data discretization, or role setting as shown in Fig. 7.



ID: idc

LABEL: ac_auditiva

idc	ac_auditiva	NR
0	0	1
1	0	1
2	0	1
3	0	1
5	0	1
6	0	1
7	0	1
8	0	1
9	0	1
10	0	1
11	0	1
12	0	1
13	0	1
14	0	1
15	0	1
16	0	1

Atribut: varsta_m_n

Alias: Varta mama

lm_m	varsta_t_n	Varta mama	str_fam
S.C.OSCAR	28	26	0
PRIMARIA UDE...	23	21	0
PRIMARIA UDE...	23	21	0
	28	26	0
A.F.ACATRINEI	16	16	0
	33	31	0
LIC.DE ARTA	24	21	0
-	24	21	0
	27	24	0
SPITAL	28	22	0
JUDECATORIE	28	25	0
BAZAR	26	23	0
STAR MOD	25	21	0
ITALIA	31	29	0
ASIGURARI	16	16	0

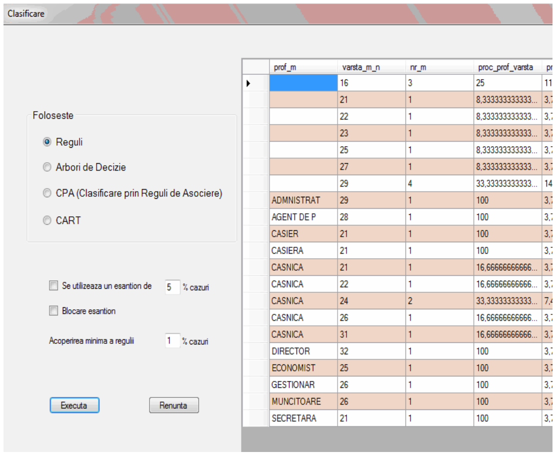
Figure 7. Logo-DM – Data transformatiom

For the first version of the system, we have considered, as modeling techniques, classification, and association rules mining.

For convenience of operating, interfaces have been dedicated exclusively to methods that, after some tests made during the implementation, have been shown to provide the best performance over a sample of real data set.

As shown in Fig. 8, for classification were considered: rules-based classification, decision trees, classification by association rules (CBA) and the CART algorithm.

For association rules, we used an implementation of the Apriori algorithm to detect frequent itemsets from which, subsequently, we build association rules. Obviously, it is possible to adjust the values for support and confidence. It is also possible to choose, from a set of items, those that will be contained in the rules or will be placed in the rule's consequent. Fig. 9 shows the interface that allows all these operations.



Foloseste

● Reguli

○ Arbori de Decizie

○ CPA (Clasificare prin Reguli de Asociere)

○ CART

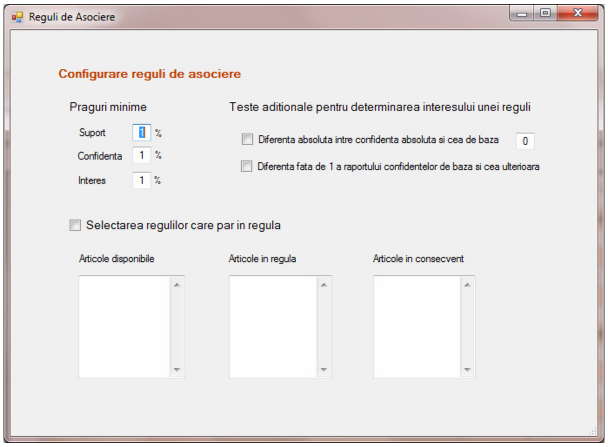
□ Se utilizeaza un esantion de 5 % cazuri

□ Blocare esantion

Acoperirea minima a regulii 1 % cazuri

prof_m	varsta_m_n	nr_m	proc_prof_varsta	pr
	16	3	25	11
	21	1	8,333333333333...	3,7
	22	1	8,333333333333...	3,7
	23	1	8,333333333333...	3,7
	25	1	8,333333333333...	3,7
	27	1	8,333333333333...	3,7
	29	4	33,33333333333...	14
ADMNISTRAT	29	1	100	3,7
AGENT DE P	28	1	100	3,7
CASIER	21	1	100	3,7
CASIERA	21	1	100	3,7
CASNICA	21	1	16,66666666666...	3,7
CASNICA	22	1	16,66666666666...	3,7
CASNICA	24	2	33,33333333333...	7,4
CASNICA	26	1	16,66666666666...	3,7
CASNICA	31	1	16,66666666666...	3,7
DIRECTOR	32	1	100	3,7
ECONOMIST	25	1	100	3,7
GESTIONAR	26	1	100	3,7
MUNCITOARE	26	1	100	3,7
SECRETARA	21	1	100	3,7

Figure 8. Logo-DM – Classification interface



Configurare reguli de asociere

Praguri minime

Suport 1 %

Confidenta 1 %

Interes 1 %

Teste aditionale pentru determinarea interesului unei reguli

□ Diferenta absoluta intre confidenta absoluta si cea de baza 0

□ Diferenta fata de 1 a raportului confidentelor de baza si cea ulterioara

□ Selectarea regulilor care par in regula

Figure 9. Logo-DM – Association rules mining interface

D. *Experimental Results*

In present, we are at the stage where, although the amount of data is still low for data mining, the system can be tested on the available data.

At this point, we have considered a real data set containing more than 300 cases described by 102 features related to personal and familial anamnesis and to complex speech examination, expresed in Romanian terms. After the feature selection process achieved through an unsupervised information-based method [13], we have obtained a set of 52 relevant features.

Fig. 10 shows a decision tree built on this data set. It is a classification model in which class label is represented by the diagnosis. The calculated model accuracy, obtained using a test dataset that is about 10% from the training set is 56,67%, as shown in Fig. 11.

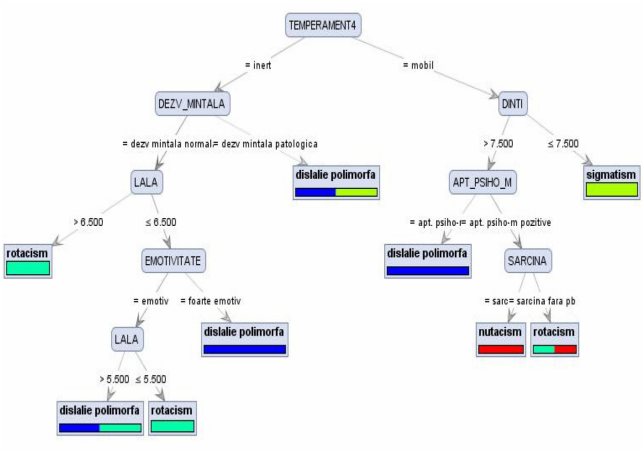
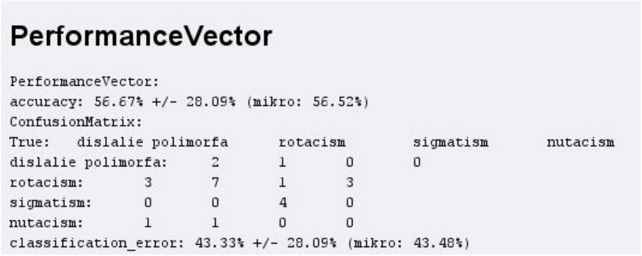


Figure 10. A classification model



PerformanceVector

```
PerformanceVector:
accuracy: 56.67% +/- 28.09% (mikro: 56.52%)
ConfusionMatrix:
True:   dislalie polimorfa      rotacism        sigmatism       nutacism
dislalie polimorfa:     2       1       0       0
rotacism:       3       7       1       3
sigmatism:      0       0       4       0
nutacism:       1       1       0       0
classification_error: 43.33% +/- 28.09% (mikro: 43.48%)
```

Figure 11. Classification performance

Speech therapy experts have analyzed this model and admitted that in addition to the complex examination of language, there are features related to the child's evolution or temperament, such as excitement ("emotivitate") that may lead to a specific diagnosis, such as :"dislalie polimorfa", or "rotacism", or "stigmatism", as shown in pattern presented above.

V. Conclusion and Future Work

Speech therapy is a complex process that must be personalized according to the characteristics of each patient, especially since they are in very high proportion children.

The use of information technology in order to assist the speech therapy has some immediate benefits and allows the collection of a considerable amount of data related to personal and familial anamnesis, to the complex evaluation, to the therapeutically applied schemes and to the results of the various stages of therapy.

Studies have shown that it should be appropriate to apply some data mining methods on these data, such as classification, clustering, and association rules, because they lead to patterns that can increase the efficiency of speech therapy.

In this paper, we have presented the first version of a data mining system, called Logo-DM that was implemented on data collected by TERAPERS system. Its aim is to try to increase the efficiency of therapy of dyslalia that is assisted by TERAPERS.

Furthermore, we have proposed to finish testing the association rules mining, and to complete the data mining kernel with clustering algorithms.

References

[1] SpeechviewerIII– http://www.synapseadaptive.com/edmark/prod/sv3 [retrieved: June 2013] .

[2] R. Laboissière, D. J. Ostry, and A. G. Feldman, "The control of multi-muscle systems: Human jaw and hyoid movements", Biological Cybernetics, vol.74, 1996, pp. 373-384.

[3] http://www.speech.kth.se/multimodal/ARTUR/index.html [retrieved: May, 2013].

[4] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data", Communications of the ACM, vol. 39(11), 1996, pp. 27-34.

[5] S. Anand, A. Patrick, J. Hughes, and D. Bell, "A data mining methodology for crosssales", Knowledge Based Systems Journal. Vol. 10, 1998, 449–461.

[6] K. Cios, A. Teresinska, S. Konieczna, J. Potocka, and S. Sharma, "Diagnosing myocardial perfusion from SPECT bull's-eye maps – a knowledge discovery approach", IEEE Engineering in Medicine and Biology Magazine, special issue on Medical Data Mining and Knowledge Discovery, vol. 19(4), 2000, pp. 17–25.

[7] P. Cabena, P. Hadjinian, R. Stadler, J. Verhees, and A. Zanasi, "Discovering Data Mining: From Concepts to Implementation", Prentice Hall, 1998.

[8] R. Wirth and J. Hipp, "CRIPS-DM: Towards a standard process model for data mining", In Proceedings of the Fourth International Conference on the Practical Applications of Knowledge Discovery and Data Mining, Manchester, 2000, pp. 29–39.

[9] A. Brice, "Children with communication disorders", Arlington, VA: ERIC Clearinghouse on Disabilities and Gifted Education, 2001.

[10] M. Danubianu, S. G. Pentiuc, O. Schipor, M. Nestor, and I. Ungurean, "Distributed Intelligent System for Personalized Therapy of Speech Disorders", Proceedings of The Third International Multi-Conference on Computing in the Global Information Technology ICCGI, August 2008, pp. 166-170.

[11] M. Danubianu, I. Tobolcea, and S. G. Pentiuc, "Advanced Technology in Speech Disorder Therapy of Romanian Language", Journal of Computing, NY, ISSN: 2151-9617, Vol: 1, no.2009, pp. 1-6.

[12] M. Danubianu, S.G. Pentiuc, I. Tobolcea, and T. Socaciu, "Model of a data mining system for personalized therapy of speech disorder", Journal of Applied Computer Science, no. 6(3), 2009, pp. 28-32.

[13] M. Danubianu, V. Popa, and I. Tobolcea, "Unsupervised Information-Based Feature Selection for Speech Therapy Optimization by Data Mining Techniques", Proceedings of ICCGI2012, June 2012, pp. 206-211.

Arabic Meaning Extraction through Lexical Resources:

A General-Purpose Data Mining Model for Arabic Texts

Giuliano Lancioni, Ivana Pepe, Alessandra Silighini,
Valeria Pettinari
Department of Foreign Languages, Literature and
Civilizations, Roma Tre University
Rome, Italy
giuliano.lancioni@uniroma3.it, iva.pepe@stud.uniroma3.it,
ale.silighini@gmail.com, val.pettinari2@stud.uniroma3.it

Ilaria Cicola
EPHE
Paris, France
ilaria.cicola@etu.ephe.fr
Department Italian Institute of Oriental Studies, Sapienza
University of Rome
Rome, Italy

Leila Benassi, Marta Campanelli
Department Italian Institute of Oriental Studies, Sapienza University of Rome
Rome, Italy
benassileila@gmail.com, marta.campanelli@hotmail.it

***Abstract*— A general-purpose data mining model for Arabic texts (Arabic Meaning Extraction through Lexical Resources, ArMExLeR) is proposed which employs a chained pipeline of existing public domain and published lexical resources (Stanford Parser, WordNet, Arabic WordNet, SUMO, AraMorph, A Frequency Dictionary of Arabic) in order to extract a weakly hierarchised, single-predicate level, representation of meaning. This kind of model would be of high impact on the study of the computational analysis of Arabic for there is no such comparable tool for this language, and will be a challenge for the nature of its specificities. One should, in fact, cope with the unique writing system that is mostly consonant-based and doesn't always mark vowels explicitly. This is crucial when you want to analyze an Arabic corpus for the same consonantal ductus may be read in several ways.**

Keywords-Arabic data mining; content extraction; automatic parsing techniques; ontologies.

I. INTRODUCTION[*]

Data mining from Arabic texts presently suffers a series of shortcomings, some related to the specificities of Arabic texts and writing system [12, 13], some deriving from the scarcity, or plain lack, of lexical resources for Arabic analogous for what can be found for other languages [14].

Other tools routinely used as helpers in data mining cannot be successfully employed in analyzing Arabic texts as well, partly for these very reasons: e.g., statistical Machine Translation (MT) tools generally perform poorly for Arabic, both for the paucity of parallel texts and text memories and for the specificities of the language (the only other Semitic language with a reasonable amount of written texts available in electronic form, Modern Hebrew, for historical reasons has become closer to Indo-European languages in both lexicon and syntax) [15].

To overcome these difficulties, our project capitalizes on the use of existing Arabic lexical resources that are linked to larger, general-purpose resources, by devising specific strategies to fill the gaps in these resources. Resources are aligned through a pipeline which is fed by the input text and outputs, after several rings in the chain, a relatively hollow semantic representation that allows for further data mining operations, thanks to the Suggested Upper Merged Ontology (SUMO) format.

Next sections will discuss [II] the tools used in the project, [III] the workflow of the system, [IV] an example derivation, [V] test results and [VI] some conclusions and suggestions for further developments.

II. TOOLS

The ArMExLeR project employs a variety of tools. Some of them are shortly described in this section before tackling their role within the system.

A. *Stanford Parser*

The Stanford Parser is a statistical parser that is programmed in order to find the grammatical structure of the sentences. It analyses a text, parsing the phrases (constituency parser) and then finds out the Verb and then its Subject or Object (dependency parser). It is a probabilistic parser which uses knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Statistical parsers still make some mistakes, but their advantage is that they always give an answer that could be later corrected by a human. The lexicalized probabilistic parser implements a factored product model, with separate Probabilistic Context-Free Grammar (PCFG) phrase structure and lexical dependency experts, whose preferences are combined by efficient, exact inference, using an A* algorithm. Or the software can be used simply as an accurate unlexicalized stochastic context-free grammar

[*] All authors have contributed equally to this work , but since it refers to a modular project, Lancioni should be mainly credited for secs. 3 and 5, Pepe for sec. 2B, Silighini for sec. 4, Pettinari for sec. 2C, Cicola for secs. 1 and 2A, Benassi for sec. 6, Campanelli for sec. 2D.

parser and either of these yields a good performance statistical parsing system. As well as providing an English parser, the Stanford parser can be and has been adapted to work with other languages such as Chinese (based on the Chinese Treebank), German (based on the Negra corpus) and Arabic (based on the Penn Arabic Treebank). Finally, this parser has also been used for other languages, such as Italian, Bulgarian, and Portuguese.

Although the parser provides Stanford Dependencies output as well as phrase structure trees for English and Chinese, this component has to be implemented for Arabic. So, for now, we have an analysis of the sentences that cannot trace the subject or the object of a verb in a sentence, but we have a reliable parsing of constituents that could be used to deepen the analysis with other tools.

The Arabic Parser from Stanford can only cope with non-vocalized texts, the tokenization being based on the Arabic used in the Penn Arabic Treebank (ATB) and is based on a whitespace tokenizer. Segmentation is done based on the Buckwalter analyzer (morphology).

The character encoding is set on Universal Character Set (UCS) Transformation Format—8-bit (UTF-8), but it may be changed if needed. The normalization of the text is needed in order to analyze the text, because otherwise the parser cannot recognize the Arabic ductus, that is the representation of consonants and long vowels which are the only obligatory component of Arabic script, and often the only one actually present in texts (auxiliary graphemes, such as short vowels and consonant reduplication markers, must be deleted). For the part-of-speech (POS) tags, the parser uses an “augmented Bies” tag set that uses the Buckwalter morphological analyzer and links it to a subset of the POS tags used in the Penn English Treebank (sometimes with slightly different meanings) [1]. Phrasal categories are the same from the Penn ATB Guidelines [16]. As mentioned, there is no tool in the parser itself that can normalize or segment an Arabic ductus, so one is compelled to employ other tools in order to perform these tasks.

B. *AraMorph*

AraMorph [2] is a program designed to allow the morphological analysis for Arabic texts in order to segment Arabic words in prefixes, stems and suffixes according to the following rules:

- the prefix can be 0 to 4 characters in length;
- the stem can be 1 to infinite characters in length;
- the suffix can be 0 to 6 characters in length.

Each possible segmentation is verified by asking the software to check if the prefix, the stem and the suffix exist in the embedded dictionary. In fact, the program has three tables containing all Arabic prefixes, all Arabic stems and all Arabic suffixes, respectively. Indeed, if all three components are found in these tables, the program checks if their morphological categories are listed as compatible pairs in three tables (table AB for prefix and stem compatibility; table AC for prefix and suffix compatibility; table BC for stem and suffix compatibility). Finally, if all three pairs are found in their respective tables, the three components are defined suitable and the word is confirmed as valid.

Hereafter (Fig. 1), we put in evidence an example of an Arabic word and analyzed by AraMorph:

```
WORD NO. 10223:     الثوري     17 occurrences

UNVOCALIZED TRANSCRIPTION:  Al+vwry+
INPUT STRING:                   الثوري
SOLUTION:                Al+vaworiy~+    ال*ثَوْرِيّ*
     vaworiy~_1          [ثور]
ENGLISH GLOSS:                  the+revolutionary+

POS ANALYSIS:
     Al/DET+vaworiy~/ADJ+
```

Figure 1. Excerpt of an AraMorph analysis of Meedan Memory

However, AraMorph presents some problems regarding the analysis of texts types that do not match to ideal text genre targeted by Buckwalter (newspaper texts and other Modern Standard Arabic non-literary texts). Indeed, the program shows three major weaknesses:

- it does not either fully or sparsely analyze vocalized texts;
- it does reject many words attested in some textual types which are not contained either in the sample of the text corpora chosen by Buckwalter, or in the lookup lists of AraMorph;
- there is neither any stylistic nor chronological information in the lookup lists; the same for a lot of transliterated foreign named entities which cannot be found in classical texts and in modern literary texts, giving rise to a number of false positives.

In order to overcome these problems, a group of researchers on linguistics at Roma Tre University had modified the original AraMorph in a new algorithm named “Revised AraMorph” (RAM), within a project of automatically analysis of ḥadīṯ corpora (SALAH project) [7]. The modifications, which are implemented to solve the weakness previously listed, are respectively:

- a mechanism which takes into account the vowels present in the text in order to reduce ambiguity linked to non-vocalized texts;
- a file with additional stems automatically extracted from Anthony Salomé Arabic-English dictionary (a work from the end of 19th century encoded in TEI-compliant XLM format) [11] and with additional lists of prefixes and suffixes with the relative combination tables of most frequent unrecognized tokens;
- a mechanism which removes automatically items in order to allow them matching to contemporary foreign named entities, especially proper names and place-names. In the other hand, the items above are not included in Salomé’s dictionary (this way Arabic named entities which can be found in Classical texts are retained for the analysis).

C. A Frequency Dictionary of Arabic

Starting from the analysis of a 30-million-word corpus, Buckwalter and Parkinson's Frequency Dictionary of Arabic (FDA) [3] lists 5,000 most frequent Arabic words from Modern Standard Arabic as well as most important words from Egyptian, Levantine, Iraqi, Gulf and Algerian dialects. Each entry in the dictionary is organized as follows: headword and English translation(s), a sample sentence or context, English translation of the sample sentence or context and statistical information. The latter represents information about word dispersion figure and raw or absolute frequency, i.e., all the variants and inflected forms belonging to a specific lemma considered as an entry. The dictionary also provides important information about morphology, syntax, phonetics and orthography as well as usage restrictions and register variation.

Word ranking proceeds according to the value of a final adjusted frequency which is produced by multiplying word frequency and dispersion figure. Finally, the rank-order goes from the high-scoring lemma to the lower-scoring one.

An example of how an entry is generally arranged (information follows FDA's definitions) is in Fig. 2:

RANK FREQUENCY: 3835
HEADWORD: وَصْفَة
PART OF SPEECH: n.
ENGLISH EQUIVALENT: description, portrayal; (Medical) prescription; (Food) recipe
SAMPLE SENTENCE: كتب الطبيب المناوب لكل واحد منهم وصفة طبية
ENGLISH TRANSLATION: The doctor on duty wrote a medical prescription for each one of them
RANGE COUNT: 62
RAW FREQUENCY TOTAL: 434

Figure 2. Example of an FDA entry

D. Arabic WordNet

Arabic WordNet (AWN) is a lexical resource for Modern Standard Arabic based on the widely used Princeton WordNet (PWN) for English [5]. There is a straightforward mapping between word senses in Arabic and those in PWN, thus enabling translation to English on the lexical level. Each concept is also provided with a deep semantic underpinning, since, besides the standard Wordnet representation of senses, word meanings are defined according to SUMO.

However, AWN represents only a core lexicon of Arabic, since it has been built starting from a set of base concepts. In this sense, being the mapping with PWN relatively poor, this project uses in fact an AWN augmented model (AAWN) which extends this core WordNet downward to more specific concepts using lexical resources such as Arabic Wikipedia (AWp) and Arabic Wiktionary (AWk).

Wikipedia is by far the largest encyclopedia in existence with more than 4 million articles in its English version (English Wikipedia) contributed by thousands of volunteers and experimenting an exponential growing in size.

Arabic Wikipedia has over 224,000 articles. It is currently the 23rd largest edition of Wikipedia by article count and the first Semitic language to exceed 100,000 articles. The growing of Arabic Wikipedia is, however, very high so it seems that in a relatively short time the size of Arabic Wikipedia could correlate with the importance (of the number of speakers) of Arabic language. Wikipedia basic information unit is the "Article" (or "Page"). Articles are linked to other articles in the same language by means of "Article links". Wikipedia pages can contain "External links", that point to external URLs, and "Interwiki links", from an article to a presumably equivalent article in another language. There are in Wikipedia several types of special pages relevant to our work: "Redirect pages", i.e., short pages which often provide equivalent names for an entity, and "Disambiguation pages", i.e., pages with little content that links to multiple similarly named articles. A significant category of specific (non-ambiguous) concepts that can be drawn from Arabic Wikipedia in order to enrich AWN is that of Named Entities (locations, persons, organizations, etc.) that, once extracted from the mentioned resource, will be attached to existing Named Entities in PWN. In this operation, an important role is played by the "interwiki links" between Arabic and English Wikipedia, as shown in Fig. 3.

On the other hand, Wiktionary is a collaborative project to produce a free-content multilingual dictionary. It aims to describe all words of all languages using definitions and descriptions in English. It is available in 158 languages and in Simple English.

Designed as the lexical companion to Wikipedia, Wiktionary has grown beyond a standard dictionary and now includes a thesaurus, a rhyme guide, phrase books, language statistics and extensive appendices. It aims to include not only the definition of a word, but also enough information to really understand it. Thus etymologies, pronunciations, sample quotations, synonyms, antonyms and translations are included.

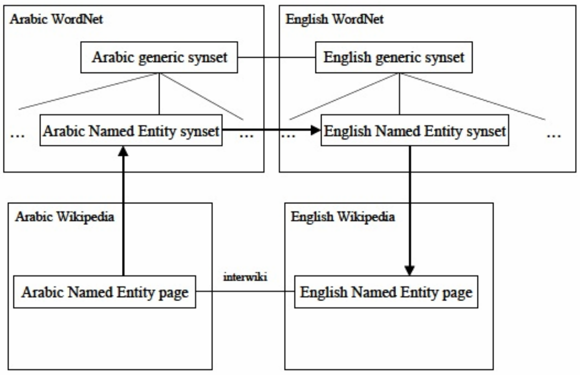


Figure 3. Relations between Arabic and English version of WordNet and Wikipedia

Wiktionary has semi-structured data. Its lexicographic data should be converted to machine-readable format in order to be used in natural language processing tasks.

Wiktionary data mining is a complex task. There are the following difficulties: the constant and frequent changes to data and schema, the heterogeneity in Wiktionary language edition schemas and the human-centric nature of a wiki.

III. WORKFLOW

Fig. 4 shows the general workflow of ArMExLeR.

The input to the system is Modern Standard Arabic written texts. The first analytical step is performed through the Stanford Word Segmenter (SWS) [4], which segments words into morphemes according to the ATB standard. SWS is not necessarily the best possible segmenter (e.g., it segments suffix pronouns, but not the article), but it is the best choice for the pipeline model, since it outputs an ATB-compliant segmentation, which is required by subsequent components.

The word-segmented input is submitted to the parsing component, Stanford Parser (SP), which statistically parses the input according to a factored model. Since, options for Arabic in SP are more limited than for English, it is not possible to get a dependency analysis, which would be more useful for content extraction. However, getting a standard parsing through the output of the (most probable) syntactic tree for an input sentence, is an invaluable contribution to a better semantic understanding of its element: e.g., identifying the subject and the object(s) of a (di)transitive verbs helps the system identify argument roles in relation to the verb - e.g., which argument is the agent and which the patient, - although linking of syntactic roles to argument roles is notoriously a nontrivial process.

An important role in the pipeline is played by the AraMorph component. The original AraMorph (AM) model is used by SP in order to lemmatize Arabic words; on the other hand, the RAM model [7] is used to select possible readings according to choices made by the syntactic parser.

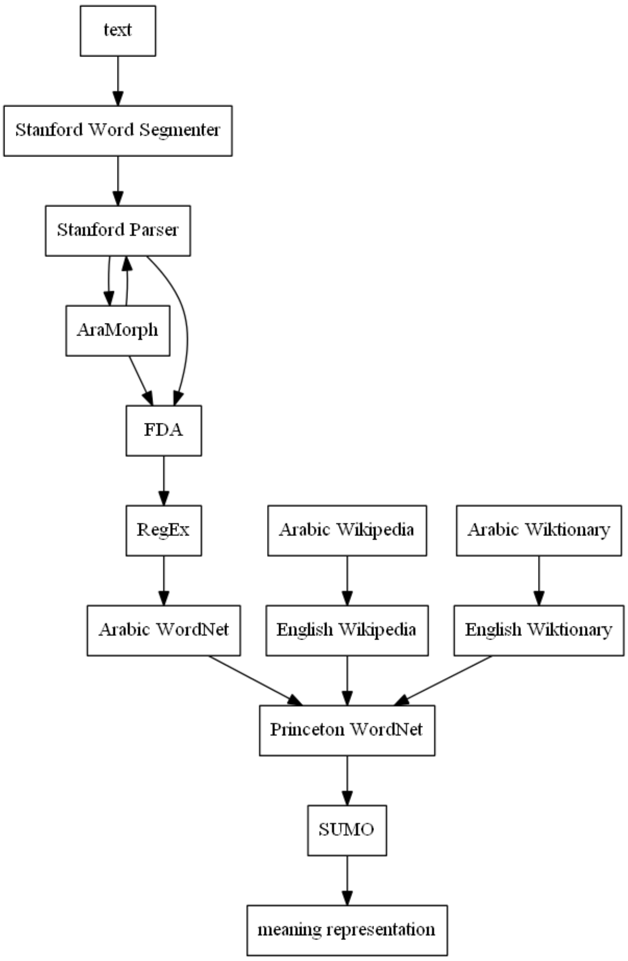


Figure 4. General workflow of ArMExLeR

In order to simplify the semantic linking task, in this step we worked on a subset of the analyses output by the parser, by filtering verb heads and the nominal heads of their arguments (plus possible introducing prepositions) through a regular expression component (RegEx).

The linking between RAM and FDA and between the latter and AWN realized by our research team is able to select the most probable reading and to link it to an AWN synset.

At this point, the system is fully within the semantic representation component: AWN is linked to the standard, Princeton WordNet 3.0 (PWN), which places the synset into a rich network of semantic relations. On its turn, PWN is entirely linked to SUMO, which allows the system to output a semantic representation in terms of ontologies, which can feed other components.

Since AWN is rather poor compared to the standard PWN, we use in fact an Augmented AWN model (AAWN) where AWN is supplemented by nonambiguous items drawn from AWp titles and sections, and AWk translations, linked to PWN by automatic linking [8, 9, 10]. This minimizes cases where a solution clearly exists, but it risks to be lost owing to limitations in AWN (which was designed to represent a core lexicon of Arabic only).

IV. An Example Derivation Process

The process can be better understood through an example. Let us start from one example (Fig. 5) drawn from the FDA corpus.

SENTENCE: نشرت الصحف قصيدة شوقى التى كتبها عن باريس بعد انتهاء الحرب الأولى

ENGLISH TRANSLATION: The newspapers published Shawqi's ode which he wrote about Paris after the end of World War I

Figure 5. Example sentence from the FDA corpus

The sample sentence is fed to SMS, which segments some of its graphic words into tokens (Fig. 6: tokens resulting from segmentation and other normalization steps are in boldface).

نشرت الصحف قصيدة شوقى التى **كتب ها** عن باريس بعد انتهاء الحرب **الاولى**

Figure 6. SMS tokenization of the sentence in Figure 5.

This segmented form of the text is input into the SP, which outputs (Fig. 7) a syntactic analysis.

The RegEx component extracts out of this syntactic tree the "core predications" (CPs: verb head and nominal heads of arguments), in order to simplify the generation of the semantic representation. This part of the system is clearly provisional, and it is likely to be widely improved in further development of the project. CPs extracted by the system are highlighted in light blue in the example.

The automatic WordNet-SUMO linking allows the system to immediately translate the PCs in terms of SUMO predicates.

Since SP has no dependency output available for Arabic — besides its general underperformance in dealing with Arabic texts compared to English ones,— such a parsing does not identify argument roles proper: e.g., in VSO clauses like the main clause in this example, we just have a sequence of NPs where nothing assures one of them is an agent, a patient, and so on. However, a general strategy that links roles output by this step to roles in entries for the relevant verbal concept in SUMO feeds back this step by assigning roles from the last to the first argument (owing to the general null subject property of Arabic).

CPs extracted by the RegEx component are linked to WordNet synsets through FDA (which selects the most frequent lemma in case of multiple possibilities) and AAWN. Synsets detected in the example are listed in Fig. 8.

نشرت = publish$_v$2(
    الصحف = ,
    قصيدة = poem$_n$1).

Figure 7. AAWN sysnets for an example entry.

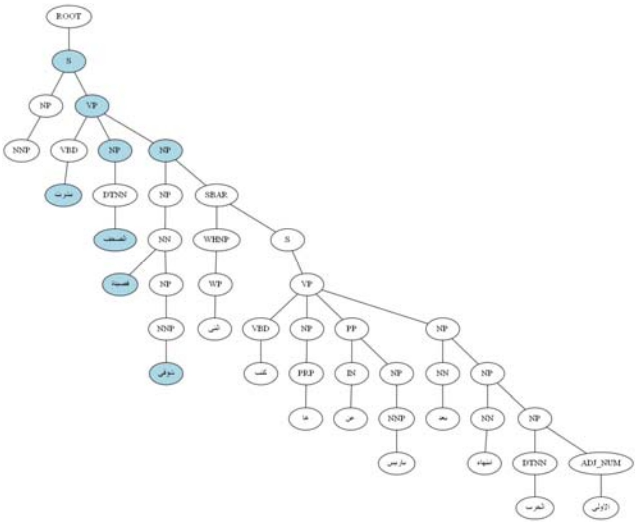


Figure 8. SP analysis of the sentence in Figure 5.

In this case, the result is (Fig. 9):

Publication (Corporation, Text).

Figure 9. SUMO representation for the example in Figure 7.

That is, the finer semantic relation has been transformed, and generalized, into a relation between SUMO concepts, which is expected to produce a better performance in data extraction.

V. Results

Since the system relies on a complex pipeline of components which are only partially under the control of our research team, it is difficult to establish the best evaluation strategy for the results of the project. We decided to separate the output of the segmenter and parser components —which are taken "out of the box" from SWS and, respectively, SP— from the output of other parts of the system.

First, the ArMExLeR has been run against the whole Meedan translation memory (~20,000 Arabic-English sentence couples downloadable from [17]). Running performances are relatively slow since optimization has not been a core concern in this stage of the project yet.

Then, 350 analyses have been randomly extracted and assigned to two different members of our research team each for a single run of evaluation. While the test corpus might seem small, the relative homogeneity of the Meedan translation memory makes it large enough for our purposes, without requiring too many resources during the testing stage.

In 96 cases (27.4%), the SWS/SP output was discarded because it was judged significantly wrong (e.g., because the main verb had been misinterpreted as a noun) by both evaluators. Of the remaining 254 cases, a further 58 (16.6%) were discarded because they did not contain any predication without anaphoric elements (which are not in the scope of the current model).

The evaluation of the system was performed on the remaining 196 cases (56% of the original sample). The analysis was deemed correct if the verb and at least one other PCs were regarded as properly assigned by at least one of the evaluators. This choice was motivated by the inherent problem in role-assignment caused by the lack of a dependency module for Arabic in SP (while such a module is available for English): therefore, the list of arguments and their relative order is not yet reliable. Only one agreement was deemed sufficient because a relatively high degree of disagreement between annotators has always been noticed for WordNet-related semantic projects (such as SemCor).

Results are summarized in Table I:

TABLE I. RESULT SUMMARY

error rate	60.54%
precision	64.90%
recall	74.56%
F measure	1.39

Comparing these results with other systems is not easy, since the ArMExLeR system evaluation applies to a specific subset of relations at the end of a relatively complex, automatic pipeline, which is not the case for other systems in Arabic text data mining. Therefore, we shall defer cross-comparison of our system to further research.

VI. CONCLUSIONS AND FURTHER DEVELOPMENTS

The ArMExLeR project shows a number of interesting features, which pave the way to further refinements and developments.

First, the system performs reasonably well, despite some shortcomings in some of the elements of the pipeline, which shows the feasibility of a predominantly symbolic, rather than purely statistic, approach to content extraction, especially in the case of a morphologically complex language such as Arabic.

Second, a partial syntactic analysis reveals itself to be sufficient to extract a reasonable amount of information from corpus texts. This is encouraging, since it is expected that a fuller match between syntax and semantics (especially when a fuller argument extraction component is developed, which includes nominalization, a highly prominent feature in Arabic texts) can bring significant improvements.

Third, the results of the project demonstrate that a very complex pipeline of several independent projects can work provided a consistent way to link chains can be found. This stresses the importance of developing links between existing lexical resources in order to capitalize on their interconnection.

Further developments in the project will include — besides optimization, in order to allow researcher for tests on larger data sets, and refinements in the evaluation stage, to allow a finer assessment of the contribution of the single components— strategies for anaphora resolution, analysis of inter-clausal relations (in order to avoid wrong interpretations of counterfactuals and other “possible worlds” structures) and the development of links to other existing resources, such as the Arabic version of VerbNet.

REFERENCES

[1] http://www.ldc.upenn.edu/Catalog/docs/LDC2010T13/atb1-v4.1-taglist-conversion-to-PennPOS-forrelease.lisp

[2] T. Buckwalter, Buckwalter Arabic Morphological Analyzer Version 1.0. Philadelphia:Linguistic Data Consortium, 2002.

[3] T. Buckwalter and D. Parkinson, A Frequency Dictionary of Arabic. London and New York: Routledge, 2011.

[4] S. Green and J. DeNero, “A class-based agreement model for generating accurately inflected translations”, in ACL ’12, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 146-155.

[5] C. Fellbaum, “WordNet and wordnets”, in: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 2005, pp. 665-670

[6] S. Green and Ch. D. Manning, “Better Arabic parsing: baselines, evaluations, and analysis”, in COLING ’10, Proceedings of the 23rd International Conference on Computational Linguistics, pp. 394-402.

[7] M. Boella, F. R. Romani, A. Al-Raies, C. Solimando, and G. Lancioni, “The SALAH Project: segmentation and linguistic analysis of hadīth arabic texts. information retrieval technology lecture notes” in Computer Science vol. 7097, Springer, Heidelberg, 2011, pp 538-549.

[8] Ch. M. Meyer and I. Gurevych, “What psycholinguists know about chemistry: aligning Wiktionary and WordNet for increased domain coverage”, in Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 883-892.

[9] E. Niemann and I. Gurevych, “The people’s web meets linguistic knowledge: automatic sense alignment of wikipedia and wordnet”, in: Proceedings of the International Conference on Computational Semantics (IWCS), Oxford, United Kingdom, 2011, pp. 205-214.

[10] E. Wolf and I. Gurevych, “Aligning Sense Inventories in Wikipedia and WordNet”, in: Proceedings of the First Workshop on Automated Knowledge Base Construction (AKBC), Grenoble, France, 2010, pp. 24-28.

[11] H. A. Salmoné, An Advanced Learner’s Arabic-English Dictionary. Beirut: Librairie du Liban, 1889.

[12] N. Habash, O. Rambow, and R. Roth, “MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization”, in Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, 2009, pp. 102-109.

[13] I. Turki Khemakhem, S. Jamoussi and A. Ben Hamadou, “Integrating morpho-syntactic features in English-Arabic statistical machine translation”, in Proceedings of the Second Workshop on Hybrid Approaches to Translation”, Sofia, Bulgaria, 2013, pp. 74-81.

[14] M. Daoud, D. Daoud and Ch. Boitet, “Collaborative construction of Arabic lexical resources”, in Proceedings of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, 2009, pp. 119-124.

[15] S. Izwaini, “Problems of Arabic Machine Translation: evaluation of three systems”, in Proceedings of the International Conference on the Challenge of Arabic for NLP/MT, The British Computer Society (BSC), London, 2006, pp. 118-148.

[16] Arabic Treebank Guidelines, http://www.ircs.upenn.edu/arabic/guidelines.html. Accessed October 2013.

[17] https://github.com/anastaw/Meedan-Memory. Accessed October 2013.