

Springer Proceedings in Complexity

Juanzi Li · Guilin Qi
Dongyan Zhao · Wolfgang Nejdl
Hai-Tao Zheng *Editors*

Semantic Web and Web Science

Springer Proceedings in Complexity

For further volumes:
<http://www.springer.com/series/11637>

Juanzi Li • Guilin Qi • Dongyan Zhao • Wolfgang
Nejdl • Hai-Tao Zheng
Editors

Semantic Web and Web Science



Springer

Editors

Juanzi Li
Department of Computer Science
and Technology
Tsinghua University
Beijing
China, People's Republic

Dongyan Zhao
Institute of Computer Science & Technology
Beijing
China, People's Republic

Hai-Tao Zheng
Tsinghua Campus H202B
Shenzhen City
China, People's Republic

Guilin Qi
School of Computer Science & Engineering
Southeast University
Nanjing
Jiangsu, China, People's Republic

Wolfgang Nejdl
L3S Research Center
Leibniz University Hannover
Hannover, Germany

ISSN 2213-8684
ISBN 978-1-4614-6879-0
DOI 10.1007/978-1-4614-6880-6
Springer New York Heidelberg Dordrecht London

ISSN 2213-8692 (electronic)
ISBN 978-1-4614-6880-6 (eBook)

Library of Congress Control Number: 2013933998

© Springer Science+Business Media New York 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

In the last few years, the rapidly developing web has led to revolutionary changes in the whole human society. Nowadays, many scholars and experts are devoting them to the work of applying the semantic web theories into specific practice, while in turn improving the semantic web standards and technologies according to the demand in practice.

Following the success of fifth Chinese Semantic Web Symposium (CSWS 2010), we continued to organize the sixth Chinese Semantic Web Symposium in Shenzhen, China. We also organized the first Chinese Web Science Conference (CWSC 2012), which is collocated with CSWS 2012. The theme of Joint Conference of CSWS 2012 and CWSC 2012 is “web science and semantic web”. Web science involves full scope of Web-related researches and applications, and it integrates the web-related interdisciplinary researches into a new field of scientific research. This joint conference aims to promote expansions from the semantic web to web science and then to discuss the core technologies of next generation web, such as the web and swarm intelligence, a new generation of semantic search, semantic and web security, and so on.

This volume contains the papers presented at CSWS 2012 and CWSC 2012. The conference received 59 submissions, of which 54 were research papers and 5 were poster papers. These submissions cover a wide range of topics, including semantic search, ontology reasoning, social semantic web, and sentiment analysis. Each submission is assigned to three PC members to review. After a rigorous reviewing process, 25 research papers were selected for publication (the acceptance rate is around 46%) and 9 papers were accepted as poster papers.

We would like to thank the excellent work done by the program committee. Each of the PC members was assigned three to four papers to review and their timely and professional reviews were helpful for us to select submissions with high quality. We also thank some people who are involved in organizing the conference,

especially Professor Jim Hendler, Professor Wendy Hall, Professor Nigel Shadbolt, Yong Jiang, Hui Ma, Hao Chen, and Shaozhou Bai.

Beijing, People's Republic of China

Juanzi Li

Nanjing, People's Republic of China

Guilin Qi

Beijing, People's Republic of China

Dongyan Zhao

Hannover, Germany

Wolfgang Nejdl

Shenzhen City, People's Republic of China

Hai-Tao Zheng

Contents

Enhancing Software Search with Semantic Information from Wikipedia	1
Xiaoli Ma and Bo Yuan	
Using Semantic Technology to Improve Recommender Systems Based on Slope One	11
Rui Yang, Wei Hu, and Yuzhong Qu	
Finding, Extracting, and Building Academic Linked Data	25
Peng Wang and Xiang Zhang	
Proactive Recommendation Based on \mathcal{EL} Concept Learning	41
Jianfeng Du, Shuai Wang, Bohong Lin, Xiaoli Yao, and Yong Hu	
Impact of Multimedia in Sina Weibo: Popularity and Life Span	55
Xun Zhao, Feida Zhu, Weinling Qian, and Aoying Zhou	
Ontology-Based Model and Procedure Creation for Topic Analysis in Chinese Language	67
Dong Han and Kilian Stoffel	
On the Temporal Dynamics of Influence on the Social Semantic Web	75
Thomas Gottron, Olaf Radcke, and Rene Pickhardt	
A Detailed Analysis of the Quality of Stream-Based Schema Construction on Linked Open Data	89
Thomas Gottron and Rene Pickhardt	
Building Large-Scale Knowledge Base for Relations from Text	103
Junfeng Pan, Haofen Wang, and Yong Yu	
Co-mention and Context-Based Entity Linking	117
Qian Zheng, Juanzi Li, Zhichun Wang, and Lei Hou	

An Approach of Text Sentiment Analysis for Public Opinion Monitoring System	131
Min Zeng, Yujiu Yang, and Wenhuan Liu	
Music Recommendation Based on Label Correlation	143
Hequn Liu, Bo Yuan, and Cheng Li	
Inferring Public and Private Topics for Similar Events	153
Xubo Wen, Xiaoli Ma, Huan Xia, and Juanzi Li	
SemreX: A Semantic Association-Based Scientific Literature Sharing System	161
Pingpeng Yuan, Hai Jin, Yi Li, Binlin Chang, Xiaomin Ning, and Li Huang	
Consequence-Based Procedure for Description Logics with Self-Restriction	169
Cong Wang and Pascal Hitzler	
SAPOP: Semiautomatic Framework for Practical Ontology Population from Structured Knowledge Bases	181
Xinruo Sun, Haofen Wang, and Yong Yu	
Exploring Information Flow Patterns Between News Portals and Microblogging Platforms	187
Bo Zhang, Jinchuan Wang, and Lei Zhang	
A Two-Step Non-redundant Subspace Clustering Approach	201
Hai-Tao Zheng, Hao Chen, Yong Jiang, Shu-Tao Xia, and Huiqiu Li	
Exploiting Ontologies to Rank Relationships Between Patents	215
Hai-Tao Zheng, Nan Ma, Yong Jiang, Shu-Tao Xia, and Hui-Qiu Li	
Search Results Diversification Based on Swap Minimal Marginal Contribution	223
Hai-Tao Zheng, Shaozhou Bai, Shu-Tao Xia, Yong Jiang, and Huiqiu Li	
Who Are We Talking About? Identifying Scientific Populations Online ..	237
Julie M. Birkholz, Shenghui Wang, Paul Groth, and Sara Magliacane	
Study of Ontology Debugging Approaches Based on the Criterion Set BLUEP²CI.....	251
Qiu Ji, Zhiqiang Gao, and Zhisheng Huang	
NJVR: The NanJing Vocabulary Repository	265
Gong Cheng, Min Liu, and Yuzhong Qu	

Visualizing RDF Data Profile with UML Diagram	273
Huiying Li and Xiang Zhang	
Empirical Study of POI-Oriented Focused Crawler	287
Xin Fan, Jun-sheng Zhou, Can-yu Cheng, Yi-chu Zhou, and Di Yin	
Semantic Word Similarity Learned from Heterogenous Knowledge Bases	299
Yiling Liu, Yangsheng Ji, Chong Gu, Shouling Cui, and Jiangtao Jia	
A Workload-Based Partitioning Scheme for Parallel RDF Data Processing	311
Mengdong Yang and Gang Wu	
Chinese Microblog Sentiment Analysis Based on Semi-supervised Learning	325
Shaojie Zhu, Bing Xu, Dequan Zheng, and Tiejun Zhao	
Lexicon-Based Sentiment Analysis on Topical Chinese Microblog Messages	333
Anqi Cui, Haochen Zhang, Yiqun Liu, Min Zhang, and Shaoping Ma	
Research on Indexing Page Collection Selection Method for Search Engine	345
Liyun Ru, Zhichao Li, Yingying Wu, and Shaoping Ma	
The Chinese Bag-of-Opinions Method for Hot-Topic-Oriented Sentiment Analysis on Weibo	357
Jingang Wang, Dandan Song, Lejian Liao, Wei Zou, Xiaoqing Yan, and Yi Su	
A Semantic-Driven Music Recommendation Model for Digital Photo Albums	369
Jiansong Chao, Haofen Wang, Wenlei Zhou, Weinan Zhang, and Yong Yu	
The DReW System for Nonmonotonic DL-Programs	383
Guohui Xiao, Thomas Eiter, and Stijn Heymans	
Accessing Information About Linked Data Vocabularies with vocab.cc ...	391
Steffen Stadtmüller, Andreas Harth, and Marko Grobelnik	
Qualitative Cognition for Uncertainty Knowledge Using Cloud Model ...	397
Yuchao Liu, Lin Li, and Juanzi Li	

Enhancing Software Search with Semantic Information from Wikipedia

Xiaoli Ma and Bo Yuan

Abstract Software is becoming ubiquitous, from desktop computers to smart phones, and has created significant impact on the quality of our everyday life. Sharing and reusing high-quality software can save tremendous amount of time and efforts that otherwise would need to be reinvented. The challenge is how to efficiently search through a potentially huge database of software and return the most relevant results. In this paper, we present a prototype of semantic software search engine that exploits the semantic information from Wikipedia, one of the largest online knowledge repositories as the result of collaborative intelligence. We propose a technique to replace the original concept space by an extended concept space extracted from Wikipedia to incorporate commonsense knowledge into software search. Experimental results show that this strategy can achieve better performance over traditional software search based on the original concept space.

1 Introduction

To facilitate the sharing of software applications across diverse disciplines, it is often desirable to develop a mechanism to automatically collect software distributed on the Web and make them easily accessible by users. Currently, the most widely used method is keyword matching. However, understanding precisely the intention of users based on search keywords still remains as a major challenge. Fortunately, recent studies in semantics show that semantics can help extend our understanding of the original content. For example, by using ontology in search, especially in organizing resources and understanding search words, search engines can better understand the meaning of various concepts and produce more targeted outputs [1].

X. Ma • B. Yuan (✉)

Intelligent Computing Lab, Division of Informatics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, People's Republic of China
e-mail: thumxl@gmail.com; yuanb@sz.tsinghua.edu.cn

Previous studies show that good correlation between computed relatedness scores and human judgments can be achieved by using vectors of Wikipedia¹-based concepts to represent the original texts [2]. As concepts represent meaning units for constructing knowledge, the concept space can, at least to some extent, serve as the commonsense and domain-specific knowledge in search applications. Wikipedia is the largest online encyclopedia and has been successfully exploited to assist the research in knowledge engineering and machine learning. Recently, it has been predominately used as additional text features for knowledge discovery and visualization in topic modeling, text categorization and clustering, semantic analysis, and other text processing tasks [3–6].

In this paper, we employed similar ideas in software search by using Wikipedia-based concept vectors to specify software applications. A semantic index is also built using this concept space. With the help of this fusion strategy, information from multiple sources can be unified to achieve a more complete understanding towards a specific software application, compared to its original description information. During the search process, the relatedness between software applications (represented as concept vectors) is calculated to rank the search results.

The contribution of this paper is twofold. First, we presented extended concept space construction (ECSC), a novel approach to forming an extended semantic representation of the original concept based on Wikipedia. This method can be also applied to other tasks requiring semantic information without building a comprehensive semantic system. Second, we applied ECSC in the task of software search and achieved promising results compared to traditional software search engines in terms of the consistency with the search results produced by human users.

The rest part of this paper is organized as follows. The strategy for extending the concept space of software for indexing and semantic search is detailed in Sect. 2. The major experimental results are presented in Sect. 3, along with a description of the data collection and evaluation procedure. A number of existing software search platforms and related studies are discussed in Sect. 4 and this paper is concluded in Sect. 5 with some directions for future work.

2 Semantic Search Using ECSC

It is well known that useful semantic information can be extracted by analyzing the log files of search engines to provide better search services. However, in many cases, the log files are not made accessible to the public. Instead, a more practical approach is to use publicly available knowledge resources such as Wikipedia to acquire the semantic information to improve the search experience. More specifically, given a software description, a concept space (a vector of concepts) is constructed to represent its essential attributes (e.g., what each term is about and what it is).

¹<http://www.wikipedia.org/>

For example, the description of Weka² (Waikato Environment for Knowledge Analysis) may read as “A suite of machine learning software developed by the Machine Learning Group at University of Waikato, containing a collection of popular machine learning algorithms for data mining tasks.” Ideally, a user that is interested in a certain data mining algorithm should be able to retrieve this software record using the name of the algorithm or even its alias as the keyword, even if it does not explicitly appear in the original description. In other words, the software search engine should be able to achieve a more general understanding of the software on top of its existing description.

In Wikipedia, each article is treated as a concept and each concept is represented by a vector of words that occur in the article. The strength of association between words and concepts can be computed by WLVM (Wikipedia Link Vector Model) [7] using the hyperlink structure of Wikipedia. The semantic relatedness of two Wikipedia articles is defined by the angle between the vectors of the links within them. A software package called WikipediaMiner³ contains an implementation of WLVM, which was used in our work to compute the relatedness between concepts.

Using ECSC, each software application is annotated as a set of concepts (original concept space) and each concept is mapped into a weighted sequence of Wikipedia concepts ordered by their relevance (i.e., use Wikipedia concepts to augment the bag of concepts that represent the software) using WikipediaMiner. Concepts with low levels of relatedness are discarded. Finally, all sets of concepts are merged into a vector of concepts (extended concept space) as the new software description. When searching for software, the semantic relatedness between two software applications is calculated by comparing their vectors of concepts, for example, using the cosine metric. In the meantime, with the help of ECSC, for the same input keyword, it is now possible to retrieve software records that are closely related but otherwise would not have been possible to be retrieved. The overall process of ECSC is shown in Fig. 1.

To demonstrate how our approach works, the top 10 individual Wikipedia concepts most relevant to a given concept (e.g., SVM and Bayesian probability) is shown in Table 1. Table 2 shows the original concepts in the description of Weka as well as the extended new concepts. Table 3 shows the comparison of the list of concepts of Weka and Mlpy,⁴ a Python open source machine learning library.

With the concepts of software applications, in our work, inverted index was built using Apache Lucene⁵ along with other important attributes as the index content. In order to efficiently rank the search results, different weights were assigned to attributes. For instance, in Eq. (1), $v_{ij} = 1$ if the j th software record contained the i th

²<http://www.cs.waikato.ac.nz/ml/weka/>

³<http://wikipedia-miner.cms.waikato.ac.nz/>

⁴<http://mlpy.sourceforge.net/>

⁵<http://lucene.apache.org/>

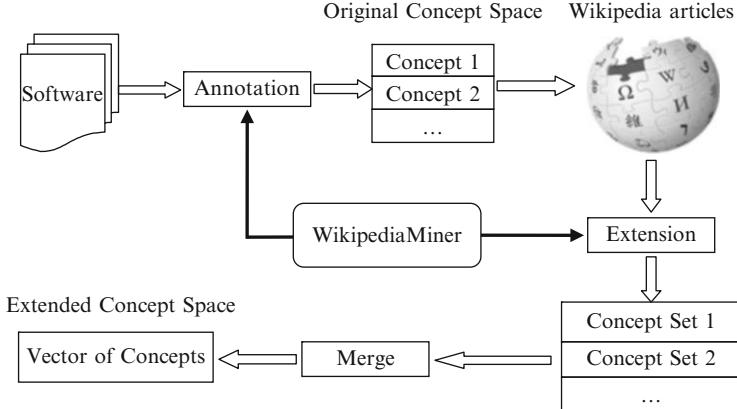


Fig. 1 A diagram of the overall process of ECSC

Table 1 The top 10 concepts most relevant to SVM and Bayesian probability

#	SVM	Bayesian probability
1	Statistical classification	Bayes' theorem
2	Decision tree learning	Frequency probability
3	Naïve Bayes classifier	Prior probability
4	Supervised learning	Decision theory
5	Perceptron	Principle of indifference
6	Linear classifier	Probability interpretations
7	Kernel methods	Statistical inference
8	Multiclass classification	Bayesian inference
9	Discriminative model	Frequentist inference
10	Document classification	Posterior probability

Table 2 The original and extended concepts most relevant to Weka

#	Original concepts	Extended concepts
1	Association rule learning	Apriori algorithm
2	Clustering	Principal component analysis
3	Feature selection	Cross-validation (statistics)
4	K-means clustering	K-medoids
5	Predictive modeling	Naïve Bayes
6	RapidMiner	Association rule learning
7	Boosting	Bootstrap aggregating
8	C4.5 algorithm	ID3 algorithm
9	Decision tree learning	Random forest
10	Statistical classification	Support vector machine

keyword and $v_{ij} = 0$ otherwise; for $W[P_{i,j}]$, its value was 0.5 for words in title and authors, 0.4 for words in the concept space, and 0.1 for other attributes.

$$\text{Rank}(j) = \sum_{i=1, j=1}^{i,j} v_{i,j} \times W[P_{i,j}] \quad (1)$$

Table 3 Comparison of the concepts of Weka and Mlpy

#	Weka	Mlpy
1	RapidMiner	Support vector machine
2	Predictive modeling	Least squares
3	Decision tree learning	Ridge regressions
4	C4.5 algorithm	Linear discriminant analysis
5	Feature selection	Perceptron
6	Clustering	Logistic regression
7	Association rule learning	Hierarchical clustering
8	Boosting	Partial least squares
9	Statistical classification	K-means
10	K-means clustering	Principal component analysis
11	ID3 algorithm	Fisher discriminant
12	Apriori algorithm	Regression
13	Bootstrap aggregating	Python
14	Random forest	Classification
15	K-medoids	Clustering
16	Support vector machine	Dimensionality reduction

3 Empirical Evaluation

We implemented our ECSC approach using an English Wikipedia dump⁶ dated 12 March 2012. After parsing the Wikipedia XML dump, we obtained 16.0 GB of text articles. Upon removing narrow and overly specific concepts (those having fewer than 100 words and fewer than 5 in-links or out-links), 3,636,122 articles related to software were left. We processed the text of these articles by removing stop words and rare words, and stemming the remaining words, which yielded 39,524 distinct terms to be used for representing Wikipedia concepts as attribute vectors. The concept space contained over 20,000 concepts and 40,000 relatedness scores.

Totally over 3,000 software applications were collected from the Web, including more than 2,000 applications retrieved from software repositories and communities such as SourceForge⁷ and Mloss⁸ and around 1,000 applications crawled from various Web sites such as universities, research institutes, and personal blogs. Figure 2 shows the metadata of software crawled from the Web.

For evaluation purpose [8], we compared the results based on ECSC to results produced by human users. We used 40 software applications as the test samples and identified 20 most related software applications for each of them using three methods. The motivation was to investigate that whether the new concept space

⁶<http://dumps.wikimedia.org/>

⁷<http://sourceforge.net/>

⁸<http://mloss.org/software/>

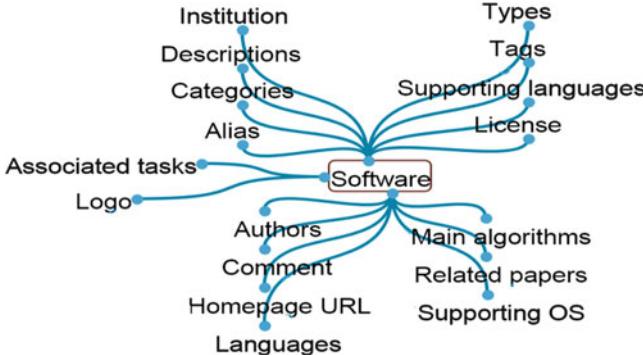


Fig. 2 The schema of software profile

based on ECSC can bring measurable benefits to the quality of search results. In other words, we were interested in finding out whether ECSC can help the search engine better understand the scope of software applications. With the lack of a well-accepted quantitative performance metric, we chose to use human results as the benchmark against which to evaluate different methods.

Let S_1 be the set of software applications selected using the original software description while S_2 be the set of software applications selected using ECSC. Furthermore, we asked a group of research students in our lab as volunteers to manually select a set of 20 software applications that they believed were most related to each test sample, referred to as S_3 .

Ideally, S_1 and S_2 should be similar to S_3 (containing a large portion of shared software items), and in practice it would be interesting to compare the size of the intersection between S_1 and S_3 (the performance of the original method) with the size of the intersection between S_2 and S_3 (the performance of ECSC). Figure 3 shows that, over the 40 test samples, ECSC (solid line) produced consistently better results than the original method (dashed line), which confirms that the extended concept space can bring measurable benefit to software search services.

More specifically, ECSC demonstrated notable improvement in the correlation between computed software relatedness and human judgment, compared to the origin system: the average proportion of common software items using ECSC was $16.8/20 = 84\%$, while the average proportion of common software items using the origin method was $12.4/20 = 62\%$. Note that the performance had certain level of fluctuation over the set of test samples, which may be due to the insufficient amount of related information available in Wikipedia articles for some software applications. Also, different software applications may have quite different concept space sizes (both original and extended concepts), which may have some impact on the advantage of ECSC over the original method.

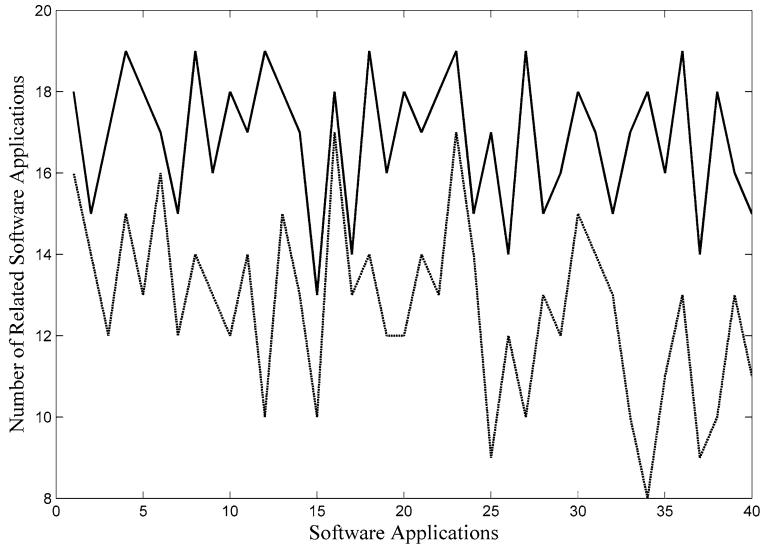


Fig. 3 The comparison between ECSC (*solid line*) and the original method (*dashed line*) over 40 software applications. The vertical axis shows the number of software applications returned that are shared with the results produced by human users

4 Related Work

SourceForge is a popular repository mainly for free or open source software, providing free services for controlling and managing software projects and limited search functions including keyword search for titles, category, license, programming languages, and operating systems. Github⁹ is a platform similar to SourceForge with revision control systems for both paid private repositories and free open source projects. It also provides keyword search for titles and contents of code as well as category selection such as programming languages and developers.

Mloss (machine learning open source software) is dedicated to building a comprehensive open source machine learning environment and provides keyword search for titles with optional filters such as author, submitter, tag, license, programming language, and operating system. The major motivation of Mloss is that, given the large number of machine learning algorithms available in the literature, their implementations are often not open to the research community, resulting in low usability and weak interoperability. It is believed that publishing existing and freshly developed algorithm toolboxes along with short articles can be highly valuable to the development of machine learning and the general scientific communities.

⁹<https://github.com/>

Table 4 The category information of the concept of machine learning in Wikipedia

Machine learning	
• Applied machine learning	• Inductive logic programming
• Artificial intelligence conferences	• Machine learning algorithms
• Bayesian networks	• Markov models
• Classification algorithms	• Neural networks
• Cluster analysis	• Machine learning researchers
• Computational learning theory	• Support vector machines
• Data mining and machine learning software	• Kernel methods for machine learning
• Decision trees	• Latent variable models
• Dimension reduction	• Learning in computer vision
• Ensemble learning	• Log-linear models
• Evolutionary algorithms	• Loss functions

Understanding and incorporating the semantic information of software is critical for the effective search of software. For this purpose, Wikipedia has been widely used as the knowledge base to provide semantic information [9, 10]. For example, the titles of articles in Wikipedia are often treated as concepts and topics. Selected Wikipedia concepts can be augmented as extra text in clustering short texts to improve the accuracy [4]. Concepts derived from Wikipedia can be also used as a high-dimensional concept space for representing the semantic meaning of text [2].

Furthermore, each article in Wikipedia belongs to at least one category and there is structured information in Wikipedia such as namespace and category trees that can be used as taxonomy¹⁰ (see Table 4 for an example). Category itself is also a reliable and useful source for topic discovery and relatedness computing [11]. For example, semantic distances can be defined as a function of the number of edges in the taxonomy along the path between conceptual nodes [12].

5 Conclusion

In this paper, we proposed a novel approach called ECSC to computing the semantic relatedness of software applications using extended concept space. With the aid of Wikipedia articles, which contain many concepts and interlinks of concepts, we calculated the relatedness of concepts using WLVM and selected a vector of concepts closely related to a given concept, effectively extending the original concept space of software applications.

Compared to other methods based on taxonomy and semantic Web technologies, our approach does not require building a domain knowledge repository by human experts, which can be expensive and difficult to maintain. Instead, our approach can

¹⁰<http://en.wikipedia.org/wiki/Wikipedia:Categorization>

take the advantage of the comprehensive and authoritative knowledge in Wikipedia accumulated through collaborative intellectual work.

Empirical evaluation on 40 test samples confirms that ECSC can consistently lead to substantial improvement in accuracy when searching for a set of software applications closely related to a given software application. Compared to the traditional method based on the original concept space (description of software), the search results using ECSC can be better correlated with human judgments. Furthermore, due to the explicit use of domain concepts, the search results are also easy to be understood by human users. In the future, we will consider incorporating other semantic techniques into software search as well as applying ECSC to other application domains.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No. 60905030). The authors are also grateful to Prof. Juanzi Li for her kind help.

References

1. Waitelonis, J., Sack, H., Hercher, J., Kramer, Z.: Semantically enabled exploratory video search. In: 3rd International Semantic Search Workshop, Article No. 8 (2010)
2. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: 20th International Joint Conference on Artificial Intelligence, pp. 1606–1611 (2007)
3. Coursey, K., Mihalcea, R.: Topic identification using wikipedia graph centrality. In: 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pp. 117–120 (2009)
4. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 787–788 (2007)
5. Yang, J., Han, J., Oh, I., Kwak, M.: Using wikipedia technology for topic maps design. In: 45th Annual Southeast Regional Conference, pp. 106–110 (2007)
6. Medelyan, O., Witten, I.H., Milne, D.: Topic indexing with wikipedia. In: AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, pp. 19–24 (2008)
7. Milne, D.: Computing semantic relatedness using wikipedia link structure. In: New Zealand Computer Science Research Student Conference (2007)
8. Tumer, D., Shah, M.A., Bitirim, Y.: An empirical evaluation on semantic search performance of keyword-based and semantic search engines: Google, Yahoo, Msn and Hakia. In: Fourth International Conference on Internet Monitoring and Protection, pp. 51–55 (2009)
9. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic wikipedia. In: 15th International Conference on World Wide Web, pp. 585–594 (2006)
10. Kaptein, R., Serdyukov, P., De Vries, A., Kamps, J.: Entity ranking using wikipedia as a pivot. In: 19th ACM International Conference on Information and Knowledge Management, pp. 69–78 (2010)
11. Chernov, S., Iofciu, T., Nejdl, W., Zhou, X.: Extracting semantic relationships between wikipedia categories. In: First Workshop on Semantic Wikis – From Wiki to Semantics (2006)
12. Strube, M., Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using wikipedia. In: 21st National Conference on Artificial Intelligence, pp. 1419–1424 (2006)

Using Semantic Technology to Improve Recommender Systems Based on Slope One

Rui Yang, Wei Hu, and Yuzhong Qu

Abstract Slope One is a family of algorithms proposed for collaborative filtering and has been widely deployed on websites' recommender systems. Compared to SVD, LSI, Similarity Fusion, or some other commonly used algorithms, Slope One often gives better performance in usability, realizability, and efficiency. However, its prediction accuracy is sometimes lower than other expensive methods, because it is a collaborative filtering model only based on average rating difference and cannot meet some special or individual requirements. The user's and item's features are also not well considered. In this paper, we propose a new approach for enhancing Slope One using semantic technologies. We explore the implicit relationships between items based on the Linked Data and some measures for computing the semantic distances. The relevance information can be utilized to adjust the weighting when computing the prediction ratings. The approach is easy to be implemented and does not increase the complexity of Slope One hardly. A preliminary experiment is conducted and shows that our approach outperforms the traditional weighted Slope One scheme.

1 Introduction

Recommender systems [1] have made contributions to the success of personalized websites as they can automatically and efficiently choose the items or services suitable to user's interest from huge datasets. Among recommendation approaches,

R. Yang

Department of Computer Science and Technology, Nanjing University, China
e-mail: ryang@mail.nju.edu.cn

W. Hu (✉) • Y. Qu

Department of Computer Science and Technology, Nanjing University, China

State Key Laboratory for Novel Software Technology, Nanjing University, China
e-mail: whu@nju.edu.cn; yzqu@nju.edu.cn

collaborative filtering (CF) [9, 15], which tries to predict the utility of items for a particular user based on the items previously rated by other users, is most popular used nowadays.

As the most concise form of nontrivial and rating-based CF models, Slope One algorithm [4, 11] shows quite an easy way to build a CF model based on average rating difference, and it has been widely deployed in some successful real-world Web applications such as inDiscover (an MP3 recommender system) and Value Investing News (a stock market news site). But it does not perform as well as some other expensive algorithms in the rating prediction (the most commonly used evaluation method in a large share of the papers in the field of recommender system) [1, 8]. As studies indicate, the core idea of Slope One is that the average difference may cover up the personality. Unfortunately, the user's personality cannot be covered in some especial cases.

Since Tim Berners-Lee, father of the Semantic Web, published the famous article in Scientific American [2], the Semantic Web (SW) has become a hot research field. Nowadays, more and more theoretical researches are converting to practical applications. In the meanwhile, the Linking Open Data (LOD) cloud [3] has grown considerably. The room for end-user applications instead of the professional tools like semantic search engines and APIs that consume Linked Data is being paid more and more attention. Thanks to the LOD, in the research field of recommender system, we can easily get much more open domain knowledge than before. How to use these semantic data to improve the recommender systems has become a new research hot spot, and it is the main focus of this paper.

In this paper, we propose a new approach to enhance recommender systems based on Slope One. A movie recommender system, whose original data source is MovieLens dataset, is developed to illustrate this approach particularly. In specific, the movie recommender system uses Linked Data (in particular, DBtropes [10]), which offers a vast amount of information of various types of film or television works to compute the prediction ratings. We provide some methods to show how to map resources to Linked Data from a relational database. With these mapping methods, we combine these closed and open data and compute the semantic distance (can also be called the implicit relationship between the items) [5, 7]. We borrow ideas from content-based recommender systems [1] to improve the original Slope One prediction algorithm. Our goal in this paper is not to propose a method to replace the original Slope One in the field recommender system or compare the accuracy of a wide range of CF algorithms. We just want to make an attempt to combine the semantic technologies with traditional CF and increase the number of accuracy without reducing computationally efficiency and simplicity. Results on a preliminary experiment show that our approach outperforms the original Slope One scheme and the scheme only based on semantic distance.

The rest of this paper is structured as follows. Section 2 discusses related studies. In Sect. 3, we introduce the semantic distance calculation method and our approach to map semantic distance to traditional datasets and improve the original Slope One scheme. Experimental results on the MovieLens dataset are reported in Sect. 4. Finally, Sect. 5 concludes the paper with future work.

2 Related Work

State-of-the-art recommender systems can be generally categorized as content-based, collaborative, and hybrid [1, 18]. We focus on the collaborative methods in this paper because they are trendy and not a lot of researchers have combined them with Semantic Web technologies to provide recommendations until now. In the following of this section, we will briefly introduce some mainstream collaborative methods and some Semantic Web technologies which can be used in recommendation systems.

2.1 Collaborative Methods

User-Based Collaborative Filtering. Collaborative filtering systems predict the utility of items for a particular user based on the items previously rated by other users. User-based collaborative filtering is said to be the oldest algorithm in this field [1, 15]. It was proposed in 1992 and first applied to spam filtering system. Nowadays, [Digg.com](#) uses it to recommend suitable news to every particular user to solve the information overload problem.

For the online recommendation systems, when a user a needs personalized recommendation service, we can look for some other users who have similar interests with a first and then push the items these users favored (but a did not browse before) to a as recommendations. This idea is the core of user-based collaborative filtering. So we can summarize the above into two steps:

1. Find out the set S of users having similar interests with the target user.
2. Find out the items that users in S like and make a recommendation list based on them.

Step 1's key is to compute the interest similarity of any pair of users. Collaborative filtering usually calculates interest similarity based on behavior similarity, i.e., let $N(u)$ be the set of items that user u has positive feedback and $N(v)$ be the set of items that user v has positive feedback, we can compute the similarity between user u and user v by the Jaccard coefficient:

$$w_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}, \quad (1)$$

or by the cosine similarity:

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}}. \quad (2)$$

After we get the similarity between the users, user-based CF finds out K users whose interest similarity with the target user is the lowest, then recommend their favorite items to the target user. A general equation can help us measure user u 's interest in item i :

$$p_{ui} = \sum_{v \in S(u, K) \cap U(i)} w_{uv} r_{vi}, \quad (3)$$

where $S(u, K)$ contains K users whose interest is most similar to the user u and $U(i)$ contains users who have rated item i . r_{vi} is the rating of user v to item i or some other value that represents how user v likes item i , sometimes it can also be simplified as 0 and 1 (dislike or like).

Item-Based Collaborative Filtering. Item-based CF is the most widely used algorithm in industry. The world's largest e-commerce site Amazon and online video-sharing site YouTube are using recommender systems based on it. Unlike user-based, item-based CF tries to recommend items to users that are similar to items they like before [1, 15]. For example, this algorithm recommends *The Hobbit* to a user just because *The Lord of the Rings* is in the user's favorite list. It is worth noting that item-based CF calculates the item-to-item similarity based on user behavior instead of items' features. Item A and item B are very similar because most of the users who like A also like B .

After we get the item-to-item similarity, a general equation can help us measure user u 's interest in item i :

$$p_{uj} = \sum_{v \in S(j, K) \cap I(u)} w_{ji} r_{ui}, \quad (4)$$

where $S(j, K)$ contains K items which are most similar to item j while $I(u)$ contains the items user u like. w_{ji} here is the similarity between item j and i .

2.2 Weighted Slope One

There is a kind of widely deployed item-based CF algorithms called Slope One in industry. The Slope One algorithms are based on predictors in the form of $f(x) = x + b$. According to user ratings on items, we can get the regression line between any pair items. It is a very simple algorithm because it just uses average difference between the two items' ratings as the single free parameter b [11]. Its final prediction is calculated as follows:

$$p_{ui} = \frac{\sum_{j \in I(u)-i} (\sum_{x \in S_{ji}} \frac{v_{xi} - v_{xj}}{|S_{ji}|} + v_{ui}) |S_{ji}|}{\sum_{j \in I(u)-i} |S_{ji}|}, \quad (5)$$

where S_{ji} is the set of users that have rated both items j and i , I_u is the set of items that user u has rated, and v_{ui} represents the rating of user u to item i .

Different from the original Slope One algorithm, the above equation underlines the number of ratings available from each user. Thus, it is also called the weighted Slope One scheme. Apart from that, a third variant, bipolar Slope One, has also been studied. It tries to resolve another problem that praise and negative feedback on the user's decision-making influence is different. It firstly divides the items into those that the user has rated positively and those that have been rated negatively and then applies the weighted Slope One scheme separately and uses the weighted average as the final result.

In this paper, we will pay more attention to the weighted Slope One because it outperforms the others in MovieLens's datasets and is also the most popular one in industry [4, 11].

2.3 Recommender Systems in the Semantic Web

In the Semantic Web area, researchers try to improve recommender systems with semantic technologies mainly based on the Linking Open Data (LOD) cloud and the content-based recommender system model (different from the item-based model, the content-based model usually only uses items' features to compute the recommendations). We can easily fetch much useful attribute information about the items in our system from LOD, and sometimes this information is not readily available especially when we are starting to build recommender systems. Then we use it to compute the similarity between the items and build the item-based recommender system. For example, the datasets offered by MovieLens only give out titles, years, IMDB's URLs, and genres about movies, while DBtropes, a Linked Data wrapper for [TVTropes.org](#), provides much more information about directors, casts, writers, summaries, and even the users' comments. The information wrapped by Linked Data manifests in the form of RDF triples. So mining the connections between the items would not be a very difficult task with such a rich source of data and the good form. Researchers have convinced that they can calculate a more accurate similarity between items through Linked Data compared with the traditional method. In this paper, we also reuse some of the excellent algorithms.

However, CF-based recommender systems in general perform much better than content-based ones. This is mainly because content-based algorithms ignore the user behavior, thus ignoring the law contained in the items popularity and user behavior. Therefore, its accuracy is relatively low [4].

3 Our Approach

In this section, we will introduce how to use semantic technologies to improve the weighted Slope One scheme. In order to build the movie recommender system which uses both traditional datasets and Linked Data, we follow four steps (see Fig. 1):

1. Identify the relevant subset from LOD (in the form of RDF triples).
2. Map the URIs in RDF triples to the items in traditional datasets and reduce the RDF data for computational optimization.
3. Use the LDSD algorithm to compute the semantic distances and insert them into traditional datasets as item-to-item similarities.
4. Integrate the similarities into the original weighted Slope One scheme and compute the recommendations.

3.1 Linked Data Used in Our Recommender System

Technically, Linked Data is a model for publishing structured data online and dereferenceable URIs are used as identifiers. We give a definition of Linked Data dataset firstly in order to define algorithms using it later [12].

Definition 1 A *Linked Data dataset* is a graph $G = (R, L, I)$, where $R = \{r_1, r_2, r_3, \dots, r_n\}$ is a set of resources and $L = \{l_1, l_2, l_3, \dots, l_n\}$ is a set of typed links, both resources and typed links are identified by dereferenceable URIs. $I = \{i_1, i_2, i_3, \dots, i_n\}$ is a set of instances of these links between resources such as $i_k = \langle l_j, r_a, r_b \rangle$.

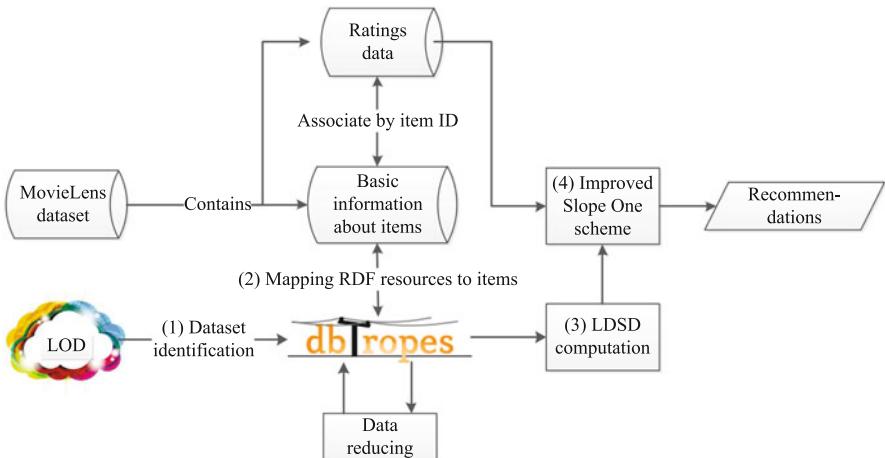


Fig. 1 Overview of the approach

```

http://dbtropes.org/.../TheMatrix
  rdf:type http://dbtropes.org/ont/TVTItem ;
  rdfs:label "The Matrix" ;
  rdfs:comment "What is the Matrix?..." ;
  http://skipforward.../hasFeature http://dbtropes.../int_101087a ;
    http://dbtropes.../int_10435f29 ;
    http://dbtropes.../int_108bef7e ;
  rdfs:seeAlso http://dbtropes.../EnterTheMatrix ;
    http://dbtropes.../Starwars ;

```

Fig. 2 An example for an RDF sentence in DBtropes

[Linkeddata.org](#), a widely known and used website, exists to provide a home for, or pointers to, resources across the Linked Data community. In order to meet our needs, we mainly use DBtropes as the source data. There are more than 10,000,000 RDF statements, 22,000 items, 22,000 feature types and 1,750,000 feature instances in DBtropes and [DBTropes.org](#) provides a download link to all Internet users. All data are in the form of RDF triples.

Example 1. Figure 2 depicts some RDF triples in DBtropes. From it we can see that there is a direct link between the resource *The Matrix* and the resource *Starwars*, and the links like [http://skipforward.net/.../hasFeature](#) can help us to find the relationships. All the information like this is useful to calculate similarities between items.

3.2 Mapping Semantic Distance to Traditional Datasets

In order to use Linked Data dataset and the traditional dataset like MovieLens dataset (providing with rating data and basic information about items and users) together, we have to establish the correspondence between the two datasets that associates each resource in a set with a resource in another set. In this paper, we use the simplest method, string matching, to accomplish this task because DBtropes published its resources in a very structured form. Every film entity is identified by a dereferenceable URI like [http://dbtropes.org/.../TheMovieTitle](#). The last fragment of the URI is the movie title. Thus, we can get the movie titles from MovieLens dataset and easily change the spelling style to make the two successfully matched.

It is hard to give out a general mapping method. We believe that the right method is the best method. For example, if we want to build a book recommender system instead of the movie recommender system, we can use ISBN numbers which can be easily found in both traditional datasets and Linked Data to accomplish the mapping task. For a paper recommender system, the paper titles, authors, journals, and institutes may be helpful. When two or more features are considered, we can also use the Vector Space Model (VSM) and set support threshold to do this work.

3.3 Semantic Distance

As said before, MovieLens' dataset does not provide us with plenty of information about movie's features or some other "noncritical" details. So we have to mine the implicit relationships between items from semantic datasets (Linked Data). Based on Definition 1, the work in [12] defined a *Linked Data Semantic Distance* (LDSD) measure to compute the distance between two resources published as Linked Data. The series of algorithms have been used in some item-based recommender systems (also called content-based) and convinced to be efficient and well performed in some cases. Since [12, 13] has discussed a lot about LDSD, we directly give out the equations here instead of describing and interpreting the details of it. At a glance, the following definitions identify four dimensions to compute the semantic distance between two resources r_a and r_b :

Definition 2 C_d is a function that computes the number of direct and distinct links between resources in a graph G . $C_d(l_i, r_a, r_b)$ equals 1 if there is an instance of l_i from resource r_a to resource r_b , 0 if not. By extension, C_d can be used to compute the total number of direct and distinct links from r_a to r_b ($C_d(n, r_a, r_b)$) as well as the total number of distinct instances of the link l_i from r_a to any node ($C_d(l_i, r_a, n)$).

Definition 3 C_{io} and C_{ii} are two functions that compute the number of indirect and distinct links, both outgoing and incoming, between resources in a graph G . $C_{io}(l_i, r_a, r_b)$ equals 1 if there is a resource n that satisfies both $\langle l_i, r_a, n \rangle$ and $\langle l_i, r_b, n \rangle$, 0 if not. $C_{ii}(l_i, r_a, r_b)$ equals 1 if there is a resource n that satisfies both $\langle l_i, n, r_a \rangle$ and $\langle l_i, n, r_b \rangle$, 0 if not. By extension, C_{io} and C_{ii} can be used to compute the total number of indirect and distinct links between r_a and r_b ($C_{io}(n, r_a, r_b)$ and $C_{ii}(n, r_a, r_b)$, outgoing resp. incoming) as well as the total number of resources n linked indirectly to r_a via l_i ($C_{io}(l_i, r_a, n)$ and $C_{ii}(l_i, r_a, n)$, outgoing resp. incoming).

We select two measures [12] from the LDSD series to compute the similarity (can be also called the semantic distance here). Equation (6) is the first similarity measure named $LDSD_d$ and it only considers direct incoming and outgoing links. Equation (7) is the second similarity measure named $LDSD_i$ and it takes indirect links into account.

$$LDSD_d(r_a, r_b) = \frac{1}{1 + C_d(n, r_a, r_b) + C_d(n, r_b, r_a)}, \quad (6)$$

$$LDSD_i(r_a, r_b) = \frac{1}{1 + C_{io}(n, r_a, r_b) + C_{ii}(n, r_a, r_b)}. \quad (7)$$

3.4 Recommendation Algorithm

Based on the definitions and equations above, we can finally define the prediction algorithm. The prediction rating of user u to item i is computed as follows:

$$p_{ui} = \frac{\sum_{i \in I_u - i_j} (\sum_{x \in S_{ji}} \frac{v_{xi} - v_{xj}}{|S_{ji}|} + v_{ui}) \log \frac{|S_{ji}|}{LDSD(i_i, i_j)}}{\sum_{i \in I_u - i_j} \log \frac{|S_{ji}|}{LDSD(i_i, i_j)}}, \quad (8)$$

where $LDSD(i_i, i_j)$ here can be $LDSD_d(i_i, i_j)$, $LDSD_i(i_i, i_j)$ or $LDSD_{cw}(i_i, i_j)$. We verify their quality according to the experiment in the next section. Equation (8) is a nonlinear transformation method based on weighted Slope One; it directly changes the calculation method of weights in the original equation. The new weights consider the influence of relationships between items and the relationships are just semantic distance which we mined from the Linked Data. We try to add the weight if there is a firm relationship between item i_i and i_j and decrease it in the opposite situation. In fact, we learn from Term Frequency-Inverse Document Frequency(TF-IDF) [17] when computing the weights.

Corresponding to the nonlinear method above, we also defined a so-called linear method. Formally speaking:

$$p_{ui} = (1 - \alpha) \times \frac{\sum_{i \in I_u - i_j} (\sum_{x \in S_{ji}} \frac{v_{xi} - v_{xj}}{|S_{ji}|} + v_{ui}) |S_{ji}|}{\sum_{i \in I_u - i_j} |S_{ji}|} + \alpha \times \frac{\sum_{i \in I_u - i_j} (\sum_{x \in S_{ji}} (v_{xi} - v_{xj}) + v_{ui} |S_{ji}|) \frac{1}{LDSD(i_i, i_j)}}{\sum_{i \in I_u - i_j} \frac{1}{LDSD(i_i, i_j)}}. \quad (9)$$

The first half of Eq. (9) is the original weighted Slope One, and the latter half takes the LDSD instead of the number of ratings available from each user into account. Free parameter α is used to adjust the proportion. From some angles this method is more scalable. By adjusting the parameter, we can observe the influence of semantic distance in more detail.

4 Evaluation

There are three main experimental methods for evaluating the effects of recommender systems—offline experiment, user study and online experiment [14]. Because our system is not a real commercial system, we mainly focus on the offline experiment.

Table 1 RMSE and MAE results

Methods	RMSE	MAE	Run time
Weighted Slope One (WSO)	1.077	0.789	16.518s
Nonlinear transformation of WSO with $LDSD_d$	1.077	0.789	17.100s
Nonlinear transformation of WSO with $LDSD_i$	1.092	0.802	17.079s
Linear transformation of WSO with $LDSD_d$	1.067	0.764	16.573s
Linear transformation of WSO with $LDSD_i$	1.068	0.771	16.612s

4.1 Prediction Accuracy

Prediction accuracy is usually used to evaluate the capability of predicting user behavior by a recommender system or a recommendation algorithm. This indicator is the most important indicator of recommender system offline experiment.

Rating Prediction. The accuracy of rating prediction is commonly calculated by Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [1, 14]. For a user u and an item i , let r_{ui} be the actual rating that u gives i while \hat{r}_{ui} be the prediction rating given by recommendation algorithm, RMSE and MAE are defined as follows:

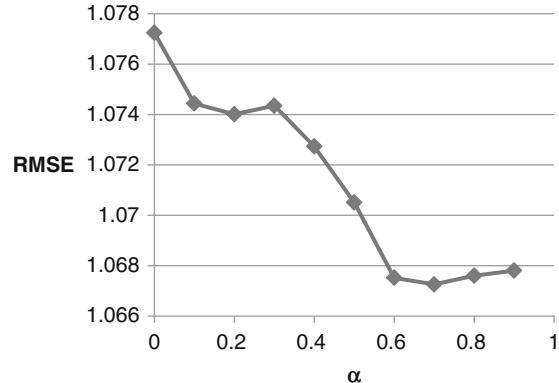
$$RMSE = \sqrt{\frac{\sum_{u,i \in T} (r_{ui} - \hat{r}_{ui})^2}{|T|}}, \quad MAE = \frac{\sum_{u,i \in T} |r_{ui} - \hat{r}_{ui}|}{|T|}, \quad (10)$$

where T is the testing set. Netflix thinks RMSE increases penalties for wrong prediction (by square). Therefore, RMSE is more demanding. Researches have shown that if a scoring system is set up based on the integer rating, rounding the prediction ratings would improve MAE results [16].

We select five methods to do the comparison. The methods include the original weighted Slope One (WSO), nonlinear and linear transformation of WSO combined with two kinds of $LDSD$. We predict more than 5,500 ratings given by 442 users in each test.

The results shown in Table 1 can be analyzed from three perspectives. Firstly and most obviously, linear transformation of WSO with $LDSD_d$ has the best performance. Linear transformation of WSO with $LDSD_i$ also has a significant improvement. It indicates that our approach has capability to make rating prediction more accurate. Secondly, two nonlinear methods do not perform well. From this point we can see that replacing the weight directly does not improve the prediction and even declines sometimes. At last, results of run time tell us none of the transformations change the computational complexity.

By comparing the second and third rows (or the fourth and fifth), we found that $LDSD_d$ contains more effective information than $LDSD_i$. Direct links give firmer relationships, while indirect links may bring about some “noises.” For linear transformation of WSO with $LDSD_d$, we also observed that the change of results

Fig. 3 RMSE— α **Table 2** Precision results

Method	Precision	Recall	Coverage
Weighted Slope One (WSO)	36.93%	7.14%	48.27%
Soft trans.WSO with $LDSD_d$	39.72%	7.68%	48.19%

when free parameter α varies. In Fig. 3, RMSE is 1.077, which equals the result of original WSO when α is zero and reaches the lowest point when α is in the vicinity of 0.65. This leads us to conclude that the linear integration indeed has a positive effect. The decreasing RMSE shows that $LDSD$ has found the correct implicit relations and is helping us to make more accurate predictions. Separately using semantic distances or original weight is not in line with our expectations and the actual results have convinced it.

Top-N Recommendation. Another important evaluation method is the Top- N recommendation. When a website provides a recommendation service, it in general gives the user a personalized recommendation list, and this is so-called Top- N recommendation [6]. Top- N recommendation's prediction accuracy can be measured by precision/recall.

Let $R_N(u)$ be the recommendation list which contains N items, $T(u)$ be the set of items actually chosen by user and U be the set of all users:

$$\text{Recall} = \frac{\sum_{u \in U} |R_N(u) \cap T(u)|}{\sum_{u \in U} |T(u)|}, \quad \text{Precision} = \frac{\sum_{u \in U} |R_N(u) \cap T(u)|}{\sum_{u \in U} |R_N(u)|}. \quad (11)$$

Movie recommender systems' ultimate goal is to recommend the movies that users most want to watch but not to predict the ratings. So Top- N recommendation is more in line with the actual requirements of applications. The results in the Table 2 show that the linear transformation of WSO with $LDSD_d$ performs best in this test, i.e., our approach improves the accuracy of the recommendations. Also, our approach does not ignore the influence of user behavior.

4.2 Coverage

Coverage describes a recommender system's ability to explore the items long tail. The simplest definition of the coverage is the proportion of the collection of items that the recommender system can recommend to total items [14]. Let I be the set of all items, the coverage is defined as follows:

$$\text{Coverage} = \frac{|\bigcup_{u \in U} R(u)|}{|I|}. \quad (12)$$

The content providers may pay more attention to the coverage. One hundred percent coverage means that the recommender system can recommend each item to at least one user. So a good recommendation system not only needs to have high user satisfaction but also has high coverage. The results in Table 2 indicate that our approach improves the prediction accuracy on the basis of not reducing the coverage. So the diversity of recommendations is preserved.

5 Conclusion

Using semantic technologies to improve recommender system is an emerging research area. In this paper, we proposed a method to integrate recommender system based on Slope One with semantic technologies, which outperforms the original schemes and does not undermine the simplicity and efficiency. We also provided an implementation of our approach and conducted experiments on a well-known movie dataset.

In future work, we look forward to proposing more general mapping methods especially at the instance level, in order to meet more kinds of application requirements. We will study new approaches for incremental semantic distance computation to support dataset update. Additionally, we hope to integrate not only Slope One CF algorithm but also some sophisticated CF algorithms with Linked Data and Semantic Web technologies.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 61003018 and 61021062, in part by the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20100091120041, and also in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK2011189.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Sci. Am.* **5**, 29–37 (2001)

3. Bizer, C., Heath, T., Idehen, K.U., Berners-Lee, T.: Linked data on the web. In: Proceedings of WWW, 2008
4. Cacheda, F., Carneiro, V., Fernandez, D., Formoso, V.: Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *IEEE Trans. Web* **5**(1), 2:1–2:33 (2011)
5. Cross, V.: Fuzzy semantic distance measures between ontological concepts. In: Proceedings of IEEE Annual Meeting of the Fuzzy Information, 2004
6. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of RecSys, 2010
7. Ge, J., Qiu, Y.: Concept similarity matching based on semantic distance. In: Proceedings of SKG, 2008
8. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inform. Syst.* **22**(1), 5–53 (2004)
9. Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: Proceedings of CHI, 1995
10. Kiesel, M., Grimnes, G.A.: DBTropes – A linked data wrapper approach incorporating community feedback. In: Proceedings of EKAW, 2010
11. Lemire, D., MacLachlan, A.: Slope One predictors for online rating-based collaborative filtering. In: Proceedings of SDM (2005)
12. Passant, A.: Measuring semantic distance on linking data and using it for resources recommendations. In: Proceedings of AAAI Spring Symposium Linked Data Meets, 2010
13. Passant, A.: dbrec – Music Recommendations Using DBpedia. In: Proceedings of ISWC, 2010
14. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender Systems Handbook, pp. 257–298, 2011
15. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**(5), 1–19 (2009)
16. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Res.* **30**(1), 79–82 (2005)
17. Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L.: Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inform. Syst.* **26**(3), 13:1–13:37 (2008)
18. Yao, J., Yao, J., Yang, R., Chen, Z.: Product recommendation based on search keywords. In: Proceedings of WISA, 2012

Finding, Extracting, and Building Academic Linked Data

Peng Wang and Xiang Zhang

Abstract This paper addresses the problem of finding and extracting academic information from conference Web pages, then organizing academic information as ontologies, and finally generating academic linked data by matching these ontologies. The main contributions include (1) a topic-crawling method and lightweight crawling method based on search engine is presented. Crawling seeds, relevant websites filter, and crawling update strategy are discussed. (2) A new vision-based approach for extracting academic information is proposed. It first segments Web pages into text blocks and then classifies these text blocks into predefined categories. The initial classification results are improved by post-processing. Finally, academic information is extracted from the classified text blocks. (3) A global ontology is used to describe the background domain knowledge, and then the extracted academic information of each website is organized as local ontologies. Finally, academic linked data is generated by matching all local ontologies.

1 Introduction

With the popularity of semantic Web technologies and the emergence of intelligent applications such as semantic search, more and more plain or semi-structured Web data need to be reorganized as semantic data, which is the foundation of many intelligent applications. Linked data is such large-scale semantic data. Recently, more and more linked data such as DBpedia [1], Freebase, and Google knowledge graph is used in many fields including knowledge engineering, machine translation, social computing, and information retrieval. Academic linked data is important for academic social network analysis and mining [2]. However, current academic linked data is mainly based on database like DBLP or researcher homepages on

P. Wang (✉) • X. Zhang

School of Computer Science and Engineering, Southeast University, Nanjing, China

e-mail: pwang@seu.edu.cn; x.zhang@seu.edu.cn

the Web [3], and it mainly describes paper publication information. Therefore, academic activity knowledge is not included by current academic linked data. Academic conferences websites not only contain much paper information but also contain much academic activity information including research topic, conference time, location, participants, and awards. Obtaining such information is not only useful for predicting research trends and analyzing academic social network but also is the important supplement to current academic linked data. Since academic conferences Web pages are usually semi-structured and content are diverse, there is no effective way to automatically find, extract [4, 5], and organize the academic information to linked data.

This paper addresses and implements a 3-phase method for crawling, extracting academic information from conference Web pages, and then generating academic linked data. To crawl academic conference websites, we design a topic crawler based on search engine. To extract academic information, we first use a new algorithm to segment the page into text blocks and then classify each text block into predefined categories based on its vision features and semantic features. To generate linked data, we first build a global ontology as background knowledge and then generate a local ontology for each conference website; finally, all ontologies are matched as an academic linked data.

2 Crawling Academic Conference Websites

In order to find academic information, we need to crawl conference websites automatically. It is a topic-crawling problem. Most Web pages are linked by hyperlink between them. For the websites in a domain, there are some hubs and authorities [6]. Hubs refer to the websites which have many hyperlinks to authority pages. Authorities are the websites that there are many hyperlinks link to them. In addition, in a special domain, the link density between pages are higher than the link density between these pages and pages of other domains [7], namely, a Web community is a collection of Web pages in which each member page has more hyperlinks within the community than outside it. We still use hyperlink to crawl more websites in academic conference domain. To avoid checking all Web pages one by one, we design a topic crawler based on Google search engine API. This method can improve the crawling efficiency. Figure 1 shows our crawling model, which has five main steps as follows:

1. *Input processing*: Given some seeds, which contain short name and full name, we generate a series of queries with year and run these queries by search engine, and then we can obtain URLs of conferences.
2. *Crawling and downloading*: We download pages according to URLs returned in step 1 and save pages into local database. Meanwhile, we extract links to external websites.

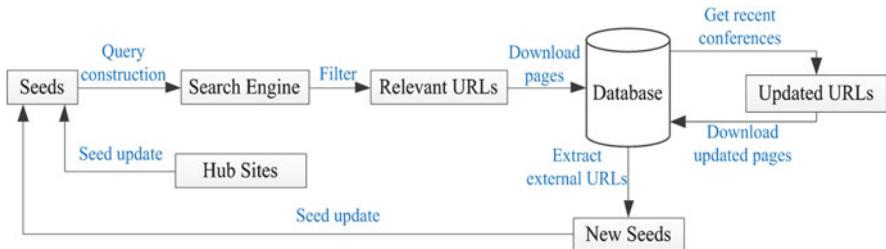


Fig. 1 The lightweight topic-crawling model

3. *Filtering relevant pages*: For each external URL, we calculate its relevant score to determine whether it is a relevant site. If a URL is relevant, we will extract conference short name and full name and add it as a new seed.
4. *Hub conference pages*: If there is no seed, a manually maintained page which contains a list of conference URLs is imported. New seeds can be obtained from these hub pages.
5. *Crawling updating*: We will crawl obtained URLs periodically; the lasted conference pages will be re-crawled more quickly than older ones.

Crawling Seeds: The principle of our crawler is also employing the link between different conference websites. Namely, links to external websites can help us to find new conference websites. Our initial seeds only contain conference short names and full names. A seed is a 2-tuple: $\langle \text{ShortName}, \text{FullName} \rangle$. For a given seed, we can construct a series of query key words like $\text{ShortName} + \text{Year}$ then use Google API to find the right conference websites. In the query key words, Year is decrease from 2012 by step 1. If we cannot find relevant conference websites in sequent 3 years, we stop to construct new query key words. For example, $\langle \text{IJCAI}, \text{International Joint Conference on Artificial Intelligence} \rangle$ is a seed, and then $\text{IJCAI}2012$, $\text{IJCAI}2011$, and $\text{IJCAI}2010$ are the corresponding query key words. When a conference website is found, we analyze all external URLs. If a URL links to a new conference, we can add this conference into seed queue. Since we don't know the real link distribution between academic conference websites, initial seeds should cover conferences in different rank.

Relevant Website Filtering: For query key words, we only examine top 5 results returned by Google API. We need to determine which result is an academic conference websites that we find. We use a three-level filtering method based on SVM to find relevant conference websites. First level is the preliminary topic filter based on URL string, short name, and full name of a website. Second level uses a collected document of conference key words to filter websites. For remaining websites, the third-level filter extracts key words of the main page and then calculates similarity between a conference background knowledge document and these key words to determine whether the websites are relevant. This three-level filter can remove the unrelated websites efficiently.

Crawling Update Strategy: According to the fact that Web pages updates satisfy Poisson distribution, we calculate the average update periods of Web pages in a website as its initial update period. Then we crawl a website periodically and adjust its crawling update period according to changes of Web pages.

Conference Hub Websites: To compensate for limitations of academic conference websites' lack of links between them, we maintain some conference hub websites, which contain a lot of academic conference list and corresponding external URLs. Once there is no seed to be crawled, we periodically visit these conference hub websites and add these conferences as new seeds.

3 Extracting Academic Information

We propose a new approach to extract useful academic information from conference Web pages automatically. First, given a sample conference Web page, it is segmented into a set of text blocks using a hybrid method which combines vision-based segmentation method and DOM-based segmentation method. Second, text blocks are classified into predefined categories, in which each text block is represented by several features including vision features and semantic features. Third, post-processing on the initial classification results can improve the classification accuracy by introducing semantic analysis and context analysis.

3.1 Page Segmentation Algorithm

To extract the academic information, we first segment the text on Web pages into blocks by VIPS [8], which is a vision-based page segmentation algorithm. VIPS can use Web page structures and some vision features, such as background color, text font, text size, and distance between text blocks, to segment a Web page. In VIPS, a Web page Ω is represented as a triple $\Omega = (O, \Phi, \delta)$. $O = (\Omega_1, \Omega_2, \dots, \Omega_N)$ is a finite set of blocks. All blocks are not overlapped. Each block can be recursively viewed as a sub-Web page associated with substructure induced from the whole page structure. $\Phi_i = (\varphi_1, \varphi_2, \dots, \varphi_T)$ is a finite set of separators, including horizontal separators and vertical separators. δ is the relationship of every two blocks in O and can be expressed as $\delta = O \times O \rightarrow \Phi \cup \{\text{NULL}\}$. Figure 2 shows the segmented results of ACL2011 main page, where the page is segmented into two blocks VB1 and VB2, which are separated by φ_1 . VB1 and VB2 are then segmented into more small blocks. Text blocks can be constructed as a vision tree, which assures that all leaf nodes only contain text information.

VIPS can obtain good segmentation results for most Web pages, but we find it will lose important information when dealing with some Web pages. Therefore, we introduce DOM-based analysis to improve the VIPS segmentation results. We propose a hybrid page segmentation algorithm to combine the VIPS algorithm

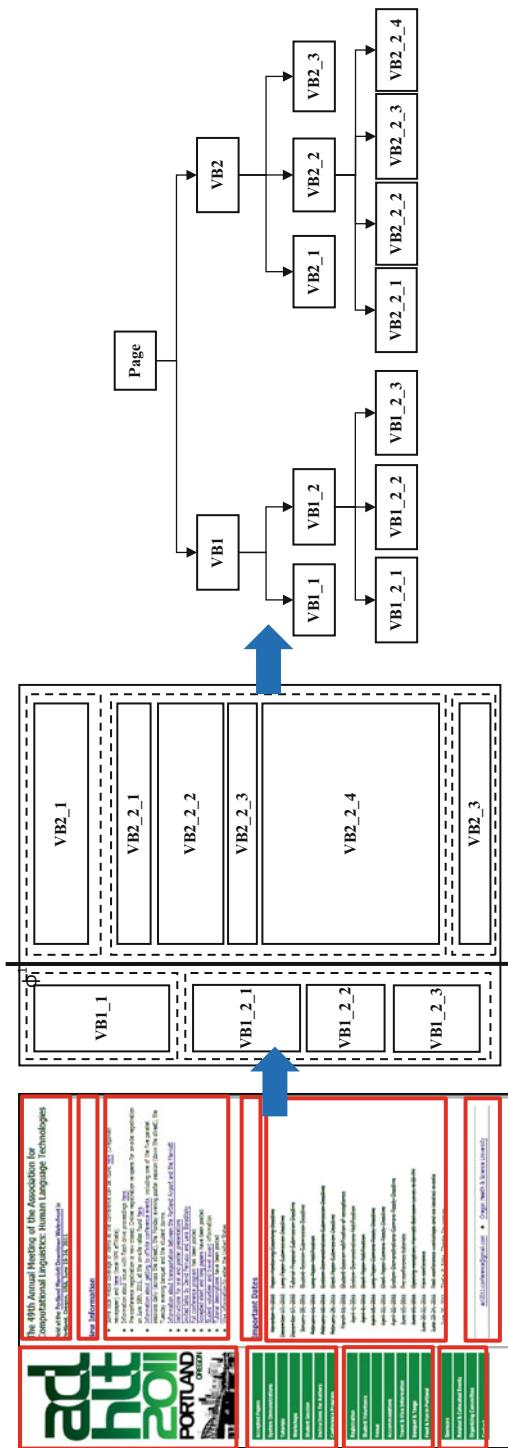


Fig. 2 An example of VIPs

and the DOM-based algorithm. First, our algorithm segments an input Web page into text blocks using VIPS algorithm. Second, it removes the noise blocks in the input Web page (navigation blocks, copyright blocks, etc.) using some heuristic rules. Third, it segments the input Web page into text blocks using DOM-based algorithm. Finally, it combines the text blocks got by VIPS and DOM-based algorithm to archive a more complete segmentation result. We call the new vision tree of text blocks produced by our algorithm the complete vision tree.

Algorithm 1: Hybrid Page Segmentation Algorithm

Input: a conference Web page P

Output: an array of segmented texts ST[]

```

1      begin
2          ST1[]←VIPS(P) //Segment the pages by VIPS
3          foreach ST1[i]□ST1[] do
4              if(ST1[i] in HeaderArea || NavigationArea|| CopyrightArea) then
5                  remove ST1[i] from ST1[]
6              end //Remove non-text blocks
7              ST2[]←DOMPS(P) //Analyzing the page by DOM structure
8              int recordJ= -1//Record last visited position in ST2[]
9              foreach ST1[i]□ST1[] do
10                 j←FindInST2(ST1[i],ST2[],recordJ)//Find blocks of ST1[i] in ST2[]
11                 if (j == -1) continue //No blocks of ST1[i] in ST2[]
12                 if (j > recordJ+1)
13                     ProcessLost(ST2[],recordJ+1,j-1) //Add lost blocks
14                 if ( ST1[i] == ST2[j] ) ST[].add( ST1[i] ) //Add text blocks to ST[]
15                 else ST[].add( ST2[j] )
16                 recordJ← j
17             end
18         end
19         Function FindInST2(ST1[i],ST2[],recordJ)
20         begin
21             j← recordJ+1
22             while (j <ST2[],size)
23                 if (ST2[j] StarWith ST1[i]) return j
24                 j++
25             return -1
26         end
27         Function ProcessLost(ST2[], recordJ+1, j-1)
28         begin
29             k←recordJ+1
30             while (k≤j-1)
31                 ST[],add(ST2[k])
32         end

```

3.2 Text Block Classification

The complete vision tree preserves some structure information of Web pages. However, it is not enough for extracting the academic information from text blocks. This paper transforms the academic information extraction problem to a classification problem. Namely, we classify text blocks; then academic information can be extracted easily from the classification results.

There are three types of academic information on a conference website: (1) information about conference events – conference name, time, location, submission deadline, submission URL, and accepted papers; (2) information about conference topics – call for papers (or workshops/research papers/industrial papers), topics of interests, sessions, tracks, and so on; and (3) information about related people and institute – organizers, program committee, authors, companies, universities, countries, and so on. Therefore, we divide the text blocks into 10 categories: (1) **DI** describes date information; (2) **PI** describes location information; (3) **AR** refers to top-level information such as research area; (4) **TO** refers to research topics, and an AR block may have some corresponding TO blocks; (5) **PO** describes the role of people in a conference such as being the speaker and chair; (6) **PE** refers to information of a person; (7) **PA** is the information about papers; (8) **CO** refers to the blocks which are combined by the above 7 categories of blocks; (9) **R** refers to the interested blocks but do not belong to any categories; and (10) **N** refers to the blocks that do not only belong to any categories but are also not related to academic information. Figure 3 shows each category and corresponding examples.

DI(dateItem)	Author notification: May 19, 2011
PI(placeItem)	Palm Springs, California
PO(position)	Program committee
PE(peopleItem)	<ul style="list-style-type: none"> • Noah Smith Carnegie Mellon University, USA
AR(area)	Call for Papers: Industrial and Applications Track
TO(topic)	<ul style="list-style-type: none"> ✍ Citation Analysis, Social Networks for IR ✍ Distributed IR, Peer to Peer Search
PA(paper)	Assigning Documents to Master Sites in Distributed Search Roi Blanco (<i>Yahoo! Research</i>)
CO(collection)	The 21 st ACM International Conference on Information and Knowledge Management (CIKM 2012) will be held from October 29 to November 2, 2012 in Maui, USA. CIKM is a well-known top tier and premier ACM conference in the areas of information retrieval, knowledge management and database. Since 1992, it has successfully brought together leading
N(not related)	Become a Sponsor of AAAI-12
R(related)	ACM CIKM review is double-blind. Therefore, please anonymize your submission. Papers cannot exceed 10 pages in length. There is no short paper track. The

Fig. 3 Examples for all text block categories

For a text block, we can construct its feature vector according to vision [9], key words, and text content. For example, given a text block “Paper Submission Due: Friday, May 6, 2011 (23:59 UTC - 11)” and its HTML source code “Paper Submission Due: Friday, May 6, 2011 (23:59 UTC - 11)”, its feature can be constructed as (1) Vision features: isTitle=false, isHeader=false, startWithLi=true, left=(280-0)/950=0.3 (page width:950, left margin: 0, text left margin:280), with=640/950=0.7 (text width: 640); (2) Key word features: nearestTitle=DI (its nearest and isTitle=true blocks is about date information), dateNum=2 (it contains 2 date words: Submission and Due), paperTypeNum=1 (it contains 1 key word about paper: Paper); (3) Text content features: fontSize=0, fontWeight=0, textLength=58, textLink=0, wordNum=11, nameNum=5, wordToName=11/5=2.2.

After getting features of text blocks, we use Bayes Network to classify text blocks. The classified results can be improved by post-processing, which includes repairing wrong classified results and adding missed classified results. For wrong classified results, we check the features of each text blocks to repair them. For missed classified results, we further introduce some context features and some heuristic rules to find missed classified blocks. Till now, we can easily extract academic information from these text blocks. For most text blocks, they are the academic information to be extracted.

4 Building Academic Linked Data

4.1 Global Ontology and Local Ontology

To obtain the academic linked data, we need to organize the extracted academic information. Therefore, we first manually build a global ontology as background knowledge of academic domain, then automatically construct local ontologies for each conference website.

After investigating a lot of conference websites and some ontologies related to academic domain, we manually construct a global ontology for describing the knowledge of academic conference. The global ontology contains 97 concepts and 27 properties. Figure 4 shows part of global ontology. Besides the hierarchy, concepts can be related by properties. For example, property *hasAuthor* can link two concepts *Paper* and *Author*, which are domain and range of *hasAuthor*. Global ontology has no instances, and it is stored as an ontology language RDF file.

For a local ontology, its concepts and properties are contained in global ontology. We don't consider the new knowledge which is not described in global ontology. Therefore, for an extracted academic information, it is either a concept or an instance of local ontology. However, not all extracted information can be translated to concepts or instances directly. Therefore, the concepts and instances should

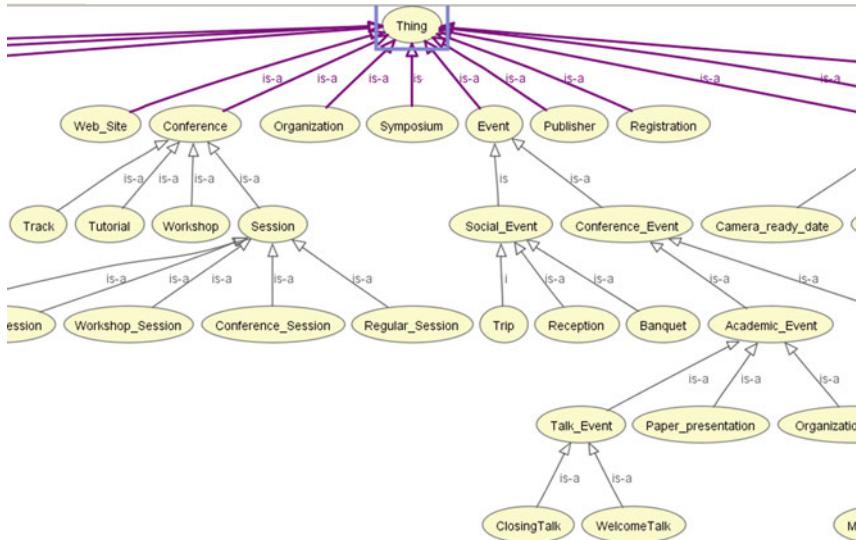


Fig. 4 Part of global ontology

be determined by the context of the academic information. For example, a paper information usually appears in PA text blocks and contains the title and authors, so it is an instance of concept Paper; all authors are instances of concept Author, titles will be the property values, and authors will be the values of *hasAuthor*. Through the above process, we can generate a local ontology for each conference website.

4.2 Linked Data Generating

In order to link all isolate local ontologies as the academic linked data, we need to match these ontologies. The linguistic-based method is a popular ontology-matching technique. For the reason that text in ontology can describes some semantics, the linguistic-based matching method can discover matching results by calculating similarities between text documents. For an academic conference local ontology, it contains regular and abundant text; therefore, the linguistic-based method is suitable. We use the ontology-matching API provided by ontology-matching system Lily [10] to discover matching results between local ontologies. Lily is an excellent matching system and can produce high-quality matching results.

Our ontology-matching strategy is calculating matches for each two ontologies, then associating all ontologies into the linked data by these matches. This strategy

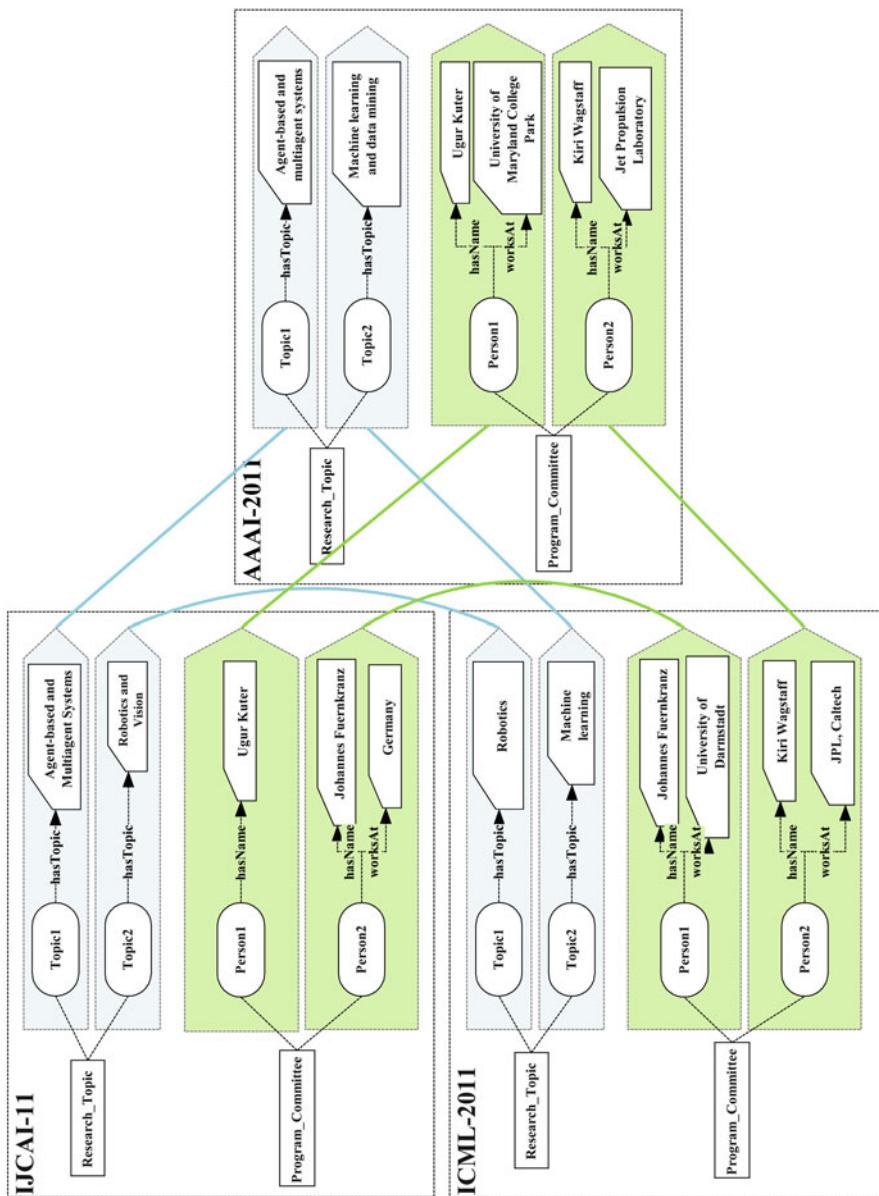


Fig. 5 Part of linked data

has the benefit of handling a number of generated local ontologies, but its disadvantage is consuming a lot of time for matching many ontologies. Figure 5 shows part of linked data for three conferences. If two instances are matched, they can be combined in linked data.

5 Experiment Results

5.1 System Implementation and Dataset

We realize the system in Java and some open-source APIs. The Web page segmentation module is implemented in C#. We also use Weka, an open-source machine learning library, to classify text blocks. Our experimental results are obtained on a PC with 2.40GHz CPU, 2GB RAM, and Windows 7.

We collect 50 computer science conference websites in the computer science field, which has 283 different Web pages and 10,028 labeled text blocks. In order to evaluate our approach, 10 students manually tag all the text blocks as reference results. The results are saved in CSV files.

We randomly select 10 sites which contain 62 pages as training dataset for constructing Bayesian network model. Other 40 conference websites are used as test dataset. We use Precision, Recall, and F1-Measure as criteria to measure the system performance.

5.2 Experimental Results and Analysis

In Fig. 6, red nodes are 252 rank A conferences, green nodes are 96 rank B conferences, and blue nodes are 113 rank C conferences. We can observe two facts: (1) There are many links between academic conference websites. (2) High-quality conferences such as rank A have few links to other rank conferences, and low-quality conferences such as rank C have a lot of links to rank A and rank B conferences. It means that if we want to crawl high-quality conferences as much as possible, we should have enough seeds of rank A.

Figure 7 further shows the internal link graph of 252 rank A computer science conferences. We can see that only few nodes have a lot of links to other nodes. If these nodes are the conferences with long history, then new conferences will have links to previous conferences. It means we can easily crawl same series conferences, but it is difficult to crawl the conferences which are not related.

Figure 8 shows that our crawling method can steadily find more than 3,109 conference websites during 62 h. It means that we can find 50 new conference websites in 1 h. Actually, our system has crawled thousands of computer science conferences

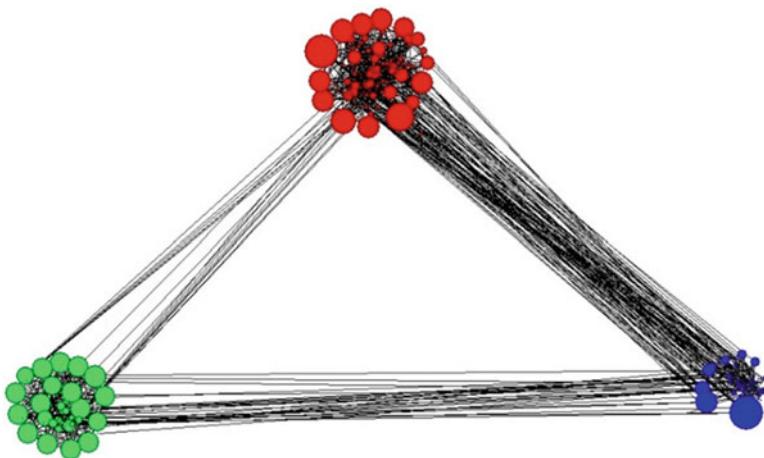


Fig. 6 Link graph between conferences of different rank

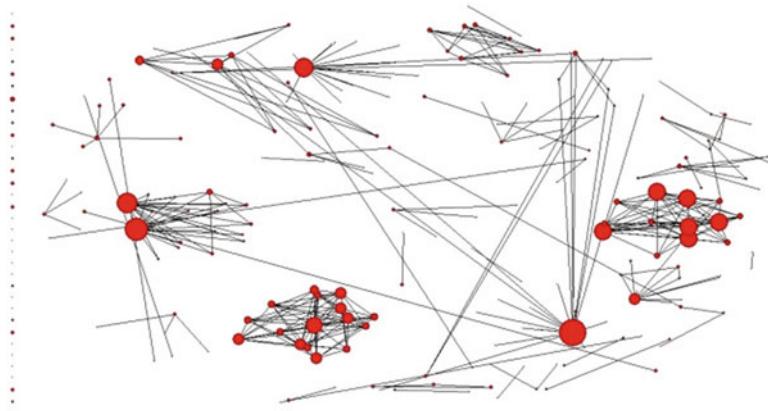


Fig. 7 Link graph of rank A conferences

in one month, and these conferences almost cover the existing conference lists which are maintained manually.

Table 1 shows the results of text block classification on 20 randomly selected websites. We have two conclusions: (1) The initial classification results only have an average of 0.75 precision, 0.67 recall, and 0.68 F1-measure. Therefore, the post-processing, the classification results have improved to an average of 0.96 precision, 0.98 recall, and 0.97 F1-measure. Therefore, the post-processing plays key roles in academic information extraction. (2) Some text blocks like DI, PO, PE, and TO, which have clear vision and text content features, have better classification results. The average F1-measure on these blocks is 0.99.

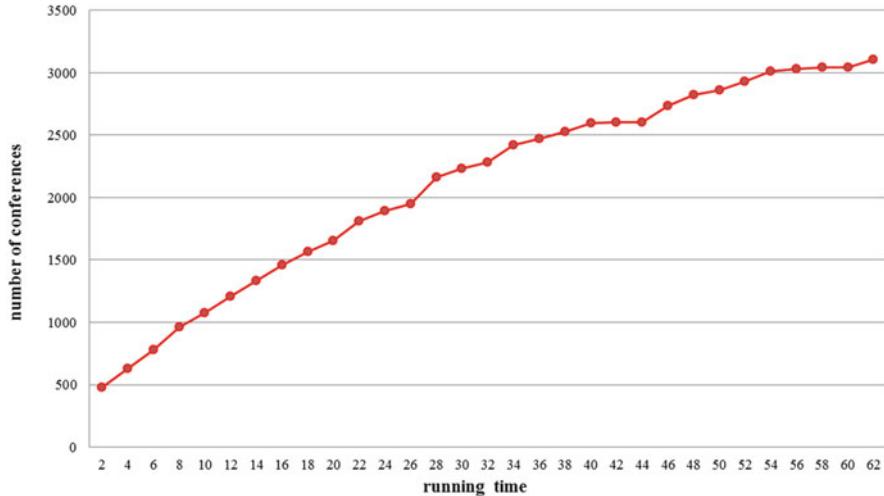


Fig. 8 Insert caption to place caption below figure

Table 1 Results of text block classification

	Initial classification result			Post-processing result		
	P	R	F1	P	R	F1
DI	0.95	0.75	0.84	0.96	0.99	0.98
AR	0.84	0.91	0.87	0.92	0.98	0.95
TO	0.90	0.41	0.57	0.99	0.99	0.99
PO	0.80	0.69	0.74	0.99	0.98	0.99
PE	0.85	0.60	0.71	0.99	0.99	0.99
PI	0.79	0.72	0.75	0.97	0.97	0.97
CO	0.35	0.80	0.49	0.91	0.95	0.93
PA	0.5	0.5	0.5	1	1	1
Avg.	0.75	0.67	0.68	0.96	0.98	0.97

Although we notice that some domain specific rules are used in the post-processing, we have to point out that similar heuristic rules could be obtained for other domains. Therefore, the post-processing is a general way to improve the extraction quality, and we need to construct efficient heuristic rules.

Table 2 shows the results of generated 50 local ontologies; we examine the ontology elements according to conference, date, place, people, topic, and related document. The generated ontology elements have at least 0.91 precision, 0.94 recall, and 0.95 F1-measure.

Table 2 Results of generated local ontologies

	Conference	Date	Place	People	Topic	Doc
Correct	120	1410	100	1500	1260	20
Error	10	0	10	10	0	0
Missed	0	30	0	90	0	0
Precision	0.92	1.00	0.91	0.99	1.00	1.00
Recall	1.00	0.98	1.00	0.94	1.00	1.00
F1	0.96	0.99	0.95	0.97	1.00	1.00

6 Conclusion and Future Work

This paper addresses the problem of finding and extracting academic information from conference Web pages, then organizing academic information as ontologies, and finally generating academic linked data by matching these ontologies. The main contributions include the following: (1) A lightweight topic-crawling method based on search engine is presented. (2) A new approach to extract academic information is proposed. (3) A global ontology is used to describe the background domain knowledge, and then the extracted academic information of each website is organized as local ontologies. Finally, academic linked data is generated by matching local ontologies.

During the academic information extraction phase, some heuristic rules are used in the post-processing. These rules are very important for producing good extraction results, but they are not suitable for dealing with the pages in other domain. In the future work, we will try to use learning methods to generate these rules automatically. In addition, when matching thousands of local ontologies, there are three obvious problems: (1) Most information of local ontologies is instance data; we need to use instance matching method. (2) The matched ontology will be very large; therefore, we will face to match a small local ontology to a large ontology. (3) When local ontologies are changed, we need to maintain the academic linked data.

Acknowledgements This work is supported by the NSF of China (61003156 and 61003055) and the Natural Science Foundation of Jiangsu Province (BK2009136 and BK2011335).

References

1. Bizer, C., Lehmann, J., Kobilarov, G., et al.: DBpedia – a crystallization point for the Web of Data. *J. Web Semant.* **7**, 154–165 (2009)
2. Tang, J., Zhang, J., Yao, L., Li, J., et al.: ArnetMiner: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV (2008)
3. Tang, J., Zhang, D., Yao, L.: Social network extraction of academic researchers. In: Proceedings of 2007 IEEE International Conference on Data Mining, Omaha, NE (2007)

4. Chang, C.-H., Kayed, M., Gergis, M.R., Shaalan, K.: A survey of web information extraction systems. *IEEE Trans. Knowl. Data Eng.* **18**, 1411–1428 (2006)
5. Laender, A., Ribeiro-Neto, B.A., da Silva, A.S., Teixeira, J.S.: A brief survey of web data extraction tools. *SIGMOD Record* **31**, 84–93 (2002)
6. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA (1998)
7. Flake, G.W., Lawrence, S., Lee Giles, C., Coetzee, F.M.: Self-organization and identification of web communities. *IEEE Comp.* **35**, 66–71 (2002)
8. Cai, D., Yu, S., Wen, J.-R., Ma, W.-Y.: VIPS: a vision-based page segmentation algorithm. Microsoft Technical Report (2003)
9. Liu, W., Meng, X., Meng, W.: ViDE: a vision-based approach for deep web data extraction. *IEEE Trans. Knowl. Data Eng.* **22**, 447–460 (2010)
10. Wang, P., Xu, B.: Lily: ontology alignment results for OAEI 2009. In: The 4th International Workshop on Ontology Matching (OM2009), Washington, DC (2009)

Proactive Recommendation Based on \mathcal{EL} Concept Learning

Jianfeng Du, Shuai Wang, Bohong Lin, Xiaoli Yao, and Yong Hu

Abstract This paper proposes a novel knowledge-based approach to proactive recommendation. It exploits \mathcal{EL} concept learning to automatically model user preferences and is different from traditional knowledge-based approaches that require users to input explicit needs. Given an item being browsed, a set of marked items and an integer threshold k which is used to determine user preferences (where individuals are treated as items), the approach learns *recommendatory restricted \mathcal{EL} concepts* and returns unmarked instances of these concepts as recommendations. A *recommendatory restricted \mathcal{EL} concept* is a most specific *restricted \mathcal{EL} concept* that has at least k marked items, the item being browsed and at least one other unmarked item as instances. Intuitively, a *recommendatory restricted \mathcal{EL} concept* models a maximal set of user preferences. To guarantee that a learned concept has a finite size and the learning process is efficient, the proposed approach does not handle general \mathcal{EL} concepts but only *restricted \mathcal{EL} concepts* which have restrictions on the number of nested quantifiers and the inner occurrence of existential restrictions. This paper treats the problem of learning *recommendatory restricted \mathcal{EL} concepts* as the problem of maximal frequent itemset mining (MFI-mining) and presents an efficient MFI-mining algorithm to learn these concepts. Experimental results demonstrate the feasibility of the proposed approach in terms of efficiency and scalability.

J. Du (✉)

Guangdong University of Foreign Studies, Guangzhou 510006, China

State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

e-mail: jfd@gdufs.edu.cn

S. Wang • B. Lin • X. Yao • Y. Hu

Guangdong University of Foreign Studies, Guangzhou 510006, China

e-mail: shuaishuai@gmail.com; 956767243@qq.com; 470407938@qq.com;

henryhu200211@163.com

1 Introduction

Recommender systems [7] try to learn user preferences over time and to automatically suggest products to fit the user needs. They are widely used in E-commerce sites to suggest products to customers and to improve the look-to-buy ratio.

Collaborative filtering and content-based filtering are two popular recommendation approaches. Collaborative filtering aggregates product ratings from customers to recommend new products. Amazon.com is a well-known example that exploits a collaborative filtering approach. In its book section, the system encourages direct feedback from customers about books they already read. Afterwards, the system pushes recommendations to the customer for books he/she might like. Content-based filtering exploits the past and current preferences of a specific user to build new recommendations to the user. For example, NewsDude [3] observes what online news stories a user has read and learns to present the user with articles of his/her interest. Content-based systems are usually implemented as classifier systems based on machine learning techniques. In collaborative filtering a large number of history user–system interactions are required to build reliable recommendations. Content-based systems exploit only the current user’s data to build a recommendation, but they require a description of user preferences that is provided as input to build recommendations. Both approaches deliver poor recommendations when not trained with lots of product ratings or patterns of user preferences. This limitation motivates a third approach, knowledge based, which tries to better use existing knowledge to build a more accurate model by requiring less training instances.

The knowledge-based approach is considered complementary to the above two approaches [4]. There are two basic specifics of knowledge-based recommender systems: case based and constraint based [7]. Both are similar in some aspects: user requirements are collected, repairs for inconsistent requirements are proposed whenever no solutions are found, and recommendation results are explained. The major difference lies in the way to compute solutions. Case-based recommender systems determine recommendations on the basis of similarity metrics, whereas constraint-based ones predominantly exploit explicit knowledge on how to relate user requirements with item features.

The knowledge-based approach does not depend on a base of user ratings, hence avoiding the *ramp-up problem* in collaborative filtering and content-based filtering. This problem can be described as follows: until there are a large number of users whose habits are known, the system cannot be useful for most users; until a sufficient number of rated items has been collected, the system cannot be useful for a particular user [4]. However, either case-based or constraint-based recommender systems require users to input explicit requirements and are hard to provide proactive recommendation. Proactivity means that the system pushes recommendations to the user whenever the current situation seems appropriate, e.g., when the user keeps browsing a product for a few seconds. The following example shows a scenario where proactive recommendation is needed.

Example 1. A user is looking for a rental room. There are six rental rooms available whose descriptions are given below, where HEMC is short for High Education Mega Center, SYSU for Sun Yat-sen University, and GDUFS for Guangdong University of Foreign Studies.

Room ₁	1,000 yuan per month for renting, towards the south, at the 3rd floor
Room ₂	1,200 yuan per month for renting, towards the east, at the 9th floor
Room ₃	1,400 yuan per month for renting, towards the south, at the 5th floor, near HEMC Line 3 which reaches the HEMC sector of GDUFS
Room ₄	1,600 yuan per month for renting, towards the south, at the 9th floor
Room ₅	1,800 yuan per month for renting, towards the east, at the 6th floor, near Metro Line 1 which reaches the HEMC sector of SYSU
Room ₆	2,000 yuan per month for renting, towards the south, at the 8th floor, near HEMC Line 1 which reaches the HEMC sector of GDUFS

The user has marked Room₁ and Room₃ as his possible choices and has been browsing Room₆ for a few seconds since certain features of Room₆ are attracting him. But the user cannot guarantee that Room₆ is also a possible choice. In this situation, a recommender system should push some recommendations with explanations to the user to help him enlarge the set of possible choices. \square

A new approach to proactive recommendation is required for similar scenarios as that shown in the above example. This approach will take an item being browsed and a set of marked items as input and return a set of recommendations which are unmarked items that have as many potentially preferred features as possible. To develop such an approach, we assume that a combination of features is potentially preferred if it is owned by the item being browsed and sufficiently many marked items. We introduce a parameter k to specify the minimal number of marked items for ensuring that a combination of features is preferred by a considering user. We need to lay a logical foundation for the proposed approach so that explanations for recommendations can be generated by existing logic-based reasoning methods. To this end we covert descriptions of items to description logic (DL) [2] assertions. We choose the DL \mathcal{EL} [1] to express converted assertions because \mathcal{EL} allows for efficient concept learning based on the search of \mathcal{EL} -description trees [6] while it has sufficient expressivity to model concepts in many applications such as life science.

In this paper, we do not focus on how to convert descriptions of items to \mathcal{EL} assertions, because the conversion can be done by existing information extraction techniques. We only give an example below to show what kind of data that the proposed approach works on.

Example 2. The descriptions of rental rooms, given in Example 1, are converted to the following \mathcal{EL} assertions on which the proposed approach will work.

Room ₁	Room(rm_1), $\exists \text{hasPrice}.\text{AtMost1500}(rm_1)$, $\text{TowardsSouth}(rm_1)$, $\exists \text{atFloor}.\text{AtMost6}(rm_1)$
Room ₂	Room(rm_2), $\exists \text{hasPrice}.\text{AtMost1500}(rm_2)$, $\text{TowardsEast}(rm_2)$, $\exists \text{atFloor}.\text{AtLeast7}(rm_2)$
Room ₃	Room(rm_3), $\exists \text{hasPrice}.\text{AtMost1500}(rm_3)$, $\text{TowardsSouth}(rm_3)$, $\exists \text{atFloor}.\text{AtMost6}(rm_3)$, $\text{locatesNear}(rm_3, \text{HEMCLine3})$, $\text{Route}(\text{HEMCLine3})$, $\text{reaches}(\text{HEMCLine3}, \text{GDUFS_HEMC})$, $\text{HEMCNorthLocation}(\text{GDUFS_HEMC})$
Room ₄	Room(rm_4), $\exists \text{hasPrice}.\text{MoreThan1500}(rm_4)$, $\text{TowardsSouth}(rm_4)$, $\exists \text{atFloor}.\text{AtLeast7}(rm_4)$
Room ₅	Room(rm_5), $\exists \text{hasPrice}.\text{MoreThan1500}(rm_5)$, $\text{TowardsEast}(rm_5)$, $\exists \text{atFloor}.\text{AtMost6}(rm_5)$, $\text{locatesNear}(rm_5, \text{MetroLine1})$, $\text{Route}(\text{MetroLine1})$, $\text{reaches}(\text{MetroLine1}, \text{SYSU_HEMC})$, $\text{HEMCNorthLocation}(\text{SYSU_HEMC})$
Room ₆	Room(rm_6), $\exists \text{hasPrice}.\text{MoreThan1500}(rm_6)$, $\text{TowardsSouth}(rm_6)$, $\exists \text{atFloor}.\text{AtLeast7}(rm_6)$, $\text{locatesNear}(rm_6, \text{HEMCLine1})$, $\text{Route}(\text{HEMCLine1})$, $\text{reaches}(\text{HEMCLine1}, \text{GDUFS_HEMC})$, $\text{HEMCNorthLocation}(\text{GDUFS_HEMC})$

The conversion is done by extracting structured information based on a predefined vocabulary and by discretizing numeric values. \square

The proposed approach works on an \mathcal{EL} -ABox which is constituted by those \mathcal{EL} assertions converted from descriptions of items. We also refer to individuals in an \mathcal{EL} -ABox as items. The proposed approach learns most specific concepts, each of which has at least k marked items, the item being browsed and at least one other unmarked item as instances. Intuitively, a most specific concept to be learned (simply called a target concept) has as many preferred features as possible and can be used to deliver at least one recommendation, where an instance of a target concept is regarded as a recommendation to a considering user if it has not been marked or is not being browsed by the user. The approach does not restrict that marked items be explicitly specified by users. Instead, it allows marked items to be automatically determined by a recommender system. For example, we can assume that an item is marked by a user if it has been browsed by the user for a sufficiently long time.

We notice that a target concept may not have a finite size if it is expressed in full \mathcal{EL} . Hence, we introduce two syntax restrictions on a target concept to make it finite while ensuring the learning process to be efficient. The first restriction says that the number of nested quantifiers occurring in a target concept is bounded by a user-specified constant. The second restriction says that there is at most one existential restriction in $\{C_1, \dots, C_n\}$ for every occurrence of existential restriction $\exists r.(C_1 \sqcap \dots \sqcap C_n)$ in a target concept. We call an \mathcal{EL} concept with the above two restrictions a *restricted \mathcal{EL} concept*. The proposed approach only learns target concepts that are restricted \mathcal{EL} concepts. We call such target concepts *recommendatory restricted \mathcal{EL} concepts*.

The most challenging problem of the proposed approach is how to efficiently learn target concepts. To tackle this problem we treat the problem of learning target concepts as the problem of maximal frequent itemset mining (MFI-mining) for which there exist a number of efficient methods. A recommendatory restricted \mathcal{EL} concept is a maximal *frequent* conjunction of concept names or existential

restrictions, where a concept is said to be frequent if its instances include at least k marked items, the item being browsed, and at least one other unmarked item.

We adapt an efficient MFI-mining algorithm proposed in [8] to enumerating all recommendatory restricted \mathcal{EL} concepts. We also empirically verify the proposed approach on Lehigh University Benchmark (LUBM) [5] ontologies with assertions from 2/4/6/8/10 universities. Each LUBM ontology has also a TBox. To take TBoxes into account, we compute an ABox completion for each LUBM ontology by adding assertions that are entailed by the ontology. We then conduct 20 test scenarios for computing all target concepts in an ABox completion, where in each scenario, the marked items and the item being browsed are randomly generated. Experimental results show that the proposed approach is feasible in terms of efficiency and scalability.

2 Syntax and Semantics of Recommendatory Concepts

We first give some preliminaries on the description logic (DL) \mathcal{EL} . Let N_I , N_C , and N_R be disjoint sets of individual names, concept names, and role names, respectively. \mathcal{EL} concepts are built according to the syntax rule $C ::= \top | A | C \sqcap D | \exists r.C$, where $A \in N_C$, $r \in N_R$, and an \mathcal{EL} concept of the form $\exists r.C$ is also called an *existential restriction*. An \mathcal{EL} -TBox is a finite set of *concept inclusions* of the form $C \sqsubseteq D$ or $C \sqsubset D$, where C and D are \mathcal{EL} concepts. An \mathcal{EL} -ABox is a finite set of *assertions* of the form $C(a)$ or $r(a, b)$, where C is an \mathcal{EL} concept, $r \in N_R$, and $a, b \in N_I$.

The semantics of \mathcal{EL} is defined by means of *interpretations* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a nonempty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns binary relations on $\Delta^{\mathcal{I}}$ to role names, subsets of $\Delta^{\mathcal{I}}$ to concept names, and elements of $\Delta^{\mathcal{I}}$ to individual names. The interpretation function $\cdot^{\mathcal{I}}$ is extended to other concepts in a usual way (see [2]). An interpretation \mathcal{I} satisfies a concept inclusion $C \sqsubseteq D$ (resp. $C \sqsubset D$) if $C^{\mathcal{I}} \sqsubseteq D^{\mathcal{I}}$ (resp. $C^{\mathcal{I}} \subset D^{\mathcal{I}}$); it satisfies an assertion $C(a)$ (resp. $r(a, b)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$). An interpretation \mathcal{I} is a *model* of an \mathcal{EL} -ABox \mathcal{A} if it satisfies all assertions in \mathcal{A} .

A concept inclusion or an assertion α is said to be *entailed* by an \mathcal{EL} -ABox \mathcal{A} , written $\mathcal{A} \models \alpha$, if α is satisfied by every model of \mathcal{A} . It is clear that the existence of assertions does not impact the entailment of concept inclusions, thus $\mathcal{A} \models \alpha$ actually means $\emptyset \models \alpha$ for any concept inclusion α . An individual $a \in N_I$ is called an *instance* of a concept C in \mathcal{A} if $\mathcal{A} \models C(a)$. The *most specific concept* [6] of an individual $a \in N_I$ in an \mathcal{EL} -ABox \mathcal{A} is defined as the \mathcal{EL} concept C such that $\mathcal{A} \models C(a)$ and for all \mathcal{EL} concepts D , $\mathcal{A} \models D(a)$ implies $\mathcal{A} \models C \sqsubseteq D$.

It was shown [6] that the most specific concept of an individual may not have a finite size. For example, the most specific concept of a in the \mathcal{EL} -ABox $\{r(a, a)\}$ is $\exists r \dots \exists r. \top$, which has infinitely many nested quantifiers. Recall that a target concept to be learned is a most specific \mathcal{EL} concept that has the item being browsed, sufficiently many marked items and at least one other unmarked item as instances.

Thus, a target concept can possibly have an infinite size. To guarantee that a learned target concept has a finite size and the learning process is efficient, we introduce *d-restricted* \mathcal{EL} concepts defined below.

Definition 1. Given an integer d , an \mathcal{EL} concept C is called *d-restricted* (or simply *restricted*) if the maximal number of nested quantifiers occurring in C is at most d and there is at most one existential restriction in $\{C_1, \dots, C_n\}$ for every occurrence of existential restriction $\exists r.(C_1 \sqcap \dots \sqcap C_n)$ in C .

The proposed approach actually learns *recommendatory d-restricted* \mathcal{EL} concepts, namely, *frequent d-restricted* \mathcal{EL} concepts that are as specific as possible, which are defined below. The requirement that a target concept C to be learned is frequent and is as specific as possible implies that C can model a maximal set of user preferences.

Definition 2. Given an \mathcal{EL} -ABox \mathcal{A} , a set S of marked items, an item being browsed a_0 , and an integer k not larger than the cardinality of S , where all items are individuals appearing in \mathcal{A} and a_0 is possibly in S , a *d-restricted* \mathcal{EL} concept C is said to be *frequent* in \mathcal{A} w.r.t. S, a_0, k if $\mathcal{A} \models C(a_0)$, $\mathcal{A} \models C(a)$ for some unmarked item $a \notin S \cup \{a_0\}$, and $\mathcal{A} \models C(a_1), \dots, \mathcal{A} \models C(a_k)$ for k different marked items $a_1, \dots, a_k \in S$; moreover, C is called a *recommendatory d-restricted* \mathcal{EL} concept in \mathcal{A} w.r.t. S, a_0, k if there is no frequent *d-restricted* \mathcal{EL} concept D in \mathcal{A} w.r.t. S, a_0, k such that $\mathcal{A} \models D \sqsubset C$.

The following example further explains Definition 2 and shows the usefulness of recommendatory *d-restricted* \mathcal{EL} concepts in recommender systems.

Example 3. Let \mathcal{A} be an \mathcal{EL} -ABox constituted by all \mathcal{EL} assertions given in Example 2. According to Example 1, we have assumed that a user U sets the item being browsed as rm_6 and marked items as rm_1 and rm_3 . Suppose $k = 1$ and $d = 2$. This means that a combination of features, expressed as an \mathcal{EL} concept, is considered to be preferred to the user U if it has rm_6 , at least one item in $\{rm_1, rm_3\}$ and at least one item in $\{rm_2, rm_4, rm_5\}$ as instances; moreover, a recommendatory *d-restricted* \mathcal{EL} concept should not contain any existential restriction in which the number of nested quantifiers is more than 2. It is not hard to see that there are two recommendatory *d-restricted* \mathcal{EL} concepts in \mathcal{A} w.r.t. $\{rm_1, rm_3\}, rm_6, k$. They are $C_1 = \text{Room} \sqcap \exists \text{hasPrice}.\top \sqcap \exists \text{atFloor}.\top \sqcap \exists \text{locatesNear}.\text{(Route} \sqcap \exists \text{reaches.HEMCNorthLocation})$ and $C_2 = \text{Room} \sqcap \text{TowardsSouth} \sqcap \exists \text{hasPrice}.\top \sqcap \exists \text{atFloor}.\top$, respectively. The unmarked but not being browsed instance of C_1 is rm_5 , while the unmarked but not being browsed instance of C_2 is rm_4 . Both rm_5 and rm_4 are reasonable recommendations to the user U , because rm_5 has a maximal combination of features which is preferred to U (rooms near a traffic route which reaches some HEMC north locations), while rm_4 has another maximal combination of features which is preferred to U (rooms towards south). \square

Although the proposed approach needs to collect marked items before learning recommendatory *d-restricted* \mathcal{EL} concepts, it does not have a ramp-up problem. In fact, when a considering user to whom we deliver recommendations has not marked any item or has only marked a few items, we can set the parameter k

as the number of items that have been marked by the user, so that the proposed approach still works. The proposed approach can also take TBoxes into account. It is clear that the information of a TBox \mathcal{T} can be largely propagated to an ABox \mathcal{A} by adding assertions entailed by $\mathcal{T} \cup \mathcal{A}$, forming an *ABox completion* which is $\mathcal{A} \cup \{A(a) \mid A \in N_C, a \in N_I, \mathcal{T} \cup \mathcal{A} \models A(a)\} \cup \{r(a, b) \mid r \in N_R, a \in N_I, b \in N_I, \mathcal{T} \cup \mathcal{A} \models r(a, b)\}$. Given an ontology with a TBox, the proposed approach can work on the ABox completion rather than on the ABox of the given ontology so as to make use of the information on the TBox.

3 Learning Recommendatory Restricted \mathcal{EL} Concepts

To develop a method for learning all recommendatory d -restricted \mathcal{EL} concepts, we introduce *most specific d -restricted \mathcal{EL} concepts* defined below.

Definition 3. Given an \mathcal{EL} -ABox \mathcal{A} and an individual a , a *most specific d -restricted \mathcal{EL} concept* of a in \mathcal{A} is a d -restricted \mathcal{EL} concept C such that $\mathcal{A} \models C(a)$ and for all d -restricted \mathcal{EL} concepts D , $\mathcal{A} \models D \sqsubset C$ implies $\mathcal{A} \not\models D(a)$.

For any individual a , there is a unique most specific d -restricted \mathcal{EL} concept of it up to equivalence. That is, if both C and D are most specific d -restricted \mathcal{EL} concepts of a in \mathcal{A} , then $\mathcal{A} \models C \sqsubseteq D$ and $\mathcal{A} \models D \sqsubseteq C$; otherwise, $C \sqcap D$ is a d -restricted \mathcal{EL} concept such that $\mathcal{A} \models C \sqcap D(a)$ and either $\mathcal{A} \models C \sqcap D \sqsubset C$ or $\mathcal{A} \models C \sqcap D \sqsubset D$, contradicting that both C and D are as specific as possible.

By observing that a d -restricted \mathcal{EL} concept is expressed as a conjunction of concept names or d -restricted existential restrictions, we define a *component* of an individual a in \mathcal{A} as a concept name or a d -restricted existential restriction which is more general than the most specific d -restricted \mathcal{EL} concept of a in \mathcal{A} . The most specific d -restricted \mathcal{EL} concept of a in \mathcal{A} is semantically equivalent to the conjunction of *all* components of a in \mathcal{A} , while a recommendatory d -restricted \mathcal{EL} concept in which a is the item being browsed is semantically equivalent to the conjunction of *some* components of a in \mathcal{A} . It implies that a recommendatory d -restricted \mathcal{EL} concept in \mathcal{A} w.r.t. some S, a, k is actually a maximal conjunction of components of a in \mathcal{A} such that $\mathcal{A} \models C(a)$, $\mathcal{A} \models C(b)$ for some $b \notin S \cup \{a\}$, and $\mathcal{A} \models C(a_1), \dots, \mathcal{A} \models C(a_k)$ for k different $a_1, \dots, a_k \in S$. Hence, all recommendatory d -restricted \mathcal{EL} concepts in \mathcal{A} w.r.t. S, a, k can be discovered by an MFI-mining algorithm in which a component of a in \mathcal{A} is treated as an *item*, while a conjunction of components is treated as an *itemset*.

Based on the aforementioned idea, we develop a three-step method for computing recommendatory d -restricted \mathcal{EL} concepts in \mathcal{A} w.r.t. S, a, k .

In the first step, the most specific d -restricted \mathcal{EL} concept of a in \mathcal{A} is computed. This can be done by the method proposed in [6] which is previously used to compute the *d-approximation* of a in \mathcal{A} . The *d-approximation* C of a in \mathcal{A} is the most specific \mathcal{EL} concept such that $\mathcal{A} \models C(a)$ and the maximal number of nested quantifiers occurring in C is at most d . The most specific d -restricted \mathcal{EL} concept

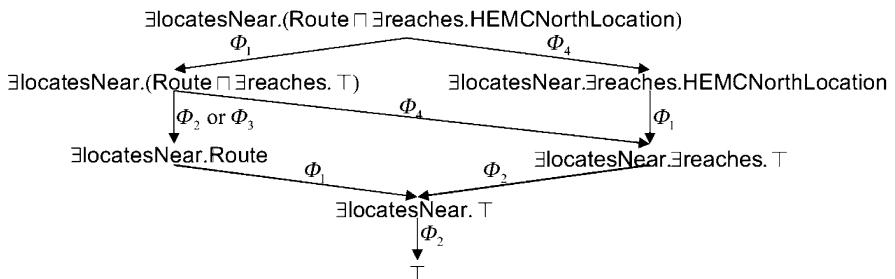
is almost the same as the d -approximation except that it also has a restriction on the inner occurrence of existential restrictions.

In the second step, all different components of a in \mathcal{A} are computed. To do this, we introduce a refinement operator Φ which takes a d -restricted existential restriction C as input and returns the set of concept names (including the top concept) and existential restrictions that are strictly more general than C in \mathcal{A} . The result of $\Phi(C)$ is defined recursively as follows, where A is a concept name, C_1 and C_2 are concept names or conjunctions of concepts, and $\xi(C')$ returns a concept simplified from C' by replacing all occurrences of $\top \sqcap D$ with D :

- $\Phi(C) \stackrel{\text{def}}{=} \Phi_1(C) \cup \Phi_2(C) \cup \Phi_3(C) \cup \Phi_4(C).$
- $\Phi_1(C) \stackrel{\text{def}}{=} \bigcup \{\Phi(C') \cup \{\xi(C')\} \mid C' \text{ is obtained from } C \text{ by replacing an occurrence of } \exists r.A \text{ with } \exists r.\top\}.$
- $\Phi_2(C) \stackrel{\text{def}}{=} \bigcup \{\Phi(C') \cup \{\xi(C')\} \mid C' \text{ is obtained from } C \text{ by replacing an occurrence of } \exists r.\top \text{ with } \top\}.$
- $\Phi_3(C) \stackrel{\text{def}}{=} \bigcup \{\Phi(C') \cup \{\xi(C')\} \mid C' \text{ is obtained from } C \text{ by replacing an occurrence of } \exists r.(C_1 \sqcap C_2) \text{ with } \exists r.C_1\}.$
- $\Phi_4(C) \stackrel{\text{def}}{=} \bigcup \{\Phi(C') \cup \{\xi(C')\} \mid C' \text{ is obtained from } C \text{ by replacing an occurrence of } \exists r.(C_1 \sqcap C_2) \text{ with } \exists r.C_2\}.$

We say that a concept D in $\Phi(C)$ is a *descendant* of C ; furthermore, D is a *child* of C if D is a descendant of C and there is no descendant of C which has D as a descendant. An example on how to apply Φ is provided below.

Example 4. Consider Example 2 again. Let $d = 2$ and \mathcal{A} be the set of all assertions appearing in Example 2. The most specific d -restricted \mathcal{EL} concept of rm_6 in \mathcal{A} is $\text{Room} \sqcap \exists \text{hasPrice}.MoreThan1500 \sqcap \text{TowardsSouth} \sqcap \exists \text{atFloor}.AtLeast7 \sqcap \exists \text{locatesNear}.(\text{Route} \sqcap \exists \text{reaches}.HEMCNorthLocation)$. By applying the refinement operator Φ to $C = \exists \text{locatesNear}.(\text{Route} \sqcap \exists \text{reaches}.HEMCNorthLocation)$, we can obtain all descendants of C . The following graph shows the relationship between C and all its descendants, where $C_1 \xrightarrow{\Phi_i} C_2$ denotes that C_2 is a child of C_1 and is one-step obtained from C_1 by applying Φ_i .



□

The following proposition shows the soundness and completeness of $\Phi(C)$ in terms of finding all concepts that are strictly more general than C . The soundness of this proposition is obvious, while the completeness follows from the fact that a concept that is more general than C can be obtained from the \mathcal{EL} -description tree [6] of C by gradually deleting a label in a node or deleting an edge.

Proposition 1. *Let \mathcal{A} be an arbitrary \mathcal{EL} -ABox \mathcal{A} and C a d -restricted existential restriction, then for all \mathcal{EL} concepts D , $D \in \Phi(C)$ if and only if $\mathcal{A} \models C \sqsubset D$.*

Let the most specific d -restricted \mathcal{EL} concept of a in \mathcal{A} be normalized to $A_1 \sqcap \dots \sqcap A_m \sqcap C_1 \sqcap \dots \sqcap C_n$, where A_1, \dots, A_m are concept names and C_1, \dots, C_n are d -restricted existential restrictions, and then by Proposition 1, the set of components of a in \mathcal{A} is $(\{A_1, \dots, A_m, C_1, \dots, C_n\} \cup \Phi(C_1) \cup \dots \cup \Phi(C_n)) \setminus \top$.

In the last step, all the different components of a in \mathcal{A} are ordered according to the parent-child relations, yielding a sorted list $L = \langle C_1, \dots, C_n \rangle$ such that children are placed after parents, and then `EnumerateHBRConcepts`(L, \mathcal{A}, S, a, k) is called to compute all recommendatory d -restricted \mathcal{EL} concepts in \mathcal{A} w.r.t. S, a, k . The algorithm `EnumerateHBRConcepts`, shown in Fig. 1, is adapted from an efficient MFI-mining algorithm proposed in [8], where a *minimal hitting set* for a set \mathcal{S} of itemsets is a minimal itemset I such that $I \cap I' \neq \emptyset$ for all itemsets I' in \mathcal{S} . In this algorithm, minimal hitting sets are used to facilitate the computation of target concepts (i.e., recommendatory d -restricted \mathcal{EL} concepts), where a target concept is treated as an itemset and every component constituting the target concept is treated as an item. The following example illustrates how the algorithm `EnumerateHBRConcepts` works.

Example 5. Continue Example 4. Let rm_6 be the item being browsed, $S = \{rm_1, rm_3\}$ be the set of marked items and $k = 1$. The set of components of rm_6 in \mathcal{A} can be sorted according to the parent-child relations, yielding a sorted list $L = \langle \text{Room}, \exists \text{hasPrice}. \text{MoreThan1500}, \exists \text{hasPrice}. \top, \text{TowardsSouth}, \exists \text{atFloor}. \text{AtLeast7}, \exists \text{atFloor}. \top, \exists \text{locatesNear}. (\text{Route} \sqcap \exists \text{reaches}. \text{HEMCNorthLocation}), \exists \text{locatesNear}. (\text{Route} \sqcap \exists \text{reaches}. \top), \exists \text{locatesNear}. \exists \text{reaches}. \text{HEMCNorthLocation}, \exists \text{locatesNear}. \text{Route}, \exists \text{locatesNear}. \exists \text{reaches}. \top, \exists \text{locatesNear}. \top \rangle$, in which children are placed after parents. The work flow of `EnumerateHBRConcepts`(L, \mathcal{A}, S, a, k) is shown in the following graph. A rounded rectangle corresponds to a call to the procedure `FindHBRConcept`, where the content is the result (a diagnosis and a target concept) of this call. It stretches out a set of labeled branches, each corresponding to an element of the diagnosis (line 7 of `ConstructHBRConcept`). A circle indicates a reuse of a previously computed diagnosis. If some labels in the current path appear in the diagnosis in the circle (line 6 of `ConstructHBRConcept`), the circle stretches out an unlabeled branch; otherwise, the circle stretches out a set of labeled branches, each corresponding to an element of the diagnosis (line 7 of `ConstructHBRConcept`). A rectangle shows why the current path is pruned: “not frequent” means that the concept constituted by all labels in the current path is not frequent (line 2 of `ConstructHBRConcept`); “not an MHS” means that the set of labels in the current path is not a minimal hitting set for the set of diagnoses gone through by the current path (line 7 of `ConstructHBRConcept`).

Main Procedure EnumerateHBRConcepts($\langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$)

Input: A sorted list of concepts $\langle C_1, \dots, C_n \rangle$, an \mathcal{EL} -ABox \mathcal{A} , a set S of marked items, an item a which is being browsed, and an integer k .

Output: A set of target concepts.

Comments: The maximum subscript m of target concepts, target concepts T_0, \dots, T_m and diagnoses D_0, \dots, D_m are used globally, where the subset of $\{C_1, \dots, C_n\}$ consisting of concepts not general than T_i is called a *diagnosis* and denoted by D_i .

- 1: $m \leftarrow 0$;
- 2: ConstructHBRConcept($0, \emptyset, \langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$);
- 3: **return** $\{T_0, \dots, T_m\}$;

Procedure ConstructHBRConcept($i, M, \langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$)

- 1: **if** $i = m$ **then**
- 2: **if** not frequent(M, \mathcal{A}, S, a, k) **then exit**;
- 3: FindHBRConcept($i, M, \langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$);
- 4: $m \leftarrow m + 1$;
- 5: **end if**
- 6: **if** $D_i \cap M \neq \emptyset$ **then** ConstructHBRConcept($i + 1, M, \langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$)
- 7: **else for** each $e \in D_i$ such that $M \cup \{e\}$ is a minimal hitting set of $\{D_0, \dots, D_i\}$ **do**
- 8: ConstructHBRConcept($i + 1, M \cup \{e\}, \langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$);
- 9: **end if**

Procedure FindHBRConcept($i, M, \langle C_1, \dots, C_n \rangle, \mathcal{A}, S, a, k$)

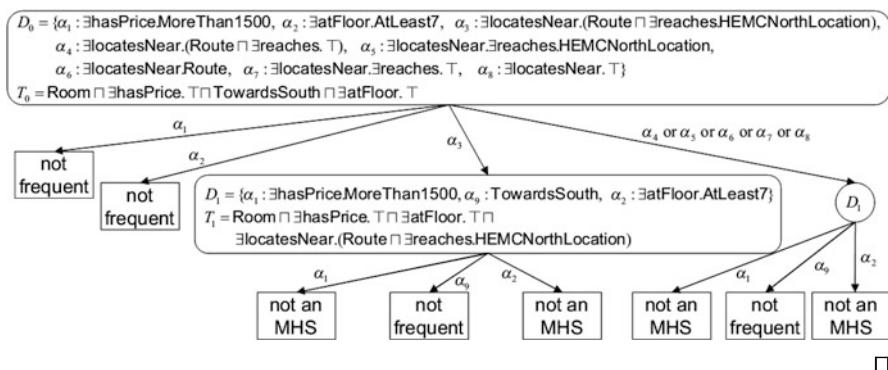
- 1: $D_i \leftarrow \emptyset; M' \leftarrow M$;
- 2: **for** j from 1 to n with $C_j \notin M'$ and C_j is not a child of any concept in M' **do**
- 3: **if** frequent($M' \cup \{C_j\}, \mathcal{A}, S, a, k$) **then** $M' \leftarrow M' \cup \{C_j\}$
- 4: **else** $D_i \leftarrow D_i \cup \{C_j\}$;
- 5: **end for**

6: $T_i \leftarrow C'_1 \sqcap \dots \sqcap C'_m$ where $\{C'_1, \dots, C'_m\} = M'$;

Boolean Function frequent($\{C_1, \dots, C_m\}, \mathcal{A}, S, a, k$)

- 1: $S_{item} \leftarrow \{b \mid \mathcal{A} \models C_1(b)\} \cap \dots \cap \{b \mid \mathcal{A} \models C_m(b)\}$;
- 2: **return** $a \in S_{item}$ and $|S \cap S_{item}| \geq k$ and $S_{item} \setminus (S \cup \{a\}) \neq \emptyset$;

Fig. 1 The algorithm for computing all target concepts



□

Analogously as shown in [8], `EnumerateHBRConcepts(L, \mathcal{A}, S, a, k)` is correct (see Proposition 2 below) and needs up to $|L| \cdot n_{diag} + n_{MHS}$ calls to the function `frequent` (which predominates the whole algorithm in terms of the execution time), where n_{diag} is the number of diagnoses (i.e., the number of target concepts) and n_{MHS} is the number of minimal hitting sets for the set of diagnoses.

Proposition 2. *`EnumerateHBRConcepts(L, \mathcal{A}, S, a, k)` returns the set of all recommendatory d -restricted \mathcal{EL} concepts in \mathcal{A} w.r.t. S, a, k .*

4 Implementation and Evaluation

As mentioned before, to make the proposed approach work better for an ontology with a TBox, we need to compute the ABox completion of the ontology. Since the proposed approach does not rely on the completeness of the ABox completion, we use a simple method for approximating the ABox completion. That is, we translate the given ontology to a set of first-order clauses, remove all clauses that have function symbols or have more than one positive literal, and compute the unique least model of the set of remaining clauses. Then the least model is a subset of the ABox completion. We implemented the above method by translating clauses to SQL statements and applying MySQL to retrieve instantiated clauses. We also implemented the proposed approach by treating instance retrieval for components (i.e., computing $\{b \mid \mathcal{A} \models C_i(b)\}$ in line 1 of the function `frequent`) as SQL queries to the back-end MySQL database that stores the ABox completion. All the above implementations were written in JAVA.

We conducted preliminary experiments for the proposed approach, focusing on efficiency and scalability only. The test set is composed of Lehigh University Benchmark (LUBM) [5] ontologies with assertions from 2/4/6/8/10 universities. For each test ontology we generated 20 test scenarios. In each scenario, we randomly generated 10 marked items and the item being browsed, where all these items are instances of the concept name **Person** which has more instances than other concept names. Each test scenario was carried out against different (1/5/10) numbers of k , i.e., the minimal number of marked items for supporting user preferences, yielding totally 300 cases. All experiments were conducted on a 2.66GHz Pentium Dual Core PC running Windows XP with 1GB RAM allocated to the Java Virtual Machine.

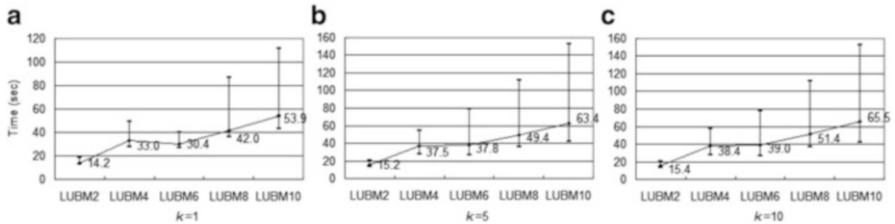
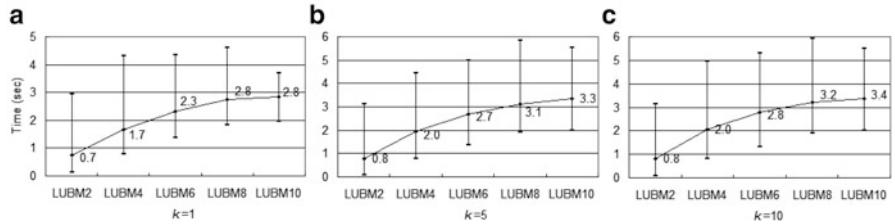
We consider the computation of the ABox completion as a preprocessing step as it only needs to be done once. The statistics of all test ontologies including the execution time for the preprocessing step is shown in Table 1.

The execution time for computing all target concepts from the stored ABox completion against different values of k and different test ontologies is shown in Fig. 2. The scalability is quite good: in general the execution time increases smoothly when the size of ABox increases, while it only differs slightly for different values of k except that for a few cases, the execution time is extremely long because these cases make many calls to the function `frequent`.

Table 1 The statistics of all test ontologies

Ontology	$ N_C $	$ N_R $	$ N_I $	#Person	$ \text{ABox} $	$ \text{ABox}_{\text{comp}} $	Time (sec)
1. LUBM2	43	32	38,334	19,120	230,061	314,692	77
2. LUBM4	43	32	78,579	39,697	477,784	652,735	164
3. LUBM6	43	32	118,500	60,176	722,953	987,305	275
4. LUBM8	43	32	163,552	83,694	1,001,418	1,367,700	416
5. LUBM10	43	32	207,426	106,409	1,272,575	1,737,937	603

Note: #Person is the number of instances of the concept Person; $|\text{ABox}|$ is the number of assertions in the original ABox; $|\text{ABox}_{\text{comp}}|$ is the number of assertions in the computed ABox completion; Time is the execution time for computing the ABox completion

**Fig. 2** The max/avg/min execution time against different ABoxes**Fig. 3** The max/avg/min execution time excluding the time for accessing the MySQL database in line 1 of the function frequent

Although the efficiency may not be acceptable for a practical recommender system, we found that most of the execution time is spent on instance retrieval for components (i.e., computing $\{b \mid \mathcal{A} \models C_i(b)\}$ in line 1 of `frequent`), which is implemented as SQL queries to MySQL. The inefficiency is mainly caused by the time-consuming table joins in executing these queries. If we pre-execute the SQL queries and cache their results for reuse, the execution time spent on instance retrieval for components can be neglected. Excluding this portion of execution time yields promising results as shown in Fig. 3. All cases finish in six seconds and the average execution time is no more than 3.4 s. This implies that the approach can be made practical by preprocessing SQL queries.

5 Conclusion and Future Work

This paper has proposed a novel knowledge-based approach to proactive recommendation, which is based on learning \mathcal{EL} concepts with well-defined properties. Differing from traditional knowledge-based approaches, the proposed approach does not require users to input explicit needs and delivers dynamic recommendations that are adapted to ongoing user–system interactions. Compared to collaborative or content-based filtering, the proposed approach has no ramp-up problem and is suitable for bootstrapping a recommender system.

Our experiments have shown that the approach is feasible in terms of efficiency and scalability. But this is only the first step for verifying the approach. We plan to further verify it with other common metrics, such as accuracy, diversity, novelty, and coverage, through real-life applications in our future work.

Acknowledgements This work is partly supported by NSFC grants (61005043 and 71271061) and the Undergraduate Innovative Experiment Project in Guangdong University of Foreign Studies.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the \mathcal{EL} envelope. In: Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI), pp. 364–369, 2005
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
3. Billsus, D., Pazzani, M.J.: A hybrid user model for news story classification. In: Proc. of the 7th International Conference on User Modeling (UM), 1999
4. Burke, R.: Knowledge-based recommender systems. In: A. Kent (ed.): Encyclopedia of Library and Information Systems, **69**(32), (2000)
5. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. *J. Web Semant.* **3**(2–3), 158–182 (2005)
6. Küsters, R., Molitor, R.: Approximating most specific concepts in description logics with existential restrictions. In: Joint German/Austrian Conference on AI (KI/ÖGAI), pp. 33–47, 2001
7. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer, New York (2011)
8. Satoh, K., Uno, T.: Enumerating maximal frequent sets using irredundant dualization. In: Proc. of the 6th International Conference on Discovery Science (DS), pp. 256–268 (2003)

Impact of Multimedia in Sina Weibo: Popularity and Life Span

Xun Zhao, Feida Zhu, Weineng Qian, and Aoying Zhou

Abstract Multimedia contents such as images and videos are widely used in social network sites nowadays. Sina Weibo, a Chinese microblogging service, is one of the first microblog platforms to incorporate multimedia content sharing features. This work provides statistical analysis on how multimedia contents are produced, consumed, and propagated in Sina Weibo. Based on 230 million tweets and 1.8 million user profiles in Sina Weibo, we study the impact of multimedia contents on the popularity of both users and tweets as well as tweet life span. Our preliminary study shows that multimedia tweets dominant pure text ones in Sina Weibo. Multimedia contents boost popularity of tweet as well as users. Users who tend to publish many multimedia tweets are also productive with text tweet. Finally, we demonstrate that tweets with multimedia contents survive longer than text tweets. Our research demonstrates the impact of multimedia in Sina Weibo with respect to how it affects the popularity, life span of tweets, and the popularity of user. The results could be leveraged by social-media-based marketing and decision-making.

X. Zhao (✉)

Master Candidate, School of Information Systems, Singapore Management University
e-mail: jackzhaoxun@gmail.com

F. Zhu

Assistant Professor, School of Information Systems, Singapore Management University
e-mail: fdzhu@smu.edu.sg

W. Qian • A. Zhou

Professor, Software Engineering Institute, East China Normal University
e-mail: wnqian@sei.ecnu.edu.cn; ayzhou@sei.ecnu.edu.cn

1 Introduction

The recent years have seen social network services gaining ever-increasing popularity as a result of people's growing communication demand as well as Internet's permeation into everyone's daily life. These services have profoundly changed the way people acquire knowledge, share information, and interact with one another on a societal scale.

Microblogging services, such as Twitter and Sina Weibo, allow users to publish short messages called "tweet" or "weibo" which contains no more than 140 characters. Each user may "follow" another user to receive all up-to-date messages published by that user and get "followed" by other users to spread his messages. One can also use "@" to address a user directly. The ease of usage and succinct nature of tweets have made possible the swift propagation of news and messages in Twitter network [6].

The huge number of users, together with the staggering amount of content people generated every day in these microblogging sites, has led researchers to analyze the syntactics and semantics underlying these social network services. Kwak et al. [7] points out that Twitter is more of a news media than a social network. Cha et al. [1] points out that follower count alone could not reflect the popularity of users.

Previous research on microblogging services relies mainly on textual information and social link information. However, what has as yet been largely neglected is another aspect of the microblogging data, the multimedia content, which has manifested its importance with the ever-increasing volume of the data and the profound changes it has given rise to the information diffusion throughout the network. As the saying goes—a picture is worth a thousand words. Nowadays, social media users find it much more convenient and enjoyable than ever before to express their opinions by posting pictures, attaching video clips rather than just typing a message. Mobile social network application developers also introduce features to allow users to take pictures and then upload them through a simple click. Compared with text information, multimedia contents are more eye-catching and entertaining. The result is that multimedia contents such as audio tracks, images, and videos command viral popularity everywhere they go ranging from personal blogs to video sharing sites and to social network services. For example, according to our findings, more than 30% of tweets published in Sina Weibo contain image links. Less measurable but no less profound is the ever-growing attention people paid to multimedia content, which is demonstrated by our results that, compared against tweets of pure text, tweets with multimedia content are retweeted by users for a much longer period of time, which we call they *survive* longer.

We focus our study in this paper on Sina Weibo, a popular, Twitter-like microblogging service platform that originated from China. It features more than 300 million active users in February 2012. Besides microblogging features as those provided by Twitter, Sina Weibo has incorporated multimedia-friendly features such as attaching images as well as short URL links to a tweet. Our Sina Weibo dataset contains more than 1.8 million users and 230 million tweets, of which 111 million

are original tweets. We show that (I) the majority of the tweets in Sina Weibo are tweets containing multimedia contents, (II) multimedia contents such as images and short URLs linked to videos are not often used simultaneously, and (III) multimedia contents generally survive a longer period of time.

2 Related Work

On the one hand, previous research on social media provides a rough tour guide of popular microblogging services. Kwak et al. [7] uses a huge data to illustrate the user composition, trending topics, etc., of Twitter. Java et al. [6] studies the underlying motivation of certain user activity. On the other hand, some works focused on detailed problems based textual information and social link information of the data. Various methods have been proposed to discover certain event or topic [13, 15], discover the community structure [4], or trace the way information is propagated in social networks [5, 10].

Another line of work analyzes multimedia contents associated with semantic geographic annotation. Crandall et al. [2] developed methods to determine the location of a photo. This line of work is based on data that contains just multimedia contents.

To the best of the author's knowledge, this is the first work combining and comparing the textual and multimedia information in microblogging service.

3 Research Questions

Two basic elements in microblog services such as Sina Weibo are users and tweets. Users are creators and consumers of tweets. On one hand, users generate tweets by composing, publishing, or reposting tweets. On the other hand, users consume tweets by reading, reposting, and replying tweets. In traditional text world, the generation and consumption process is quite straightforward. However, if we take multimedia content into consideration, would some previously identified patterns change? Specifically, we consider the following two dimensions:

1. Tweet generation

- (a) Would multimedia content influence the popularity of users?
- (b) Are users who publish more tweets also inclined to publish more tweets with multimedia content?

2. Tweet consumption

- (a) Would multimedia content influence the popularity of tweets?
- (b) Is multimedia content related to the life span of tweets?

Road Map. The following sections are organized as follows: Firstly, we give a description of our dataset in Sect. 4. We then analyze in Sect. 5 the composition of multimedia content in Sina Weibo. We explore the correlation between multimedia content and popularity. We also analyze whether users exhibit same taste for publishing text tweets and multimedia tweets. Finally, in Sect. 6 we illustrate the life span comparison between tweets with multimedia content and text content, respectively.

4 Dataset Description

We use a corpus of data containing 230 million tweets published by 1,812,701 users from January 2011 to July 2011. In this set of tweets, 111 million are original tweets, while the rest are retweets and replies. The majority of the tweets are written in Chinese.

Based on the genre of multimedia content a tweet contains, we divide tweets into the following classes:

1. Text tweet. Text tweets are tweets which only contain text information.
2. Image tweet. In Sina Weibo, there is a feature in each tweet indicating whether this tweet has an image link.
3. URL tweet. URLs are links other than images which embed in the text body of the tweet.

Image tweet and URL tweet together forms the concept of multimedia tweet.

On the other hand, Sina Weibo allows users to choose whether to include a URL link specifying a homepage, favorite links, or other microblog account in their profile. For ease of discussion, we categorize the set of users into 2 types, referred as URL users and No URL users based on whether there is a URL link embedded in their profiles or not.

5 Multimedia Contents Popularity

In terms of the form of a tweet, a tweet is either an original tweet, a reply, or a retweet. Original tweets are tweets directly composed by the user and reflect the original intention of that tweet, while retweets are just reposts of original tweets and replies are commentaries about the original tweet started with an “@”. Replies and retweets are widely used as measures of popularity of the original tweets [7, 12]. To study the composition of multimedia content in Sina Weibo, we distinguish between the set of general tweet and popular tweet. General tweets consist of all the original tweets in our dataset, and popular tweets are a subset of general tweets which receive a considerable amount of retweets. Cha et al. [1] has reported that popular tweets are more likely to be posted by celebrities and news medias.

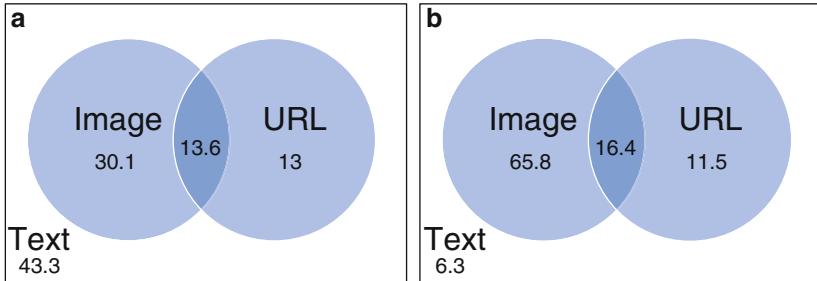


Fig. 1 Venn diagram for composition of multimedia tweet. **(a)** General tweet composition, **(b)** Popular tweet composition

Zhao et al. [16] has reported that the topics of popular tweets are different from ordinary tweet. Yu et al. [14] finds out that the trends in Sina Weibo are created due to the retweet of multimedia content such as jokes, images, and videos. Our analyses further support this point.

5.1 Original Tweets Content Composition

Out of 230 million tweets in our dataset, 127 million are replies or retweets. Replies and retweets are comments and replicates of original tweets. They can be used as measures for popularity of original tweets [1], but they do not have any content value. For original tweets, which are not replies nor retweets, we divide them into 3 categories, namely, text tweet, image tweet, and URL tweet as previously categorized. We also select another group original tweet which received more than 1,000 retweets for comparison. We call this set of tweet popular tweet. Figure 1 shows multimedia content (image and URL) composite more than 50% in both setting. In more detail, image tweets dominate in general tweet composition, with more than 40%, and the dominance is more profound in popular tweet setting with text tweet only composite 6.3% in popular tweets. This shows that while text tweets do exist in a considerable amount, the majority of trending tweets in Sina Weibo are multimedia content tweets. Interestingly, we also see no small overlap between image tweets and URL tweets, which indicate the usage of multimedia is integrative and simultaneous.

A similar approach is to use the number of replies to define the popularity of a tweet. To our regret, our data does not contain reply information. Thus, we do not provide popularity analysis based on replies in this article.

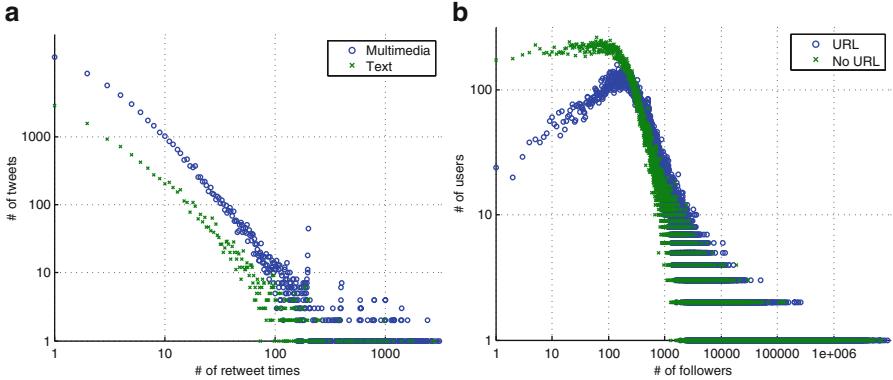


Fig. 2 Tweet and user popularity distribution. **(a)** Tweet popularity distribution, **(b)** User popularity distribution

5.2 Tweet and User Popularity

To understand the interplay between popularity and multimedia content, we need to examine the popularity each tweet and user dissents and the difference between multimedia content and plain text. To measure the popularity of tweet and user, we follow the convention in [1] and use retweet times as a measure of tweet popularity and follower count for user popularity.

Figure 2a displays tweet popularity distribution of 1,000,000 randomly selected tweets. The overall distribution approximately fits a power law pattern [3]; most of the tweets receive very few retweets and only a few tweets receive large number of retweets. The number of tweets from different popularity level differs by orders of magnitude. Interestingly, we also observe a long tail in both multimedia setting and text setting when $retweettimes > 100$. This abnormal pattern indicates the number of very popular tweet is larger than power law distribution suggests, reflecting that very popular tweets do exist in a considerable amount. This finding has important implications for microblog-based marketing. Marketers would get a great payoff by aiming at those top popular tweets.

The proportion of multimedia tweets in these 1 million tweets is 61.8%, which is consistent with our previous composition analysis in general setting. With retweet number set, the number of multimedia tweet is larger than text tweet. While with tweet number set, retweet times of multimedia tweet is also larger. This reflects that multimedia tweets are more popular than text tweet in terms of absolute number and retweet times.

Sina Weibo allows users to include another type of multimedia content right into their profile. In their profile, a user could put a URL-specified homepage link, blog site, or other microblog account. Based on whether a user puts such URL links in their profile, we divide users into two groups. For simplicity, we use URL to refer to the set of users who have such information and No URL for those who do not.

Follower count could be used as a measure for user popularity [1]. Figure 2b also shows a power law pattern when $200 < followercount < 1,000$ for both set of users, as the number of users decreases exponentially with follower count increase. We also observe a long tail when $followercount > 1,000$, indicating the number of very popular users is more than the power law pattern suggests. For URL distribution, we find a global maximum at $follower = 200$. While before URL reach its peak, the number of No URL users is always bigger than URL users. We conjecture that this may result from the fact that URL users tend to engage more effort in maintaining their Weibo account as well as interacting with their friends, making the number of inactive (less followers) users less than No URL users.

5.3 Comparing User Activeness

For multimedia content lovers, are they also craving in posting a lot of text tweets? Specifically, are users who publish most multimedia tweets also the ones who publish most text tweets? The amount of tweet a user posts can be used as an indicator of user activeness [1]. We get the number of text tweets and number of multimedia tweets for each user in the previous setting. Rather than directly compare the number of text tweets and the number of multimedia tweets, we use the relative order of user ranks based on tweet quantity and multimedia quantity as a measure of difference. We first sort users by those two measures, so the rank 1 user in tweet quantity indicates the most active publisher. Increased ranks imply less active publishers. Users with the same number of tweet would receive the average rank amongst them. Once each user receives a rank from these two measures, we could compare their rank difference. We use Spearman's rank correlation coefficient [11]

$$\rho = 1 - \frac{6 \sum (x_i - y_i)^2}{N^3 - N} \quad (1)$$

as a measure of the strength of association between two rank sets, where x_i and y_i are ranks of users based on two measures in a dataset of N users. The coefficient assesses how well a monotonic function could describe the relationship between two variables, without making any other assumptions about the particular nature of the relationship between the variables. The closer ρ is to $+1$ or -1 , the stronger the correlation. A perfect positive correlation is $+1$, and a perfect negative correlation is -1 .

The results in Fig. 3 show a moderate strong correlation (above 0.6) between ranks of multimedia tweet quantity and text tweet quantity for all pairs. However, if we narrow our focus on top 500 users, those who rank top 500 in tweet quantity, the correlation becomes stronger. Further narrowing on even top users lead to even higher correlation, indicating users who publish most tweets also publish most multimedia tweets, especially for most active users.

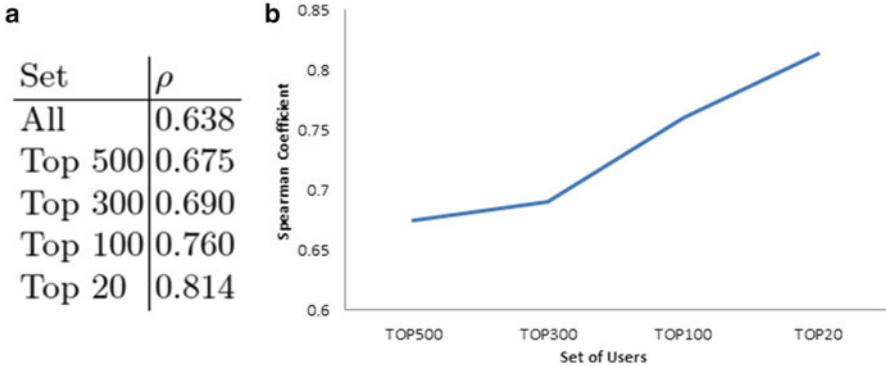


Fig. 3 Spearman correlation of user post activeness. (a) Spearman correlation samples, (b) Spearman correlation trend

6 Life Span Analysis

Many factors, such as user popularity and topic of tweet [9], could affect the life span of a tweet. Previous studies [6, 7] have reported that messages in microblogging services such as Twitter spread and disappear rather fast. Lardinois [8] reported that instead of a social media, Twitter is indeed a broadcast medium with virtually all retweets happens within the first hour after the original tweet. Figure 4a shows how retweet times changes for a typical tweet as time passes in [8]. It quickly receives a lot of retweets after its birth and slowly lose its attention.

Interestingly, in our Sina Weibo data, we find that some of the tweets remain viral and repeatedly get reposted for a long period of time. To get the temporal effect of multimedia contents, we set up the following experiment: We first select all trending tweets, including retweets and original tweets, which get at least 500 retweets in July 2011. Then we filter out retweets and get the original tweet id of these trending tweets. Finally, we go to previous months and check the publish time of these original trending tweets. We only track 6 months backwards, which start from January to June, for original tweets found prior to January is too small for statistical analysis.

For comparison, we also separate the trending tweets into three categories: text tweet, image tweet, and URL tweet. Figure 4 shows the bar plot of how many original tweets within each category are found in each month from January to June.

According to [10], life span of memes, or new topics, follows exponential decay. In this article, we follow this convention and model the life span of tweet as the form

$$N(t) = N_0 e^{-bt} \quad (2)$$

where $N(t)$ is the quantity at time t , N_0 is the initial quantity and b the decay rate.

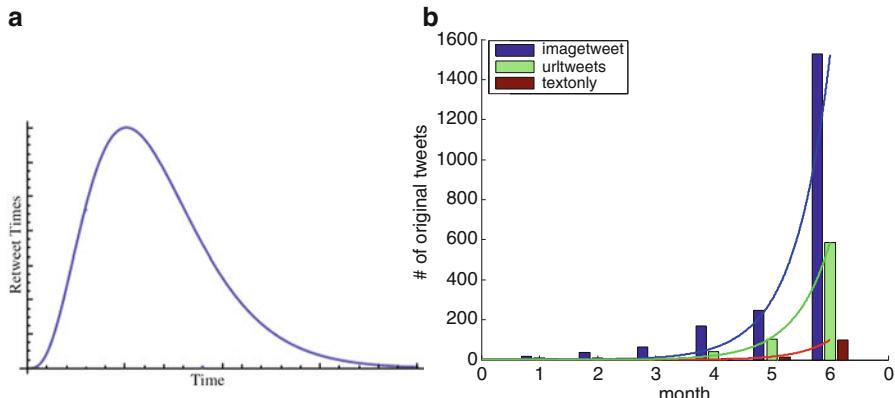


Fig. 4 Tweet and user popularity distribution. (a) Gamma distribution representing a typical tweet life span in Twitter, (b) Number of original tweets found in each month and fitting result

Table 1 Decay rate coefficient with error range

Category/Coefficient	N_0	b	τ
Text	0.001678($-0.004541, 0.007897$)	1.831(1.211,2.451)	0.546
Image	0.08929($-0.231, 0.4096$)	1.624(1.022,2.226)	0.616
URL	0.02766($-0.02104, 0.07636$)	1.660(1.365,1.955)	0.602

Based on this assumption, we use formula (2) to fit the data observed in Fig. 4b. We use nonlinear least square method to fit the data. Table 1 shows the result coefficient of the fitting, where $\tau = \frac{1}{b}$ is defined as the mean life span of tweets.

The amount of original tweets in Fig. 4b shows all three groups drop exponentially from June to January. The amount of image tweets are always dominant in each month followed by URL tweets, further suggesting multimedia content's power of attracting retweets over text. The decrease rate, however, is a bit different among three groups. As in Table 1, text tweets have the largest decay rate, followed by URL and image tweets, which implies image tweets have the longest life span, followed by URL tweets and text tweets.

In Table 1, there is a significant gap between life span of text tweet and the other two multimedia groups, while the difference between image tweet and URL tweet is marginal. This shows a fundamental difference of content virality as well as popularity between multimedia tweets and text tweets. This is because the rich information and eye-catching nature makes multimedia tweets more viral than text tweets, thus enabling them to spawn a longer period of time after they first get published. For comparison in the two multimedia groups, image tweets show a slightly longer life span than URL tweets. We conjecture that this is because pictures are directly embedded in the tweet, which gives users a direct visualization, while URLs are more often appeared as links, and content illustration is dependent on the text information rather than multimedia itself.

The error range of text group is larger than the other two groups. We conjecture that this is caused by the small amount of data in text tweet. Only a handful of text tweets are found in the beginning months of the year.

In order to get a larger sample size, we also set different retweet popularity threshold (100, 200, 300, etc.) in this experiment. In all of these attempts, the program would not finish running because of large sample size. Our findings point out that multimedia contents have a longer life span than traditional text messages.

7 Conclusion

In this paper, we show that multimedia tweets composite a large proportion in Sina Weibo. Moreover, we demonstrate multimedia contents influence the popularity of tweet and user by boosting the retweet times of a tweet and the follower number of a user. The number of highly popular tweets exists in a larger scale than power law pattern suggests. Multimedia contents help to promote retweets and follower account of user. Users who publish large number of text tweets are the ones who publish a lot of multimedia tweets. Finally, we study the correlation between multimedia contents and tweet life span. Multimedia tweets such as image tweets and URL tweets have a longer life span than text tweet.

Acknowledgements This work is partially supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. This work is also supported by the China National Basic Research (973 Program) under grant number 2010CB731402 and China National High-tech R&D Program (863 Program) under grant number 2012AA011003.

References

1. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in twitter: The million follower fallacy. In: Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM), 2010
2. Crandall, D.J., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: Proceedings of the 18th International Conference on World Wide Web, pp. 761–770. ACM, New York (2009)
3. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets. Cambridge University Press, Cambridge (2010)
4. Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., Han, J.: On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 813–822. ACM, New York (2010)
5. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. ACM Trans. Knowl. Discovery Data **5**(4), 21 (2012)
6. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, pp. 56–65. ACM, New York (2007)

7. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web, pp. 591–600. ACM, New York (2010)
8. Lardinois, F.: The short lifespan of a tweet: Retweets only happen within the first hour. In: <http://www.readwriteweb.com>
9. Lerman, K., Ghosh, R.: Information contagion: An empirical study of the spread of news on digg and twitter social networks. In: Proceedings of 4th International Conference on Weblogs and Social Media (ICWSM), 2010
10. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 497–506. ACM, New York (2009)
11. Spearman, C.: The proof and measurement of association between two things. *Am. J. Psychol.* **15**(1), 72–101 (1904)
12. Wang, D., Li, Z., Salamatian, K., Xie, G.: The pattern of information diffusion in microblog. In: Proceedings of The ACM CoNEXT Student Workshop. ACM, New York (2011)
13. Weng, J., Lee, B.S.: Event detection in twitter. In: Proceedings of 5th International Conference on Weblogs and Social Media (ICWSM), 2011
14. Yu, L., Asur, S., Huberman, B.A.: What trends in chinese social media. arXiv preprint arXiv:1107.3522, 2011
15. Zhao, W., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. *Adv. Inform. Retrieval* 338–349 (2011)
16. Zhao, X., Jiang, J., He, J., Song, Y., Achananuparp, P., LIM, E.P., Li, X.: Topical keyphrase extraction from twitter. In: Proceedings of 49th Annual Meeting of the Association for Computational Linguistics, 2011

Ontology-Based Model and Procedure Creation for Topic Analysis in Chinese Language

Dong Han and Kilian Stoffel

Abstract This paper focuses on setting up a methodology to create models and procedures for the functions and processes involved in topic analysis, especially for the business documentation in the Chinese language. Ontologies of different types are established and maintained containing the annotated and evolved knowledge. Extraction, transforming, and loading methods are adapted from approaches which are used to set up a data warehouse for standardized data models, exploited as the basis of a large variety of analysis. Topic discovery is conducted based on Latent Dirichlet Allocation for different usage. An interactive tool is implemented to support the proposed design and realistic demands.

1 Introduction

For enterprises, a large number of documents are involved in the daily business, and the size of the documents is also increasing with the pervasive usage of electronic forms. Therefore, the motivation is high to facilitate their operation, not only to get an intuition of the content of the documents but also to get a more profound comprehension of the semantics of the documents and the relationship between groups of keywords. If this objective can be achieved, then it will be much more effective for enterprises to integrate this type of operations into their processes of decision making, auditing, and market promotion. This idea, furthermore, can be adapted to a wide range of specific domains such as financial analysis, quality control, and logistics management. In many cases, latent topics are only implicitly presented in the documents. The classical way to handle this problem is to conduct statistical studies. This approach is particularly well suited for the cases in which

D. Han (✉) • K. Stoffel

Information Management Institute, University of Neuchâtel Pierre-à-Mazel 7,
CH-2000 Neuchâtel, Switzerland

e-mail: dong.han@unine.ch; kilian.stoffe@unine.ch

a relatively small number of words are used very frequently. If, however, a larger number of words are distributed in the documents with latent links, then some novel approaches are required.

Meanwhile, as the economic development in China progresses, business based on Chinese language is getting influential in the global market. Documents written in Chinese are more difficult for traditional approaches which are based on the assumption of alphabets. We proposed a method to analyze the Chinese language in [1] with data mining approaches. As research advances, we have set up a more profound mechanism, leveraging the specific characteristics of Chinese language applied in different domains including financial management and enterprise management.

The contribution and innovation of this paper is to propose a methodological framework by taking advantages of data-driven approaches. It fully considers the reusability of the existing systems from the point of view of the data and avoids redundancy and repetition of functions. Domain strategies can also be estimated by applying this framework to verify their effectiveness. A prototypical tool has been implemented as a working platform to validate the proposed design. This prototype has already been deployed in real user cases to verify the effectiveness of the proposed methods.

2 Theory and Ontology Establishment

Considering the qualitative features of the data involved in this paper, we select *Grounded theory* [2] as the guideline to develop our new approach. This approach is appropriate for the research of Chinese documentation since there is usually plenty of data at the beginning of an analysis without conclusive statements. Inheriting from the principles of Grounded theory, three types of ontologies are set up: *Project ontology* is a high-level ontological framework to define the objects and their relationships to ensure the compatibility. *Reference ontology* contains the incorporated knowledge from domain experts. *Code ontologies*, as depicted in Fig. 1a, record the raw data from the primary texts and the user's annotation. OWL is utilized as the ontology language, and depending on the concrete scenarios, we can use *lite*, *DL* and *full* standards.

Whilst generic ontologies are mainly designed for English and other alphabetical languages, ontologies based on the Chinese language are established considering its typical features. *Speech ontologies* represents the speech of the words of the original texts as well as the annotations. We use the speech ontologies here not aiming to interpret the sentences from one language to another. Instead, we would just like to comprehend the structures of the sentences for their subjects, verbs, objects, etc. This serves as the fundamentals of the further analysis. *Auxiliary ontologies* is a set of the bag of keywords which are highly influential to the grammar of the languages. They contain the major auxiliary and assistant words in the Chinese language, such as the words to express tense, tone, and meanings. Also the words for negation and interrogation are listed in these ontologies. *Localization ontologies* are

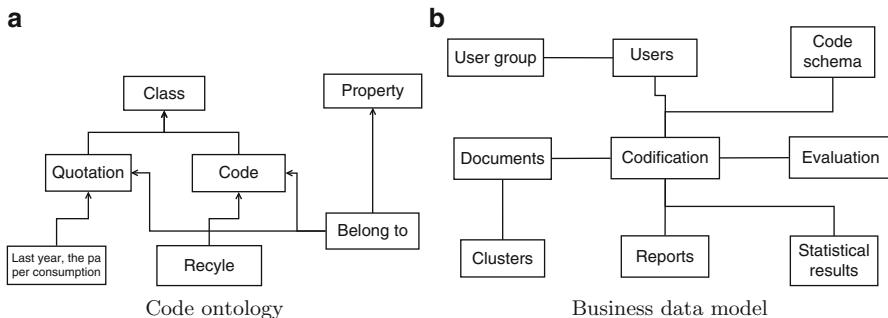


Fig. 1 Code ontology and business data model. (a) Code ontology, (b) Business data model

set up to represent the knowledge about localization on top of the language per se. For example, the regional divisions in China include North, Central, and South with different dialects, terms, and expressions. Ontologies in this category are exploited with both language and domain knowledge to produce derived information. These ontologies play an important role in analyzing the documentation in Chinese language [1]. They also serve as the input of the ETL processes presented in the following section.

3 ETL and Data Warehousing

In order to extract information from the ontological knowledge, we need to formulate a set of dimensions as the properties of this knowledge. For each ontology file, two tuples, external and internal respectively, are set up. All the extractions are supposed to be based on these tuples. Correspondingly, two kinds of extractions are designed—global extraction and local extraction. *Global extraction* aims at handling external information of the submissions made by experts and is not going inside the files. The purpose of this step is to reorganize all the data in a structured way, labeling each file with its properties in accordance with the tuple previously defined. Once we finish this step, all the knowledge is stored in a uniform format, each file with its name containing the attributes in tuple 1. To facilitate this process of global extraction and to conform with industry standards, we use Powershell [3] scripts to carry out the extractions.

Next, *local extraction* is conducted. Local extraction aims to retrieve internal information from the ontologies following the definition of tuple 2. The results of this step are divided into two categories with respect to their data structures: (1) *Relational tables*. In a relational table, each attribute represents one dimension of the attributes, for example, company, year, code, and code frequency. As a format well fitting the schema of relational databases, relational table is straightforward to be imported into a database management system for advanced queries. (2) *Pivot*

tables. A second type of tables used in the system is pivot tables. A pivot table, used for data manipulation such as aggregation and sorting of the data, is mainly for the purpose of data output and graphics in tabular forms [4].

$$Q_{\text{external}} = \{\text{group}, \text{user}, \text{project}, \text{document}, \text{year}, \text{file}\} \quad (1)$$

$$Q_{\text{internal}} = \{\text{file}, \text{quotation}, \text{coordinates}, \text{codes}, \text{ratings}\} \quad (2)$$

With the two types of extractions, it is sufficient to extract and maintain most of the useful information from the ontologies. Moreover, subjects are derived from the business scenario, with data loaded via the ETL processes. These subjects are presented to depict a general view of the data involved in a project. Above that, a business data model is established to characterize the entities in each subject area in a finer granularity in order to reveal more details as described in Fig. 1b. Furthermore data marts can be established and data mining methods can be carried out on top of these models.

4 Topic Analysis

The steps presented above facilitate the processes of topic analysis composed of two parts: data annotation and systematic analysis. Data annotation is carried out at the beginning by the users in interaction with the primary documents and recorded in the form of ontologies. This will provide refined data on top of the original texts. Once the annotation has been started, we need to extract the terms which are significant to the texts. For the terms, certain features can be extracted in respect to different dimensions of interest. With enough flexibility of feature construction, the main functionality to be provided during this step is the aggregation. For each document d , as an example, $(\mathbf{Q}_d, \mathbf{C}_d)$ is created as a matrix recording the quotations and their codes in this document. They are used as input of the term-topic model. There are several topic models which have been applied to discover the topics from documents. We propose to use LDA as it has advantages over other methods as shown in [5]. The formulas given by [6] illustrate the basic idea of LDA (see Fig. 2). One of the most important tasks for LDA is to estimate the parameters involved in the model, and the effectiveness of the parameter estimation highly influences the output of the topic discovery and analysis. A list of algorithms, such as the variational EM method [6] and the Gibbs sampling algorithm [7], are considered as the candidates to be leveraged to estimate the parameters involved in the LDA processes. The topics will then be generated based on the parameter estimation of these algorithms.

1. Choose $N \sim Poisson(\lambda)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the N words w_n :
 - (a) Choose a topic $z_n \sim Multinomial(\theta)$.
 - (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n .

Fig. 2 LDA procedures (Source: David M. Blei, Andrew Y. Ng, and Michael I. Jordan, *Latent Dirichlet Allocation*, Journal of Machine Learning Research 3 (2003), p. 996 [6])

5 Implementation

A prototype, namely, *Qualogier*, has been implemented as a working platform and test bed of the proposed approach (see Fig. 3). Based on ICEPpdf [8], it takes PDF files as the primary documents, providing operations on these files such as turning pages, zooming in/out, printing, and extracting the texts. Users are able to select sentences and phrases from the texts of interest in the form of quotations and then assign codes to them. All the user behaviors are recorded in ontologies and then, together with the original texts, modeled as the background for establishing the data warehousing models. Furthermore different utilities are provided for exporting and outputting information using ETL methods. The topics generated by LDA are presented along with the original documents to show the key concepts based on their semantics and the users' annotations. The system also supports ontology inference based on forward chaining and backward chaining methods to produce new facts, for example, recommendations to the users. This system has been deployed in our research project for 40 domain experts to evaluate the sustainability performance of different firms in the form of a case study. These experts have various operating systems, domain knowledge, and research subjects. They are working with the proposed system in an interactive way as shown in Fig. 4. They select and highlight reports from the firms and submit them to the system. Based on the analytical system feedback, the experts will proceed with further studies. This system has several advantages compared to other similar systems like ATLAS.ti [9].

6 Conclusion and Future Work

In this paper, a methodology is presented based on ontologies, ETL, data warehousing, and LDA to retrieve information from the original data and model the entire working processes for documentations in the Chinese language. In the future, we will conduct more evaluation approaches and user experiments to verify the effectiveness of the methodology presented in this paper.

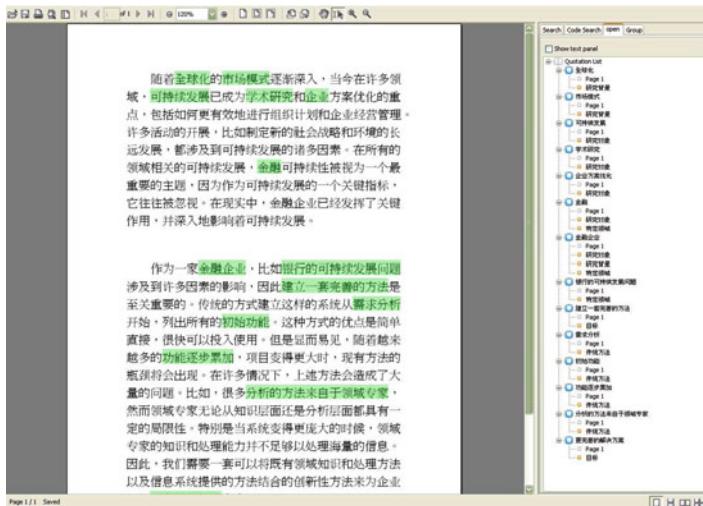


Fig. 3 Screenshot of the system

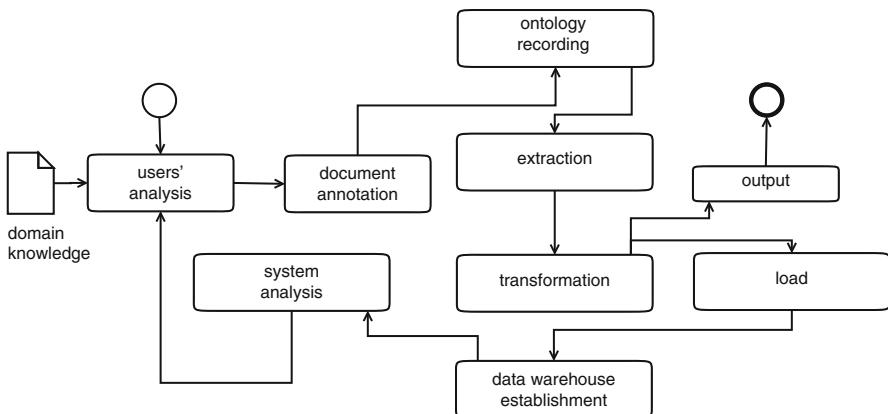


Fig. 4 Workflow of the entire processes

Acknowledgements This work is supported by Swiss National Science Foundation(SNSF) project “*Formal Modelling of Qualitative Case Studies: An Application in Environmental Management*” (Project No. CR21I2_132089/1).

References

1. Han, D., Stoffel, K.: Ontology based qualitative methodology for Chinese language analysis. In: Proceeding of the 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2012

2. Glaser, B., Strauss, A.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Transaction, Piscataway, United States (1967)
3. Microsoft: Powershell. <http://technet.microsoft.com/en-us/library/bb978526.aspx>
4. Wikipedia: Pivot table. http://en.wikipedia.org/wiki/Pivot_table
5. Gimpel, K.: Modeling topics. *Inform. Retrieval* **5**, 1–23 (2006)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
7. Griffiths, T.L., Steyvers, M.: Finding scientific topics. In: *Proceedings of the National Academy of Science*, 5228–5235, 2004
8. Icepdf: Icepdf. <http://www.icepdf.org/>
9. Han, D., Stoffel, K.: An interactive working tool for qualitative text analysis. In: *Proceeding of the 12th Francophone International Conference on Knowledge Discovery and Management*, 2012

On the Temporal Dynamics of Influence on the Social Semantic Web

Thomas Gottron, Olaf Radcke, and Rene Pickhardt

Abstract The factors indicating influence on the social semantic web have been analysed in various approaches. We address the still open question of the temporal dynamics of these factors. We focus in particular on emerging, hot topics and the communities of microblog users around them. By training a model to predict the influence over several time segments we are able to analyse how the impact of 16 factors denoting influence develops over time. As datasets we employ a collection of microblog messages around two emerging new topics: the Egyptian revolution and the case of Dominique Strauss-Kahn being accused for sexual assault. In conclusion we demonstrate that factors indicating influence are relatively stable—even in emerging communities. In particular the characteristics of an influential user remain stable, while the characteristics of the contents they publish are less constant. However, we also observe some minor differences in our two cases that can be explained by the context of the particular topics.

1 Introduction

Social media and social networks are an enormous resource for user-generated content, social interaction and collective knowledge. Especially the semantics encoded in an explicit or implicit way into the generated contents are a rich source for analysis and utilizing publicly available knowledge. The social aspect of these contents, however, entails the risk and opportunity that the available information is geared towards the opinion of influential users. Accordingly, the analysis of influence in social networks is central to various recent publications

T. Gottron (✉) • O. Radcke • R. Pickhardt

WeST – Institute for Web Science and Technologies University of Koblenz-Landau
56070 Koblenz, Germany

e-mail: gotttron@uni-koblenz.de; oradcke@uni-koblenz.de; rpickhardt@uni-koblenz.de

(e.g. [2, 8, 9, 11, 15]). These studies, however, typically focus on a snapshot of the data and ignore temporal aspects and dynamics of influence. While for established topics and communities, the assumption might be justified that the factors denoting influence are relatively stable, it is unclear if this also holds for emerging topics and the communities that discuss them. This is of particular importance, if one considers the fact that nowadays many breaking news and hot topics are discussed on social networks first and before they hit the classical media.

In this paper we fill this gap by investigating the time dynamics of influential factors in communities emerging around new topics. We base our analysis on the microblogging service Twitter, which is renowned for its users discussing and disseminating hot topics and news fast and quickly. Twitter also has the advantage that it is quite well analysed and understood regarding the aspects of influence and information propagation. Therefore, various factors indicating influence on Twitter have been identified already. This allows us to use a common and established notion of influence in microblog environments: influence can be measured by the degree of information propagation. Which factors actually contribute to the propagation of information items can be analysed by training prediction models. These models consider the features of users or contents and their impact on the likelihood of a microblog post—in the context of Twitter commonly referred to as a *tweet*—to be spread among a community. The actual impact of a feature is encoded in the derived model by means of feature weights. Generally speaking, higher weights of a feature denote that this factor is more characteristic for influence. In order to investigate the temporal dynamics of influence factors, we iteratively train influence prediction models over several time segments. This series of models allows for a comparison and analysis of the impact of features and their evolution over time.

In this paper we have analysed two datasets collected around emerging topics: the Egyptian revolution and the case of Dominique Strauss-Kahn being accused for sexual assault. In both cases, we have observed a surprisingly stable behaviour of influential factors also among emerging communities. In particular user features turned out to be very stable. There are, however, some minor differences which can be explained from the context of the events.

We proceed as follows. In the next section, we review related work, including a survey of established notions of influence and prediction models for influence in the context of microblogging. In Sect. 3 we describe and motivate the approach we take in this paper. There, we also review in more detail the choice of features and prediction models and present our analytical approach regarding the analysis of temporal dynamics. In Sect. 4 we explain our experimental setting, in particular also introducing the datasets. We present and discuss our results in Sect. 5, before concluding the paper with a summary in Sect. 6.

2 Related Work

In the last years, the microblogging service Twitter¹ has received a lot of attention in the scientific community. Influence of users and news propagation have been at the centre of attention in many of the relevant publications. The commonly agreed notion of influence is defined by the ability to cause a reaction in other people. For Twitter influence can be interpreted as causing other user to reply to a user's microblog posts [10, 15], as the ability to have your posts spread in the network [2, 6, 8, 10–12], the number of followers [5, 14, 16] or PageRank-alike computations over the social network [9, 18].

On the methodological side, these works typically aim at predicting the influence of a microblog post or a user using classification or ranking algorithms. In this context, the general approach is to employ a large general sample of Twitter posts and perform a classical dataset split to obtain a training and evaluation set. Temporal aspects in the data are—if at all—mainly considered to differentiate the post's context in terms of day of the week (working day vs. weekend) or time of the day (working hours, night time, free time hours) [8]. Few works concentrate on how specific topics are reflected or discussed on Twitter [15]. However, the specific topics were rather used to identify a closed community or to determine the boundaries of a dataset.

Regarding the modelling of the semantics in social networks or online communities, there is less work. The most prominent ontology to describe the social web certainly is SIOC [3]. Another ontology for capturing and describing the semantics of social media, social networks and user behaviour was introduced by Angeletou et al. in [1]. This ontology allows for describing various features of users, contents and interaction. Among these features are also notions like the *impact* of users and posted contents. This behaviour ontology has also been used successfully to model Twitter as a component of the social semantic web [15].

3 Measuring Influence in Microblogging Communities

In the previous section we gave an overview of different notions of influence on Twitter. The most commonly accepted notion of influence is defined by being able to spread information via microblog posts. The motivation behind this definition is to consider the spread of tweets as a sign for (a) the provision of important or interesting information and as sign of (b) being able to cause other people to react and propagate this information. This notion of influence can easily be measured on Twitter. The function of retweeting a post is frequently used on the microblog service and effectively spreads the posts of one user throughout the social

¹<http://twitter.com/>.

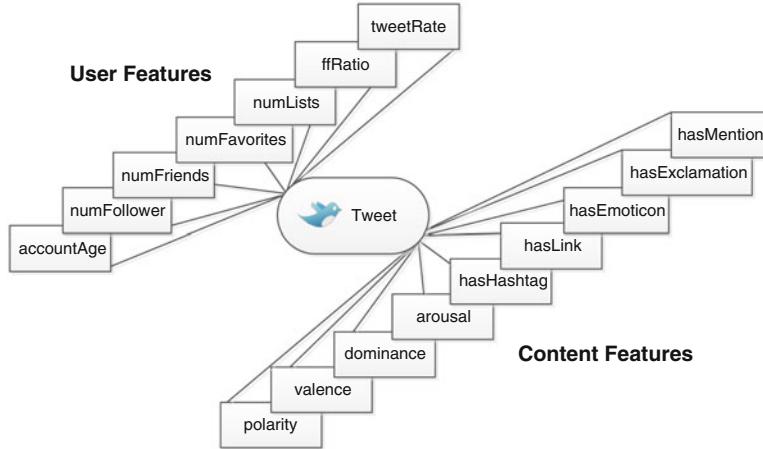


Fig. 1 Features used in our analysis

network. Therefore, we use the observation of a microblog post being retweeted as an indication for influence.

Also the question of how to derive the factors that cause influence has been analysed well. Prediction models have been employed at large in order to discover which factors actually contribute to the propagation of information and, thereby, denote influence. These models are trained on the observations of a tweet's content as well as the user features of its author and how these affected the likelihood of retweets. Technically, this means that in order to be able to model influence factors, we need to predict retweets based on user and content features.

3.1 Feature Denoting Influence

Regarding the features to describe the microblog posts, we employed a set of features which has been derived from related work. In the context of this paper we consider both: features of a post itself and features of the authoring user, which are then attributed to the post as well. All features are summarized in Fig. 1.

User features were mainly considered in work analysing the social network and can be computed as follows:

accountAge The age of the author's user account. The Twitter API provides the creation date of an account which in combination with the creation date of a tweet can be used to determine the age of the user account when the tweet was written. The motivation to consider this feature is to differentiate between novice and established users.

numFollower The number of users following the author of a tweet. Intuitively, users with many followers have a wider reach and, therefore, a higher potential for influence according to our definition.

numFriends The number of friends, i.e. how many other users the author follows.

This number can give indications about the nature of user. Human users have a limited capability of being able to (seriously) pay attention to the updates posted by other users. Too many as well as too little friends might indicate that a user does not represent a human user but rather a company profile, a news broadcast or an automatically acting software agent.

numFavorites The number of incoming tweets a user has marked as favourites.

The point in considering this feature is that it entails a usage of advanced UI features in Twitter and might, thus, indicate experienced power users.

numLists The number of lists the author is a member of. On Twitter, users can create lists of other users. If a user has been added to a list, this usually is a sign for expertise in a certain area, being a member of a well-defined group of people or being merited some particular status.

ffRatio The follower-to-friend ratio is an aggregated value of the number of followers and friends. It is used in related work in addition to the pure values in order to compare the number of incoming and outgoing links in the network of a user.

tweetRate The average rate of tweets per day. The value of this feature is determined globally for the entire lifetime of user profile and can help to identify more or less active users.

Content features were typically introduced in work about the analytics of microblog posts and can be computed as follows:

hasHashtag The presence of a hashtag in a tweet. Hashtags start with the # symbol and are used to mark and organize tweets.

hasLink The tweet contains a link to a URL. These external links can provide additional information, might refer to a discussed object (image, article) or direct the user elsewhere in the web. We do not distinguish between shortened and standard URLs.

hasMention A binary value, whether a tweet mentions another twitter user. This so-called mentioning is done by including the other user's name preceded by an @ symbol.

hasExclamation Presence of exclamation marks or question marks in a tweet. Some related work gave hints in the direction that strong expressions or questions caused more reaction in other users.

hasEmoticon Presence of an emoticon (smiley) in a tweet. Emoticons can indicate humour, irony or sentiments like happiness or sadness.

Valence, Arousal, Dominance Sentiment (pleasure vs. displeasure, excited vs. calm, weak vs. strong) of the tweet according to the ANEW dictionary [4].

Polarity Sentiment (positive vs. negative) of the tweet using the SentiStrength approach [17].

3.2 Prediction Models to Describe Influence

Also for the purpose of selecting a suitable model for describing influence on the observed data, we considered successful methods in the related work. We discovered three main approaches, which have been applied to the detection or prediction of influence:

- Rule-based approaches (J48) [15]
- Logistic regression [11]
- Bayesian learning, in particular naive Bayes (NB) [6]

All of these methods are established methods from the machine learning domain, and there are various implementations available. Also the quality of the derived models can easily be assessed on observations retained for evaluation. Metrics like accuracy or AUC (area under receiver operating characteristic curve) are well established and commonly used to compare models to each other, but also to give an indication on the absolute performance. A detailed discussion of these models and metrics would be beyond the scope of this paper. We refer to standard textbooks in this field, such as [19].

Once a prediction model has been trained on observed tweets and the quality of the model is assured, it can be analysed to obtain insights into the more indicative factors of influence. The accuracy of learned rules or the values of feature weights denote which factors are more and which are less indicative for influence.

3.3 Evolution of Influence Over Time

In order to analyse the dynamics of features over time we need to take the time axis into consideration. To this end, we split up the available observations along the time dimension in several segments. Each of these segments needed to be still large enough to allow for a reasonable analysis. At the same time we needed several segments in order to be able to detect eventual trends or developments over time.

Based on such a segmentation we identified two approaches to observe dynamics in the influence factors:

Evolution of feature weights. Each of the individual time segments of the data is used to train and evaluate a prediction model using a classical 10 times tenfold cross-validation setting. This allows for comparing the weights of the features in the model over time and considers their temporal evolution. A decrease or increase of features weights over time indicates a changed impact of a feature on the likelihood of a tweet being influential. In this way, it is possible to directly observe changes in the factors that denote influence.

Evolution of prediction performance. A second option is to train a model on the earliest data segment and apply and evaluate it on the remaining, subsequent segments. The quality of this model can then be compared to models that were

trained locally on the later segments. This approach provides the chance to observe the prediction stability of the models, i.e. if a model trained at the initial stage of a topic can still be used later on, when the topic and its community have evolved. In this way it is possible to indirectly observe temporal dynamics of the prediction models.

4 Experiments

As mentioned in the introduction we focus in this paper on emerging, hot topics. Therefore, we identified two events that caused a sudden and widespread discussion of the topics on Twitter: the Egyptian revolution and the case of Dominique Strauss-Kahn being accused for sexual assault.²

4.1 Datasets

The dataset addressing the Egyptian revolution was extracted from the publicly available TREC Tweets2011 corpus.³ This corpus is composed of approximately 16 million tweets and was released for the purpose of evaluating retrieval tasks on microblogs. The corpus is particularly interesting in our case, because it is covering the time of the Egyptian revolution in early 2011. This event was strongly discussed on Twitter, and its emergence was not foreseen. Thus, we considered this a good example for a new and hot topic which caused interaction between users. For the remainder of the paper we will refer to this dataset as the *Egypt* dataset.

The second dataset we constructed ourselves using the Twitter streaming and search API.⁴ With the event of Dominique Strauss-Kahn being arrested under the suspect of sexual assault on the 14th of May 2011, we started to collect tweets related to this topic. Also here we observed a strong discussion of the matters on Twitter. Furthermore, also this topic emerged newly and could not have been foreseen. Neither were the main individuals involved in this case at the attention of a wider public before this event, which supported also here the hypothesis that the discussion was not carried on from an existing topic. We will refer to this dataset as the *DSK* dataset.

For both datasets we had to select tweets from a much larger corpus (the TREC corpus and the tweets being published over the streaming API). The selection was

²In the preparation of this work we considered various other datasets, but had to discard them because they were either too small, did not cover a long enough time span or did not permit to extract the features we were interested in.

³<http://trec.nist.gov/data/tweets/>.

⁴<http://dev.twitter.com/>.

Table 1 Characteristics of the employed datasets

Dataset	Topic	Time	Hashtag, keywords	No. of Tweets
Egypt	Egyptian revolution	Jan 25–Feb 8, 2011	#jan25, mubarak #dsk, strauss- kahn,	22,050
DSK	Arrest of Dominique Strauss-Kahn	May 15–Jul 16, 2011	strausskahn	3,230

based on specific hashtags or keywords contained in the post’s content that indicated our topics of interest. The hashtags were manually chosen based on their accuracy in identifying relevant tweets on a smaller sample of data. Furthermore, we restricted the datasets to tweets in English language. This restriction has been necessary as some of the features are language dependent (e.g. the sentiment orientation). To this end, we used an approach for language classification on short texts [7], which has proven suitable for identifying the language of tweets. Table 1 summarizes the characteristics of our two datasets.

As an additional check to avoid a topic drift or bias in our datasets, we compared the number of tweets per day with figures from Google insights for search.⁵ We observed a very high correlation of the volume of tweets with the volume of search traffic on Google on the corresponding topics. Figure 2 shows the two curves for the DSK dataset, which have a correlation value of 0.88.

The features listed in Sect. 3.1 were obtained via the Twitter API or were computed on the data available in the corpora. For features with a skewed distribution of values (e.g. the number of followers having a power law-like distribution), it was necessary to clean the data from outliers. Outliers distort the distribution which can have implications for the derivations of prediction models. Therefore, we identified extreme values for a feature and artificially adjusted these values to a reasonable range. To this end we considered the distribution of values and marked observations outside a 1.5 range of the interquartile range (IQR) as outliers. These values of the outlier observations were then modified and change to the maximum or minimum value of the 1.5 IQR, respectively.

4.2 Choice of Model and Feature Selection

In order to decide which of the methods for constructing influence prediction models, we found in related work and listed in Sect. 3.2 performed best, we conducted a preliminary test on our data. To this end, we compared the performance of J48, logistic regression and NB by running a 10 times tenfold cross-validation

⁵<http://www.google.com/insights/search/>.

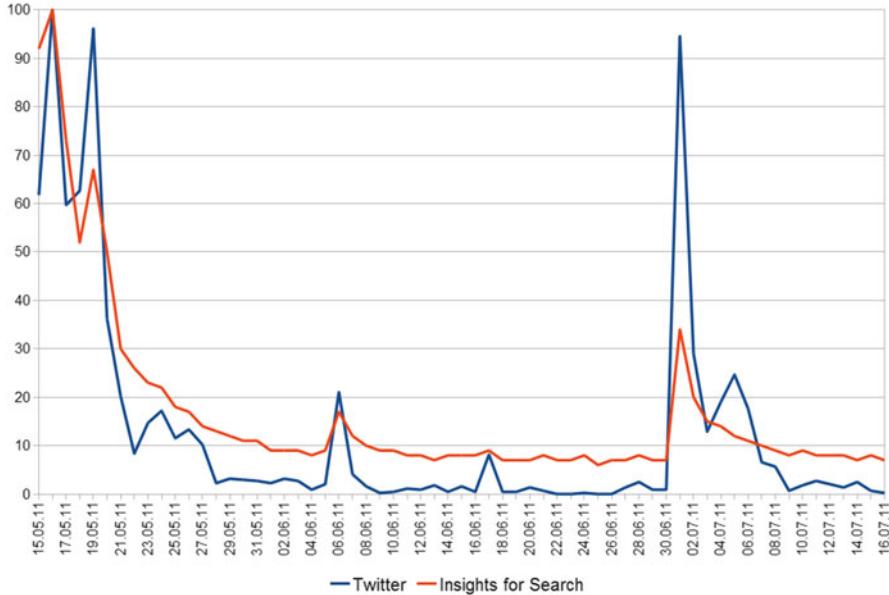


Fig. 2 Comparison of the volume of tweets in the DSK dataset and search requests on Google over time

Table 2 Performance of different models

Dataset	Metric	J48	NB	log. Regression
Egypt	Accuracy	84.18%	81.37 %	84.96%
	AUC	0.67	0.78	0.81
DSK	Accuracy	91.34%	89.44 %	91.34%
	AUC	0.82	0.93	0.93

experiment on the complete datasets. Table 2 shows the results of this analysis. It can be seen that logistic regression performs best on our datasets under the aspects of both: accuracy and AUC. Therefore, we decided to use logistic regression for the further experiments.

In this step we also identified features which were strongly correlated to each other as well as features which showed no correlation with the observation of retweets. We observed the features Polarity and Valence to be highly correlated which makes perfectly sense as they effectively measure the same notion of sentiment in texts. Such redundancy is not required for the data analysis and can actually even lead to wrong interpretations of the feature weights. In conclusion we considered to exclude the feature *Polarity* from our models. A further analysis of the features *hasEmoticon*, *hasMention* and *hasExclamation* indicated them to be independent from the observation of retweets. In a second iteration, we trained and evaluated the same prediction models without the four features identified as

superfluous and obtained results of equal quality. Therefore, we decided to drop the features for the further analysis.

In a next step we computed the division of our datasets into time segments. For the Egypt dataset we divided the available tweets in ten segments over the observed time period. Each segment contains 2,205 tweets. For the DSK dataset we had less data available. Thus, though actually covering a longer time span, we could only create five time segments of 646 tweets each.

5 Results

We will now look at the time dynamics of features denoting influence of users. We start with the larger Egypt dataset, before looking at the DSK dataset.

Observations on the Egypt Dataset. Figure 3 shows the change of feature weights in the logistic regression model over the time of the ten data segments. To the left, in Fig. 3a we see how the weights of the user features evolve. The weights remain relatively stable over the full time period. Only towards the end, in the last two time segments, we observe a drop of the weights for *numFriends* and an increase in the weight of *accountAge*. The content features in Fig. 3b on the right, instead, behave very differently. While the model weights of the features *hasLink* and *hasHashtag* remain relatively stable, the weights of the sentiment features fluctuate quite strong. One explanation might be the changing tone in the tweets as the events of the Egyptian revolution evolved. However, we also suspect that data sparsity has an influence. Given that the tweet posts are relatively short and contain few words, dictionary-based approaches for sentiment detection might not cover a large enough vocabulary. This problem holds for both approaches: SentiStrength and ANEW.

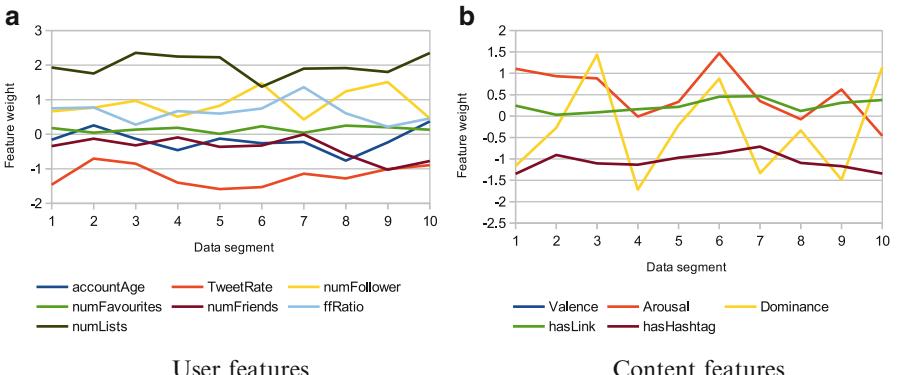


Fig. 3 Development of the weights of user and content features over time on the Egypt dataset. **(a)** User features, **(b)** Content features

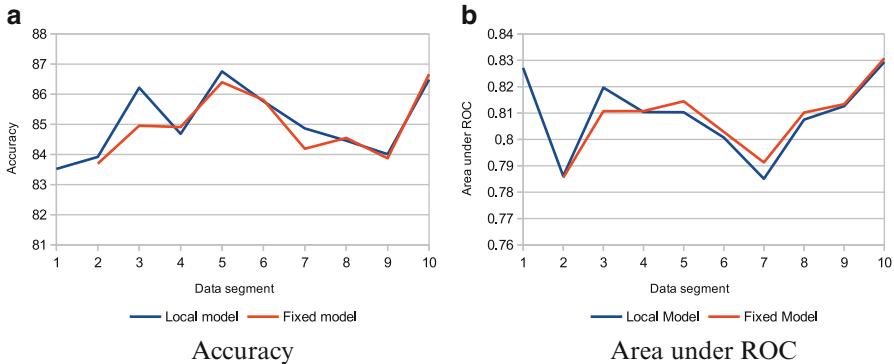


Fig. 4 Comparison of the performance of local models trained and evaluated on each individual data segment vs. a model trained on the first data segment applied to subsequent segments of the Egypt dataset. (a) Accuracy, (b) Area under ROC

The plots in Fig. 4 demonstrate the stability of the local models trained on each time segment and the performance of the model trained on the first segment and applied to subsequent time segments. Under both metrics—accuracy as well as AUC—we observe that the influence of prediction model is surprisingly stable. Even in our setting of hot emerging topics we can see that a prediction model trained at the initial phase of the community does not perform significantly different from a continuously updated model adjusted to the current state of the community at later points in time.⁶

Observations on the DSK Dataset. Also on the DSK dataset we can see a similar behaviour as for the Egypt dataset. The user features in Fig. 5a are more stable than the content features in Fig. 5b. Again, the content features representing the sentiment analysis are highly unstable.

One particularity can be observed on the user features in the last data segment. Here we observe a quite consistent change of the weights for all the features. An explanation can be the twist in the story of Dominique Strauss-Kahn in this case. On June 30th doubts about the victims credibility came up, and the legal case was eventually dismissed. These events fall into our last data segment. Given this strong turn of the events, it is debatable whether this can be seen as a change of influential features in this topic and its community, or whether it is actually a new topic and a new community involved in the discussion altogether.

⁶Please note that performance of the fixed model is noted only from data segment 2 onwards. On the first segment it is equivalent to the local model.

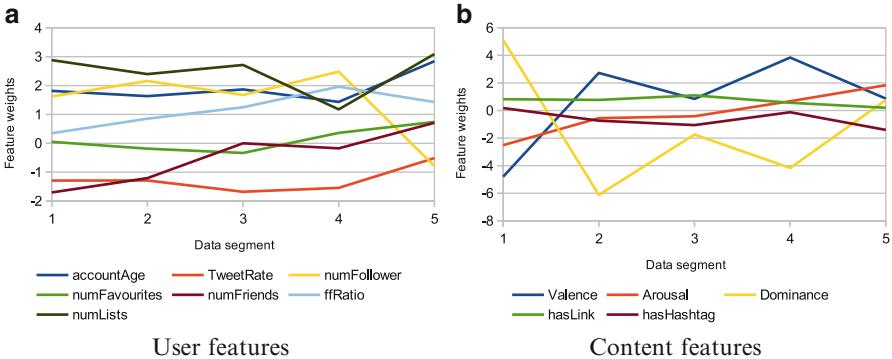


Fig. 5 Development of the weights of user and content features over time on the DSK dataset.
(a) User features, **(b)** Content features

6 Conclusion

In this paper we analysed the time dynamics of features describing influence on emerging topics of the social semantic web. To this end, we employed two datasets settled in the context of newly and suddenly emerging topics. Thereby, we could observe the behaviour of communities that did not exist in this form before our observation. We used various user and content features to build a model describing the influence of users and messages in these communities. Comparing the impact and contribution of the features to the model over time, we observed a surprisingly stable behaviour even in new communities. Some minor deviations across the observed datasets can be explained by the context of the events they were based on.

In future work we intend to build and apply user role classification methods on the author data. These models provide a more fine-grained distinction of the users in categories like initiators, supporters or aggregators. With an extension of our datasets, we will then analyse how the role of users under the constraint of certain topics evolves over time. Another future work might be the identification of other features for influence that have not been evaluated in our temporal analysis. Finally, it might be interesting how content propagation depends on the time the users actually perceive microblog posts. By using temporal top-k retrieval mechanisms [13], we can efficiently reconstruct the timeline of all users at any given time. This might give insights into the question whether tweets which stay long at the top ranks of the timelines of many users are more likely to be propagated. Such an observation would indicate that news propagation as an influence metric also depends strongly on the temporal and social context of users.

Acknowledgements The research leading to these results has received partial funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST and grant agreement no. 287975, SocialSensor.

References

1. Angeletou, S., Rowe, M., Alani, H.: Modelling and analysis of user behaviour in online communities. In: Proceedings of the 10th International Conference on the Semantic Web - Volume Part I, pp. 35–50. ISWC'11. Springer, Berlin, Heidelberg (2011), <http://dl.acm.org/citation.cfm?id=2063016.2063020>
2. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 65–74. WSDM '11. ACM, New York, NY (2011), <http://doi.acm.org/10.1145/1935826.1935845>
3. Bojars, U., Breslin, J.G., Peristeras, V., Tummarello, G., Decker, S.: Interlinking the social web with semantics. IEEE Intell. Syst. **23**, 29–40 (2008)
4. Bradley, M.M., Lang, P.J.: Affective norms for English words (ANEW): Instruction manual and affective ratings. Tech. rep., The Center for Research in Psychophysiology, University of Florida (1999)
5. Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in Twitter: the million follower fallacy. In: Proc. Int. Conf. on Weblogs and Social Media, pp. 10–17, 2010
6. Che Alhadi, A., Gottron, T., Kunegis, J., Naveed, N.: Livetweet: Monitoring and predicting interesting microblog posts. In: ECIR'12: Proceedings of the 34th European Conference on Information Retrieval, pp. 569–570, 2012
7. Gottron, T., Lipka, N.: A comparison of language identification approaches on short, query-style texts. In: ECIR '10: Proceedings of the 32nd European Conference on Information Retrieval, pp. 611–614, Mar 2010
8. Hong, L., Dan, O., Davison, B.D.: Predicting popular messages in Twitter. In: Proc. Int. World Wide Web Conf., pp. 57–58, 2011
9. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: Proc. Int. World Wide Web Conf., pp. 591–600, 2010
10. Leavitt, A., Buchard, E., Fischer, D., Gilbert, S.: The influentials: New approaches for analyzing influence on Twitter, 2009, <http://www.webecologyproject.org/wp-content/uploads/2009/09/influence-report-final.pdf>
11. Naveed, N., Gottron, T., Kunegis, J., Che Alhadi, A.: Bad news travel fast: A Content-based analysis of interestingness on twitter. In: WebSci '11: Proceedings of the 3rd International Conference on Web Science, 2011
12. Naveed, N., Gottron, T., Kunegis, J., Che Alhadi, A.: Searching microblogs: Coping with sparsity and document quality. In: CIKM'11: Proceedings of 20th ACM Conference on Information and Knowledge Management, pp. 183–188, 2011
13. Pickhardt, R., Gottron, T., Scherp, A., Staab, S., Kunze, J.: Efficient graph models for retrieving top-k news feeds from ego networks. In: SocialCom'12: Proceedings of ASE/IEEE International Conference on Social Computing, 2012
14. Romero, D.M., Galuba, W., Asur, S., Huberman, B.A.: Influence and passivity in social media. Machine Learning and Knowledge Discovery in Databases, pp. 18–33, 2011
15. Rowe, M., Angeletou, S., Alani, H.: Predicting discussions on the social semantic web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) The Semantic Web: Research and Applications. Lecture Notes in Computer Science, vol. 6644, pp. 405–420. Springer, Berlin/Heidelberg (2011)
16. Suh, B., Hong, L., Pirolli, P., Chi, E.H.: Want to be retweeted? large scale analytics on factors impacting retweet in Twitter network. In: Proc. Int. Conf. on Social Computing, pp. 177–184, 2010
17. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. J. Am. Soc. Inform. Sci. Tech. **63**(1), 163–173 (2012), <http://dx.doi.org/10.1002/asi.21662>
18. Weng, J., Lim, E.P., Jiang, J., He, Q.: TwitterRank: Finding topic-sensitive influential twitterers. In: Proc. Int. Conf. on Web Search and Data Mining, pp. 261–270, 2010
19. Witten, I.H., Frank, E.: Data Mining, 1st edn. Morgan Kaufmann, San Francisco (2000)

A Detailed Analysis of the Quality of Stream-Based Schema Construction on Linked Open Data

Thomas Gottron and Rene Pickhardt

Abstract The continuously increasing volume of linked open data (LOD) is a challenge when it comes to processing this data. Using the output of an RDF graph traversal (e.g. an LOD crawl) as a linearisation of the data can serve as a basis for a stream-based processing approach. SchemEX (Konrath et al., J. Web Semantics 2012, to appear) utilises such an approach to efficiently compute a schema-based index structure for looking up relevant data sources. In this paper we conduct a detailed analysis of the impact of the stream-based approach regarding the accuracy of the computed schema. We investigate the impact of parameter choices as well as the impact of the analysed data set under several application-motivated metrics. It can be observed that all three factors have an influence on the quality of the schema. In particular, we found that excessive use of blank nodes has a negative impact when using SchemEX to answer complex queries in the deviations. However, stream-based schema approximation is quite accurate. The deviation in the schema elements is at most 10%; the information encoded in the schema deviates by even less than 4%.

1 Introduction

With the linked open data (LOD) movement the publication of RDF data on the web has gained enormous momentum. The amount of data available in semantic formats increases by the minute. This poses a challenge for large-scale processing of this data. Search indices, statistics or aggregation over LOD requires the processing of a huge volume of this public RDF data. It quickly becomes infeasible to download all the available data and process it offline in a random access fashion, i.e. with

T. Gottron (✉) • R. Pickhardt

WeST – Institute for Web Science and Technologies University of Koblenz-Landau
56070 Koblenz, Germany

e-mail: gottron@uni-koblenz.de; rpickhardt@uni-koblenz.de

the ability to access every individual RDF triple in an efficient manner whenever needed. Therefore, alternative processing paradigms have been investigated in the last years.

Recent approaches have considered such paradigms as parallel processing [1, 2], sampling [3, 7] or stream-based processing of RDF data [5, 10, 11]. Parallel or distributed processing approaches address the increasing data volume with more computational power. Sampling approaches reduce the size of the data graph such that the remainder can still be processed on a single machine. Streaming approaches use a data graph traversal (e.g. a crawl of the LOD cloud) to obtain a linear representation of the data and operate on this flat structure. The latter two approaches achieve scalability of graph processing at the trade-off of reducing the accuracy of their results. The loss of accuracy in these cases is at best evaluated in a theoretic way or only on small-scale settings.

With the work at hand we fill this gap for the stream-based approach of SchemEX [10, 11]. SchemEX is designed to extract schema information from RDF triples on the LOD cloud. The extracted schema is the basis for an index of the LOD cloud describing which kind of data can be found where. It uses both explicit schema information, i.e. the explicitly stated `rdf:type` of resources, as well as implicit type information, i.e. what properties are defined for the resources. For smaller data sets the complete schema information can be computed easily. Querying each resource in the data set for all its properties, all its types as well as the types of all other resources it is linked to provide all information necessary for building a SchemEX index. For very large data sets, however, this random access to all information becomes more and more expensive. Therefore, the stream-based approach of SchemEX computes a schema by considering a small window over resources encountered during a traversal of the data graph. The stream-based approach used for the SchemEX index has been compared to a gold standard schema computed with full and random access to the data. On a real-world data set of 11 million triples sampled from the LOD cloud, SchemEX has been run with various configurations to investigate the effect of the parameter settings on the retrieval performance of the schema index.

There are three needs to extend this analysis. First of all, by now SchemEX is being used also in other settings. This entails the availability of various metrics for different application scenarios. Second, the data analysed needs to be extended in its size (number of triples) and in the number of data samples to check that previous observations do in fact generalise. Finally, the parameter configuration space can be extended to better-performing settings, which can still be run with reasonable resources on a single machine. Therefore, in this paper we investigate the accuracy of a stream-based schema computation on RDF data using the SchemEX approach on various data samples over different data volume and different stream window sizes and compare it under several application-motivated metrics to the actual gold standard schema.

We proceed as follows: In the next section we review related work of large-scale RDF processing, schema extraction and approximation. Then we give a brief introduction into SchemEX and applications of the SchemEX index in Sect. 3.

In Sect. 4 we explain our evaluation setup and present and discuss the results in Sect. 5. We conclude with a summary and our plans for future work.

2 Related Work

Most applications operating on LOD have to challenge the enormous data volume of publicly available RDF data. Approaches of parallel processing based on the map-reduce paradigm have been used to clean and analyse governmental data [1] or to create VoID descriptions [2] of LOD data sets. Visualisation of RDF data [3] or estimations of data volume on the semantic web [7], instead, have employed sampling techniques.

The approach of SchemEX [10, 11] is to use a LOD graph traversal as input for the computation of a schema over RDF data. The traversal, e.g. as created by a crawler such as LDSPider [9], provides a stream of RDF triples. Using a windowing technique over this stream allows for an efficient processing of the data. The result is a close approximation of the true schema which serves as index structure for identifying data sources relevant to an information need given in the form of a SPARQL query [6]. Further analysis employed a SchemEX index for a comparison of the information encoded in explicit and implicit schema information as well as the redundancy of both kinds of information [5]. We will present SchemEX in more detail in the next section.

Schema construction over semi-structured data is a long-established field. Based on the OEM model [15], the derivation of a schema from data has been analysed in various research papers [4, 14]. Using a bi-simulation process allows for detecting typical pattern in the data that correspond to a schema. Adapting the process to stratified bi-simulation provides the possibility to compute an approximative schema [13] in order to reduce the schema size. Wang et al. [16], instead, proposed to cluster nonequivalent but similar objects during the process of schema extraction to reduce computational complexity.

Graph linearisation approaches can also be found in other contexts for detecting frequent graph patterns [18]. A very efficient algorithm is based on DFS coding [17] which leverages a unique sequential representation of a graph obtained via a depth first search (thus DFS). Frequent graph patterns can then be discovered in this linear encoding. DFS coding has also been applied on RDF data to estimate the frequency of data patterns [12].

3 An Overview of SchemEX

The original purpose of SchemEX [10, 11] is to link schematic information to data sources which provide resources conforming to this schema. Data sources are, e.g. static RDF documents and SPARQL end points [8]. The central concepts of SchemEX are type clusters (TC) and equivalence classes (EQC). A TC contains all

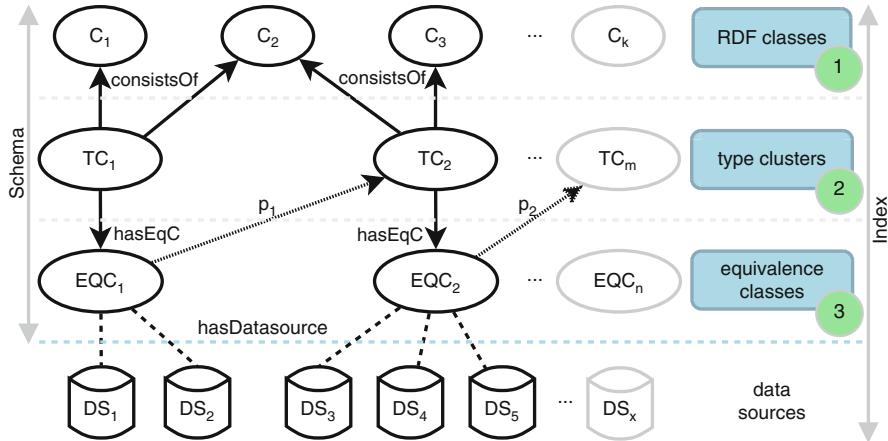


Fig. 1 SchemEX index structure with three layers leveraging RDF typings and property sets

data sources which provide resources conforming to the same well-defined set of types/classes. The EQC divide the data sources in each TC into disjoint subsets, defined by the set of properties the resources in this data source share and in which TC the object of the property link lies. The EQC also allow for deriving property clusters (PC), i.e. groups of resources which share the same set of properties without any restrictions on their types. An overview of the information contained in a SchemEX index is shown in Fig. 1.

Since its introduction, the information encoded by SchemEX is analysed under other aspects as well. Current works investigate the entropy of the distribution of TC and PC, as well as their joint distribution and the redundancy encoded in a schema considering both kinds of information [5]. Also the option to derive type information of resources with given properties or vice versa is considered. Such derivations can be used to recommend relevant vocabulary to a linked data engineer.

As already mentioned in the introduction, for large data sets SchemEX can be computed very efficiently using a stream-based approach. In this case, the analytical component is operating in a single-pass fashion over a stream of RDF triples. By using a windowing technique, it is possible to obtain a very good approximation of the true schema using commodity hardware. However, the windowing technique entails a certain risk of loss of schema information: If the information regarding one resource extends beyond the window size, this information is considered only partially for the construction of the schema.

4 Evaluation Setup

Stream-based schema construction causes errors in the schema, in the sense that there are deviations from the true schema of the analysed data. The amount of errors depends on the considered data set and the size of the window used when accessing the stream of RDF triples. The impact of the errors depends on the application scenario. Given the various applications considered for the SchemEX index so far, there is a range of metrics which can be considered for evaluating the accuracy of the schema.

4.1 Evaluation Metrics

The following metrics are directly related to the application of SchemEX as a look-up service for LOD data sources, for deriving types from properties and vice versa as well as a basic count of the elements in the schema.

Count Statistics. Here, we consider the number of essential elements in the schema information of SchemEX. This entails counting the number of observed types, properties, TC, PC and EQC. To evaluate the accuracy of the counts, we compute the relative deviation of the figures from the count of the corresponding element in the gold standard schema. Note that such a simple count is a rough metric. Even if an approximated schema contains as many, for instance, TC as the gold standard, this does not mean that the specific composition of the TC nor the contained data sources are equivalent.

Querying. In a retrieval setting we evaluate the accuracy with which the approximated schema can answer queries. A query defines a graph pattern of interest, and SchemEX is used to retrieve a list of data sources providing resources conforming with this pattern. Given the lack of real-world queries, we simulate all queries that lead to nonempty result sets on the gold standard schema. This approach follows previous work in evaluating SchemEX [11] and can be considered a very exhaustive test. Given the different levels of the schema, the queries can have different complexities. From simply asking for all data sources containing resources of a given type to resources satisfying exactly a given set of types (i.e. a TC) or properties (i.e. a PC), to resources satisfying a common superset of types (TCS) or even to resources of a given type set linking to resources of another given type set via given properties (i.e. EQC). Given this equivalence to schema elements, we will accordingly talk about type, TC, PC, TCS or EQC queries. Based on the result

sets for these queries we can compute classical information retrieval metrics such as recall, precision and F1. We aggregate the metrics over all possible queries using a macro-average.

Schema Information. We consider the joint distribution of TC and PC as main schema characteristics of the data [5]. The information in SchemEX allows for an estimation of this joint distribution based on the volume of data sets stored in the schema index. By computing the marginal entropy of TC and PC, the joint entropy of TC and PC together and a normalised variation of mutual information (MI), we obtain aggregate values for the distribution. These aggregate values can be compared relative to the same values computed on the gold standard.

4.2 Evaluation Methodology

To observe the impact of different data samples we need several real-world data sets. In this way we can observe the behaviour over different samples of data, which cover different sources in the LOD graph and have different characteristics. These data sets can then be analysed under various settings for the window size in the stream-based schema construction and compared to the gold standard schema. To additionally see how the deviations develop within the data set when more triples are processed, the data sets will be analysed at various degrees of their data being processed.

5 Results

We used the approach described in the previous section for evaluating the accuracy of stream-based schema extraction. As data set we employed the RDF data provided for the Billion Triples Track (BTC) of the Semantic Web Challenge 2011. This data set has the advantage to represent a sample of real-world data taken from the LOD cloud. Furthermore, the data set is large enough to sample smaller data sets from it.

5.1 Data Sets

We use samples taken from the BTC 2011 data set¹ consisting of a total of 2.12 billion triples. The BTC data is provided in chunks of 10 million triples each. On commodity hardware, the computation of a SchemEX gold standard is feasible

¹ Available from: <http://km.aifb.kit.edu/projects/btc-2011/>.

Table 1 Characteristics of the segments from the BTC 2011 data set used for evaluation

Segment	No. of triples	BTC2011 chunks	TC	EQC	Types	Properties
BTC-Seg-1	20,000,000	000 and 001	3,130	16,229	5,448	6,537
BTC-Seg-2	20,000,000	005 and 006	257	2,840	680	281
BTC-Seg-3	20,000,000	010 and 011	2,099	8,072	1,809	2,228
BTC-Seg-4	20,000,000	015 and 016	1,340	7,220	1,264	1,009
BTC-Seg-5	20,000,000	020 and 021	2,899	10,187	2,501	1,012
BTC-Seg-6	20,000,000	025 and 026	980	6,260	1,372	275
BTC-Seg-7	20,000,000	030 and 031	16,421	25,695	18,683	3,461
BTC-Seg-8	20,000,000	035 and 036	3,076	10,372	3,748	810
BTC-Seg-9	20,000,000	040 and 041	1,814	16,460	2,526	349
BTC-Seg-10	20,000,000	045 and 046	3,543	10,180	3,770	830
BTC-Seg-11	20,000,000	050 and 051	578	4,303	1,349	349
BTC-Seg-12	20,000,000	055 and 056	424	3,038	1,222	242
BTC-Seg-13	20,000,000	060 and 061	137	1,865	775	153
BTC-Seg-14	20,000,000	065 and 066	3,815	10,907	4,168	512
BTC-Seg-15	20,000,000	070 and 071	7,905	32,852	12,166	2,690
BTC-Seg-16	20,000,000	075 and 076	2,703	7,699	3,115	365

for up to 20 million triples. We created 16 such segments of 20 million triples. Table 1 gives an overview of the data sets as well as their characteristics on the gold standard. For each segment we computed schemata stepwise from the 1st million to the full 20 million snapshot, thus giving us 200 schemata to compute. Each of these schemata was computed as perfect schema (gold standard) and with the stream-based approximation using a window size of 1k, 10k, 25k, 50k, 100k, 250k and 500k resources. So, for each schema we had eight configurations. In total we computed $16 \cdot 20 \cdot 8 = 2,560$ SchemEX indices.

5.2 Observations

We will start to look at the variation of our quality metrics over the data segments. To this end, we consider only the performance of the schema approximation based on the largest window size of 500k triples, which constitutes the best approximation we considered in our evaluations. Figure 2 shows the accuracy for count statistics (Fig. 2a), retrieval performance (Fig. 2b) and entropy computation (Fig. 2c) over all our data sets.

When looking at the performance regarding the count statistics, we observe deviations of no more than 10% from the actual numbers. The count of types and properties is perfect. This is obvious as for counting types and properties it is sufficient to see one triple at a time. These counts do not depend on combinations of triples and are, thereby, independent of the employed window size. Regarding the more complex structures, i.e. TC, PC and EQC, this is not the case. Here, it is necessary to observe all triples relevant to a resource within the window buffer. For TCs this means to see *all* the type information, and for the PC count we need to

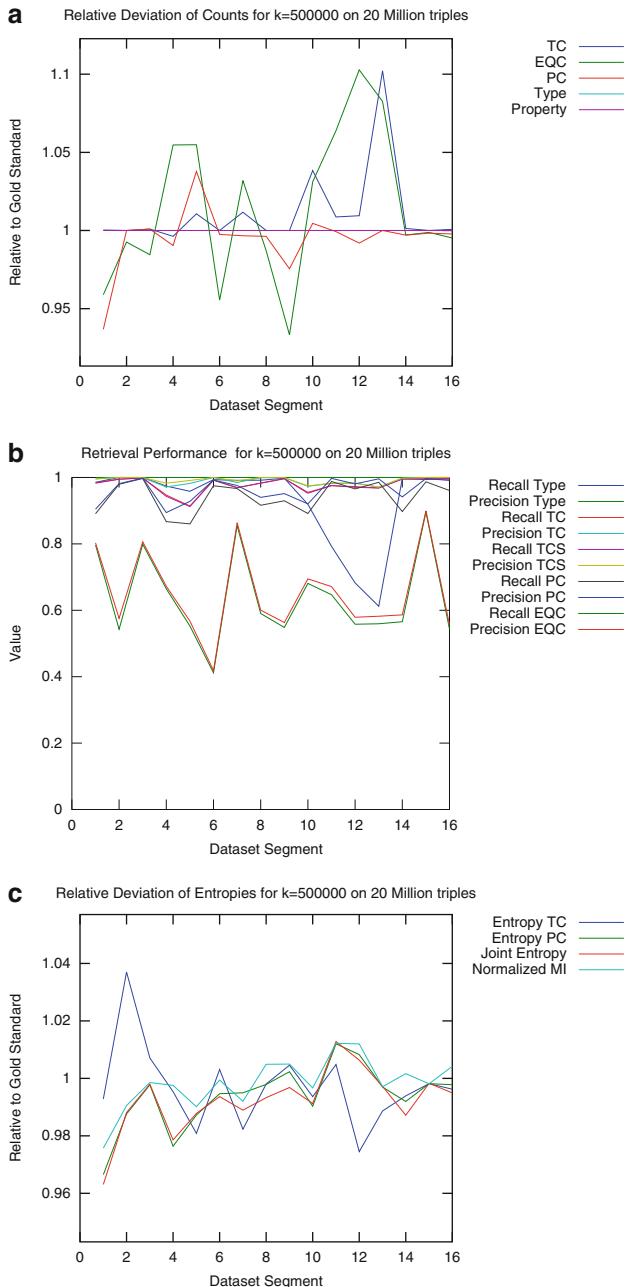


Fig. 2 Performance metrics over all data segments for a window size of 500k. (a) Count statistics relative to gold standard, (b) Retrieval performance, (c) Entropy values relative to gold standard

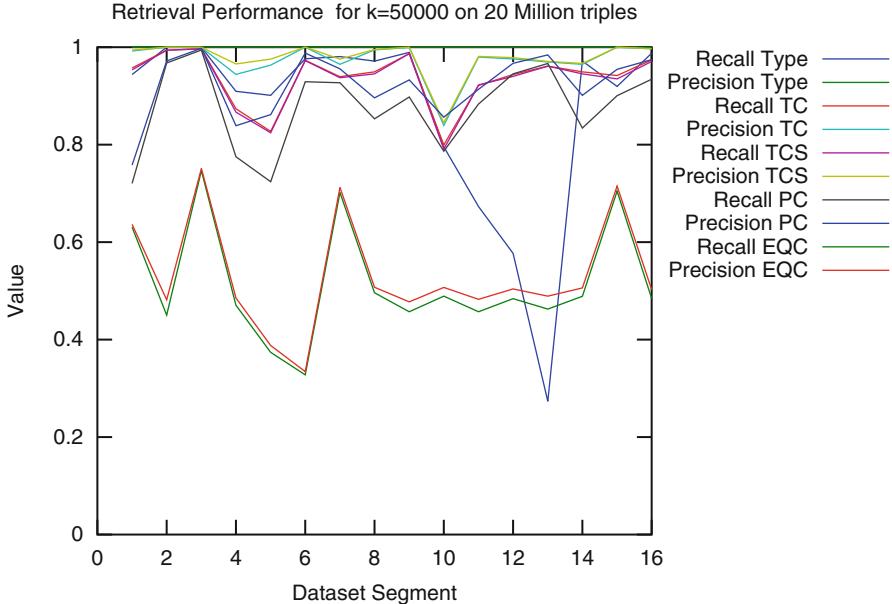


Fig. 3 Retrieval performance over all data segments for a window size of 50k

see *all* the properties within the window buffer. The correct detection of EQCs even requires to see all the types, the properties related to the considered resource as well as all the type information for linked resources. Given the increasing complexity of the structures, we can observe that also the count statistics deviate more for the EQC than for the PC and TC. Also regarding the retrieval metrics we observe similar tendencies. Retrieval performance for the simpler type TC, TCS and PC queries is very high in general with values between 0.89 and 1.0 for recall and precision in Fig. 2b. Again, more complex EQC queries show a higher fluctuation in performance and vary more. For both, recall and precision, we observe values in the range between 0.42 and 0.86. To examine this behaviour, we will look in more detail at the examples of data segments BTC-Seg-6 and BTC-Seg-15 below, as these segments show the worst and best performance, respectively. The approximation of the entropy values (Fig. 2c), instead, is quite accurate over all data segments. Relative to the entropy values of the perfect schema, there is no deviation of more than 4%.

The quality of the schema approximation with smaller settings for the window size is very high as well, but—as expected—do not reach the level of the 500k window. However, the general behaviour we observed was very similar. The retrieval performance of schema indices based on the default SchemEX setting of a window size of 50k resources is shown in Fig. 3. We can see that the problematic data segments are the same as for the higher window size settings. The absolute

recall and precision values, instead, are lower and range between 0.72 and 0.99 for the simpler query types and between 0.33 and 0.76 for the challenging EQC queries.

As under the aspects of count statistics and entropy values the approximated schemata provide very good results over all data segments, we now investigate the extreme cases in more detail, in order to understand which reasons might be behind these observations. To this end, we take a closer look at the data segment BTC-Seg-6 (Fig. 4a) and BTC-Seg-15 (Fig. 4b). The figures show the F1 metric performance of EQC queries for all the choices of windows size and when considering an increasing portion of triples in the data segments.²

While for data segment BTC-Seg-15 we can observe a value of the F1 metric which is relatively stable and actually increases suddenly after approximately 12 million triples, the F1 values behave just the opposite for data segment BTC-Seg-6. Already starting at a lower level, the most intriguing part is the sudden and significant drop after processing 9 million triples. This behaviour is quite atypical and we did not observe it for any of the other data segments. For the other data segments we also observed a relative constant development of the metric and a relative stability after having processed 5 to 10 million triples.

Also, this unusual change in performance is observed only for EQC queries. As can be seen in the example of in BTC-Seg-6 in Fig. 5, the F1 metric of TC and PC queries is actually very high. Thus, it seems we are facing a problem in detecting the EQC elements in the schema correctly.

A manual inspection of the data provided an explanation for the drop of performance. At this position the data in this segment mainly consists of automatically created FOAF profile information. These profiles are encoded using blank nodes to model the friends of a person. The blank nodes then link to the actual public FOAF profile of the friend via an `rdfs:seeAlso` property. However, the large volume of blank nodes quickly fills the window of resources considered on the RDF stream. Therefore, the definition of the type of the referenced resource comes at too large distance in the data stream for still being useful. In conclusion the derived EQC pattern is incomplete and causes wrong responses to the queries. This also explains that the entropy values are not affected by this problem. Given that for estimating the joint distribution of TC and PC, it is not necessary to know the types of the objects in a triple, here the lack of this information has no impact (c.f. Fig. 6).

However, aside from this one peculiar case in the data, the approximated schemata are relatively accurate and performed very well under the considered metrics. Furthermore, given that the use of blank nodes—and in particular such an excessive use—in linked open data is disregarded and considered bad practice, we expect not to encounter this problem very often and even less in the future. Alternatively, a solution to this problem would be to extend SchemEX computation to treat blank nodes separately.

²Essentially, the rightmost values of the curves correspond to the metric value we displayed in the plots above of a situation of having processed the complete 20 million triples of the data segment.

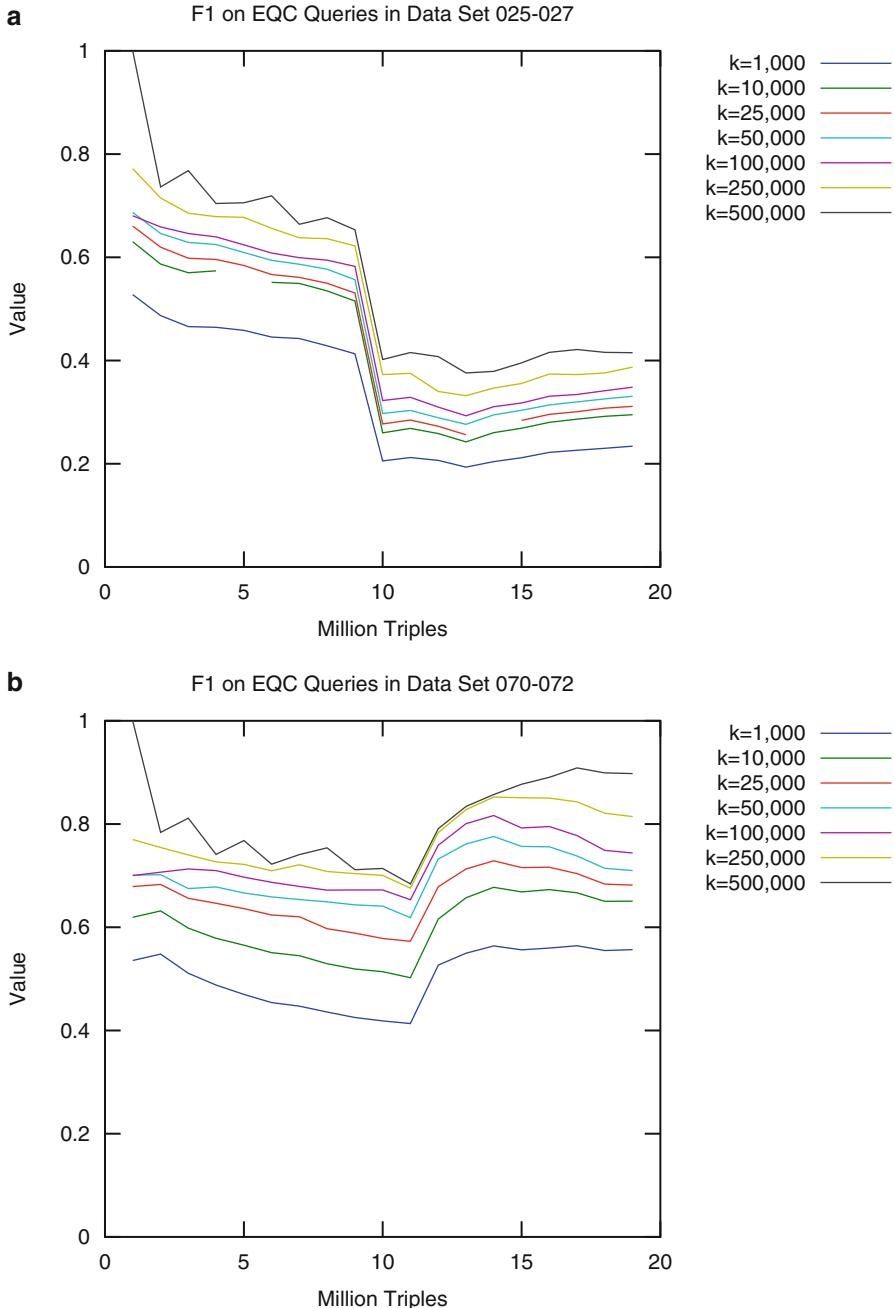


Fig. 4 Development of F1 metric on EQC queries for different window size settings and increasing number of triples. **(a)** Data segment BTC-Seg-6, **(b)** Data segment BTC-Seg-15

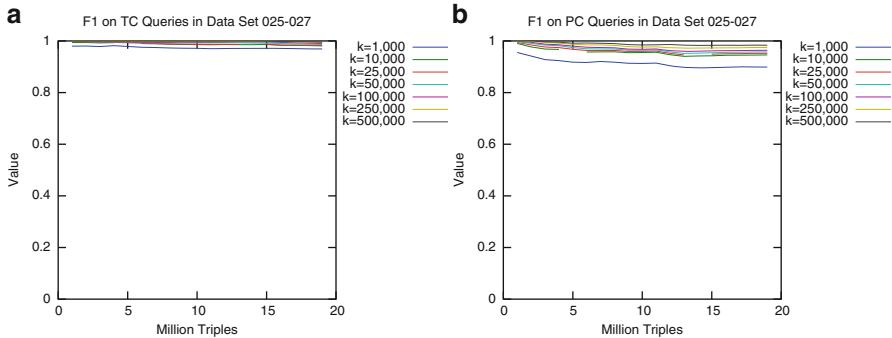


Fig. 5 Development of F1 metric on TC and PC queries in BTC-Seg-6. **(a)** Queries of type TC, **(b)** Queries of type PC

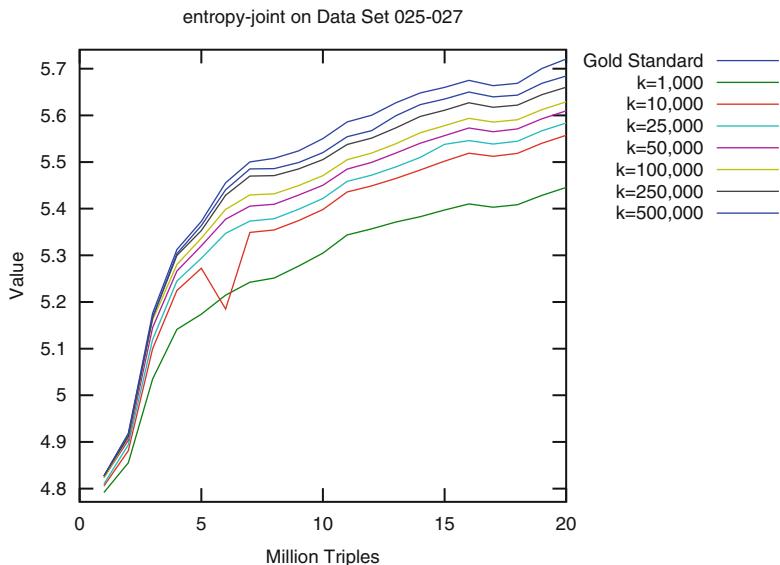


Fig. 6 Joint entropy of data segment BTC-Seg-6 with increasing data set size

6 Conclusion

In this paper we investigated the accuracy of stream-based schema construction from RDF data as one scenario for scalable semantic web data processing. The derived schema can be leveraged for several scenarios, such as indexing, vocabulary suggestion and analysis of the distribution of type and property combinations. We analysed the accuracy of a stream-based schema construction vs. an exact schema computation under several scenarios and with appropriate metrics. We also

investigated the impact of parameter choice and the deviation over iteratively increasing data sets. Running the example on several subsamples of a much larger data set taken from the BTC 2011, we demonstrate that the quality is not constant and depends on the input data. Massive use of blank nodes in LOD data sets can cause a significant drop of retrieval performance on the challenging EQC queries. However, for all other metrics, we observed very good values and little loss of accuracy. Deviation in count statistics reaches at most 10%; the deviations in entropy lie in the range of at most 2–4%. Recall and precision of other query types typically lies in the range 0.89 and 1.0.

As future work we will address the treatment of blank nodes in SchemEX. Given the impact and the (surprisingly) frequent use of blank nodes in some domains of the LOD cloud, we will store them in a separate cache during the stream-based schema computation. This will counterbalance the loss of information for proper resources identified by a full URI. Thereby, we should at least gain good performance for resources which conform to the best practices of LOD publishing.

Acknowledgements The research leading to these results has received partial funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST and grant agreement no. 287975, SocialSensor.

References

1. Böhm, C., Freitag, M., Heise, A., Lehmann, C., Mascher, A., Naumann, F., Ercegovac, V., Hernandez, M., Haase, P., Schmidt, M.: Govwild: integrating open government data for transparency. In: Proceedings of the 21st International Conference Companion on World Wide Web, pp. 321–324. WWW '12 Companion. ACM, New York, NY (2012)
2. Böhm, C., Lorey, J., Naumann, F.: Creating void descriptions for web-scale data. *Web Semant. Sci. Serv. Agents World Wide Web* **9**(3), 339–345 (2011)
3. Gallego, M., Fernández, J., Martínez-Prieto, M., de la Fuente, P.: Rdf visualization using a three-dimensional adjacency matrix. In: SemSearch'11: Proceedings of 4th International Semantic Search Workshop, 2011
4. Goldman, R., Widom, J.: Dataguides: Enabling query formulation and optimization in semistructured databases. In: Jarke, M., Carey, M.J., Dittrich, K.R., Lochovsky, F.H., Loucopoulos, P., Jeusfeld, M.A. (eds.) VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25–29, 1997, Athens, Greece. pp. 436–445. Morgan Kaufmann, San Francisco (1997)
5. Gottron, T., Knauf, M., Schegemann, S., Scherp, A.: Explicit and implicit schema information on the linked open data cloud: Joined forces or antagonists? Tech. Rep. 06/2012, Institut WeST, Universität Koblenz-Landau (2012)
6. Gottron, T., Scherp, A., Krayer, B., Peters, A.: Get the google feeling: Supporting users in finding relevant sources of linked open data at web-scale. In: Semantic Web Challenge, Submission to the Billion Triple Track, 2012
7. Haesenblas, M., Halb, W., Raimond, Y., Heath, T.: What is the size of the semantic web? In: Proceedings of the International Conference on Semantic Systems, 2008
8. Heath, T., Bizer, C.: Linked Data: Evolving the Web Into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool (2011)

9. Isele, R., Harth, A., Umbrich, J., Bizer, C.: LDspider: An open-source crawling framework for the web of linked data. In: Poster, International Semantic Web Conference 2010. Shanghai, China (2010)
10. Konrath, M., Gottron, T., Scherp, A.: Schemex – web-scale indexed schema extraction of linked open data. In: Semantic Web Challenge, Submission to the Billion Triple Track, 2011
11. Konrath, M., Gottron, T., Staab, S., Scherp, A.: SchemEX-Efficient Construction of a Data Catalogue by Stream-based Indexing of Linked Data, Web Semantics: Science, Services and Agents on the World Wide Web, **16**(5), pp. 52–58, 2012. The Semantic Web Challenge. (2011)
12. Maduko, A., Anyanwu, K., Sheth, A., Schlickelman, P.: Graph summaries for subgraph frequency estimation. In: Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications, pp. 508–523, ESWC’08. Springer, Berlin, Heidelberg (2008)
13. Nestorov, S., Abiteboul, S., Motwani, R.: Extracting schema from semistructured data. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pp. 295–306, SIGMOD ’98. ACM, New York, NY (1998)
14. Nestorov, S., Ullman, J.D., Wiener, J.L., Chawathe, S.S.: Representative objects: Concise representations of semistructured, hierachial data. In: Proceedings of the Thirteenth International Conference on Data Engineering, pp. 79–90, ICDE ’97. IEEE Computer Society, Washington, DC (1997)
15. Papakonstantinou, Y., Garcia-Molina, H., Widom, J.: Object exchange across heterogeneous information sources. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 251–260, ICDE ’95. IEEE Computer Society, Washington, DC (1995)
16. Wang, Q.Y., Yu, J.X., Wong, K.F.: Approximate graph schema extraction for semi-structured data. In: Proceedings of the 7th International Conference on Extending Database Technology: Advances in Database Technology, pp. 302–316, EDBT ’00. Springer, London (2000)
17. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: Proceedings of the 2002 IEEE International Conference on Data Mining, p. 721, ICDM ’02. IEEE Computer Society, Washington, DC (2002)
18. Yan, X., Yu, P.S., Han, J.: Graph indexing: a frequent structure-based approach. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 335–346, SIGMOD ’04. ACM, New York, NY (2004)

Building Large-Scale Knowledge Base for Relations from Text

Junfeng Pan, Haofen Wang, and Yong Yu

Abstract Recently more and more structured data in the form of RDF triples have been published and integrated into Linked Open Data (LOD). While the current LOD contains hundreds of data sources with billions of triples, it has a small number of distinct relations compared with the large number of entities. On the other hand, Web pages are growing rapidly, which results in much larger number of textual contents to be exploited. With the popularity and wide adoption of open information extraction technology, extracting entities and relations among them from text at the Web scale is possible. In this paper, we present an approach to extract the subject individuals and the object counterparts for the relations from text and determine the most appropriate domain and range and the most confident dependency path patterns for the given relation based on the EM algorithm. As a preliminary result, we built a knowledge base for relations extracted from Chinese encyclopedias. The experimental results show the effectiveness of our approach to extract relations with reasonable domain, range, and path pattern restrictions, as well as high-quality triples.

1 Introduction

In recent years, many knowledge bases, such as DBpedia, YAGO, and Zhishi.me, have been published on the Web in the form of linked data, which are very useful for both human reading and machine consumption. Comparing with the number of different entities in these knowledge bases, there are only a few distinct relations.

J. Pan (✉) • H. Wang • Y. Yu

APEX Data & Knowledge Management Lab Shanghai Jiao Tong University
800 Dongchuan Rd., Shanghai 200240, China

e-mail: panjf@apex.sjtu.edu.cn; whfcarter@apex.sjtu.edu.cn; yyu@apex.sjtu.edu.cn

Furthermore, these knowledge bases only extract data from structured or semi-structured data sources without considering implicit knowledge from unstructured text, which is in a huge and increasing amount on the Web.

On the other hand, open information extraction, such as Machine Reading [18] and Never-Ending Language Learning [5], focuses on extracting entities and their relations from text at Web scale. This work aims to have three advantages over traditional information extraction methods: (1) less human-labeled data, (2) relations discovered automatically, and (3) less complexity and high scalability.

In this paper, we are motivated to build a knowledge base of relations extracted from text by leveraging open information extraction techniques. Moreover, for each relation, we extract not only subject-object examples but also high-level restrictions such as the domains and ranges from text. Both information are quite useful to describe relations, which can be used for further natural language processing training or high-quality ontologies for additional extraction. To extract such information, we adapt a novel algorithm based on Expectation-Maximization (EM). And an experimental relation knowledge base is built to show the effectiveness and efficiency of our algorithm.

The rest of the paper is organized as follows: We review some related work in Sect. 2. Then we give structure of our relation knowledge base in Sect. 3. The algorithm to build the relation knowledge base is proposed in Sect. 4 with some analysis. In Sect. 5 we build an experimental relation knowledge base on Chinese encyclopedias and Zhishi.me to show the efficiency and effectiveness of our algorithm. We also provide Web access to it. Finally, we conclude our work in Sect. 6.

2 Related Work

Most of the related work comes from two research areas: open information extraction and knowledge base construction.

2.1 Open Information Extraction

Open information extraction aims to extract entities and their relations from text domain in an independent way at Web scale [8]. The first open information extraction system, TextRunner [3], used a Naive Bayes model to train a self-supervised learner and build a single-pass extractor. Subsequent work has shown that other models such as CRF [4] or Markov Logic Network [22] can improve the extraction performance. Second-generation open information extraction [9] improves the results using constraints on verbs and arguments learner.

Another famous open information extraction system is the Never-Ending Language Learning project. The main method in this system is the coupled

semi-supervised learning algorithm [6]. This is an ontology-driven information extraction system as well; given some useful meta-properties, it is able to populate the Semantic Web [14].

There are some differences between previous work and our algorithm. One is although open information extraction may accept an initial ontology, it does not use background knowledge from the Web. WOE [21] uses infoboxes on Wikipedia as background knowledge, but we try to discover richer relation-dependent patterns rather than their general patterns. Another difference is that previous work focuses on the concrete subject and object of a relation. This is only a part of our knowledge base, the examples. The other part is the restrictions. Our system, related to OntExt [15], not only extracts concrete examples but also provides abstract restrictions on the relations for more insight into the relations and an ontology to help additional relation extraction.

2.2 Knowledge Base Construction

Nowadays there are a lot of knowledge bases on the Web, especially in the linked data form. Notable ones include DBpedia [1], YAGO [20], and Zhishi.me [16]. But these knowledge bases mainly focus on entities; there are only a few distinct relations. Furthermore, they do not use the knowledge in text which is available in large amount. Our relation knowledge base is complementary to the entity-oriented linked data because we extract many automatically discovered relations with rich examples and restrictions from text.

There are also some knowledge bases focusing on relations, including FrameNet [2], PropBank [17], VerbNet [19], and some verbs in WordNet [12]. Unfortunately most of these are created manually, so updating them or creating a similar new one will cost a lot and it is not able to gain benefit from the corpora at Web scale as well. Our work aims to overcome these problems, building the relation knowledge base automatically.

PRISMATIC is a large-scale lexical relation resource that is automatically created over text [11]. Its extraction process is mainly based on some simple rules, while our algorithm involves machine-learning technology.

3 The Relation Knowledge Base

3.1 Relation Knowledge Group

Our relation knowledge base *RelationKB* can be seen as a set of relation knowledge groups. Each group G has the following components: the relation v , a list of examples E_v , and three restrictions (the domain D_v , the range R_v , and the path patterns P_v).

An example of a relation v is a pair of subject and object which are linked by the relation. For instance, *(Beijing, China)* is an example of the relation *LocatedIn* because Beijing is located in China.

Furthermore, the examples are extracted from text so we need a criterion to judge whether an extracted example is correct or not. We try to find a function f_E such that $f_E(s_i, o_i | v) > f_E(s_j, o_j | v)$ for all the pairs where (s_i, v, o_i) is more correct than (s_j, v, o_j) . Then we can easily judge which extracted example is more credible.

Not only the examples but also the restrictions are very important to our knowledge base. There are three kinds of restrictions: domain, range, and path patterns.

The domain restriction of a given relation is used to state the properties of the subject, while the range restriction describes the properties of the object. In order to specify these restrictions, it is good to use some knowledge in the online encyclopedias and Linked Open Data. In such Web sites, there is always some information about categories for a given entity. For example, *Beijing* may have tags like *China*, *Capital*, and *Municipality*.

Another important kind of restriction in our relation knowledge base is the set of path patterns. Inspired by [21], we extract the path pattern from the dependency parse tree. Given a sentence containing the relation with subject and object, we can extract a dependency parse path from the subject or object word to the relation word. Similar to [21], we use generalized part-of-speech tags (i.e., all noun tags to “N”, all verb tags to “V”) in the dependency path to represent the path pattern.

The same as the examples, we also need scores for the restrictions to indicate credibility. A high score means the tag or the path pattern is strongly related to the given relation.

3.2 Relation Knowledge Base

In order to put all the above together, we should solve the following problems:

- We should extract credible subject–object pairs and their relations from text.
- We should specify credible domains, ranges, and path patterns for the relations from text.

Both problems are very challenging and they are also highly coupled. If we can extract high-quality examples, we can use the tags and the path patterns from them to get restrictions perfectly. On the other hand, if we have high-quality restrictions, we can use them to extract more accurate examples. The high degree of coupling is an important property for us to design the algorithm.

4 Building the Relation Knowledge Base

4.1 Architecture Overview

The key idea of our algorithm is to solve the two coupled problems, extracting the examples and specifying the restrictions for the relations, from text simultaneously.

Firstly we extract candidate examples from preprocessed text and candidate restrictions from the candidate examples and assign these candidates initial scores. Then we update the scores of the examples and the restrictions by each other iteratively. After a number of iterations, we get the final scores of the examples and the restrictions and form our relation knowledge base.

As shown in Fig. 1¹, the process of building relation knowledge base has three main steps: text annotation, candidates extraction, and iterative score estimation. We should also note that finally the knowledge base could be linked to Linked Open Data.

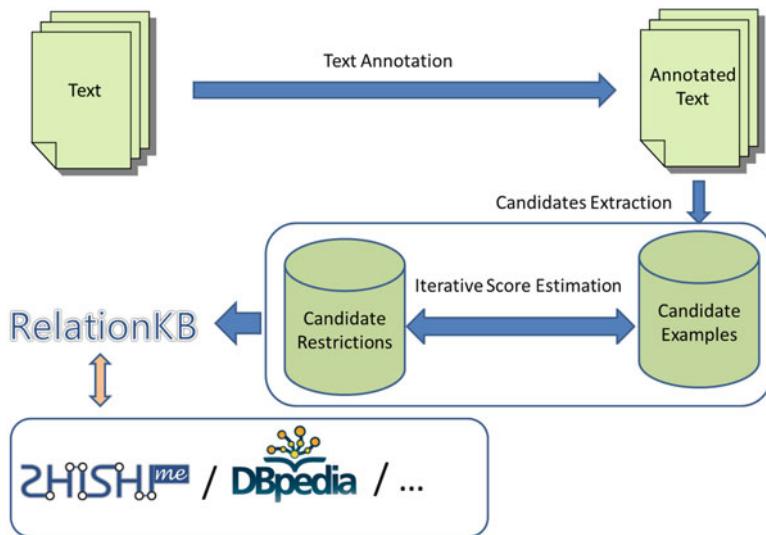


Fig. 1 The process of building relation knowledge base

¹All sites mentioned in this figure have rights and marks held by their respective owners.

4.2 Text Annotation

The text annotation step employs some NLP tools to convert the raw text into a sequence of annotated sentences. The annotations include tokenization (word segmentation for Chinese), POS tagging, and dependency parsing on top of the raw input text. The annotated text is used as the input for the candidates extraction step.

4.3 Rule-Based Candidates Extraction

According to the concept of dependency grammar [13], we can directly extract candidate examples from the annotated text. We identify the relations in the sentences according to POS tags, and for each relation, we enumerate nouns in its subtrees as subject or object candidates. We also extract candidate restrictions in the candidates extraction step. For each candidate example, we tag the head nouns of the subjects and objects to form the candidate domain and range of the relations. We also extract candidate path patterns from subject-relation-object paths on the dependency trees.

Figure 2 shows that some candidates are extracted from two annotated sentences. The left one is a sentence which means *Beijing is located in China*, and we extract one candidate example with tags and path pattern, while the right one is a sentence which means *The City Museum is located in the Museum Square*, and two candidate examples with restrictions are extracted.

We should note that in the right one, one relation *LocatedIn* is matched to two candidate examples (*City_Museum*, *Museum*) and (*City_Museum*, *Square*) in one sentence, which implies one of the candidate examples should be wrong. In this case, we get the wrong candidate examples (*City_Museum*, *Museum*) because of an error made by the dependency parser (the head of the later *Museum* is probably the *Square* not the *LocatedIn*). Sometimes the extracted candidate example is wrong even if the dependency tree is right. In order to handle these errors, we should introduce another step, score estimation of these candidates, so that we can determine the credibility of the examples and the restrictions.

4.4 EM-Based Score Estimation

In the score estimation step, we group the candidate examples and restrictions by relation and process each relation independently. For each relation v , we group the candidates from one sentence into one candidate group. Suppose there are n candidate groups for v and m_i candidates in group c_i . In one group, each candidate c_{ij} contains the extracted candidate subject s_{ij} with tags $T(s_{ij})$, object o_{ij} with tags $T(o_{ij})$, and path pattern p_{ij} . Figure 2 can be seen as two candidate groups for relation *LocatedIn*, one is a group with single candidate, while the other has two candidates.

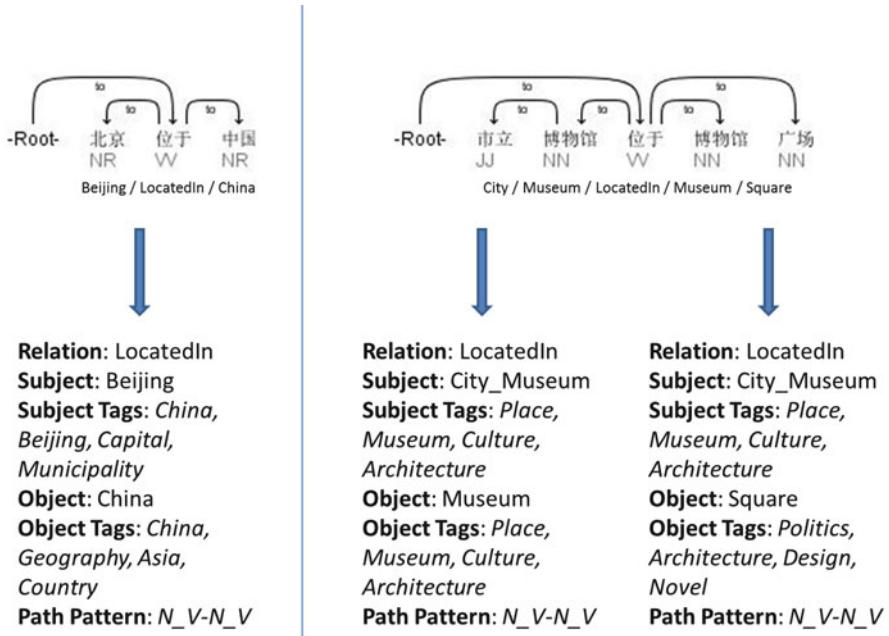


Fig. 2 Text annotation and candidates extraction. Extracted words and tags have been translated to English

We need a score function $f_E(s_{ij}, o_{ij} | v)$ to judge whether the example (s_{ij}, o_{ij}) is a credible subject-object pair for v . It is reasonable that the credibility of the example is positively correlated to the credibility of its tags and path pattern so we use Eq. (1) to estimate the score. Here $N_w(t)$ is the number of words having tag t . And we also introduce three functions $f_D(t_d | v)$, $f_R(t_r | v)$, $f_P(p | v)$ to indicate the credibility of the tag t_d in the domain, the tag t_r in the range, and the path pattern p for v .

$$f_E(s_{ij}, o_{ij} | v) = f_P(p_{ij} | v) \left(\sum_{t_d \in T(s_{ij})} \frac{f_D(t_d | v)}{N_w(t_d)} \right) \left(\sum_{t_r \in T(o_{ij})} \frac{f_R(t_r | v)}{N_w(t_r)} \right) \quad (1)$$

Intuitively, $f_D(t_d | v)$ should be the real count of t_d over the maximum possible count. We assume that each candidate group is worth 1 count (which implies the maximum possible count is n) and allocate it for each candidate in one group according to $f_E(s_{ij}, o_{ij} | v)$. The way to compute f_R and f_P is almost the same as f_D . Details are given in Eq. (2):

$$f_L(t | v) = \frac{1}{n} \sum_{cond} \frac{f_E(s_{ik}, o_{ik} | v)}{\sum_{k=1}^{m_i} f_E(s_{ik}, o_{ik} | v)}$$

$$(L, t, cond) \in \{(D, t_d, t_d \in T(s_{ij})), (R, t_r, t_r \in T(o_{ij})), (P, p, p = p_{ij})\} \quad (2)$$

According to Eqs. (1) and (2), the scores of the examples and the restrictions are interdependent. It is natural to design an Expectation-Maximization [7] algorithm to estimate them simultaneously:

1. Initialize $f_E(s_{ij}, o_{ij} | v) = \frac{1}{m_i}$.
2. Update f_D , f_R , f_P , and f_E iteratively until convergence:
 - a. **E-step:** For every tag and path pattern, update $f_D(t_d | v)$, $f_R(t_r | v)$, and $f_P(p | v)$ using Eq. (2).
 - b. **M-step:** For every example, update $f_E(s_{ij}, o_{ij} | v)$ using Eq. (1).

Finally we can build the relation knowledge base by the examples (subject–object pairs) and the restrictions (tags for domain and range and path patterns) for each relation according to the scores of these candidates.

5 Experiment on Chinese Encyclopedias and Zhishi.me

In this section, we built an experimental relation knowledge base in Chinese for evaluation on our algorithm. We extracted the examples and restrictions of relations from the Chinese online encyclopedias. We also provide some statistics and Web access to our knowledge base.

5.1 Data

We built an experimental relation knowledge base from the abstract text of all the entries in three online Chinese encyclopedias, Wikipedia in Chinese, Baidu Baike, and Hudong Baike. There are 2,517,826 pieces of text and 20,637,524 sentences in total.

In addition to the text, we use the property `Zhishi:category` at a Chinese Linked Open Data, Zhishi.me, which extracts the categories for the entities from the above three online encyclopedias. We used them to tag the extracted entities. There are a total of 985 tags and 3,109,448 words that have at least one of these tags.

5.2 Processing

We processed the data according to the three steps presented in Sect. 4: text annotation, candidates extraction, and score estimation. In the text annotation step, FudanNLP² is employed to do the word segmentation, POS tagging, and

²FudanNLP, an Open Source Chinese Natural Language Processing toolkit, <http://code.google.com/p/fudannlp/>.

Table 1 Statistics of the demo relation knowledge base: 7,097 relations in total, numbers in parentheses are standard deviations. We only keep the example with the highest score for each candidate group

Type	Avg. tags in domain	Avg. tags in range	Avg. path patterns	Avg. examples
Number	107.46(120.51)	103.13(115.82)	9.68(20.40)	276.03(4473.40)

Table 2 Running on Hadoop: The running time is the longest running time among all the mappers/reducers

	Running side	# Mappers or reducers	Running time
Word segmentation and POS tagging	Mapper	27	38 min, 23 s
Dependency parsing	Mapper	28	10 min, 14 s
Candidates extraction	Mapper	29	2 min, 5 s
EM-based score estimation	Reducer	100	5 min, 58 s

dependency parsing on the text. In the candidates extraction step, we apply the rules we defined in Sect. 4.3. We also use 16,122 verbs listed in the Chinese Proposition Bank³ to remove some incorrect relations we extracted according to the POS tag. In the score estimation step, we use the EM algorithm presented in Sect. 4.4, and our convergence condition is that the sum of the delta scores of all the restrictions for each relation is smaller than 3×10^{-6} . The average number of iterations for all the relations is 16.35 with the standard deviation 68.51. Some statistics of our demo relation knowledge base are shown in Table 1.

Our algorithm is very easy to parallelize. In the text annotation and candidates extraction step, each sentence can be processed independently, while in the score estimation step, each relation can be processed at the same time. To take advantage of this, we run our algorithm on Hadoop, a map-reduce framework. Table 2 gives details on running time.

5.3 Evaluation

We evaluate our experimental relation knowledge base for both the restrictions and the examples. We sample 40 relations from our demo; the correctness of the domain, the range, and the examples are judged by our volunteers. Since the path patterns are not easily judged by humans, we do not evaluate them directly, but the path patterns are already judged indirectly because they also influence the other components of the relation knowledge base.

For each relation, the domain, the range, and the examples can be seen as three ranked lists according to the scores of the elements in them. We use the average precision (average of the precision value obtained for the top k elements, each time a relevant/correct element is retrieved) as the metric. The mean average

³<http://verbs.colorado.edu/chinese/cpb/>.

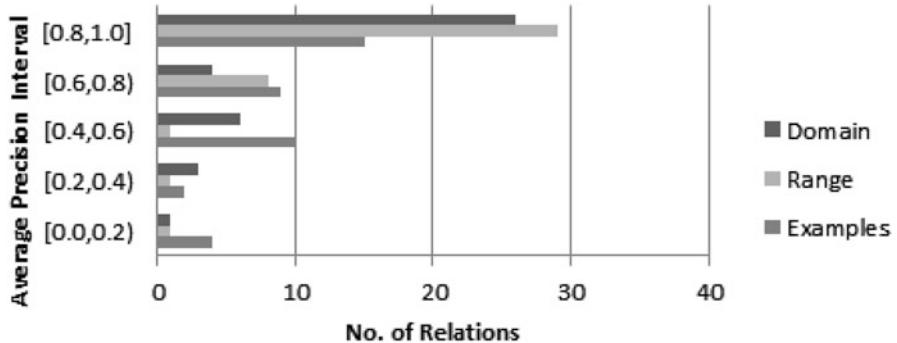


Fig. 3 The average precision distribution on the sample relations

Table 3 Statistics of subject–object pairs linked to Zhishi.me

Type	Both linked	Only the subject linked	Only the object linked	Neither linked
Number	312,055	552,940	271,306	452,691

precision is 0.788 for the domain (top 5), 0.865 for the range (top 5), and 0.657 for the examples (top 15). The average precision distribution on the sample relations is shown in Fig. 3.

We have found that errors are mainly caused by the following reasons: (1) Some words have wrong tags. (2) Some relation phrases are not actually relations (maybe just modifiers to nouns) in the sentence. (3) Some extracted noun phrases are incomplete or linked to wrong entities. These kinds of errors indicate how to improve the quality of our relation knowledge base in the future (e.g., to improve tag cleaning, relation identification, entity linking, and disambiguation).

5.4 Linking to Zhishi.me

Our experimental relation knowledge base is linked to Chinese LOD Zhishi.me in two ways. For the restriction part, the tags in domain and range are naturally from Zhishi.me. For the example part, we link the subject and object to the entities in Zhishi.me by string matching. In our experiment, we have extracted 1,718,380 unique entities, and 341,156 of them are linked to Zhishi.me. Some details about entity linking are shown in Table 3.⁴

⁴The matching result should be better if more powerful entity linking involved.

5.5 Web Access

We also provide a simple Web site for users to browse our demo relation knowledge base at <http://relationkb.apexlab.org>. We use AllegroGraph RDFStore to store the triples and provide querying capabilities. There are two services in the Web site: a lookup service and a SPARQL query service.

The lookup service is provided to search relations by their labels. For every returned relation, we give two links, one to the top restrictions of the relation and the other to the selected examples. Furthermore, we can directly use the URI of the relation to visit the top restrictions page of the relation.

In the top restrictions page, the top 10 tags with their links to Zhishi.me in the domain and range of the relation are listed in descending order by their scores, as well as the top 10 path patterns. These *See More* links are used to see all the elements of a restriction. In the selected examples page, at most 30 examples are listed. For each example, we list the subject and the object with the tags of their head nouns, the path pattern, and the score of the example, and for a subject or object which can be linked to Zhishi.me, one link to its Zhishi.me page is provided.⁵

Figure 4 shows the lookup service, the top tags in the domain and range, the top path patterns, and selected examples of the relation *LocatedIn*.

For some advanced and complex search, we can input customized SPARQL queries at the SPARQL page. We can also use [http://relationkb.apexlab.org/sparql-xml?query=\[query\]](http://relationkb.apexlab.org/sparql-xml?query=[query]) to get XML Format SPARQL query results.

6 Conclusion and Future Work

In this paper, we leverage open information extraction to build a relation knowledge base from text. To extract the examples and the restrictions for the relations are two coupled problems so we design an EM algorithm to solve them simultaneously. Also we build an experimental relation knowledge base and link it to Zhishi.me to show the efficiency and effectiveness of our algorithm and how our relation knowledge base enriches the linked data.

In the future, we are planning to use an advanced method to identify the relations (e.g., the algorithm presented in [10]) and better entity-linking algorithm to improve the quality of the relation knowledge base. In addition, it would be better if we added some structures about the relations, such as relation clusters or relation hierarchies. Finally we are also planning to use our relation knowledge to populate more and more linked data from text and update the relation knowledge base itself when populating.

⁵Unfortunately Zhishi.me is down in the recent weeks, so the links do not work now. It will be fixed when Zhishi.me comes back.

a APEX-RelationKB

LOOKUP

Try: 位于 喜欢 参加 尝试

relation
位于

Submit

1 result(s)

位于 : Restrictions Examples

Lookup Service

b 位于 (See some examples)

Top Path Patterns (See More)

N_V-N_V	0.9176358065270915
N_V-N_LC_V	0.05557019911658591
N_V-N_PU_V	0.006305865541282523
N_P-V_N_V	0.004753072734301916
N_V-N_D_V	0.002881511173547661
N_LC_V-N_V	0.0002703261331630044
N_V-N_M_V	0.0018003095052369195
N_V-N_P_V	0.0017465608301301468
N_V-N_C_V	8.070756668952107E-4
N_V-N_LC_PU_V	6.024443472955099E-4

c Top Tags in Domain (See More)

地理	2HISHP 0.31304667805457803
文化	2HISHP 0.21638883882571977
镇	2HISHP 0.16488030086913416
场所	2HISHP 0.1408827089217119
行政区划	2HISHP 0.13669393211060116
机构	2HISHP 0.1267081866536434
建筑	2HISHP 0.10834090042624372
旅游	2HISHP 0.10685605849168935
历史	2HISHP 0.10585473796618246
语言	2HISHP 0.0982370079720528

d Top Tags in Range (See More)

地理	2HISHP 0.4655906353002296
城市	2HISHP 0.17459879146288465
文学	2HISHP 0.16604679068063835
文化	2HISHP 0.15679978266692918
历史	2HISHP 0.1564619545788669
行政区划	2HISHP 0.15463118768991457
语言	2HISHP 0.15021024906358374
人物	2HISHP 0.12901172755974552
镇	2HISHP 0.09164261366483384
姓氏	2HISHP 0.08121597065193721

e 位于 (See the restrictions)

Selected Examples

- 肇庆名典商旅酒店 2HISHP 位于 广东省肇庆市端州四路
 - Subject Tags : 类型 餐厅 休闲 服务 场所 动
 - Object Tags : 语言 政治 交通 行政区划 文学 设计 社会 官制 拓扑学 物理 历史 官职 技术 文化 机构 电影 电脑 工程 设施
 - Path : N_V-N_V
 - Score : 1.563389655425646E-7
- 广州天意大酒店 2HISHP 位于 广州市番禺区大石南大路
 - Subject Tags : 类型 餐厅 休闲 服务 场所 动
 - Object Tags : 语言 政治 交通 行政区划 文学 设计 社会 官制 拓扑学 物理 历史 官职 技术 文化 机构 电影 电脑 工程 设施
 - Path : N_V-N_V
 - Score : 1.563389655425646E-7
- 深圳宝丽城大酒店 2HISHP 位于 深圳经济特区公明镇建设路
 - Subject Tags : 类型 餐厅 休闲 服务 场所 动
 - Object Tags : 语言 政治 交通 行政区划 文学 设计 社会 官制 拓扑学 物理 历史 官职 技术 文化 机构 电影 电脑 工程 设施
 - Path : N_V-N_V
 - Score : 1.563389655425646E-7
- 北京市海淀区青龙桥医院 2HISHP 位于 北京市 2HISHP
 - Subject Tags : 医院 医疗 制度 医学 场所 机构
 - Object Tags : 地理 文明 政治 设计 社会 国家 物业 首都 历史 文化 城市 北京 政府
 - Path : N_V-N_V
 - Score : 1.522002036533876E-7
- 北京市丰台区晓园中医医院 2HISHP 位于 北京市 2HISHP
 - Subject Tags : 医院 医疗 制度 医学 场所 机构
 - Object Tags : 地理 文明 政治 设计 社会 国家 物业 首都 历史 文化 城市 北京 政府
 - Path : N_V-N_V
 - Score : 1.522002036533876E-7

Domain

Range

Selected Examples

Fig. 4 The lookup service, the top restrictions and the selected examples of the relation *LocatedIn*. (a) Lookup Service, (b) Path Patterns, (c) Domain, (d) Range, (e) Selected Examples

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: a nucleus for a web of open data. In: Proceedings of the 6th International Semantic Web Conference, pp. 722–735, ISWC'07/ASWC'07. Springer, Berlin, Heidelberg (2007)
2. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Vol. 1, pp. 86–90. Association for Computational Linguistics, Montreal, Quebec (1998)
3. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 2670–2676. Hyderabad, India, January 6–12, 2007
4. Banko, M., Etzioni, O.: The tradeoffs between open and traditional relation extraction. In: Proceedings of ACL-08: HLT, pp. 28–36. Association for Computational Linguistics, Columbus, Ohio (2008)
5. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010. AAAI Press, Atlanta, Georgia, July 11–15, 2010
6. Carlson, A., Betteridge, J., Wang, R.C., Hruschka, Jr., E.R., Mitchell, T.M.: Coupled semi-supervised learning for information extraction. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 101–110, WSDM '10. ACM, New York, NY (2010)
7. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. B (Methodological)* 1–38 (1977)
8. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. *Comm. ACM* **51**, 68–74 (2008)
9. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam: Open information extraction: The second generation. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 3–10. IJCAI/AAAI, Barcelona, Catalonia, Spain, July 16–22, 2011
10. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545. Association for Computational Linguistics, Edinburgh, Scotland (2011)
11. Fan, J., Ferrucci, D., Gondek, D., Kalyanpur, A.: Prismatic: Inducing knowledge from a large scale lexicalized relation resource. In: Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, pp. 122–127. Association for Computational Linguistics, Los Angeles, California (2010)
12. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
13. Kubler, S., McDonald, R., Nivre, J.: Dependency Parsing. Morgan & Claypool (2009)
14. Mitchell, T.M., Betteridge, J., Carlson, A., Hruschka, E., Wang, R.: Populating the semantic web by macro-reading internet text. In: Proceedings of the 8th International Semantic Web Conference, pp. 998–1002, ISWC '09. Springer, Berlin, Heidelberg (2009)
15. Mohamed, T., Hruschka, E., Mitchell, T.: Discovering relations between noun categories. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp. 1447–1455. Association for Computational Linguistics, Edinburgh, Scotland (2011)
16. Niu, X., Sun, X., Wang, H., Rong, S., Qi, G., Yu, Y.: Zhishi.me: weaving chinese linking open data. In: Proceedings of the 10th International Conference on The Semantic Web - Volume Part II, pp. 205–220, ISWC'11. Springer, Berlin, Heidelberg (2011)
17. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.* **31**(1), 71–106 (2005)

18. Poon, H., Christensen, J., Domingos, P., Etzioni, O., Hoffmann, R., Kiddon, C., Lin, T., Ling, X., Mausam, Ritter, A., Schoenmackers, S., Soderland, S., Weld, D., Wu, F., Zhang, C.: Machine reading at the university of washington. In: Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, pp. 87–95. Association for Computational Linguistics, Los Angeles, California (2010)
19. Schuler, K.K.: Verbnet: a broad-coverage, comprehensive verb lexicon. Ph.D. thesis, Philadelphia, PA, USA (2005), aAI3179808
20. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706, WWW '07. ACM, New York, NY (2007)
21. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 118–127. Association for Computational Linguistics, Uppsala (2010)
22. Zhu, J., Nie, Z., Liu, X., Zhang, B., Wen, J.R.: Statsnowball: a statistical approach to extracting entity relationships. In: Proceedings of the 18th International Conference on World Wide Web, pp. 101–110, WWW '09. ACM, New York, NY (2009)

Co-mention and Context-Based Entity Linking

Qian Zheng, Juanzi Li, Zhichun Wang, and Lei Hou

Abstract Recently, online news has become one of the most important resources from which people get useful information. Linking named entities in news articles to existing knowledge bases is a critical task to facilitate readers to understand the news well. In this paper, we propose an approach for linking entities in Chinese news articles to Chinese knowledge bases. Our approach first recognizes three types of named entities (i.e., person, location, and organization) and then uses a disambiguation method to link entities occurring in news articles to entities in knowledge bases. In the disambiguation process, co-mentioned entities are used as features to compute the context similarities between entities in news and entities in knowledge bases; the disambiguation results are decided by a threshold-filtering method on the context similarities. Experiments on linking entities in Sina news to Hudong knowledge base validate the effectiveness of our approach; it achieves 84.39%, 84.02%, and 86.16% F1-scores in the task of linking person entities, location entities, and organization entities, respectively.

Q. Zheng (✉) • J. Li • L. Hou

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 People's Republic of China

e-mail: zy3205791@gmail.com; ljz@keg.cs.tsinghua.edu.cn; houlei@keg.cs.tsinghua.edu.cn

Z. Wang

College of Information Science and Technology, Beijing Normal University, Beijing, 100875 People's Republic of China

e-mail: wzc@keg.cs.tsinghua.edu.cn

1 Introduction

Recently, online knowledge bases have become the most universal resources for people to find their integrative information on the Web. In recent years, several knowledge bases appear, such as Wikipedia,¹ Hudong,² and Baidu,³ these “Wiki-like” knowledge bases allow users from all over the world to edit their Web pages to improve the coverage and integrity of the information and have accumulated large-scale entities in various domains. Meanwhile, with the fast growth of the Internet, millions of news data are posted on the Web. According to a report from CNNIC⁴ in 2011, there are 353 million people who read news on the Internet in China. Named entities, such as persons, organizations, and places, carry important information in news stories. Linking entities in news to entities in knowledge bases will provide background knowledge of news to readers and thus improve the users’ reading experience. This entity linking task is challenging due to name variations and entity ambiguity. In reality, an entity may have multiple surface forms. For example, the entity of “Tsinghua University” has its abbreviation “Tsinghua”; on the contrary, one entity mention may also refer to several different real world entities. For instance, the entity mention of “Michael Jordan” can refer to the famous basketball player, the computer science professor, or some other persons.

The problem of linking entities in unstructured text to knowledge bases is called entity linking, which is attracting increasing interests of researchers. There have already been several entity linking approaches proposed [1–10]. These approaches can be generally classified into three broad categories:

- Independent mention disambiguation [11] where each mention is mapped to a knowledge base entity independently. The main idea is to compare the context of the mention with the text metadata associated with the entity in the knowledge base [1–4]. They differ in the features used (e.g., bag of words, Wikipedia categories) and the comparison technique (cosine similarity, classifier). The main drawback is that they do not consider the interdependence between disambiguation decisions.
- Inter-document collective mention disambiguation observes that a document typically refers to topically coherent entities. They consider interdependence between disambiguation decisions within a document [4, 6–8]. They perform collective assignment of candidate mention in a document to entities and select an assignment that not only maximizes the mention context-to-entity similarity but also the coherence among the assigned entities [6, 7]. Coherence between a pair of entities is computed using the knowledge base, for example, based on the number of common Wikipedia pages that link to Wiki pages of these two entities

¹<http://www.wikipedia.org>

²<http://www.hudong.com>

³<http://baike.baidu.com>

⁴<http://zxxd.cnnic.cn/>

[6]. While some approaches model the interdependence as sum of their pair-wise dependencies [4, 6], more recent techniques model the global interdependence [7, 8].

- Semantic knowledge-based method [12] links named entities in text with a knowledge base unifying Wikipedia and WordNet, by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledge base.

Approaches in the former two groups mostly focus on linking entities to knowledge bases like DBpedia [13, 14] and YAGO [15, 16]. Their main idea is to compare the “context” (i.e., surrounding words) of the entity mentions in a document with entities’ features in a knowledge base and select the entity having the largest similarity with a certain entity mention as the result. These approaches are not effective when dealing with large-scale entities.

Approaches in the third group perform more effectively when linking English named entities. However, they did not evaluate the effectivity on “Chinese Wiki” (i.e., Hudong, Baidu) in their experiments. It should be noted that Hudong’s classification tree is inconsistent in some entities [17]; there may be a negative effect on the taxonomy feature.

In this paper, we propose an approach for linking entities in news documents to Chinese knowledge bases. Main challenges of task include:

1. Chinese Entity is hard to extract. Taking the entity “丹东(Dan Dong)” as an example, there is no “丹东” in dictionary, but there is a place called “丹东” in Liaoning province.
2. Different entities may use the same name; there are 12 searching results of “李健” in Hudong, plus the frequency of their emergence is near, not like “Michael Jordan.”
3. Same entities may use different names (i.e., 加州理工大学 vs. 加州理工学院; both mean California Institute of Technology) that may cause the redundant result.

In order to solve the above challenges, we propose a simple and effective approach for entity linking between news data and knowledge bases. In our approach, entities in the same news document are regarded as context features and used to calculate the similarity between entries and entities.

Our contributions include:

- We make full use of the content of the knowledge base result page and propose a naive but effective approach to link with news. We define some features to do the ambiguity problem in entity linking.
- We use Infobox information to fix the linking result and do the assessment of elevation.
- At the end of June this year, there are 298,996 news in Sina.⁵ And we have linked them with Hudong.

⁵<http://news.sina.com.cn>

- We have 84.39% in person, 84.02% in location, and 86.16% in organization of F-score in linking entity in Sina to Hudong.

The rest of this paper is organized as follows: Sect. 2 defines some basic entity linking related concepts; Sect. 3 gives detail explanation of the proposed entity linking algorithm; Sect. 4 describes the result of the experiments; in the end we give a conclusion and state possible future work in Sect. 5.

2 Task Definition

We first define some related concepts of news data and knowledge bases and then formulate the problem of entity linking.

Definition 1. A news article is a set of informal specification of text that refers to a brief with certain social values and the fact that recently occurred. A news article can be described as $D = \{E, W\}$, where E is the entity set in D and W is the other words occurring in D . In this paper, we use E_{per} for person entity, E_{loc} for location entity, and E_{org} for organization entity. In the following text, “per” is short for person, “loc” is short for location, and “org” is short for organization.

Definition 2. We formally represent a knowledge base link article as $A_i \in K$; for each $A_i \in K$ we have $A_i = \{t, I\}$, where t can be described as $t = \{E', W'\}$, E' is the entity set in A , and W' is the other words occurring in D' . I is the Infobox associated with attribute-value pair, and then we have $I_j = \{p_i\}_{i=1}^n \in A_j$, where $p_i = < a_i, v_i >$ is the attribute-value pair in $I_j \in A_j$. According to our definition, the Infobox like Fig.1 can be formally defined as: $I_{ThomasHanks} = \{<\text{中文名}, \text{汤姆汉克斯}>, \dots, <\text{爱好}, \text{棒球、钓鱼}>, <\text{其他信息}, \dots, >\}$. t is categorically not null and I is optional in A_i . For each t in A , there is $W = \{w_i\}_{i=1}^n \in t$. For most person, location, and organization pages in Hudong, the Infobox is quite complete and comprehensive; then that is the reason we choose these three types of entities to do the experiments in this paper.

Definition 3. Consider a word w_i ; there are two possibilities that this w_i may represent an entity, like “汤姆汉克斯 (means Thomas Hanks).” In another case, that w_i may not be an entity. Then we have $E = \{E_i\}_{i=1}^n \subseteq W$.

Definition 4. A matching news entity with Hudong (MNEH) problem is the process of matching the entity in news $E = \{e_i\}_{i=1}^n \subseteq W \in D$; e_i denote an entity name mention. For each $e_i \in E$ is characterized by its name, and it is always surrounded by contexts in the document D which contains it. A matching result link is the annotation of entity which guides people to knowledge base page. We have the result set $M = \{m_i\}_{i=1}^n$; in this paper, m_i may denote the result for person, location, and organization entities. For example, $m = \text{www.hudong.com/wiki/汤姆汉克斯} \in M$

中文名:	汤姆·汉克斯	英文名:	Thomas Hanks
籍贯:	美国加利福尼亚州康克特	出生地:	美国加利福尼亚
性别:	男	国籍:	美国
出生年月:	1956年7月9日	职业:	演艺演员 艺术制片人 艺术导演
毕业院校:	加州查伯得学院、	政党:	无
成就:	1995年 赢得了奥斯卡最佳男主角的金像奖杯 1994 获奖 最佳男演员 (1994年)《阿甘正传》 2010年10月12日，汤姆·汉克斯将以制片人的身份获得美国制片人协会的电视制作成就奖	重要事件:	《费城故事》(Philadelphia) 把汤姆·汉克斯的演艺生涯推上了第一个高峰 2000年 获奖 最佳男演员《荒岛余生》、洛杉矶影评人协会奖 LAFCA 英国影视艺术学院颁奖荣誉“史丹利·寇比力克杰出电影奖”
代表作品:	《阿甘正传》、《费城故事》、《西雅图不夜城》	身高:	180厘米

Fig. 1 Infobox of “Thomas Hanks (汤姆汉克斯)”

3 Our Approach

3.1 Hypotheses

Our approach works based on two hypotheses: the context similarity hypothesis and the co-mention hypothesis. The first hypothesis is regarding the similarity between linking page and the news content. The second hypothesis is about the entity co-mention in same linking page either in the same news. Both hypotheses are described in details as follows.

Hypothesis 1 (Context Similarity). The similarity between the linking page of an entity and the content of a news is proportional to the probability that the linking page is the correct link to this named entity mentioned in the news. In other words, if an entity is mentioned in news but there is totally no overlap or interrelation between the news content and this entity, then we can assume that this entity is not the one mentioned in that news.

Hypothesis 2 (Co-mention). If multiple entity queries in the same news content are mentioned in the same entity linking page, the chance for them to refer to the entities in the target domain is higher.

3.2 Matching Details

Our approach links three types of named entities (i.e., person, location, organization) in news articles to Chinese knowledge bases Hudong. There are two steps during the disambiguation: candidate selection and entity disambiguation. Candidate selection is to get a list of candidate linking entities from Hudong knowledge base; entity disambiguation chooses the right entity from the candidates.

3.2.1 Search Situations in Hudong

There are three types of structures of searching result in Hudong knowledge base:

Entity Pages

Entity page in Hudong knowledge base describes a single entity and may contain the information about this entity. Generally, the title of each page is the entity name, for example, title “中科院” is the organization name of Chinese Academy of Sciences, and if “中国科学院” is searched, Hudong will return the same page as “中科院.”

Polysemy Pages

Each polysemy page in Hudong knowledge base contains a list of entities, these entities could be given the same name, and the polysemy page is created to separate them. For example, the polysemy page of “李健” lists twelve entities having the same name of “李健,” including the famous singer and the football player.

Nil Pages

The nil page is the result page that Hudong cannot find the result, or the searching entity name does not exist in Hudong; this page often contains hyperlinks which link to the relative entity pages, but in Chinese that result is not that good, for example, if the searching entity name is “依内里奥,” the transliteration of the person who created “Università di Bologna”; these kinds of entity infrequently exist in Chinese KB; thus, we skip these pages.

Table 1 Co-mention weight
(e_i, e'_i) in news documents

	Entity type e_i	Entity type e'_i	Weight (e_i, e'_i)
PER	PER		1
	LOC		1
	ORG		2
LOC	PER		0
	LOC		2
	ORG		0
ORG	PER		0
	LOC		0
	ORG		1

3.2.2 Entity Disambiguation

Co-mention Weight. To make the co-mention hypotheses work, we determined the weight between different types of entities in same news as Table 1:

Table 1 shows the features on the Hypothesis 2 (co-mention) to calculate the similarity. We set the weight by the relevance it should be in the news, for example, person entity e_1 is mentioned in a news D, and then the probability of an organization occurring in a knowledge base page article A is larger than person entity e_2 mentioned in the same news D, because that organization he (or she) has joined is more relevant than a person co-mentioned in the same news with him (or her).

3.2.3 Context Similarity

Here we describe how to compute the context similarity between the linking page and the news content. For each entity e_{ij} from news articles, we get the linking page content d_q straightly. Thus, if the similarity $\langle e_{ij}, d_q \rangle$ is larger than a threshold δ , then put e_{ij} in candidate list, and if not, then drop it. This similarity is computed based on cosine similarity, which is defined as follows:

Let $v_E = \langle w_{e_1}, w_{e_2}, \dots, w_{e_t} \rangle$ and $v_{E'} = \langle w'_{e'_1}, w'_{e'_2}, \dots, w'_{e'_t} \rangle$ denote feature vectors of the entity set E and entity set E' . Their similarity is computed as

$$\text{Sim}(v_E, v_{E'}) = \frac{\sum_{i=1}^t w_{e_i} \cdot w'_{e'_i}}{\sqrt{\sum_{i=1}^t w_{e_i}^2} \sqrt{\sum_{j=1}^t w'_{e'_j}^2}} \quad (1)$$

For example, if the vector length $t = 2$ and the type of e_1 is per, type of e_2 is loc, type of e'_1 is per, and type of e'_2 is per, then we have $\text{Sim}(v_E, v'_{E'}) = 0.5$.

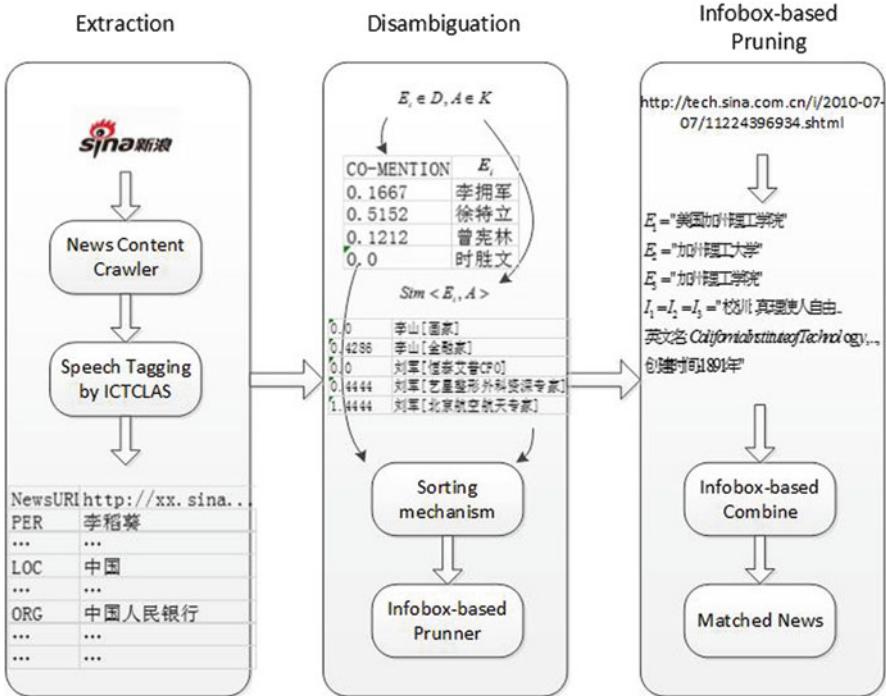


Fig. 2 Framework of matching system

3.3 Framework Overview

According to Fig. 2, the complete entity matching framework contains three main components: extraction, disambiguation, and Infobox-based pruning.

3.4 Extraction Process

We use the ICTCLAS [18] which is a popular Chinese entity extractor to achieve the goal that to recognize the entity, given a Chinese news article, ICTCLAS works based on the hidden Markov model.

We already know that there are some incorrect results like “艾观察” is tagged as a person name in a news, and during the syntactic analysis, “艾观察” may be a subject or object word in a sentence. Then the emergence of this kind of result in news article is not that inscrutable. Consequently, the news is not long enough that can make the tagging result to get worse, but in order to identify the person, location, and organization entity in news effectively, we still use ICTCLAS, and specifically, we perform error eliminating work after tagging. We use a basic way to actualize

the error eliminating work, searching the entry in Hudong, and Hudong will return a nil result page when this entry does not exist in Hudong.

At the same time, apparently, some of the entity name cannot be found in Hudong by searching the extracted name directly, but in nature, they exist as some other name in Hudong, for example, “汤姆·汉克斯,” the entry name in Hudong is “汤姆·汉克斯” or “汤姆·汉克斯。” Consequently, we do the annotated character cleaning work on the tagged result; obviously, this is much more practical to find the existing page in Hudong than before.

3.5 Disambiguation Process

The situations in disambiguation process are mentioned before; we do the disambiguation on the entity pages and polysemy pages; for scenario “entity pages,” we need to know whether the entity in entity page is the same one that appeared in the news or not, and for scenario “polysemy pages,” for each entities in every news, we rank the content similarity and choose the maximal; this maximal content similarity refers to the most likely link result in disambiguation but not always justness; thereupon, by way of making the most correct of link result, we solve these “polysemy page” problems wielding the “entity page” method for each entities in the list after all.

3.6 Infobox-Based Pruning Process

With respect to the organization result, we give an add-on method to deal with the organization process to make it different from the person and location process. For instance, there are three entities “美国加州理工学院,” “加州理工学院”, and “加州理工大学” in Hudong which all refer to “California Institute of Technology”; we combine these three entities into one; this combination conduces to gain the escalate of result, this avenue harnesses the Infobox result in disambiguating result, and for the higher similarity, we add them to a candidate list. We found 50 instances of these situations during the experiment, and there are 420 entities in the text data.

Given Hudong link result page article A and A' , we have $I = \{p_j\}_{j=1}^m \in A$ and $I' = \{p'_j\}_{j=1}^m \in A'$; Algorithm 1 shows the procedure of calculating the Infobox relatedness between I and I' .

Algorithm 1. Relatedness Between Infoboxes

Input:
The Infobox pair set, $I = \{p_i\}_{i=1}^n \in A$;
The Infobox pair set, $I' = \{p'_j\}_{j=1}^m \in A'$;

Output:
A float, R ;
1: $R = 0$;
2: $Size = \text{Max } m, n$;
3: **for** each p_i **do**
4: **for** each p'_j **do**
5: **if** $p_i = \{p'_j\}$ **then**
6: $R++$;
7: **end if**
8: **end for**
9: **end for**
10: $R = R/size$;
11: **return** R ;

Consider the Infobox content of “加州理工学院”⁶ and “加州理工大学”⁷; we have 100% Infobox relatedness between them by using Algorithm 1. It is prudent to use every pair of Infobox attribute-value pair to calculate that; during the experiments, using all these pairs to compute the similarity between two Infoboxes is unnecessary, because in Hudong, there is no analogous Infobox but equal, and there is no link result that has been pruned when their Infoboxes are unequal. In substance, the reason of that is this kind of “Chinese Wiki” frequently considers the conformity using thesaurus hence combining their linking page by artificial. Nevertheless, not all knowledge bases use an artificial way to combine synonyms, so we keep that method in order to make it useful during the future experiments.

4 Experiment

4.1 Evaluation Measures

Given that $M = \{m_i\}_{i=1}^n$ is the matching result of $E = \{e_i\}_{i=1}^n$ and $M' = \{m'_i\}_{i=1}^{n'}$ is the correct link set of that ought to be, we have

Precision: the number of the matching result samples that is equal to the exact link.

⁶<http://www.hudong.com/wiki/加州理工学院>

⁷<http://www.hudong.com/wiki/加州理工大学>

$$\text{precision}(i) = \begin{cases} 1, & m_i = m_i' \\ 0, & m_i \neq m_i' \end{cases} \quad (2)$$

Recall: the number of the existing entities in news also matched by the matching system.

$$\text{recall}(i) = \begin{cases} 1, & |e_i \cap e_i'| \neq \emptyset \\ 0, & |e_i \cap e_i'| = \emptyset \end{cases} \quad (3)$$

For the whole testing set, we use the average of these two measures to calculate the F1-score.

$$\text{precision} = \text{avg}(\text{precision}(i)); \quad \text{recall} = \text{avg}(\text{recall}(i)) \quad (4)$$

Then, we have

$$\text{F1-Score} = \frac{2 \times \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

4.2 Experiment Result

The reason that some of the entities in news articles cannot be extracted is varied; one reason is that sometimes, Chinese is used to express the English person name, and there are some special symbols between the family name and the giving name, like “-”; nevertheless, sometimes there is not; then obviously, the tagging tool cannot make the same result when the entity is not changed but it occurs in different places. Another reason is that some of the phrases in the sentence begin with a person’s first name; as an example, “施于人” is originally the last three words of the old saying “己所不欲勿施于人,” but the systematic analysis is an artist in “中央美术学院实用美术系.”

For the location entity in news, notice that in the experimental results we’ve got a great accuracy; one of the reasons for this phenomenon is that the segmentation tool contains a very comprehensive dictionary, so that the assessment of the accuracy requires more demand. In the process of the statistics for the accuracy, we just list the news which just have one location result; the linking mistake can only be found when the count of the result is small enough, in the sense that the system uses the co-mention mechanism to do the disambiguation, and if the count is larger or equal to 2, the accuracy will be close to 100%.

Some of the Hudong links exist in different sorts. Specifically, “玉树” is the place that suffered earthquake in “青海省” last year; “玉树” does not exist in Hudong as location but another entity named “玉树” which is a subject of tree, because “树” in Chinese amounts to “tree.”

Table 2 Precision of the test data

	Baseline	+Disambiguation	+Infobox pruning
PER	0.7186	0.9134(+19.5%)	0.9175(+0.41%)
LOC	0.8951	0.8986(+0.35%)	0.9042(+0.56%)
ORG	0.8017	0.8017(+0.00%)	0.9718(+17.01%)

Table 3 Recall of the test data

	Baseline	+Disambiguation	+Infobox pruning
PER	0.7998	0.7789(-2.09%)	0.7812(0.23%)
LOC	0.7708	0.7792(+0.84%)	0.7847(+0.55%)
ORG	0.6371	0.6371(+0.00%)	0.7739(+13.68%)

Table 4 F1-score of the test data

	Baseline	+Disambiguation	+Infobox pruning
PER	0.7570	0.8407(+8.37%)	0.8439(+0.32%)
LOC	0.8283	0.8346(+0.63%)	0.8402(+0.56%)
ORG	0.7100	0.7100(+0.00%)	0.8616(+15.16%)

We analyze the effectiveness during the three steps of our system, Table 2 shows the precision of the test data, Table 3 shows the recall of the test data, and Table 4 shows the F1-score of the test data. Compared with location and organization entities, person entities got a great improvement during the disambiguation step, and for location and organization, Infobox pruning step got a significant influence; this is because the vastest majority of disambiguation problem is person entity. However, in entity extraction step, some location and organization entity cannot be extracted as same name in one news article; then it causes us not to always find the right result, and then we use Infobox similarity to calculate to combine them into one link, and that also obtained a substantial upgrade in location and organization entity.

5 Conclusion and Future Work

In this paper we proposed an efficient algorithm to link news entities and knowledge-based entities. This algorithm can be used to provide background knowledge for news entities. At the same time, adding new source of information to the knowledge base is also useful.

As our future work, we will concentrate on improving the quality of the entity linking and use some other ways of entity extraction. We also want to use other knowledge base like Baidu and Wikipedia, but the Chinese in Wikipedia is not enough that we should do some alignment as preliminary problem.

Acknowledgements The work is supported by the Natural Science Foundation of China (No. 61035004, No. 60973102), 863 High Technology Program (2011AA01A207), European Union 7th Framework Project FP7-288342, and THU-NUS NExT Co-Lab and the project cooperated with Chongqing research institute of science and technology.

References

1. Bunescu, R.C., Pasca, M.: Using encyclopedic knowledge for named entity disambiguation. In: EACL'06 pp. 9–16 (2006)
2. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R.V., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In: WWW'03, pp. 178–186 (2003)
3. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: CIKM'07, pp. 233–242 (2007)
4. Milne, D.N., Witten, I.H.: Learning to link with wikipedia. In: CIKM'08, pp. 509–518 (2008)
5. Cucerzan, S.: Large-scale named entity disambiguation based on wikipedia data. In: EMNLP-CoNLL'07, pp. 708–716 (2007)
6. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of wikipedia entities in web text. In: KDD'09, pp. 457–466 (2009)
7. Hoffart, J., Yosef, M.A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: EMNLP'11, pp. 782–792 (2011)
8. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: a graph-based method. In: SIGIR'11, pp. 765–774 (2011)
9. Han, X., Zhao, J.: Named entity disambiguation by leveraging wikipedia semantic knowledge. In: CIKM'09, pp. 215–224 (2009)
10. Chakaravarthy, V.T., Gupta, H., Roy, P., Mohania, M.K.: Efficiently linking text documents with relevant structured information. In: VLDB'06, pp. 667–678 (2006)
11. Wang, C., Chakrabarti, K., Cheng, T., Chaudhuri, S.: Targeted disambiguation of ad-hoc, homogeneous sets of named entities. In: WWW'12, pp. 719–728 (2012)
12. Shen, W., Wang, J., Luo, P., Wang, M.: Linden: linking named entities with knowledge base via semantic knowledge. In: WWW'12, pp. 449–458 (2012)
13. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia – a crystallization point for the web of data. JWS' 2009, 154–165 (2009)
14. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: a nucleus for a web of open data. In: ISWC/ASWC'07, pp. 722–735 (2007)
15. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a large ontology from Wikipedia and wordnet. JWS' 2008, 203–217 (2008)
16. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW'07, pp. 697–706 (2007)
17. Wang, Z., Wang, Z., Li, J., Pan, J.Z.: Building a large scale knowledge base from Chinese wiki encyclopedia. In: JIST'11, pp. 80–95 (2011)
18. Zhang, H., Liu, Q., Zhao, J.: Chinese name entity recognition using role model. Comput. Linguist. Chin. Lang. Process., 29–602 (2003)

An Approach of Text Sentiment Analysis for Public Opinion Monitoring System

Min Zeng, Yujiu Yang, and Wenhuan Liu

Abstract With the thriving of microblog, a huge amount of people get involved in online life. This leads the government to intensify supervision on online remarks, and opinion polarity is what they care most. But microblog opinions contain several specialties from hotel remarks; they are format-free, short, and most express only one polarity. In this paper, we target on finding an appropriate polarity recognition method for public opinion supervision system. In our method, we explore new feature extraction rules which extract phizs, emotional nouns, verbs, adjectives, and bigrams as representative features. Then, we apply SVM to classify these online opinions into positive and negative class. Based on the crawled real-world datasets, our method can respectively achieve an accuracy of 80.1% and 87.4% for microblog reviews and traditional hotel remarks, so the proposed method is appropriate and effective for public opinion supervision system.

1 Introduction

The growth and popularity of opinion-rich websites, especially microblog and SNS, arouse new challenge and the need to extract opinionated information from these sites, for both market department of enterprise and government. Through many sites, people can express their personal thoughts free online, and there are 400 million online users in China now; their opinions and emotions could easily affect others through the web; this forces the government to put more attention to network supervision through public opinion monitoring system. Among all the attributes of public opinions, the sentiment trend is the most important one. But previous sentiment analysis works mainly focused on traditional long texts other than microblog opinions. We find microblog texts distinguish in many aspects, for

M. Zeng (✉) • Y. Yang • W. Liu

Graduate School at Shenzhen, Tsinghua University, 518055 Shenzhen, Guangdong, China

e-mail: jammyzm@gmail.com; yang.yujiu@sz.tsinghua.edu.cn; lwh@sz.tsinghua.edu.cn

example, they contain many web words, phizs, so we think it will help a lot in sentiment analysis if we apply suitable feature extraction strategy to microblog texts. In this paper, we explore sentiment analysis by using rules to derive polarity patterns from public opinions. As the online remarks are short and flexible in format, and the phrase format is stable and contains more grammatical information, we combine both words and bigrams as feature. Also we pay attention to the phiz in reviews, as many netizens use phizs frequently. With these suitable feature extraction strategies, we enhance the microblog sentiment classification performance obviously, and we apply different classification algorithms to corpus, and SVM is proven to achieve the best result. Also, we find different classification algorithms don't affect the result so much as the variances of feature extraction.

The rest of the paper is organized as follows: we will discuss some related works in Sect. 2, and Sect. 3 specifically explains the method applied in the experiment, Sect. 4 depicts the experiment, and conclusion is given in Sect. 5.

2 Related Works

For decades of researches in text sentiment analysis, there exist two kinds of methods, namely, machine learning method and semantic method.

In the development of machine learning method, feature extraction plays a key role in enhancing the classification performance. Earlier studies in sentiment analysis used semantic dependencies between words to predict sentiments. For example, Hatzivassiloglou and McKeown [1] had explored different forms of adjectives and their usefulness as subjectivity clues. Then, Wiebe and Wilson's [2] later work moved away from simple unigram to emotional phrase and consider grammatical meaning of reviews; this helps to enhance the accuracy. They observed that word patterns along with its modifiers could reflect intensity of phrase sentiments, for example, "indiscriminate massacre," the adjective modifier "indiscriminate" gives the bigram an overall negative emotion. By using typed dependency, the syntactic structure of the sentence or phrase will be taken into consideration during the sentiment research. Along with this thinking, the researcher did a lot of improvement in later works. Thet [4] proposed heuristic rules to compute sentiment value of sentence by using grammatical dependencies. A typical rule is {positive adjective + positive noun} -> {positive output} (e.g., "pretty cat"). Polarity pattern rules created by heuristic approaches can't cover all extraction rules, and some of the rule is less efficient which may lead to more extraction and training time.

At the same time, researchers explore semantic method, and they do a lot of work to optimize this approach. Liu Qun [3] incorporated HowNet to calculate the semantic orientation value of feature words then determine the polarity of review by the sum of values. Semantic method achieves good results in ordinary online remarks, but when it comes to microblog and BBS reviews, new features appear, and they are different from current corpus. For example, netizens use plenty of network words and phizs, for this kind of text, semantic methods like HowNet fail to achieve

high precision. For machine learning methods, the feature extraction strategy could be optimized, while for semantic methods, HowNet is not easy to extend; it didn't work well for microblog and BBS remarks. So in this study, the most important idea is to find an appropriate sentiment analysis method for different kinds of online reviews.

3 Methodology

Earlier studies in sentiment analysis tend to use complicated rule, like boost method to find subjective expression first, or use linguistic methods like case grammar, to apply to long text, but most online remarks are short and mainly made up by semantic opinion, so we don't need to distinguish subjective part from the text. Also, we carry out experiments and prove semantic method performs worse than machine learning method, especially in dealing with microblog corpus.

This method mainly contains two steps: extraction and classification. The whole framework can be depicted as the following flow-process diagram (shown as Fig. 1).

3.1 Feature Extraction and Weight Calculation

The choice of feature plays a key role in deciding precision. As illustrated in the front section, our target is to find a suitable approach to identify emotional trend of online public opinions. Of all sources of public opinions, microblog and BBS occupy biggest share. And these two sources have distinctive features. First, it's short in length; then, emotional polarity is obvious; also, netizens use lots of

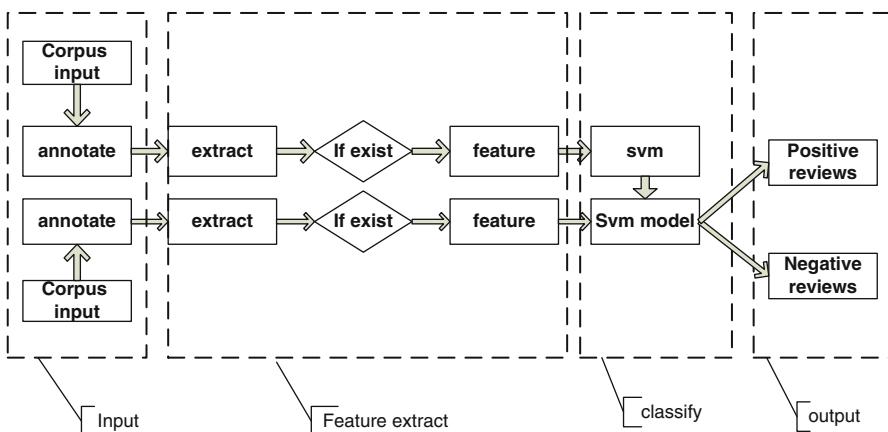


Fig. 1 Text tendency classification system operation flow

Table 1 Extraction example

Object	Original text	Extraction result
Phiz	[Astonish], he didn't make it	[Astonish]
Emotional verb	I fail to complete it	Fail
Emotional noun	We need success for this game	Success
Adjective–adverb	This bottle is big enough	Big enough
Adverb–adjective	He is a quite brilliant person	Quite brilliant

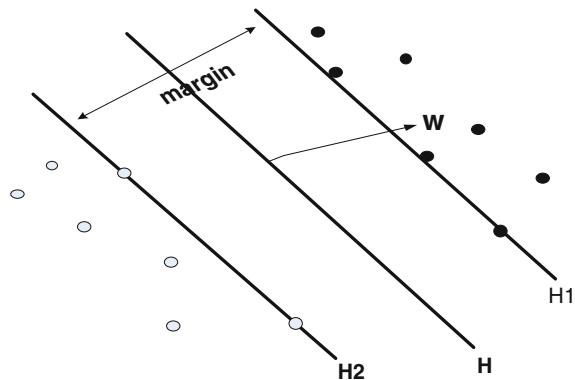
phizs, this could be extracted as important feature for emotional trend identification. According to these features, we set a set of extraction object to select suitable feature as Table 1:

Our extraction strategies are illustrated as follows. **Strategy I**: extract all phizs, namely, text between “[” and “]”. **Strategy II**: we pick the high-frequency used emotional verbs and nouns from emotional words list provided by HowNet; thus, we collect 1,200 emotional verbs and 600 emotional nouns; these cover most of verbs and nouns that contain polarity and also include web words; in this way, we can cover most of emotional expressions. **Strategy III**: as adjective often appears with adverb or negative adverb around, adverb, especially negative adverb usually effects the extent of polarity, for example, “good” and “not good” represent opposite polarity, we should treat them in a different way. So when extracting adjectives, we apply a rule that if a negative adverb appears within a three-word distance around the adjective, we will set the weight of this adjective feature as minus. Also, we extract bigram as adverb–adjective or adjective–adverb, but in order to reduce the sparsity of the feature vector, we don’t use bigram as independent feature but only consider the adverb as an influence to the feature’s weights. In this study, we extract bigrams and choose the center word as feature while modifier as an influence to the feature’s weight. We select 90 high-frequency adverbs and divide them into three classes based on their extent. The high-extent class contains 68 words, and the low-extent class contains 22 words. After comparing several classification results, we set the following rule: if a high level modifier appears around an adjective, we will multiply the feature weight by 1.2, and if a low level modifier appears around an adjective, we will multiply the feature weight by 1.1. If a negative adverb appears around the feature, we will turn the weight into the opposite number. Also, we use TFIDF algorithm to calculate the feature’s weight. The formula is as follows:

$$w_{ij} = \frac{t f_{ij} \times \log \left(\frac{N}{n_i} + 0.01 \right)}{\sqrt{\sum_{k=1}^N \left[t f_{kj} \times \log \left(\frac{N}{n_j} + 0.01 \right) \right]^2}}, \quad (1)$$

where w_{ij} indicates feature i ’s weight for class j , $t f_{ij}$ means feature frequency of i in text j , N represents the total number of training files, and n_i means the number of files that contain feature i .

Fig. 2 Two-dimensional SVM classification



3.2 Opinion Classification

After choosing the right characteristics to express the opinion of the short online text, we need to select an appropriate classifier to distinguish between different points. As we know, SVM (short for support vector machine) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Therefore, the optimal separating hyperplane maximizes the margin of the training data.

The success of SVM for text classification encourages us to deal with the opinion polarity issue via the same classification strategy. In fact, we prove SVM achieves the highest accuracy in text emotional polarity classification scene based on our experimental results.

To elaborate the procedure of classification, here, we simply describe the SVM theory. In two-dimensional situation, we can clearly see the dividing function of SVM in Fig. 2, where the black spots depict the negative opinion text, and the small circles represent the positive ones.

All the points in the figure above mean the samples, and we can see the margin maximizes when we choose hyperplane $H1$ and $H2$ as dividing plane. Thus, we define the dividing plane $H1$ and $H2$ as optimal classification hyperplanes; the corresponding vectors of the samples on optimal classification hyperplanes are called support vectors. And we can define the hyperplane H as a linear function like the following formula:

$$g(x) = wx + b. \quad (2)$$

where x is the input sample, and $g(x)$ is the dividing interface of two class. We set zero as the threshold and suppose x_i is an input sample; if $g(x_i) > 0$, we assign x_i

to class one, and if $g(x_i) < 0$, we assign x_i to the other class. We can calculate the margin in the following formula:

$$\text{margin} = 2 / \|w\|. \quad (3)$$

Thus, we can get the optimal classification hyperplane by minimizing $\|w\|$.

We can't always make a linear classification in two-dimensional space, so we need to map all samples to a higher dimensional space by using a function. In SVM, we give this mapping function a name: kernel function.

So, we can summarize our proposed framework as follows:

Input: online short texts collected from social media

Output: every online text opinion label (positive or negative)

Step 1: represent each input text in required format for SVM classification. It's depicted as follows: Vector $v = (\text{class}; \text{feature 1, weight}; \dots; \text{feature } i, \text{weight})$.

Step 2: training and getting the classification model. We need to choose several training parameter to get our model. In this experiment, we select RBF model, depicted as the following formula:

$$k(u, v) = e^{-\gamma|u-v|^2} \quad (4)$$

and set penalty factor and parameter γ in formula (4) to default value 1.

Step 3: input the test text and get the classification result.

4 Experiment

In this section, we will give introduction about our dataset and experiment results.

After observing plenty of online remarks concerned about some latest social events, we found that online remarks are short and most remarks are less than 30 words. And most are expressed in flexible format, some of them even only made up by phizs or punctuations. Also they tend to contain abundant network words. This is different from hotel and movie remarks. So if we want to find an appropriate method for public opinion supervision system, we need to create new corpora that are close to remarks from microblog and BBS.

4.1 Dataset

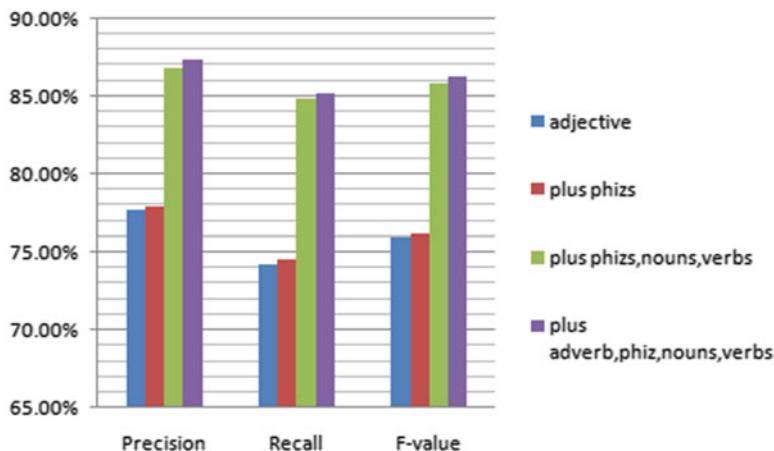
We decide to build two corpora, the first is made up by hotel remarks which include 1,500 positive reviews and 1,500 negative reviews, and the second corpus contains 2,000 remarks about QQ Circle download from Sina microblog. Thus, our corpora include two popular styles of online remarks: the phraseology of hotel remarks are in written form, while reviews from microblog are more similar to verbal expression. In the experiments, we choose 2/3 corpus as training set and the rest as testing set (Table 2).

Table 2 Comparison of two corpora

Name	Text number	Positive text	Negative text	Average text length
QQ circle	2,000	1,000	1,000	<25
Hotel remarks	3,000	1,500	1,500	>60

Table 3 Results of four feature extraction strategies on hotel remarks corpus

	Precision	Recall	F-value
Adjective	77.70%	74.24%	75.93%
Plus phizs	77.90%	74.52%	76.17%
Plus phizs, nouns, verbs	86.90%	84.89%	85.88%
Plus adverb, phiz, nouns, verbs	87.40%	85.21%	86.29%

**Fig. 3** Results of four feature extraction strategies on hotel remarks corpus

After collecting the corpus, we need to run an annotator to segment words and POS tag every word of the corpus. We use ICTCLAS developed by Chinese Academy of Sciences to do this job. ICTCLAS is easy to extend its dictionary, and it achieves a segment speed of 996KB/s with an accuracy of 98%.

4.2 Feature Extraction

In order to confirm the effect of our feature extraction strategy, we conduct four experiments. I: only extract adjectives; II: also extract phizs; III: plus extract emotional nouns and verbs; and IV: also consider adverbs around adjectives. IV is the rule illustrated in Table 1. Then we use SVM as classification method. The next table and graph depicts the experiment results of applying four emotional feature selection strategies on hotel remarks corpus (Table 3; Fig. 3).

Table 4 Results of four feature extraction strategies on QQ circle corpus

	Precision	Recall	F-value
Adjective	69.40%	67.88%	68.63%
Plus phizs	75.60%	73.23%	74.40%
Plus phizs, nouns, verbs	79.40%	78.42%	78.91%
Plus adverb, phiz, nouns, verbs	80.10%	79.23%	79.66%

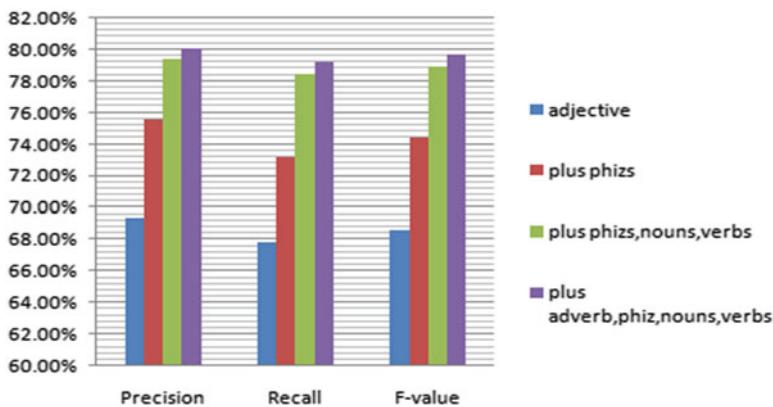


Fig. 4 Results of four feature extraction strategies on QQ circle corpus

From the graph, we can clearly see that emotional nouns and verbs play an important role in emotion trend representation for hotel remarks, but phizs are not so vital, because hotel remarks contain few phizs. Also, our feature extraction method which includes adjective, phiz, emotional noun, and verb achieves best result.

The next table and graph shows the experiment results of applying four emotional feature selection strategies on QQ circle corpus (Table 4; Fig. 4).

When we apply our feature extraction strategy to microblog reviews of QQ circle, we can see a sharp rise in result indicator after we consider phizs, emotional nouns, and verbs.

4.3 Classification Method Comparison

With data and theory given above, we carry out our experiments.

We conduct three experiments: semantic method based on HowNet, machine learning methods naïve Bayes, and the SVM experiment described in the front section.

HowNet. HowNet is made up by plenty of conceptions, which are represented by words. One conception is connected to other conceptions, and a conception contains one or several sememes. HowNet offers a method to calculate the similarity between

Table 5 Experiment result of three classification methods

Corpus	Classification method	Precision	Recall	F-value
QQ circle remarks	HowNet	53.7%	46.64%	49.92%
	Naïve Bayes	78.9%	75.82%	77.33%
	SVM	80.1%	79.23%	79.66%
Hotel remarks	HowNet	80.3%	75.56%	77.86%
	Naïve Bayes	81.7%	78.77%	80.21%
	SVM	87.4%	85.21%	86.29%

two words, so we can use HowNet to calculate emotional similarity of two words. First, we can get two words' similarity by summing all their sememes similarity value. The similarity of sememe p and sememe k is calculated as formula (5):

$$\text{Sem}(p, k) = \frac{\alpha}{\alpha + d}, \quad (5)$$

where d is the distance of p and k in the sememe system of HowNet, and α is an adjustable parameter.

In this experiment, we apply the same feature extracting rule depicted in Table 1 to extract emotional feature from annotated corpus. Then we choose forty pairs of base words, composed of forty positive words and forty negative words; we compare the emotional feature word to every base word; and then we use the following formula to get word M 's emotional trend:

$$\text{Orientation}(M) = \sum_{i=1}^k \text{similarity}(N_i, M) + \sum_{j=1}^k \text{similarity}(P_j, M), \quad (6)$$

where N_i represents negative base words, while P_j represents positive base words. If M 's orientation value is positive, it is a positive word; otherwise, it is negative word. Then, we can get the sum of every feature word's value, if it's positive, the text will be assigned to positive class; otherwise, it will be assigned to negative class.

The result of this experiment is given in Table 5.

This method works well for hotel remarks, because we extract emotional verbs and nouns and adjective phrases as emotional feature, which represent emotional polarity very well. Also we don't need to consider feature dimensional sparsity, but since there are lots of web words used in microblog, which don't exist in HowNet, and HowNet is difficult to extend, it fails to get a good result for microblog corpus.

Naïve Bayes. Naïve Bayes is a widely used method in polarity classification of short text. This method extracts emotional words as features of a text and calculates the text's possibility of belonging to every class, then assigns the text to the class with largest possibility.

Suppose x is one of feature words in a given text, $C = \{C_1, C_2, C_3, \dots, C_n\}$, C is the collection of all classes, $P(C_i/x)$ means the possibility of text containing x belonging to class C_i , $P(C_i)$ is the possibility of C_i ,

$$P(C_i) = \frac{N_i}{N}, \quad (7)$$

where N_i means number of text in class C_i , and N means total number of text from all classes. $P(x/C_i)$ means the proportion of text containing x in class C_i .

$$P\left(\frac{x}{C_i}\right) = \frac{N_x}{N_i}, \quad (8)$$

N_x means number of text containing word x in class C_i . Then, according to the Total Probability formula, we can get the following formula:

$$P\left(\frac{C_i}{x}\right) = \frac{P\left(\frac{x}{C_i}\right) \times P(C_i)}{P(x)}. \quad (9)$$

For each of feature word x_i in the given text, we can get $P(C_i/X_i)$, and since the feature variable's independent identical distribution character of Bayes hypothesis, we can get the possibility of the given text belonging to class C_i ,

$$P\left(\frac{C_i}{X}\right) = \prod_i P\left(\frac{C_i}{X_i}\right). \quad (10)$$

For each of class C_i belonging to C , we can get $P(C_i/X)$, and we assign the text to the class with the max $P(C_i/X)$.

Since short text doesn't contain as much grammatical structure as long text, the emotional words have effective representation in text's polarity, so we use emotional words as feature and Tf*Idf formula to calculate feature weights of emotional words. In this experiment, we use the same feature extracting rule as depicted in Table 1, but we ignore bigrams; we only extract emotional nouns, verbs, and adjectives.

The result is shown in Table 5. From the result, we see naïve Bayes performs good in both kinds of corpus, but QQ circle remarks are more difficult to deal with, this is probably because of the wide appearance of web words and lack of context information in short text.

SVM. We illustrate this method in the methodology part, and experiment result is shown in Table 5.

Our experiment result proves if we use feature selection rule depicted in Table 1 and use SVM as classification algorithm; we achieve best result in emotion trend recognition both for microblog corpus and hotel remark corpus.

4.4 Result and Discussion

From the experiment result, we can discover that reviews from microblog are more difficult to tackle than hotel reviews; it is because some of the microblog texts are too short and less regular in expression. And HowNet does a good job in hotel

reviews but accuracy drops a lot when it comes to microblog reviews. This proves microblog opinions are not suitable to deal in semantic way, because netizens use too many web words, while HowNet's dictionary is difficult to extend. After we take phiz, emotional noun, and verb into consideration, classification accuracy rises obviously, in both types of corpus; this proves that emotional nouns and verbs play an important role in emotion polarity representation. Also, we can see the method in which we also consider adverb in feature selection, and the use of SVM as classification algorithm achieves best performance on the collected datasets.

5 Conclusion

The objective of this research is to find an appropriate approach to fulfill polarity classification task in public opinion supervision area. We conduct the semantic method using HowNet and machine learning method including naïve Bayes and SVM. From the experiment result, we can conclude that SVM performs best when dealing with reviews from microblog and BBS. And feature selection method which extracts phizs, emotional verbs, and nouns helps to enhance the accuracy of SVM effectively. Also extracting adverbs near adjective and considering their influences to feature weights, instead of using them as feature, helps inducing dimension of feature vector while achieving higher precision.

Acknowledgements The work was supported by Guangdong Natural Science Foundation (No.9451805702004046) and the cooperation project in industry, education, and research of Guangdong province and Ministry of Education of P.R.China (No.2010B090400527). In addition, we thank the anonymous reviewers for their careful reading and very valuable comments and suggestions.

References

1. Hatzivassiloglou, V., McKeown, K., R.: Predicting the semantic orientation of adjectives[A]. In: Proceedings of 21st Conference of the American Association for Artificial Intelligence (AAAI-04)[C], pp. 174–181 (1997)
2. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, HLT-EMNLP (2005)
3. Liu, Q., Li, J.-S.: Word similarity computing based on HowNet[c]. In: The 3rd Chinese Lexical Semantics Workshop, Taipei (2002)
4. Thet, T.T., Na, J.-C., Khoo, C.S.G.: Aspect-based sentiment analysis of movie reviews on discussion boards. J. Info. Sci. **36**(6), 823–848 (2010)

Music Recommendation Based on Label Correlation

Hequn Liu, Bo Yuan, and Cheng Li

Abstract The Web is becoming the largest source of digital music, and users often find themselves exposed to a huge collection of items. How to effectively help users explore through massive music items creates a significant challenge that must be properly addressed in the era of E-Commerce. For this purpose, a number of music recommendation systems have been proposed and implemented, which can identify music items that are likely to be appealing to a specific user. This paper presents a hybrid music recommendation system based on the labels associated with each music album, which also explicitly takes into account the correlation among labels. Experimental results on a real-world sales dataset show that our approach can achieve a clear advantage in terms of *precision* and *recall* over traditional methods in which labels are treated as independent keywords.

1 Introduction

Information recommendation has become an important research area since the first paper on collaborative filtering (CF) published in the 1990s [1]. Extensive work has been conducted in both industry and academia on developing new techniques and algorithms for building recommendation systems over the last decades [2]. Recently,

H. Liu • B. Yuan (✉) • C. Li

Intelligent Computing Lab, Division of Informatics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, People's Republic of China

e-mail: qq862688@126.com; yuanb@sz.tsinghua.edu.cn; licheng1492@gmail.com

more attentions are being devoted to this topic due to the success of several practical systems deployed at Amazon¹ (book), Pandora² (music), and MovieLens³ (movie).

As the Web becomes the largest source of digital music, massive music items have been made accessible to users. For example, users can now enjoy and share music with others in online music communities, while more and more online music stores are open for music fans. Since it is an ever increasingly difficult and time-consuming task to search through the diverse music collection, it is almost compulsory for successful music Web sites to implement an efficient music recommendation system to provide good user experience (e.g., Douban⁴ and Xiami⁵).

Most of the existing music recommendation systems are based on the similar idea as other recommendation systems used in E-Commerce, relying on the historical records or behaviors of users. For example, the widely used rating matrix stores the preference of users given to various products, which can be used to discover users with similar behaviors (collaborative filtering). However, the CF method may face significant difficulty when the matrix is sparse, which is often the case in practice. On the other hand, labels or tags can be used to describe the contents or features of products and identify other products with similar properties (content-based recommendation). However, labels are traditionally treated as a set of independent keywords, and their relationships are largely ignored.

In this paper, we propose a hybrid music recommendation system based on labels and, more importantly, the correlation among labels to identify similar music items and users. Each music album is represented by a set of labels assigned to it by various users (music description), and each user is also represented by a set of labels associated with the set of albums that he/she has purchased (music preference). The correlation between two labels is defined according to the frequency that they are attached to the same user and is taken into account when calculating the distance between two sets of labels. By contrast, without this correlation information, two label sets without any common labels are regarded as being completely different. In the experiments, a real-world sales dataset from an online music store is used as the benchmark problem, and a comparative study is conducted against the traditional method without considering the correlation among labels.

The rest part of the paper is organized as follows. Section 2 gives a brief review on music recommendation and the use of label information in recommendation. The architecture of our system and the proposed techniques based on label correlation are detailed in Sect. 3. The experiment results are presented in Sect. 4, and this paper is concluded in Sect. 5 with some discussions for future work.

¹<http://www.amazon.com>

²<http://www.pandora.com>

³<http://www.movielens.umn.edu>

⁴<http://www.douban.com>

⁵<http://www.xiami.com>

2 Related Work

2.1 Music Recommendation

In general, there are two types of music recommendation systems: content-based and collaborative recommendations [2]. Content-based recommendation systems analyze the acoustic features of music and calculate the similarity between two music items in terms of acoustic features and recommend music items that are most similar to those known to be attractive to a specific user. For good recommendation accuracy, a lot of manual efforts are required to extract acoustic features.

The collaborative recommendation systems recommend items that other users in the same user group with similar preferences have purchased or favored [3]. A typical application of this method is by Amazon.com in which 20–30% of the profits are claimed to be due to recommendation. One of the major issues is that, with the increasing number of users and music items, the issue of data sparseness becomes inevitable, resulting in low accuracy and efficiency. Another fundamental challenge is that, when a new user or a new item enters in to a recommendation system based on the collaborative method, it is impossible for the system to respond properly due to the lack of historical data [2, 4, 5].

Several music recommendation systems are currently available online, which are used to recommend new music items to users or help them find potentially interesting music items. Figure 1 shows a screenshot of Douban Music.



Fig. 1 An illustration of the recommendation results in Douban Music

2.2 Recommendation Based on Label

As far as music is concerned, labeling is an effective way to describe the properties of music items and also accurately classify music items [6–9]. With the use of labels, the original user-item structure is extended to user-label-item (i.e., not only the items that a user has purchased but also the properties of the items are available). However, traditionally, labels are used as independent keywords in most of the music recommendation systems [10], and their correlation has not yet been fully exploited. The correlation information between two labels can reflect their underlying semantic relationship and can help better calculate the similarity between two music items. After all, in the era of Web 2.0, it is increasingly easier to get the labels of music items [11]. For example, they can be tagged by either music experts or through the collaborative efforts of millions of users.

3 Recommendation Based on Label Correlation

3.1 System Architecture

The general framework of the proposed music recommendation system based on label correlation is shown in Fig. 2. For new users, they are allowed to manually select a set of labels for themselves, and the system will recommend music items with similar label sets (content-based). For existing users, the system will create two recommended lists. Given the set of labels associated with the user (the set of labels of music items purchased by this user) and the label correlation information based on historical data, the first list is created by finding music items with similar label sets (content-based). The second list is created by finding a set of users with similar label sets and then selecting some items from their purchase records (CF).

3.2 Methodology

The labels of the music items purchased by a user represent this user's interest or preference. For each user, according to the sales record and the complete label list, a four-element tuple (*user*, *label*, *num_label*, *num_item*) is created as the most basic relationship between users and labels where *num_label* is the number of times that *label* has appeared in the purchased items by *user* and *num_item* is the number of purchased items by *user* (e.g., see Table 1).

The representation power of a label to a user is defined as *describe* (*user*, *label*), which describes the importance of the label to the user (Table 2):

$$\text{describe}(\textit{user}, \textit{label}) = \textit{num_label} / \textit{num_item} \quad (1)$$

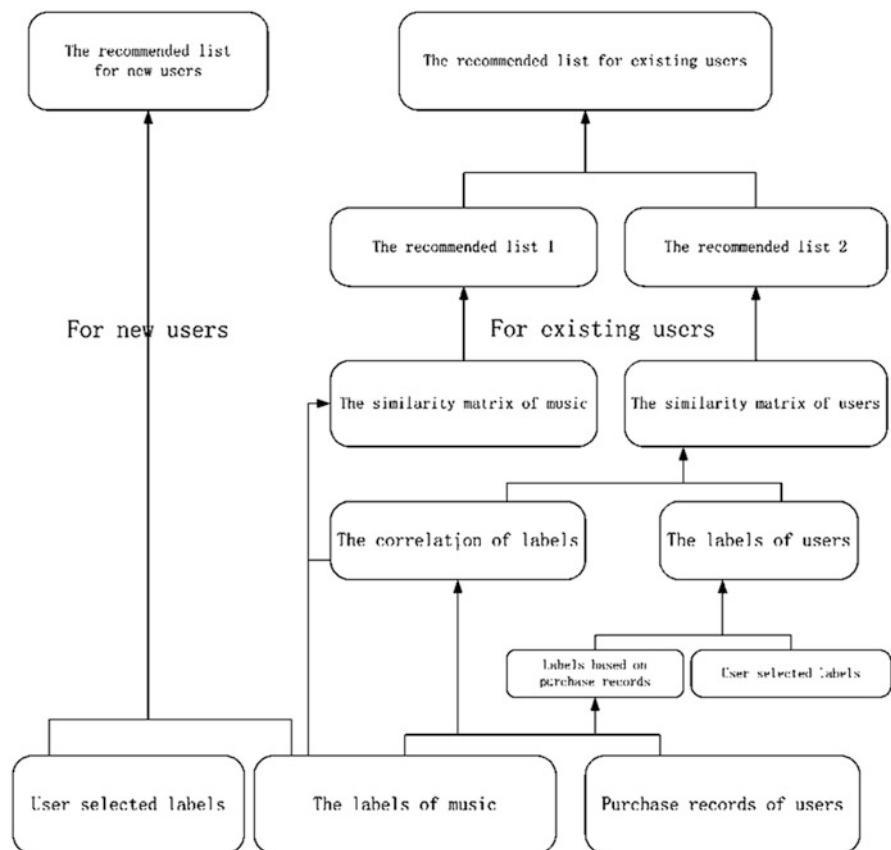


Fig. 2 The general framework of the music recommendation system based on label correlation. Note that new users and existing users are treated differently

Table 1 The four-element tuple representing the relationship among user, label, and item

User	Label	num_label	num_item
u_i	l_m	3	3
u_i	l_n	1	3
u_i	l_o	1	3
u_j	l_m	2	3
...

Table 2 The importance of labels to users

User	Label	Describe
u_i	l_m	3/3
u_i	l_n	1/3
u_i	l_o	1/3
u_j	l_m	2/3
...

For u_i , the correlation between two labels is defined as

$$\text{label_cor}_i(l_m, l_n) = \text{describe}(u_i, l_m) \text{describe}(u_i, l_n) \quad (2)$$

Note that for $m = n$, the value of label_cor_i is set to 1. Over the entire set of users, the correlation between two labels is defined as

$$\text{label_cor}(l_m, l_n) = \sum_i \text{label_cor}_i(l_m, l_n) \quad (3)$$

The similarity between two users is given by Eq. (4) where L_i and L_j are the two corresponding label sets:

$$\text{user_sim}(u_i, u_j) = \sum_{l_1 \in L_i} \sum_{l_2 \in L_j} \text{label_cor}(l_1, l_2) \text{describe}(u_i, l_1) \text{describe}(u_j, l_2) \quad (4)$$

The similarity between two music items is defined as

$$\text{item_sim}(\text{item}_i, \text{item}_j) = \sum_{l_1 \in L_i} \sum_{l_2 \in L_j} \text{label_cor}(l_1, l_2) \quad (5)$$

The similarity between a user and a music item is defined as

$$\begin{aligned} \text{user_item_sim}(u_i, \text{item}_j) &= \sum_{l_1 \in L_i} \sum_{l_2 \in L_j} \text{label_cor}(l_1, l_2) \\ &\times \text{describe}(u_i, l_1) \text{describe}(u_i, l_2) \end{aligned} \quad (6)$$

The advantage of this approach is that it is straightforward to calculate the distance between users or music items with different numbers of labels. Also, our approach is flexible as the distance calculation does not simply depend on the existence or absence of a label. Instead, all relationships among labels are taken into account, which is expected to exploit more information hidden in the label sets.

For new users, since there is no personalized information in the system, the label cloud can be used to guide the user to choose the labels for themselves. The size of each displayed label is determined by the number of times that this label has been used. According to the selected label set, the similarity between the label set and the music items can be calculated, and the most similar items will appear in the recommended list.

The recommended list for existing users consists of two parts. The first part of the list is created by finding music items similar to those that have been purchased by the user. Denote the set of purchased items by $\{\text{item}_1, \text{item}_2, \dots, \text{item}_w\}$. For each item in this set, 5 most similar music items are retrieved, resulting in $5w$ items



Fig. 3 An example of the label cloud in our system

in total. If an item appears k times, its similarity value will be multiplied by k . All items are sorted by the similarity (items that have already been purchased before are removed) and the top n items are selected. For the second part of the list, m users that are most similar to the given user are identified. Next, the similarity between the given user and each of the items that have been purchased by this set of m users is calculated. At last, all items are sorted based on the similarity, and up to 50 items are selected (items that have already been purchased before are removed). The final recommended list is created by appending the second part of the list to end of the first part of the list.

4 Experiments

4.1 Dataset

The dataset used in this paper was obtained from a LP (Long Play) album store on Taobao Marketplace,⁶ the most popular online platform in China for small businesses and individual entrepreneurs to open retail stores, which was founded by Alibaba Group.⁷ The dataset contained the purchase record of each customer and a list of all albums in stock. The time range of the dataset was from 2009-12-16 to 2012-01-12, which was divided by the time point of 2011-12-04 into the training set and the test set. Each album was labeled by a group of music funs according to its information such as singer, region, music style, melody, rhythm, and emotion.

⁶<http://www.taobao.com>

⁷<http://www.alibaba.com>

Table 3 The experiment results of recommendation based on label correlation and TF-IDF

Label correlation					TF-IDF		
<i>n</i>	<i>m</i>	<i>L</i>	Recall	Precision	<i>L</i>	Recall	Precision
2	1	5	20.32%	15.21%	5	13.20%	9.78%
2	2	8	24.71%	20.36%	8	18.52%	13.41%
3	2	10	29.63%	23.32%	10	20.61%	16.75%
4	2	10	33.54%	23.96%			
5	3	15	41.23%	22.78%	15	24.33%	14.38%
5	4	15	43.56%	22.91%			
5	5	20	44.37%	19.83%	20	25.24%	13.22%

4.2 Experimental Results

For evaluating recommendation systems, *recall* and *precision* have been widely used as the performance metric [12, 13]. Suppose *S*, *M*, and *N* are the length of the recommended list; the number of albums purchased by all users; and the number of albums purchased by all users that also appeared in the recommended list. *Recall* is defined as N/M , which is equal to the proportion of actually purchased albums that have also been recommended. *Precision* is defined as N/S , which is equal to the proportion of recommended albums that have been actually purchased.

In the experiments, we considered three parameters: *n*, *m*, *L*, representing the number of items in the first part of the recommended list, the number of similar users, and the maximum length of the recommended list, respectively. For comparison, the content-based recommendation system based on TF-IDF and Jaccard similarity [14] was implemented. The Jaccard similarity was defined as the size of intersection divided by the size of the union of the two label sets, while TF-IDF was used for measuring the importance of a label to an album.

From Table 3, we can see that with the increasing value of *n*, *m*, and *L*, the *recall* value increased monotonically, as more and more albums appeared in the recommended list. In the meantime, the *precision* value increased from around 15% to nearly 24% before dropping back to under 20%, which shows that parameter tuning is important for recommendation systems in order to achieve satisfactory performance. By contrast, our method achieved unanimously better results across various parameter settings compared to the traditional method in which each label was treated as an independent keyword.

5 Conclusion

This paper addressed a research question of significant practical value: how to effectively use label information in the scenario of music recommendation. Different from traditional recommendation systems where labels are only treated

as independent keywords, we proposed to exploit the correlation among labels and use the label set to represent both the music items and users. The similarity measures for user to user, item to item, and user to item were also defined. Experiment results on a real-world sales dataset show that, with this type of correlation information, the connection among users and music items can be better described and the quality of recommendation in terms of *precision* and *recall* were both improved compared to traditional methods based on the Jaccard distance and TF-IDF.

As to future work, we will investigate and compare other ways to define the label correlation and calculate the similarity among users and music items, which may hopefully extract more useful information from the original dataset and better reflect the interest of users. More importantly, most existing studies on recommendation are based on the assumption of stationary user behaviors by treating all purchase records as an unordered list. However, it is likely that the interest of a user on music may change during time. Consequently, we may explicitly incorporate the time factor into the framework of recommendation by tracking the change of labels associated with a user (e.g., change of frequency, new labels).

Acknowledgements This work was supported by the National Natural Science Foundation of China (No. 60905030). The authors are also grateful to the LP album store owner for providing the sales dataset.

References

1. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating “Word of Mouth”. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 210–217. ACM Press, New York (1995)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
3. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Info. Syst.* **23**(1), 103–145 (2005)
4. Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., Riedl, J.: Getting to know you: learning new user preferences in recommender systems. In: 7th International Conference on Intelligent User Interfaces, pp. 127–134. ACM Press, New York (2002)
5. Resnick, P., Varian, H.R.: Recommender systems. *Commun. ACM* **40**(3), 56–58 (1997)
6. Pazzani, M., Billsus, D.: Learning and revising user profiles: the identification of interesting web sites. *Mach. Learn.* **27**(3), 313–331 (1997)
7. Basu, C., Hirsh, H., Cohen, W.: Recommendation as classification: using social and content-based information in recommendation. In: Fifteenth National Conference on Artificial Intelligence, pp. 714–720. AAAI Press, Menlo Park (1998)
8. Cano, P., Koppenberger, M., Wack, N.: Content-based music audio recommendation. In: 13th Annual ACM International Conference on Multimedia, pp. 211–212. ACM Press, New York (2005)
9. Pampalk, E., Flexer, A., Widmer, G.: Improvements of audio-based music similarity and genre classification. In: 6th International Conference on Music Information Retrieval, pp. 628–633. London, UK (2005)

10. Nakamoto, R., Nakajima, S., Miyazaki, J., Uemura, S.: Tag-based contextual collaborative filtering. *IAENG Int. J. Comput. Sci.* **34**(2), 214–219 (2007)
11. Sinclair, J., Cardew-Hall, M.: The folksonomy tag cloud: when is it useful? *J. Info. Sci.* **34**(1), 15–29 (2008)
12. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: 2nd ACM Conference on Electronic Commerce, pp. 158–167. ACM Press, New York (2000)
13. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender system. *ACM Trans. Info. Syst.* **22**(1), 5–53 (2004)
14. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized recommendation in social tagging systems using hierarchical clustering. In: 2008 ACM Conference on Recommender Systems, pp. 259–266. ACM Press, New York (2008)

Inferring Public and Private Topics for Similar Events

Xubo Wen, Xiaoli Ma, Huan Xia, and Juanzi Li

Abstract Event detection, extraction, and tracking can help people to better understand the event that happened in the world. Previous research focuses on mining single event. In this paper, we propose a topic model to infer the public and private topic from a group of similar events. Aiming at the consistency and mapping of topics, this model discriminates public and private topics by using Bernoulli distribution to determine the source of words. Experiment on earthquake dataset shows that our proposed algorithm can induce the public and private topics acceptable by users.

1 Introduction

Event detection, extraction, and tracking focus on news streams and social content such as Twitter and Weibo. The previous research on event detection is based on the news stream within one single event. Event extraction is focused on the extraction of 5W1H or using event template. These methods ignore the information from different news streams. In this paper, we want to extract the semantic information from similar events over multiple news streams. Taking earthquakes in Haiti and Chile as an example shown in Fig. 1, the Haiti earthquake and Chile earthquake are two similar events that happened at different times, but they shared some common topics of “international aid” and “Seismic search” and owned specific topics like “history of Chile earthquake.” If we can infer these topics from multi-similar events, we can get the topic distribution for a specific kind of event such as earthquake and international sports game.

X. Wen (✉) • X. Ma • H. Xia • J. Li

Department of Computer Science and Technology, Tsinghua University,

Beijing 100084, People’s Republic of China

e-mail: wexubo@gmail.com; thumxl@gmail.com; xiahuan.cn@gmail.com;

lijuanzi2008@gmail.com

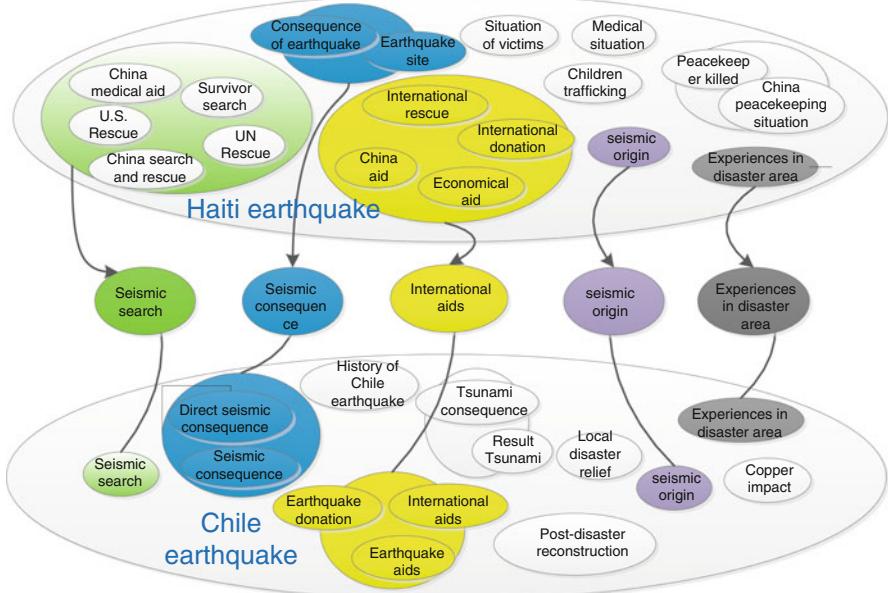


Fig. 1 Topic features of similar event

The contribution of this paper includes two aspects. Firstly, we put forward the concept of event pattern for news report of same kind events. Secondly, we propose a topic model solving the topic space consistency and public topic recognition problems at one time to infer the public and private topics.

The organization of this paper is as follows. Section 1 introduces the problem of topic inferring for similar events. In Sect. 2, we formally define the problem and explain the details and effectiveness of our model. Section 3 shows the results. Section 4 illustrates the related work. Section 5 includes the conclusion.

2 Topic Modeling for Multiple Similar Events

2.1 Problem Definition

Similar events can be periodical events with regular contents and events with same attributes. Formally, we define the following terms:

- An event set E is a collection of n events in same type denoted by $E = \{e_1, e_2, \dots, e_n\}$, where e_i is an event. Each e_i contains k_i documents which are reported when e_i happened and we denote it as $e_i = \{d_{e_1,i}, \dots, d_{e_1,k_i}\}$.

Table 1 Symbols of topic model based on similar events

Symbol	Representation
T_c, T_s	Public topic and private topic
ϕ^c, ϕ^s	The word distributions of public topic and private topic
β^c, β^s	Hyper parameters of public topic and private topic distribution
$\eta_{d,s}$	Ratio of public topic to private topic generated around one event
α^c, α^s	Hyper parameters of public topic and private topic distribution
x_{di}	Signify the position in generating process coming from public or private topic, value 0 or 1
θ_d^c, θ_d^s	Document distribution of public topic and private topic
γ^c, γ^s	Hyper parameters signifying the ratio of public topic and private topic
N_d	The number of words in document d
$c_{d,c-i}, c_{d,s-i}$	The numbers of public topic and private topic in document d without counting as position i
$m_{d,z-i}$	The number of topic z assigned to document d without counting as position i
$n_{z,w-i}$	The number of word w assigned to position i that is assigned to topic z without counting as position i

- A public topic P_b is the topic that occurred in all events in event set E . We define $P_b = \{d_{e_1,i}, \dots, d_{e_1,j}, d_{e_2,k}, \dots, d_{e_2,l}, \dots, d_{e_n,h}, \dots, d_{e_n,g}\}$.
- A private topic P_{br} is a topic which occurred only in one event or few events in E , we define $P_{br} = \{d_{e_1,i}, \dots, d_{e_1,j}, d_{e_2,k}, \dots, d_{e_2,l}, \dots, d_{e_t,h}, \dots, d_{e_t,g}\}$.
- Given event e_i , its topic space can be denoted by $TSi = \{z_{e_1,i}, \dots, z_{e_1,j}\}$, where $z_{e_1,i} = \{d_{e_1,k}, \dots, d_{e_1,l}\}$. To infer public and private topics, we define our problems below:
- How to map $TS1, TS2, \dots, TSm$ into a common topic space $TSc = \{z_1, z_2, \dots, z_c\}$, where TSc can represent the inner topic space for all events in E .
- How to decide whether $z_i \in \{z_1, z_2, \dots, z_c\}$ is a public topic or private topic.

2.2 Topic Models Based on Similar Events

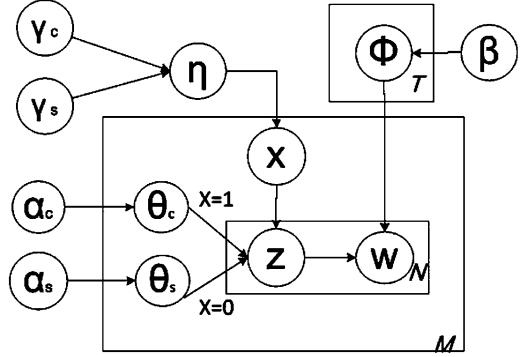
First, we list the symbols used in the model in Table 1.

In order to ascertain public topics, we pre-assume some topics to be public topic. In the generating process, let a prior distribution $\eta_{d,s} = Beta(\gamma_c, \gamma_s)$ be a parameter of Bernoulli distribution. Figure 2 shows our topic model.

Next, we will formally describe the generating process of the model:

1. For public topics, the distribution is over prior parameter β^c : $\theta^c = Dir(\beta^c)$.
2. For single event e :
 - a) For all private topics T_s , sample to attain its distribution $\theta^s = Dir(\beta^s)$.
 - b) For every document d :

Fig. 2 Topic model based on similar events



- i. Sample a Bernoulli parameter $\eta_{d,s} = \text{Beta}(\gamma_c, \gamma_s)$.
- ii. Assign document d a private topic distribution $\theta_d^s \sim \text{Dir}(\alpha^s)$.
- iii. Assign document d a public topic distribution $\theta_d^c \sim \text{Dir}(\alpha^c)$.
- iv. For every position i in document d :
 - A. With the value $\eta_{d,s}$ from i , tag value is $\chi_{di} \sim \text{Bernoulli}(\eta_{d,s})$, and χ_{di} signifies that the word in position i is from public topic distribution or private topic distribution.
 - B. Assign the word a topic z_{di} based on multinomial distribution $\text{Multinomial}(\phi_d^{z_{di}})$.
 - C. Assign the position i a word w_{di} based on multinomial distribution $\text{Multinomial}(\phi_{z_{di}}^{w_{di}})$.

Gibbs Sampling Method is used to estimate the parameters. The formulas are (1) and (2): “ $-i$ ” means that the word in position i does not count. The solving process exhibits in formulas (3), (4), and (5).

$$p(x_{di} = s, z_{di} = t) \propto \frac{c_{d,s-i} + \gamma^s}{N_d + \gamma^s + \gamma^c} \frac{\mathbf{m}_{d,z-i} + \boldsymbol{\alpha}_z}{\sum_{z \in T_s} \mathbf{m}_{d,z-i} + \boldsymbol{\alpha}_z} \frac{\mathbf{n}_{z,w-i} + \boldsymbol{\beta}^s}{\sum_w \mathbf{n}_{z,w-i} + \boldsymbol{\beta}^s} \quad (1)$$

$$p(x_{di} = c, z_{di} = t) \propto \frac{c_{d,c-i} + \gamma^c}{N_d + \gamma^s + \gamma^c} \frac{\mathbf{m}_{d,z-i} + \boldsymbol{\alpha}_z}{\sum_{z \in T_c} \mathbf{m}_{d,z-i} + \boldsymbol{\alpha}_z} \frac{\mathbf{n}_{z,w-i} + \boldsymbol{\beta}^c}{\sum_w \mathbf{n}_{z,w-i} + \boldsymbol{\beta}^c} \quad (2)$$

$$\theta_{d,z}^x = \frac{\mathbf{m}_{d,z} + \boldsymbol{\alpha}_z}{\sum_{z \in T_x} \mathbf{m}_{d,z} + \boldsymbol{\alpha}_z} \quad (3)$$

$$\phi_{z,w}^x = \frac{\mathbf{n}_{z,w} + \boldsymbol{\beta}_w}{\sum_{w \in T_x} \mathbf{n}_{z,w} + \boldsymbol{\beta}_w} \quad (4)$$

$$\eta_{d,x} = \frac{c_{d,x} + \gamma^x}{N_d + \gamma^s + \gamma^c} \quad (5)$$

$$\alpha = 50/T, \quad \beta = 0.01 \quad (6)$$

The hyper parameters $\alpha, \beta, \gamma_s, \gamma_c$ are also called concentration parameter. Hyper parameters α, β are prior parameter of Dirichlet. γ_s, γ_c are identical to the positions assigned to public topics and private topics and are set to the same value. All documents in one event share the same parameter values. Based on experience value, the values of α, β are set as formula (6). As parameter β is for word list, its proper value is 0.01 for a word list of 100,000 words.

3 Experiment and Result

To get the inner structure of event and event pattern of similar events, the procedure is divided into two parts: for single event, the clustering method is used to get the inner structure; for different events of the same kind, public topics are merged to attain the event pattern.

In order to verify the effectiveness of topic model over distinguishing public topics and private topics, the model is run on the events which are in the similar event sets to verify whether the results met our expectations and whether the constraint on public topics and private topics by human is working.

3.1 Dataset

The datasets are the similar event sets of earthquake and the details are in Table 2.

3.2 Experiment Details

In the experiments, hyper parameters α, β are set according to formula (6). After assigning different values to combination parameters γ_s, γ_c and comparing the results, the model is not sensitive to them. With overall consideration of the length of documents (i.e., the average size of word list for each document), γ_s, γ_c are set to 50.

The number of topics is set by the size of the datasets to ensure topic space is neither too large to be trivial nor too small to distinguish the difference. There are

Table 2 Number of news reports in “earthquake”

Location of the earthquake	Chile	Haiti	Japan	Total
Number of news reports	632	2,070	893	3,595

Table 3 Keywords in topics of “earthquake”

Earthquake		
	Topic	Keywords
Public topics	Medical condition	Victims, hospital, medical, doctors, survivors, rescued, dead bodies
	Earthquake losses	Company, insurance, economic structures, facilities, disaster
	International assistance	Reconstruction, provided assistance, society, meetings, reconstruction
	Earthquake donation	Disaster, events, people donations, donations, received, published
Private topics	Tsunami affected	The Pacific, alarm, Hawaii, impact, waves, residents, Chile
	Chile earthquake	Chile, Santiago, local, capital, time, earthquake
	Nuclear leak	Nuclear power plant, reactors, Naoto Kan Electric Power Company

3,595 documents in earthquake, so public topic number is set 25 by experience point, and the private topic numbers of the three events are 5, 15, and 7.

3.3 Results and Analysis

In this section, we verify whether the experiment results are in line with our expectations manually, which means that public topics are the events that frequently occurred and the private topics are specific events. After observing the keywords in each public topic, these topics can be seen mainly associated with the earthquake’s general topic. For private topics, the keywords are reflected with the particular topics of earthquake in Table 3.

After analyzing the earthquake topic, the public topics are in the upper part, and the private ones are in the lower part. In Table 3, the keywords are found in the topics as follows: Firstly, the public topic is of general topics about events such as medical events and losses caused by the earthquake, in line with our expectations. Secondly, private topic is specific to a particular event topic, such as the nuclear leakage caused by the earthquake in Japan.

Based on above analysis, after dividing topic space into public topics and private topics artificially in the model, the experimental results show that the actual public topics and private topics can be distinguished and meet our expectations.

4 Related Work

Topic detection and tracking (TDT) includes three subtasks: the segmentation task, the detection task, and the tracking task. Most research work around TDT is based on the data of single event on multiple time slots. The earliest work on topic detection began in 1998 by [1]. Later, topic model such as PLSA and LDA emerged and soon were applied in topic detection [2, 3].

In [4], Hong et al. presented a method for topic detection over different news reports centered on single event. Based on the work [5], the author recommended a parameter representing the ratio of public topics to private topics relevant to the text streams to distinguish public topics in different text streams. So this is the source for our model on similar event set.

5 Conclusion

In this paper, the problems of topic detection and topic mapping in the similar events are solved at the same time using LDA expansion model. After verifying the actual data from the experiments, the theories that the news can be divided into the public topics and the private topics are broadly in line with expectations.

Acknowledgements The work is supported by the Natural Science Foundation of China (No. 61035004, No. 60973102), 863 High Technology Program (2011AA01A207), European Union 7th framework project FP7-288342, and THU-NUS NExT Co-Lab and the project cooperated with Chongqing research institute of science and technology.

References

1. Yamron, J., Yang, Y.: Topic detection and tracking pilot study final report. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218. Lansdowne, VA (1998)
2. Blei, D.M., Lafferty, J.D.: Correlated topic models. In: Advances in Neural Information Processing Systems 18, Vancouver, BC (2005)
3. Li, W., McCallum, A.: Pachinko allocation: dag-structured mixture models of topic correlations. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 577–584. ACM, Pittsburgh (2006)
4. Hong, L., Dom, B., Gurumurthy, S., Tsoutsouliklis, K.: A time-dependent topic model for multiple text streams. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 832–840. ACM, San Diego (2011)
5. Paul, M.: Cross-collection topic models: automatically comparing and contrasting text. Bachelor Degree. Advisor: Girju, R. Department of Computer Science. University of Illinois at Urbana-Champaign (2009)

SemreX: A Semantic Association-Based Scientific Literature Sharing System

Pingpeng Yuan, Hai Jin, Yi Li, Binlin Chang, Xiaomin Ning, and Li Huang

Abstract Access to scientific literature information is a very important, as well as time-consuming, daily work for scientific researchers. However, more and more literatures are available. It imposes a challenge to literature database. Current literature systems mainly paid attention to a few explicit relationships among literature entities. In this paper, we present SemreX—a semantic association-based literature sharing system based on semantic web technologies. The concept of semantic association is proposed to reveal explicit or implicit relationships between semantic entities so as to facilitate researchers retrieving semantically relevant information. For the purpose of expression of semantic association, we propose a semantic association data model. Since it is very important for identification of semantic association to identify entities correctly, we develop some methods to identify entity names correctly. To discover some implicit semantic association, we propose two kinds of classification methods: SVM-k-NN used to classify those literatures based on category trees and Wikipedia-based classification using Wikipedia as knowledge base to classify literatures.

1 Introduction

With the advances of science and technology, more and more literatures are available. Thus, it is very important for researchers to retrieve scientific literature efficiently and easily. Currently, there are many literature retrieval systems available, such as ACM [1], IEEE Xplore [2], CiteSeerX [3], Libra [4], CiteULike [5], and

P. Yuan • H. Jin (✉) • Y. Li • B. Chang • X. Ning • L. Huang

Services Computing Technology and System Lab, Cluster and Grid Computing Lab,
School of Computer Science and Technology, Huazhong University of Science and Technology,
Wuhan 430074, China

e-mail: hjin@hust.edu.cn; joehust@gmail.com; decstery@gmail.com; ningxm@hust.edu.cn;
lihuang0101@gmail.com

DBLP [6]. However, those systems are generally based on explicit relationship among literatures, such as author publications and publication sources. Those result in losing a large amount of relevant information implied in literatures, categories, and social network which users are interested in. Those implicit relationships available will help users find valuable information more quickly.

Semantic web not only contains resources but also includes the heterogeneous relationships among them. Here, we introduce a semantic association-based literature sharing system—SemreX. The concept of semantic association is proposed to reveal explicit or implicit relationships among literature entities, such as authors, papers, publications, and categories so as to facilitate users retrieving semantically relevant information, as well as context of literature entities.

The rest of the paper is organized as follows. The brief introduction of SemreX is given in Sect. 2. Section 3 presents our data model: semantic association-based data model. Section 4 presents the semantic data storage system of SemreX. It is very important to identify entities correctly; in Sect. 5, we propose some of our methods to reach the goal. Classifying literatures can indicate many implicit relationships; we present two approaches in Sect. 6. Finally, conclusions are given in Sect. 7.

2 System Overview

Since there exists semantic association among any entities and semantic association indicates abundant information, SemreX needs to discover and manage those semantic associations efficiently. To reach the goal, SemreX automatically extracts and provides metadata and their explicit association from literatures. Moreover, SemreX also discovers implicit relationship of literature entities, implied in content and format of paper. Those explicit and implicit relationships are used to evaluate and cluster literature entities. Evaluated entities and their relationships are important criterions to match query term and rank search result or show interesting results to users.

SemreX is equipped with the following functions: automation extraction of literature metadata [14], semantic associate-based storage, literature classification and ranking, literature retrieval [15], and semantic data visualization. We also develop a highly efficient storage system for semantic data. Currently, SemreX manages metadata of more than 1 million literatures integrated from publicly available datasets including SwetoDblp [7], DBpedia [8], and metadata extracted from personal profiles. The dataset contains more than 1.2 million literatures and 600,000 authors. Moreover, the dataset is still growing.

3 Semantic Association Data Model

In this section, we first formally define the data model that our work builds upon. The data model is based on RDF/RDFS, but it is possible to extend our work to other formalisms as well.

There exists explicit and implicit relationship among entities. Although those relationships can be represented by RDF, it is not enough and requires the ability to transparently represent some data associated with RDF triples. For example, what is the probability that the relationship is correct or valid? We extend the RDF semantics and model the semantic web data as a directed graph. Before the presentation of semantic association data model, we introduce the graph definition for RDF data model.

Suppose T is an RDF graph, $T=(V^t, E^t, l_v^t, l_e^t)$, where $V^t = \{v_x: x \in \text{subj}(T) \cup \text{obj}(T)\}$; l_v^t is the node-labeling function, $l_v^t(v_x) = x$; and $E^t = \{(s, o) | (s, p, o) \in T\}$ is the edge-labeling function, $l_e^t(s, o) = p$.

Now, we define semantic association data model formally in the following.

Suppose labeling function set $F = \{f_i | i \in N, i=0, 1, \dots, n-1\}$ are statement-labeling functions. The triples consist of RDF graph $L = (V^l, E^l, l_v^l, l_e^l)$, where $V^l = S_t \cup A_v$, $S_t = \{st_t: t \in T\}$, $A_v = \{a_x: x = f_i(t), t \in T\}$, $l_v^l(v_x) = x$, $E = \{(t, f_i(t)) | t \in T\}$, $l_e^l(t, f_i(t)) = f_i$. f_i can be any functions, for example, may be functions which indicate the possibility of statement or weight functions.

In SemreX, when ranking resources globally, f_i is defined as a weight function indicating the association strengths of subject and object (or property value) in a statement. The ranking mechanism takes relationship analysis and the edge weight functions into account. Since an extended random surfer has different transition probabilities along different types of association in this data model, therefore, weights should rationally be determined to support Markovian walk [9].

4 Highly Efficient Semantic Data Storage

In order to manage RDF data efficiently and support RDF data analytical processing applications better, we also design an RDF store—DBLink.

DBLink consists of 4 layers: persistent layer, physical storage layer, RDF graph model layer, and interface layer. Due to volatility of memory, the persistent layer maps buffers in main memory to files and therefore provides a data persistent mechanism for SemreX. Thus, the data in memory can be preserved after the system is shut down. Every table managed by physical storage layer is mapped to a file. During startup of the storage system of SemreX, it reads map files and constructs tables of physical storage layer. When data are updated, the data are written into map files.

Physical storage layer manages tables and indexes. The tables and indexes are stored in buffers. Buffers are the basic component exchanged between all layers and components in the system. Each buffer that needs to be persistent is mapped to a file in file system, and upper-level tables, indexes, and temporary query execution results are stored in buffers. The tables in which physical storage layer manages include variable length table, fixed length table, and temporary table. The physical storage layer manages two kinds of RDF data: RDF instance and RDF schema.

The RDF graph model layer is a logical model layer, which provides a global view of RDF data. The view is implemented by the union of all the decomposed data stored in the tables of physical storage layer. On the top of the RDF graph model layer, there are several interfaces for the upper application, including import/export utilities, graph manipulation interface, and query interface.

5 Named Entity Identification

Before discovering semantic association among entities, it is very important to identify entities correctly. However, due to abbreviation and other reasons, it is very difficult to identify entities correctly. There are several reasons for ambiguity. One reason is abbreviations. Ambiguity also occurs in names of authors and organizations.

For the purpose of identifying named entities correctly, we adopt finite automata to identify named entities from papers. Considering the semantic completeness of content, we adopt the leftmost matching approach to identify named entities. Some identified named entities are abbreviations. According to the analysis on occurrence of full names and their abbreviations, SemreX uses regular expressions to extract the mapping between the full name of entities and their abbreviation from papers and their description. Moreover, considering some special cases, for example, Web Ontology Language is commonly abbreviated to OWL, we use the identified abbreviations of literature as local context to identify other abbreviations of literature.

Since there may exist several full names which have same abbreviation, it is necessary for name disambiguation to consider adjacent entities, which are called as global context. Global context includes category of adjacent entities and co-occurrence frequency of entities. SemreX combines global context with Hidden Markov Model to achieve the goal.

6 Multi-Class Literature Classification

With the huge number of literatures available, there is a growing need for literature categorization so as to help users manage and utilize those literatures. Literature categorization plays a key role in organizing the massive sources of literature information. Literature categorization is the process of assigning documents to a set of previously fixed categories. In many domains, such as libraries, presses, and supermarkets, many category trees are defined and used. For example, ACM defines a category tree which contains nine categories. Further, those categories are divided into subcategories. Recently, a large number of knowledge bases edited by volunteers all over the world are available on the web. They can incorporate latest information due to volunteers' efforts. However, the concepts of those knowledge

Table 1 Precision, recall, F-measure of MSVM-k-NN

Categories	Precision	Recall	F-measure
Hardware	83.33%	83.33%	0.83
Computer systems organization	92.59%	83.33%	0.88
Software	71.05%	90.00%	0.79
Data	78.79%	86.67%	0.83
Theory of computation	73.53%	83.33%	0.78
Mathematics of computing	86.96%	66.67%	0.75
Information systems	82.61%	63.33%	0.72
Computing methodologies	56.82%	83.33%	0.68
Computer applications	100.00%	60.00%	0.75
Average	80.63%	77.78%	0.78

bases are organized using graph structures. In SemreX, we use two kinds of technologies for literature classification: one is a tree-based text classification; another is a graph-based text classification.

Up to now, many popular algorithms have been applied to text categorization, such as Naïve Bayes, k -Nearest Neighbor, and Support Vector Machine. Among these methods, k -NN and SVM are used commonly and achieve better performance both in theory and practices [10]. Compared with other categorization methods, SVM has apparent advantages in avoiding overfitting of the result. But SVM is actually a binary classifier. However, in practice many problems are multi-class categorization problem. So SVM cannot be directly used in solving multi-categorization [11]. k -NN is an example-based text categorization algorithm. However, the determination of k has not yet got a good solution. Here, we adopt an approach named as Multi-class SVM- k -NN (MSVM- k -NN) which is the combination of SVM and k -NN [12]. In the approach, SVM is first used to identify category borders, and then k -NN classifies documents among borders. MSVM- k -NN can overcome the shortcomings of SVM and k -NN and improve performances of multi-class text classification. For our experiments, we use a dataset that contains 270 papers from ACM Digital Library. Table 1 shows the experimental results. According to Table 1, the average F-measure is 0.78, while the best average F-measure is 0.88. Moreover, the best and worst precisions are 1 and 0.56. The worst precision is in the category “Computing Methodologies.”

Literature classification can also be boosted by considering the neighborhood of terms in a graph structure (e.g., semantic association of terms in a knowledge base such as Wikipedia). Semantic association includes explicit relationship and implicit relationships. Thus, it is required to capture not only explicit relationship but also implicit semantic association such as texts which belong to the same categories. Here, we utilize Wikipedia to discover the implicit relationship among terms. Each article in Wikipedia describes a topic (or concept). Each article belongs to at least one category. Besides, categories are nested in a directed acyclic graph [13]. There exist three kinds of relationship among concepts of Wikipedia: synonyms, hyponyms, and associative. According to those semantic associations, we can relate those terms which are literally different.

To adopt Wikipedia as knowledge bases to classify texts, we need to extract terminologies of Wikipedia from documents and then adjust the representation of document according to Wikipedia. Documents can be represented as vector $v = (<\text{category tags}>, <\text{candidate concepts}>, <\text{related concepts}>)$, where *category tags* means that those words of d are category tags of Wikipedia and *candidate concepts* and *related concepts* indicate that those words of d are concepts of Wikipedia [16]. Then we calculate term frequency of each concept occurring in literatures. After each document is expressed as the above forms, we can generate candidate categories based on the voting mechanism.

7 Conclusions

In this paper, we present SemreX—a semantic association-based literature sharing system. The concept of semantic association is proposed to reveal explicit or implicit relationships between semantic entities so as to facilitate researchers retrieving semantically relevant information. For the purpose of expression of semantic association, we propose a semantic association data model. We develop methods to identify entity name correctly and classify literatures so as to reveal implicit semantic association.

Acknowledgements The research is supported by National Science Foundation of China grant No.61073096 and 863 program under grant No.2012AA011003.

References

1. ACM Digital Library, <http://portal.acm.org/portal.cfm>
2. IEEE Xplore, <http://ieeexplore.ieee.org/Xplore/home.jsp>
3. Microsoft Academic Search, <http://libra.msra.cn/>
4. CiteSeerX, <http://citeseerx.ist.psu.edu/>
5. CiteULike, <http://www.citeulike.org/>
6. DBLP Computer Science Bibliography, <http://dblp.uni-trier.de/>
7. Aleman-Meza, B., Hakimpour, F., Arpinar, I.B., Sheth, A.P.: SwetoDblp ontology of computer science publications. *J. Web Semant.* **5**(3), 151–155 (2007)
8. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives Z.: DBpedia: a nucleus for a web of open data. In: Proceedings of ISWC'07/ASWC'07, pp. 722–735. Springer, Berlin (2007)
9. Ning, X., Jin, H., Wu, H.: RSS: a framework enabling ranked search on the semantic web. *Inf. Process. Manag.* **44**(2), 893–909 (2008)
10. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. In: Proceedings of ICML 2004, pp. 321–328. Morgan Kaufmann, San Francisco (2004)
11. Chung, Y., Choi, E.H.C., Liu, L., Shhukran, M., Shi, D., Chen F.: A new hybrid audio categorization algorithm based on SVM weight factor and euclidean distance. In: Proceedings of CEA'07, pp. 152–157. World Scientific and Engineering Academy and Society, Wisconsin (2007)

12. Yuan, P., Chen, Y., Jin, H., Huang, L.: MSVM-k-NN: combining SVM and k-NN for multi-class text classification. In: Proceedings of WSCS'08, pp. 133–140. IEEE Press, New York (2008)
13. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In: Proceedings of AAAI'06, pp. 1301–1306. AAAI Press, Menlo Park (2006)
14. Chen, H., Jin, H., Ning, X., Yuan, P., Wu, H., Guo, Z.: SemreX: a semantic similarity based P2P overlay network. *J. Softw.* **17**(5), 1170–1181 (2006)
15. Ning, X., Jin, H., Wu, H.: SemreX: towards large-scale literature information retrieval and browsing with semantic association, In: Proceedings of ICEBE'06, pp. 602–609. IEEE Computer Society, Washington (2006)
16. Wang, P., Carlotta, D.: Building semantic kernels for text classification using Wiki. In: Proceedings of ACM SIGKDD'08, pp. 713–721. ACM, New York (2008)

Consequence-Based Procedure for Description Logics with Self-Restriction

Cong Wang and Pascal Hitzler

Abstract We present a consequence-based classification procedure for the description logics with self-restriction constructor. Due to the difficulty of constructing a concept inclusion model for self-restriction, we use a different proof by showing that all the completion rules can simulate all the corresponding ordered resolution inferences.

1 Introduction

Description logics (DLs) [2] are a family of logic-based formal languages, which provide theoretical foundation for ontology languages, such as OWL 2, the ontology language for the Semantic Web of W3C recommendation.¹ DLs serve as the basis for modeling and reasoning of ontologies. One of the key DL reasoning tasks is ontology classification, whose goal is to compute the hierarchical representation of subclass relations between the concepts in an ontology.

Most of the currently available ontology reasoners are based on model-building procedures such as the tableau [6] and the hyper-tableau [11] calculi. Such procedures classify an input ontology by iterating over all necessary pairs of concepts and trying to build a model of the ontology that violates the subsumption relation between them. Due to the unnecessary nondeterminism and the construction of large models, tableau methods usually cannot be scalable. Although hyper-tableau method improves the performance significantly, it is too complex to deal with

¹<http://www.w3.org/TR/owl2-overview/>.

C. Wang (✉) • P. Hitzler
Kno.e.sis Center, Wright State University, Dayton, OH, USA
e-mail: wang.156@wright.edu; pascal.hitzler@wright.edu

some tractable fragments of DLs efficiently.² Instead of building countermodels for candidate subsumption relations, the reasoning procedures for tractable DLs, such as \mathcal{EL} -family, were discovered to be able to derive subsumption consequences explicitly using inference rules. These rules are designed to produce all implied subsumption relations, while guaranteeing that only a bounded number of axioms are derived. This method is often called as completion rule-based procedure or consequence-based procedure.

Completion rule-based algorithm was firstly introduced for \mathcal{EL}^{++} in [1]. Later on, researchers extended it to Horn- \mathcal{SHIQ} [7] (known as CB³ reasoner), Horn- \mathcal{SROIQ} [13] and \mathcal{ALCH} [18], which is even beyond Horn DLs. Recently, researchers achieved to perform the consequence-based inference in a concurrent way [8, 9]. The concurrent classification reasoner ELK,⁴ with its availability of multi-core and multiprocessor, shows a substantial speedup by beating all the other currently existing reasoners for classifying SNOMED CT⁵ ontologies.

This paper provides a supplemental work for consequence-based procedures by extending the availability for self-restriction constructor. Since consequence-based procedures are closely related to resolution procedure [4, 10], it is not difficult to find the completion rules for self-restriction. By observing the resolution inference, one can easily establish the completion rules, such as $A \sqsubseteq \exists R.\text{Self}$ and $\exists R.B \sqsubseteq C$ can imply $A \sqcap B \sqsubseteq C$. The reason is that the corresponding resolution inference is done by resolving $\neg A(x) \vee R(x, x)$ and $\neg R(x, y) \vee \neg B(y) \vee C(x)$ to produce $\neg A(x) \vee \neg B(x) \vee C(x)$. The relationship between the two can be seen as that consequence-based procedure only performs the necessary inferences of ordered resolution procedure. The latter usually produces a large number of irrelevant clauses, which leads to inefficiency in practise.

Traditional proofs [1, 7, 9] for consequence-based procedures hardly work for self-restriction, because they are usually based on canonical model construction of concept inclusion. For example, one usually interprets a concept by all its subconcepts and interprets a role by the pair of two concepts [1, 7, 9], i.e., $A^{\mathcal{I}} = \{C | C \sqsubseteq A\}$ and $R^{\mathcal{I}} = \{\langle A, C \rangle | A \sqsubseteq \exists R.C\}$. Such proofs work well for existential restriction and universal restriction due to their semantics. But inference for self-restriction needs unifying variables, because its semantic is based on variables rather than concepts. Therefore, we apply an alternative kind of proof.

The structure of this paper is as follows. Section 2 describes some preliminaries of description logics and resolution procedure. Section 3 presents the completion rules for $\mathcal{ELH}(\text{Self})$. Section 4 extends the algorithm to deal with Horn- $\mathcal{SHI}(\text{Self})$. We will briefly discuss some possible extensions in Sect. 5. Finally we conclude.

²See the experiment comparison in [7].

³<http://code.google.com/p/cb-reasoner/>.

⁴<http://code.google.com/p/elk-reasoner/>.

⁵<http://www.ihtsdo.org/snomed-ct/>.

2 Preliminaries

In this section we define the description logic $\mathcal{ELH}(\text{Self})$ and Horn- $\mathcal{SHI}(\text{Self})$, as well as their fragment \mathcal{ELH} . Since we only focus on TBox classification task, we will not consider ABox assertions in this paper.

2.1 Description Logics

A signature of $\mathcal{ELH}(\text{Self})$ is a tuple $\Sigma = \langle N_C, N_R \rangle$ of mutually disjoint countably infinite sets of *concept names*, *role names*.

The syntax and semantics of $\mathcal{ELH}(\text{Self})$ is summarized in Table 1. The set of $\mathcal{ELH}(\text{Self})$ concepts is recursively defined using the concept constructors given in the upper part of Table 1. The terminology is a set \mathcal{O} of axioms defined in the lower part of Table 1.

The semantics of $\mathcal{ELH}(\text{Self})$ is defined using interpretations. An *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a nonempty set called *the domain of the interpretation* and $\cdot^{\mathcal{I}}$ is *the interpretation function*, which assigns to every $A \in N_C$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every $R \in N_R$ a relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. An interpretation \mathcal{I} satisfies an axiom α (written $\mathcal{I} \models \alpha$) if the respective condition of the right part in Table 1 holds; \mathcal{I} is a model of an ontology \mathcal{O} (written $\mathcal{I} \models \mathcal{O}$) if \mathcal{I} satisfies every axiom in \mathcal{O} . We say that α is a (logical) consequence of \mathcal{O} or is entailed by \mathcal{O} (written $\mathcal{O} \models \alpha$) if every model of \mathcal{O} satisfies α .

\mathcal{ELH} is the fragment of $\mathcal{ELH}(\text{Self})$ by disallowing self-restriction. Horn- $\mathcal{SHI}(\text{Self})$ extends $\mathcal{ELH}(\text{Self})$ with role transitivity, inverse role, and positive negation and disjunction and universal restriction (left hand of an axiom). However, since positive negation and disjunction can be simulated by conjunction, one can

Table 1 Semantics of $\mathcal{ELH}(\text{Self})$

Concept constructor	Syntax	Semantics
Top concept	\top	$\Delta^{\mathcal{I}}$
Bottom concept	\perp	\emptyset
Atomic concept	C	$C^{\mathcal{I}}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Existential restriction	$\exists R.C$	$\{x \in \Delta^{\mathcal{I}} y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
Self-restriction	$\exists R.\text{Self}$	$\{x \in \Delta^{\mathcal{I}} (x, x) \in R^{\mathcal{I}}\}$
Axioms	Syntax	Semantics
Concept inclusion (GCI)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Role inclusion	$R \sqsubseteq T$	$R^{\mathcal{I}} \subseteq T^{\mathcal{I}}$
$C, D \in N_C, R, T \in N_R$		

Table 2 Normal forms of \mathcal{ELH} axioms

$A \sqsubseteq \perp$	$\perp \sqsubseteq C$	$A \sqsubseteq C$	$A \sqcap B \sqsubseteq C$	$\exists R.A \sqsubseteq C$	$A \sqsubseteq \exists R.B$	$R \sqsubseteq T$
-----------------------	-----------------------	-------------------	----------------------------	-----------------------------	-----------------------------	-------------------

Table 3 The completion rules for \mathcal{ELH}

IR1	$\frac{}{A \sqsubseteq A}$	IR2	$\frac{}{A \sqsubseteq \top}$
CR1	$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C}$		
CR2	$\frac{A \sqsubseteq B \quad A \sqsubseteq C \quad B \sqcap C \sqsubseteq D}{A \sqsubseteq D}$		
CR3	$\frac{A \sqsubseteq B \quad B \sqsubseteq \exists R.C}{A \sqsubseteq \exists R.C}$		
CR4	$\frac{A \sqsubseteq \exists R.B \quad R \sqsubseteq S}{A \sqsubseteq \exists S.B}$		
CR5	$\frac{A \sqsubseteq \exists R.B \quad B \sqsubseteq C \quad \exists R.C \sqsubseteq D}{A \sqsubseteq D}$		

ignore the two constructors in Horn- $\mathcal{SHI}(\text{Self})$. For example, $A \sqsubseteq \neg C$ is equivalent to $A \sqcap C \sqsubseteq \perp$, and $A \sqsubseteq B \sqcap C$ is equivalent to two axioms $A \sqsubseteq B$ and $A \sqsubseteq C$.

2.2 Completion Rules for \mathcal{ELH}

In [1], a polynomial time classification procedure has been presented for the description logic \mathcal{EL}^{++} , which extends \mathcal{ELH} with the nominals, complex role inclusion(role chain) and “safe” concrete domains. The procedure uses a number of completion rules for deriving new concept inclusions. In Table 3, we list the completion rules relevant to \mathcal{ELH} [7]. Since the rules are applied to a normalized \mathcal{ELH} ontology \mathcal{O} that is obtained from the input ontology by structural transformation and simplification, we provide the \mathcal{ELH} normal forms in Table 2. In [1], it was shown that the rules IR1-R5 are sound and complete for classification, that is, a concept subsumption $A \sqsubseteq B$ is entailed by \mathcal{O} if and only if it is derivable by these rules.

2.3 Ordered Resolution

Ordered resolution [4] is a widely used calculus for theorem proving in first-order logic (FOL). The calculus has two parameters, an admissible ordering \succ on literals and a selection function.

Table 4 Translating $\mathcal{ELH}(\text{Self})$ into first-order logic

Translating concepts into FOL
$\pi_x(\perp) = \perp$
$\pi_x(\top) = \top$
$\pi_x(C) = C(x)$
$\pi_x(C \sqcap D) = \pi_x(C) \wedge \pi_x(D)$
$\pi_x(\exists R.C) = \exists y.[R(x, y) \wedge \pi_y(C)]$
$\pi_x(\exists R.\text{Self}) = R(x, x)$
Translating axioms into FOL
$\pi(C \sqsubseteq D) = \forall x : [\pi_x(C) \rightarrow \pi_x(D)]$
$\pi(R \sqsubseteq S) = \forall x \forall y : [R(x, y) \rightarrow S(x, y)]$
Translating KB into FOL
$\pi(KB) = \bigwedge_{\alpha \in KB} \pi(\alpha)$

An ordering \succ on literals is admissible if (1) it is well founded, stable under substitutions, and total on ground literals; (2) $\neg A \succ A$ for all ground atoms A; and (3) $B \succ A$ implies $B \succ \neg A$ for all atoms A and B. A literal L is (strictly) maximal with respect to a clause C if there is no other literal $L' \in C$ such that $(L' \succeq L)L' \succ L$. A literal $L \in C$ is (strictly) maximal in C if and only if L is (strictly) maximal with respect to $C \setminus L$ [10].

A *selection function* S assigns to each clause C a subset of negative literals of C (empty possibly); the literals are said to be *selected* if they are in $S(C)$. No other restrictions are imposed on the selection function, i.e., any arbitrary function mapping to negative literals are allowed.

With \mathcal{R} we denote the ordered resolution calculus, where $D \vee \neg B$ is called the main premise, $C \vee A$ is called the side premise, and $C\sigma \vee D\sigma$ is called conclusion:

$$\text{Ordered resolution : } \frac{C \vee A \quad D \vee \neg B}{C\sigma \vee D\sigma}$$

where (1) $\sigma = \text{mgu}(A, B)$, (2) $A\sigma$ is strictly maximal with respect to $C\sigma$ and no literal is selected in $C\sigma \vee A\sigma$, (3) $\neg B\sigma$ is either selected in $D\sigma \vee \neg B\sigma$ or it is maximal with respect to $D\sigma$ and no literal is selected in $D\sigma \vee \neg B\sigma$.

For general FOL, there is another rule needed, called *positive factoring*. It resolves two positive literals in one clause. However, since the target DLs in the paper are both Horn logics, the positive factoring rule is not required any more.

Table 4 shows the DL-to-FOL translation for $\mathcal{ELH}(\text{Self})$. The translation is straightforward based on the semantics of DL.

To be noticed, ordered resolution procedure for first-order logic is always sound and complete. However, different settings of the parameters can affect the termination of procedure significantly. Therefore, for decidable fragments of FOL, one needs careful tuning of details.

Table 5 The completion rules for self-restriction in $\mathcal{ELH}(\text{Self})$

Self1	$A \sqsubseteq \exists R.\text{Self}$	$R \sqsubseteq S$
	$A \sqsubseteq \exists S.\text{Self}$	
Self2	$A \sqsubseteq \exists R.\text{Self}$	$\exists R.B \sqsubseteq C$
	$A \sqcap B \sqsubseteq C$	

3 $\mathcal{ELH}(\text{Self})$

In this section, we establish the completion rules for the self-restriction constructors in $\mathcal{ELH}(\text{Self})$ and prove its soundness and completeness. Instead of the traditional proof by constructing a model of concept inclusion, we show that the completion rules can simulate all the possible ordered resolution inferences.

3.1 Completion Rules for Self-Restriction

The following rules Self1 and Self2 are the completion rules for self-restriction. Here, we give an informal explanation why they work. The Self1 rule is trivial. For the Self2 rule, recall what we mentioned in Sect. 1. The first-order logic clauses of the two axioms in Self2 are $\neg A(x) \vee R(x, x)$ and $\neg R(x, y) \vee \neg B(y) \vee C(x)$. Via resolving the two clauses by the ordered resolution, $\neg A(x) \vee \neg B(x) \vee C(x)$ can be produced, which is factually $A \sqcap B \sqsubseteq C$. In Sect. 3.3, we will show why these rules are sound and complete. For completion rules, since there is only a polynomial number of the axioms in $\mathcal{ELH}(\text{Self}) KB$ and all of them can be computed in polynomial time, we should also show that the ordered resolution procedure for $\mathcal{ELH}(\text{Self})$ is in polynomial time.

3.2 Resolution Procedure for $\mathcal{ELH}(\text{Self})$

Since our intuitive idea is to simulate all the possible ordered resolution inferences by the completion rules, we first need to show the ordered resolution procedure for $\mathcal{ELH}(\text{Self})$. We first give the definition of the resolution procedure by setting the two parameters, i.e., the predicate order and selection function.

Definition 1. Let \mathcal{R}_{DL} denote the ordered resolution calculus \mathcal{R} as follows:

- The literal ordering is an admissible ordering \succ such that $f \succ R \succ A$, for all function symbol f by skolemization, binary predicate symbol P , and unary predicate symbol A .
- The selection function selects every negative maximal binary literal in each clause.

Table 6 $\mathcal{ELH}(\text{Self})$ -clause types

(1) $\neg A(x)$	(7) $\neg A(x) \vee B(f(x))$
(2) $C(x)$	(8) $\neg A(x) \vee R(x, x)$
(3) $\neg A(x) \vee C(x)$	(9) $\neg R(x, x) \vee A(x)$
(4) $\neg A(x) \vee \neg B(x) \vee C(x)$	(10) $\neg R(x, y) \vee S(x, y)$
(5) $\neg R(x, y) \vee \neg A(y) \vee C(x)$	(11) $\neg A(x) \vee \neg B(f(x)) \vee C(f(x))$
(6) $\neg A(x) \vee R(x, f(x))$	(12) $\neg A(x) \vee \neg B(f(x)) \vee C(x)$

The clauses in Table 6 are all the possible clauses occurring during the ordered resolution procedure. We enumerate all possible \mathcal{R}_{DL} inferences between clauses and show that every conclusion is one of clause types of Table 6. With $[n, m] \rightsquigarrow [k]$ we denote an inference between clause type n and m resulting in clause type k . We denote the set of saturated clauses as $\Xi(KB)$.

Lemma 1. *Each \mathcal{R}_{DL} inference, when applied to $\mathcal{ELH}(\text{Self})$ -clauses, produces a $\mathcal{ELH}(\text{Self})$ -clause type in Table 6. The maximum length of each clause is 3. And the number of clauses different up to variable renaming is polynomial in $|KB|$.*

Proof. The ordered resolution inferences are possible between the following clauses: $[2, 3] \rightsquigarrow [2]$, $[2, 4] \rightsquigarrow [3]$, $[6, 5] \rightsquigarrow [12]$, $[6, 10] \rightsquigarrow [6]$, $[7, 1] \rightsquigarrow [2]$, $[7, 3] \rightsquigarrow [7]$, $[7, 4] \rightsquigarrow [11]$, $[8, 5] \rightsquigarrow [4]$, $[8, 9] \rightsquigarrow [3]$, $[8, 10] \rightsquigarrow [8]$.

(11) $\neg A(x) \vee \neg B(f(x)) \vee C(f(x))$ can only resolve with clause $\neg A(x) \vee B(f(x))$ or $B(x)$ and produce clause $\neg A(x) \vee C(f(x))$. Since ordered resolution only resolves on maximal literals, literal $\neg A(x)$ in clause type (7) can never participate. In addition, due to that every function symbol is unique after skolemization, there are no other clauses of clause type (7) containing $B(f(x))$. Since $\neg B(f(x))$ in (11) has to resolve with $B(f(x))$ or $B(x)$, (11) can only resolve with clause $\neg A(x) \vee B(f(x))$ or $B(x)$. For the same reason, (12) $\neg A(x) \vee \neg B(f(x)) \vee C(x)$ can only resolve with clause $\neg A(x) \vee B(f(x))$ or $B(x)$ and produce clause $\neg A(x) \vee C(x)$.

Any other inferences are not applicable. Therefore, every clause is one of the clause types of Table 6, and the maximum length of clauses is 3. Let c be the number of unary predicates, r the number of binary predicates, and f the number of unary function symbols in the signature of $\Xi(KB)$. Then, trivially c , r , and f are linear in $|KB|$. Consider now the maximal $\mathcal{ELH}(\text{Self})$ -clause of type 5 in Table 6. There are possibly at most rc^2 clauses of type 5, which number is polynomial in $|KB|$. For other $\mathcal{ELH}(\text{Self})$ -clause types, the bounds on the length and on the number of clauses can be derived in an analogous way. Therefore, the number of $\mathcal{ELH}(\text{Self})$ -clauses different up to variable renaming is polynomial in $|KB|$.

Corollary 1. *For a $\mathcal{ELH}(\text{Self})$ knowledge base KB , saturating $\Xi(KB)$ by \mathcal{R}_{DL} decides satisfiability of KB and runs in time polynomial in $|KB|$.*

3.3 Soundness and Completeness

Now, we are ready to show the soundness and completeness of $\mathcal{ELH}(\text{Self})$ completion rules.

Lemma 2. *Each \mathcal{R}_{DL} inference by the ordered resolution procedure for $\mathcal{ELH}(\text{Self})$ can be simulated by the corresponding completion rules.*

Proof. [2, 3] \rightsquigarrow [2] and [2, 4] \rightsquigarrow [3] can be simulated by CR1 and CR2. [6, 10] \rightsquigarrow [6] can be simulated by CR4. [7, 1] \rightsquigarrow [2] and [7, 3] \rightsquigarrow [7] can be simulated by CR5. [8, 5] \rightsquigarrow [4], [8, 9] \rightsquigarrow [3], and [8, 10] \rightsquigarrow [8] can be simulated by Self1, CR1, and Self2, respectively. For [6, 5] \rightsquigarrow [12], since we argued that (12) $\neg A(x) \vee \neg B(f(x)) \vee C(x)$ can only resolve with clause $\neg A(x) \vee B(f(x))$ or $B(x)$ and produce clause $\neg A(x) \vee C(x)$, the $\mathcal{ELH}(\text{Self})$ knowledge base must contain the following axioms: $A \sqsubseteq \exists R.B$, $\exists R.D \sqsubseteq C$, and $B \sqsubseteq D$. Therefore, such inference can be captured by CR5. Similarly, for [7, 4] \rightsquigarrow [11], the knowledge base must contain $A \sqsubseteq \exists R.B$, $B \sqcap C \sqsubseteq D$, and $B \sqsubseteq C$, which can be captured by CR5 as well. So, all of the ordered resolution inferences for $\mathcal{ELH}(\text{Self})$ can be simulated by the corresponding completion rules.

Corollary 2. *The completion rules for $\mathcal{ELH}(\text{Self})$ are sound and complete.*

Proof. By Lemma 2, we know each ordered resolution inference can be captured by the corresponding completion rules. Since ordered resolution is sound and complete for FOL, hence for DLs. Therefore, the completion rules for $\mathcal{ELH}(\text{Self})$ are sound and complete.

4 Horn- $\mathcal{SHI}(\text{Self})$

Horn- $\mathcal{SHI}(\text{Self})$ extends $\mathcal{ELH}(\text{Self})$ by allowing role transitivity, inverse role, and positive occurrences of universal restrictions, i.e., $\text{Tra}(R)$, $R \sqsubseteq T^-$, and $A \sqsubseteq \forall R.B$. \mathcal{EL}^{++} allow complex role inclusion (role chain) but disallow inverse role and universal restriction constructors for complexity reasons. In [1], it was shown that adding any of the latter two constructors results in a complexity increase from PTime to EXPTIME. An intuitive explanation of the exponential blowup is that resolving axioms of the form $A \sqsubseteq \exists R.B$ and $C \sqsubseteq \forall R.D$ can possibly produce axioms $\sqcap A_i \sqsubseteq \exists R.(\sqcap B_j)$, where $\sqcap A_i$ and $\sqcap B_j$ are arbitrary conjunctions of atomic concepts.

In this section, we first introduce the normalization and the well-known technique for transitive role elimination. Then as the structure in previous section, we give out the extra completion rules for Horn- $\mathcal{SHI}(\text{Self})$, then describe the ordered resolution procedure and show the soundness and completeness.

Table 7 Rules for Horn- $\mathcal{SHI}(\text{Self})$

UR1	$A \sqsubseteq \exists R.B$	$A \sqsubseteq \forall S.C$	$R \sqsubseteq S$
	$A \sqsubseteq \exists R.C$		
UR2	$A \sqsubseteq \exists R.B$	$B \sqsubseteq \forall S.C$	$R \sqsubseteq S^-$
	$A \sqsubseteq C$		
UR3	$A \sqsubseteq \exists R.\text{Self}$	$B \sqsubseteq \forall S.C$	$A \sqcap B \sqsubseteq C$

Table 8 Horn- \mathcal{SHI} (Self)-clause types

(1) $\alpha(x) \vee \beta(f(x)) \vee A(x)$	(6) $\neg A(x) \vee R(f(x), x)$
(2) $\alpha(x) \vee \beta(f(x)) \vee A(f(x))$	(7) $\neg A(x) \vee R(x, x)$
(3) $\neg R(x, y) \vee \neg A(y) \vee C(x)$	(8) $\neg R(x, x) \vee A(x)$
(4) $\neg A(x) \vee \neg R(x, y) \vee B(y)$	(9) $\neg R(x, y) \vee S(x, y)$
(5) $\neg A(x) \vee R(x, f(x))$	(10) $\neg R(x, y) \vee S(y, x)$

$\alpha(x)$ is a disjunction $\neg A_1(x) \vee \dots \vee \neg A_n(x)$ with $A_i \in KB$
 $\beta(x)$ is a disjunction $\neg B_1(x) \vee \dots \vee \neg B_n(x)$ with $B_i \in KB$
Disjunctions $\alpha(x)$ and $\beta(x)$ may be empty
Disjunctions $\alpha(x)$ and $\beta(x)$ may contain same predicates

4.1 Normalization

The technique in this subsection can be referred in [7]. We denote a concept C is *simple* if it is of the form \perp , A , $\exists R.A$, $\exists R.\text{Self}$, $\forall R.A$, where A is an atomic concept. Every Horn- $\mathcal{SHI}(\text{Self})$ ontology \mathcal{O} can be transformed into an ontology \mathcal{O}' containing only axioms of the forms $\sqcap A_i \sqsubseteq C$, $R \sqsubseteq T$, and $\text{Tra}(R)$.

For each transitive role R , one can eliminate $\text{Tra}(R)$ by introducing a triple of axioms for every axiom $\sqcap A_i \sqsubseteq \forall R.B$ and every transitive sub-role T of R , i.e., $\sqcap A_i \sqsubseteq \forall R.B'$, $B' \sqsubseteq \forall T.B'$, and $B' \sqsubseteq B$, where B' is a fresh atomic concept.

4.2 More Rules

For Horn- $\mathcal{SHI}(\text{Self})$, one also needs to add the rules in Table 7. The UR1 and UR2 rules are the relevant rules for inference among existential restriction, universal restriction, and inverse role [7]. UR3 is the rule for inference between self-restriction and universal restriction.

4.3 Soundness and Completeness

Still, we first show the resolution procedure for Horn- $\mathcal{SHI}(\text{Self})$ but very briefly. We do not need to modify \mathcal{R}_{DL} for Horn- $\mathcal{SHI}(\text{Self})$. Table 8 describes all the possible clauses occurring during the procedure.

Lemma 3. *Each \mathcal{R}_{DL} inference, when applied to Horn- $\mathcal{SHI}(\text{Self})$ -clauses, produces a Horn- $\mathcal{SHI}(\text{Self})$ -clause type in Table 8. The number of different up to variable renaming is exponential in $|KB|$.*

Proof. (sketch) For Horn- $\mathcal{SHI}(\text{Self})$, the ordered resolution inferences are possible between the following clauses: $[2, 1] \rightsquigarrow [1]$, if $\beta(f(x))$ is empty, otherwise $[2, 1] \rightsquigarrow [2]$. $[5, 3] \rightsquigarrow [1]$, $[5, 4] \rightsquigarrow [2]$, $[5, 9] \rightsquigarrow [5]$, $[5, 10] \rightsquigarrow [6]$. $[6, 3] \rightsquigarrow [2]$, $[6, 4] \rightsquigarrow [1]$. $[7, 3] \rightsquigarrow [1]$, $[7, 4] \rightsquigarrow [1]$, $[7, 8] \rightsquigarrow [7]$, $[7, 9] \rightsquigarrow [7]$, $[7, 10] \rightsquigarrow [7]$.

Any other inferences are not applicable. Therefore, every clause is one of the clause types of Table 8. The fact of exponential blowup of the length and number of clauses is trivial by looking at clause type (1). So, it is straightforward to know that saturating Horn- $\mathcal{SHI}(\text{Self})$ $\Xi(KB)$ by \mathcal{R}_{DL} decides satisfiability of KB and runs in time exponential in $|KB|$.

Lemma 4. *Each \mathcal{R}_{DL} inference by the ordered resolution procedure for Horn- $\mathcal{SHI}(\text{Self})$ can be simulated by the corresponding completion rules.*

Proof. (sketch) $[2, 1] \rightsquigarrow [1]$ and $[2, 1] \rightsquigarrow [2]$ can be simulated by CR1 and CR2. $[5, 3] \rightsquigarrow [1]$ can be simulated by CR5, $[5, 4] \rightsquigarrow [2]$ by UR1, $[5, 9] \rightsquigarrow [5]$ by CR4, and $[5, 10] \rightsquigarrow [6]$ by UR2. $[6, 3] \rightsquigarrow [2]$ and $[6, 4] \rightsquigarrow [1]$ are by CR5 and UR2. The inference with clause type 7 can be simulated by CR4, Self1, Self2, and UR2. So, all of the ordered resolution inferences for Horn- $\mathcal{SHI}(\text{Self})$ can be simulated by the corresponding completion rules. Since ordered resolution is a sound and complete procedure for first-order logic, hence for Horn- $\mathcal{SHI}(\text{Self})$.

Corollary 3. *The completion rules for Horn- $\mathcal{SHI}(\text{Self})$ are sound and complete.*

5 Discussion

We have demonstrated the completion rules for the description logics $\mathcal{ELH}(\text{Self})$ and Horn- $\mathcal{SHI}(\text{Self})$. We believe our work can be easily extended to some even more complex DLs. For example, one can extend self-restriction for \mathcal{ALCH} [18]. Although \mathcal{ALCH} allows axioms with universal restriction appearing at left hand, axioms containing self-restriction cannot resolve with these axioms. The reason is that the first-order logic clause of left-universal-restriction axioms contains function symbol such that literal cannot unify with self-restriction literal. For example, the FOL clause of \mathcal{ALCH} normal form $\forall R.C \sqsubseteq A$ is $\neg R(x, f(x)) \vee \neg C(f(x)) \vee A(x)$. Clause of the axiom with self-restriction, such as $\neg A(x) \vee R(x, x)$, cannot resolve with it, because the variable x in $R(x, x)$ and $\neg R(x, f(x))$ cannot unify. In addition, negation and disjunction of concepts, which are allowed in \mathcal{ALCH} , do not infer with self-restriction. Therefore, we should be able to extend self-restriction for \mathcal{ALCH} .

We also conjecture that it should be also easy to extend the completion rules for $\mathcal{ELH}(\text{Self})$ to deal with nominals, i.e., $\mathcal{ELHO}(\text{Self})$. Since the DL-to-FOL translation for nominals can introduce equality literal, the calculi for reasoning in equational first-order logic, *paramodulation* or *superposition* [10, 12], are also

needed. While, since the variables x in self-restriction literal $R(x, x)$ always unify with other values together, such that it should not harm the resolution procedure by producing complex clauses. In this sense, one may also extend the algorithm to $\mathcal{ALCOH}(\text{Self})$ and Horn- $\mathcal{SHOI}(\text{Self})$.

When extending with complex RIAs (role chain), the situation becomes much more complicated. To the best of our knowledge, there is no perfect technique to deal with role composition in resolution procedure. Researchers usually force some restriction on the order of role predicate. In [3], the authors applied even more restricted order than the role regularity of OWL 2 \mathcal{SROIQ} [5]. But recently researchers find complex RIAs can be eliminated by formulating as a recursive expansion of universal restrictions [17], which is similar to the encodings of transitivity axioms as we described in Sect. 4.1. Therefore, the completion rules may even extend to \mathcal{SROIQ} by this elimination technique, together with the work for \mathcal{ALCH} [18].

6 Conclusions

In this paper, we demonstrate the completion rules for self-restriction constructor. We show this in two cases, $\mathcal{ELH}(\text{Self})$ and Horn- $\mathcal{SHI}(\text{Self})$. We believe these rules and proof technique can be extended to the more complex DLs, which will be our future work. We will also explore the completion rules for more DL constructors, like negation, disjunction, and conjunction of role [16] and concept product [15].

Acknowledgements This work was supported by the National Science Foundation under award 1017225 III:s Small: TROn—Tractable Reasoning with Ontologies.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the el envelope. In: IJCAI, pp. 364–369, 2005
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
3. Bachmair, L., Ganzinger, H.: Ordered chaining calculi for first-order theories of transitive relations. J. ACM **45**(6), 1007–1049 (1998)
4. Bachmair, L., Ganzinger, H.: Resolution theorem proving. In: Robinson and Voronkov [14], pp. 19–99
5. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible sroiq. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) KR, pp. 57–67. AAAI Press, Atlanta, Georgia (2006)
6. Horrocks, I., Sattler, U.: A tableaux decision procedure for shoiq. In: IJCAI, pp. 448–453, 2005
7. Kazakov, Y.: Consequence-driven reasoning for horn shiq ontologies. In: Boutilier, C. (ed.) IJCAI, pp. 2040–2045, 2009
8. Kazakov, Y., Kroetzsch, M., Simancik, F.: Practical reasoning with nominals in the el family of description logics. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) KR. AAAI Press, Atlanta, Georgia (2012)

9. Kazakov, Y., Krötzsch, M., Simancik, F.: Concurrent classification of el ontologies. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N.F., Blomqvist, E. (eds.) International Semantic Web Conference (1). Lecture Notes in Computer Science, vol. 7031, pp. 305–320. Springer, New York (2011)
10. Motik, B.: Reasoning in description logics using resolution and deductive databases. Ph.D. thesis (2006). <http://www dblp org/rec/bibtex/phd/de/Motik2006>
11. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. *J. Artif. Intell. Res.* **36**, 165–228 (2009)
12. Nieuwenhuis, R., Rubio, A.: Paramodulation-based theorem proving. In: Robinson and Voronkov [14], pp. 371–443
13. Ortiz, M., Rudolph, S., Simkus, M.: Worst-case optimal reasoning for the horn-dl fragments of owl 1 and 2. In: Lin, F., Sattler, U., Truszczynski, M. (eds.) KR. AAAI Press, Atlanta, Georgia (2010)
14. Robinson, J.A., Voronkov, A. (eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001). <http://www dblp org/rec/bibtex/conf/dlog/2003handbook>
15. Rudolph, S., Krötzsch, M., Hitzler, P.: All elephants are bigger than all mice. In: Baader, F., Lutz, C., Motik, B. (eds.) Description Logics. CEUR Workshop Proceedings, vol. 353. CEUR-WS.org (2008)
16. Rudolph, S., Krötzsch, M., Hitzler, P.: Cheap boolean role constructors for description logics. In: Hölldobler, S., Lutz, C., Wansing, H. (eds.) JELIA. Lecture Notes in Computer Science, vol. 5293, pp. 362–374. Springer, New York (2008)
17. Simancik, F.: Elimination of complex rias without automata. In: Kazakov, Y., Lembo, D., Wolter, F. (eds.) Description Logics. CEUR Workshop Proceedings, vol. 846. CEUR-WS.org (2012)
18. Simancik, F., Kazakov, Y., Horrocks, I.: Consequence-based reasoning beyond horn ontologies. In: Walsh, T. (ed.) IJCAI. pp. 1093–1098. IJCAI/AAAI, Atlanta, Georgia (2011)

SAPOP: Semiautomatic Framework for Practical Ontology Population from Structured Knowledge Bases

Xinruo Sun, Haofen Wang, and Yong Yu

Abstract The Semantic Web is evolving very quickly. There are already many theories and tools to model various kinds of semantics using ontologies. However after organizations completed modeling the ontology structure, the ontologies must also be filled with instances and relationships to make them practical. This process of ontology population could be hard because we are facing a cold start problem. On the other hand, the potential instances could already exist in LOD, online encyclopedia, or corporate databases in the form of structured data. We think these instances along with their related features could remedy the cold start problem a lot. We present a practical framework to verify this hypothesis. In this framework, first a semiautomated seed discovery method is used to bootstrap the population. Then, we use semi-supervised learning methods to refine and expand the seed instances. Finally the population quality is verified using an effective evaluation process.

1 Introduction

Although the Semantic Web and the Linked Data (LOD)¹ are developing very fast [2], the large amount of public RDF data available is usually not exactly what organizations want. In order to leverage the capabilities of Semantic Web technologies, the organizations often start with modeling ontologies for their specific business use cases. After that, it will cost a lot to fill the ontologies with instances or migrate from legacy databases.

¹<http://linkeddata.org/>.

X. Sun (✉) • H. Wang • Y. Yu

APEX Data & Knowledge Management Lab, Shanghai Jiao Tong University

e-mail: xrsun@apex.sjtu.edu.cn; whfcarter@apex.sjtu.edu.cn; yyu@apex.sjtu.edu.cn

On the other hand, structured data on the Internet is growing rapidly. For example, many websites are using RDFa or similars in the web pages to annotate their mentioned entities and their properties. Also LOD is evolving very quickly. It is very probable that organizations could put together one or several structured data sources to obtain lots of instances. This sounds simple, but first the instances have to be cleaned up and mapped to the correct parts in the ontology. It is challenging because the amount of instances is large, and the structured data sources are usually very diverse and contain many properties. Manually cleaning up and mapping the instances and properties to ontologies are unlikely to cover everything.

In this paper, we present a semiautomatic framework for practical ontology population named SAPOP that could help populate ontologies with instances.

SAPOP deals with this problem using a three-phase process. First a semiautomated seed discovery method is used to bootstrap the population for each category and discover reliable seed instances with the help of human labeling. Depending on the size of the ontology and how much effort the organization is willing to spend, this phase requires varying amount of labeling effort. We know labeling is expensive; all labeling effort in the framework will be used effectively.

Then, we use semi-supervised learning methods to refine and expand the seed instances. This phase will utilize the large amount of unlabeled data. This way, the manual clean up and mapping efforts are largely reduced. Finally, after an effective parameter selection method, we output the final instances for each category, and we evaluate the classifier performance at the same step. Note that the evaluation results could be sent back to first phase to carry out further refinement cycles if needed. In the refinement cycles, all labeled data will be used and there will be more training data so that more comprehensive and accurate instances would be discovered. In the following sections, we will describe each of the three phases in detail.

2 Problem Definition

In this paper, we focus on the problem of ontology instance population. First, we define the ontology we need to fill with.

Definition 1. An *ontology* can be viewed as a set of categories and their properties: $O = \{(c_i, P_i, G_i) | i = 1 \dots |C|\}$, where c_i is one of the $|C|$ categories in the ontology. P_i is the set of properties that each instance of c_i could have. G_i is the set of example instances associated with this category.

Next, we define the structured data source used to fill in the ontology.

Definition 2. A *structured data source* is a collection of instances: $S = \{(e_i, D_i) | i = 1 \dots N\}$. Here e_i is one of the N instances. $D_i = \{(k_{ij}, v_{ij})\}$ is the property-value pairs of that instance.

After we have the ontology and a structured knowledge base, we want to populate the ontology with the instances.

Definition 3. *Ontology instance population* is the problem of finding a partial function $F : S \rightarrow O$.

In this definition, each instance in S is mapped to one category in ontology O , when defined by F . The instances which are not defined by F means they do not belong to, or are unrelated to, ontology O . Note that we enforce each instance has at most one category associated to keep presentation simple. The framework itself could be easily extended to overcome this limitation.

3 Seed Discovery

In the first phase of SAPOP, we want to discover as many right instances as possible for each category. Instead of sampling over the entire data source and letting human labeler work on a large amount of instances, we use a semiautomated method to discover possible seed instances.

Since there are already information of the categories encoded in the ontology O , we need to make use of these information to bootstrap. Some rules for filtering the instances are constructed automatically from predefined templates described below. The rules along with their sampled instances are presented to the user to label. Compared to labeling instances, labeling rules are much more efficient. Users could also manually specify rules if they see fit.

Definition 4. A *rule* is a 4-tuple (c, k, v, \pm) . c is the category for this rule. k and v are key and value corresponding to D in structured data source. \pm is a Boolean indicating whether this rule is positive or negative.

We use the following list of rule templates in SAPOP to generate rules:

- The category names. For each category c_i , we generate a rule $(c_i, \text{"CATEGORY"}, c_i, \pm)$.
- The property names. For each property $p \in P_i$ of category c_i , we generate a rule $(c_i, p, *, \pm)$.
- The example instances. For each example $g \in G_i$ of category c_i , we generate a rule $(c_i, \text{"LABEL"}, g, \pm)$.

When the data source and the ontology are well aligned, the rules will generate enough good seed instances. In this case, we could just set $\pm = +$. Otherwise, human labelers need to spend a limited amount of effort to evaluate \pm . Here, for each rule, we present the human labelers with a sample of seed instances that would be generated by the rule if $\pm = +$. The labelers need to choose from three options: (a) all seeds are correct instances for this category, (b) all seeds are incorrect, and (c) there are both correct and incorrect seeds. After the rules are selected, the instances that satisfy the rules become seed instances for the respective categories. Seeds of correct/incorrect rules become positive/negative training data in the next section. Note that if a category has no correct rules, it would be impossible for the learn a model. In this case, users must find the seeds themselves.

4 Instances Expansion

In this second phase, we will learn a model from the training seeds gathered at the first phase and apply this model to all of the instances in the data source. The purpose of this phase is twofold. On one hand, the seeds might be incomplete. In this case, we hope there are patterns that could be learned to discover new instances. On the other hand, some seeds might be wrong. This might also be discovered by the model, because we have negative training data.

When we have gathered enough training data, both positive and negative, a supervised learning algorithm could be directly applied to learn a model for each category.

Definition 5. A *model* for category c is a function $M_c : S \rightarrow [0..1]$. Given the keys and values of one instance in S , the model will output a probability indicating if it is a positive instance of c .

Unfortunately, supervised learning algorithms are sometimes not directly applicable in our framework. This is because it is likely that the training data is not enough. There are usually too few or even no positive examples. In these cases, we come to semi-supervised learning to rescue, especially PU learning algorithms.

PU learning is the task of learning from positive and unlabeled examples. This problem assumes a two-class classification. However, the training data only has a set of labeled positive examples and a set of unlabeled examples, but no labeled negative examples. Two theorems in [3] laid the theoretical foundation for this task.

We will not present the details or choices of PU learning methods here because of space limitation. We refer to readers to a survey on this subject [11].

5 Parameter Selection and Evaluation

After we have the models and applied to the instances, we need to decide the thresholds to make judgments. Although in some models, thresholds of 0.5 are usually applied to distinguish positive and negative instances. We found that the defaults are bad in practice for two reasons. First, the training data and their features are usually very noisy, especially when these data come from Internet sources. That means the models would give unreliable scores in the middle of the spectrum. Second, even if the training data is not noisy, the training data are very likely to be imbalanced. That would cause the actual threshold to be shifted away from the default threshold. For these reasons, we need to find thresholds for each category to ensure that the predicted instances are reliable.

Definition 6. Predicted positive instances is a set S_p such that it's precision is larger than a desired constant p . The precision is defined as $\frac{|\text{true positives in } S_p|}{|S_p|}$.

We use a stratified sampling scheme to select the proper S_p . First we sort the prediction scores by the model M_c for all instances. Next we split them into b bins. For each bin, sample s instances for human to label. Assuming the percentage of correct instances in bin B_i is p_i , we select $S_p = \cup_{p_i \geq p} B_i$. The label effort would be $b \cdot s$ for each category. This could be reduced by not considering the bins coming after the first bin having $p_i < p$ in the sorted list. Note that classifier performance could be evaluated using the same p_i s: $M_p = \text{avg}_{p_i \geq p} p_i$.

6 Related Work

Ontologies filled with accurate and comprehensive instances have been proven useful in many applications, including dictionary construction for named entity recognition [7, 9], search query refinement [4], and automatic question answering systems [10]. The general methodology for ontology construction and population has been presented in [5], which includes the process of ontology population (both concepts and instances), ontology enrichment, inconsistency resolution, and evaluation. But to our knowledge, there is no publication focusing on populating large ontologies with accurate and comprehensive instances.

Since the birth of Wikipedia, it becomes possible to obtain automatically a general purpose knowledge base which contains millions of entities. Examples include DBpedia [1] and YAGO [8]. They used mapping heuristics to transform infobox and categories in Wikipedia into rdf:type relations and other types of relations between entities. When examining the data published by DBpedia or YAGO, one can soon find that the data is imperfect. The imperfection originates from mapping error or human error. Mapping error means that the heuristics used failed to map a feature in Wikipedia to the correct semantic type for an entity. Human error means that the human editor put a wrong feature or misused a feature. In SAPOP, these limitations are covered by our refinement stage that we utilize machine learning to reduce such issues.

In the machine learning field, many theories are discovered in the last two decades, which we rely on developing the practical framework for ontology population. The traditional researches of entity classification focus on finding instances for a few entity categories, e.g., person, organization, and places (e.g., [6]). Each category would require a large amount of labeled data, coupled with co-occurred text features; supervised learning methods could provide classifiers of good quality. In order to reduce the labeling effort, semi-supervised learning methods are studied in recent years. These methods add training data by automatically selecting reliable samples from unlabeled data, e.g., Co-Training, Multi-View Learning, and PU Learning [3]. However, how to apply the methods to the problem of ontology population is not well presented.

7 Future Work and Conclusion

In this paper, we presented a practical framework for ontology population. The framework is semiautomatic in that seed instances are determined semiautomatically using rule templates. Then semi-supervised learning methods are employed to expand the instances. Finally a parameter selection and evaluation process is described.

In the future, we will continue to refine the framework. Specifically, we will focus on optimizing the semi-supervised learning algorithm to better identify positive instances from the little amount of training data. Next we will improve our framework to reduce human effort when applied to a general purpose knowledge ontology, by taking into account the ontology structure, and other sources of training data, e.g., web pages and links between existing instances.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data the semantic web. In: Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007), pp. 722–735, 2007
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* **5**, 1–22 (2009)
3. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially supervised classification of text documents. In: ICML, p. 387–394
4. Pasca, M.: Acquisition of categorized named entities for web search. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, pp. 137–145, CIKM ’04
5. Petasis, G., Karkaletsis, V., Palioras, G., Krithara, A., Zavitsanos, E.: Ontology population and enrichment: state of the art. In: Palioras, G., Spyropoulos, C., Tsatsaronis, G. (eds.) *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, Lecture Notes in Computer Science, vol. 6050, pp. 134–166. Springer, Berlin/Heidelberg (2011)
6. Sriram, B., Fuhr, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering, pp. 841–842. SIGIR ’10
7. Stevenson, M., Gaizauskas, R.: Using corpus-derived name lists for named entity recognition. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 290–295. ANLC ’00
8. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. WWW ’07. ACM, New York, NY
9. Toral, A., Munoz, R.: A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. EACL (2006)
10. Wang, R.C., Schlaefer, N., Cohen, W.W., Nyberg, E.: Automatic set expansion for list question answering. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 947–954. EMNLP ’08
11. Zhang, B., Zuo, W.: Learning from positive and unlabeled examples: A survey, pp. 650–654. IEEE (May)

Exploring Information Flow Patterns Between News Portals and Microblogging Platforms

Bo Zhang, Jinchuan Wang, and Lei Zhang

Abstract News portals, CNN and SINA, for example, have been the most significant way of getting online information for a long time. However, the situation is being changed gradually when microblogging platforms are taking over. In this paper, we focus on SINA Weibo, the largest microblogging platform in China, and try to reveal the information flow patterns between Weibo and several major news portals. Specifically, we propose to use news source analysis to locate the very first origin of a given news and news flow analysis to reveal the pattern of its propagation among Weibo and portals.

Conclusions show that Weibo has become the leading media in Social News and Entertainment headlines. For serious news like Political and Military News, portals still play a more important role than Weibo, but Weibo is quickly catching up. This paper also shows how information flows in a network of Weibo and portals in different patterns with respect to different news categories.

1 Introduction

News portals have been the most significant online media channel until the recent tremendous boom of microblogging platforms came to challenge them with much faster speed.

Although there has been quite a lot of research work on information spreading and diffusion on homogenous social networks [1–9], little is known about how information flows around between microblogging sites and news portals.

B. Zhang • J. Wang • L. Zhang (✉)

Graduate School at Shenzhen, Tsinghua University, Shenzhen, China

e-mail: zhangbo0702@gmail.com; wjcthu@gmail.com; zhanglei@sz.tsinghua.edu.cn

Therefore, we try to answer the following questions in this paper:

1. Who has got the most significant impact with respect to news production and propagation, microblogging sites or new portals?
2. How is information replicated, altered, and propagated among microblogging sites and news portals?
3. Do the information flows show different patterns with respect to different news categories and different platforms?

The answers to the above questions are obtained in this paper by conducting an empirical study from data collected from SINA Weibo, China's largest microblogging platform, and ten other top news portals. Results show that Weibo is taking the lead in many daily-life-related news categories. News portals are still defending their traditional areas of Political and Military News collection and editing but Weibo is quickly catching up. Experiments also show that even in serious news categories where news portals are the major source of news origin, Weibo plays a quite important role in the information flow graph.

The rest of the paper is organized as follows. Data collection and processing details are shown in Sect. 2. News source identification algorithms and methods are presented in Sect. 3. Section 4 discusses our main findings on information flow patterns and provides answers to the previously mentioned three questions. Finally, we conclude the paper in Sect. 5.

2 Data

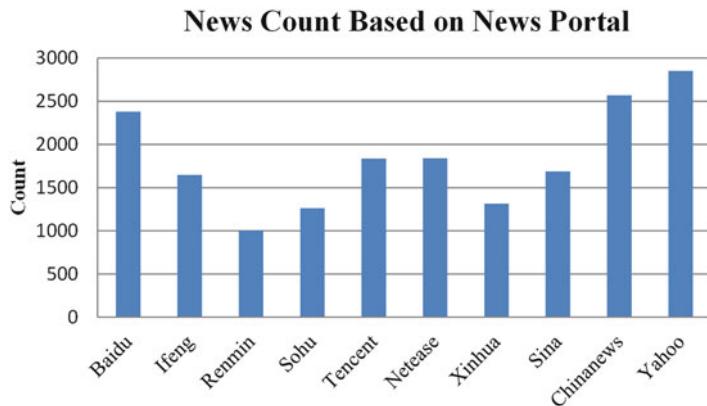
Our data collection and relevant preprocessing section consists of four parts: news portals, data collection, news classification, and detecting duplicated news. Each part is described in detail below.

2.1 News Portals

Currently, people usually obtain news information through browsing four big web portals (namely, Sina, Sohu, Netease, and Tencent) in China. However, news coverage from these four news portals is not much comprehensive sometimes. Therefore, to collect enough and comprehensive news data, we firstly need to consider more web portals which we collect news from. We choose ten web portals which simultaneously appear in news section of the whole site navigation websites which we refer to. We describe these ten web portals in Table 1.

Table 1 News portal name and URL

Name	URL	Name	URL
Baidu	http://www.news.baidu.com	Netease	http://www.news.163.com
Ifeng	http://www.news.ifeng.com	Xinhua	http://www.xinhuanet.com
Renmin	http://www.people.com.cn	Sina	http://www.news.sina.com.cn
Sohu	http://www.news.sohu.com	Chinanews	http://www.chinanews.com
Tencent	http://www.news.qq.com	Yahoo	http://www.news.cn.yahoo.com

**Fig. 1** The number of news from each web portal

2.2 Data Collection

The dataset spans a 10-day window from July 27, 2012, to August 6, 2012, and covers news information on Weibo and ten web news portals, as shown in Table 1. The dataset was obtained and parsed using the Jsoup toolkit [10], a HTML processor based on Java library. Totally, our dataset contains 19K pieces of news information. In Fig. 1, we show the number of news from each web portal. We can see that the number of news from Baidu, Chinanews, and Yahoo is relatively more, while volume of news from Renmin, Sohu, and Xinhua is comparatively less.

2.3 News Classification

As our fundamental goal is to explore the role of Weibo in news information flow according to different types of news, it is necessary for us to develop a classification schema to classify news we obtain from web portals. Our classification schema consists of two basic principles: on the one hand, news categories we choose are contained by all of the ten news portals, while on the other hand, they are either

Table 2 News category and corresponding example

Category	Example
Domestic	Chongqing police reveals details
Social	Children adoption investigation
Sport	NBA compares top five centers of Lakers
Entertainment	New entertaining photos of celebrities
Financial	Oil price rocketing
Tech	Yahoo COO: becoming the No. 2
Military	China making new jets with Ukraine

closely related to people’s daily life or relatively serious. We classify news on the basis of news categories shown in Table 2. The examples in the second column are from Sina which is a very popular news portal.

2.4 Detecting Duplicated News

There are various studies on duplicate detection. Broder [11] presented shingling method for detecting similar documents. In that paper, he used a “sketch” of “shingles” to represent each document and computed the resemblance of two documents. Charikar [12] developed *simhash* and used it to reduce dimension. Manku et al. [13] expanded Charikar’s work further and validated that Charikar’s *simhash* was practically useful for identifying near-duplicates in web documents belonging to a large-scale web page repository.

Simhash is a fingerprint technique enjoying the property that fingerprints of near-duplicates differ only in small number of bit positions. If the *simhash* fingerprints of two documents are similar, they are deemed to be near-duplicates. *Hamming distance* [14] is an appropriate method to compute the number of different corresponding bits in two bit strings.

In the paper, we utilize Charikar’s *simhash* to check duplicated news. First, we compute the value of *simhash* for each news content and then estimate the similarity of contents of each pair of news through computing *hamming distance* for each *simhash* pair. Comparison result before and after we use *simhash* to do duplicate detection is displayed in Fig. 2. From the figure, we can see that the number of different types of news is significantly reduced after we use *simhash* to check duplicated news. It means each type of news information all has plenty of redundancy. To further reveal the degree of redundancy of news information for each type, we calculate the ratio of redundancy. Specific redundancy ratios are showed in Fig. 3. We can see that all of ratios are more than 30%, except for the ratio of Tech News which is the minimum and just below 30%. Surprisingly, the Sport News is highest in terms of redundancy ratio and over 50%. It may be due to the data collection time when London is hosting 30th Olympic Games.

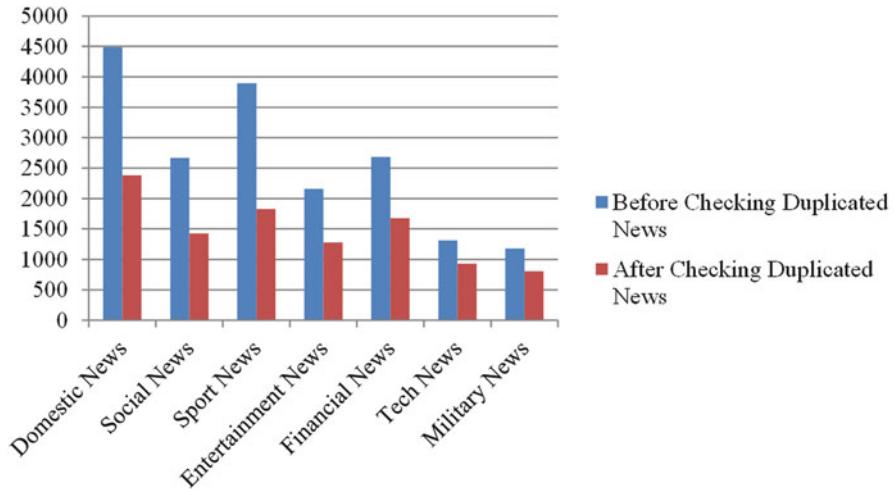


Fig. 2 The number of news for each type of news before and after checking duplicated news, respectively

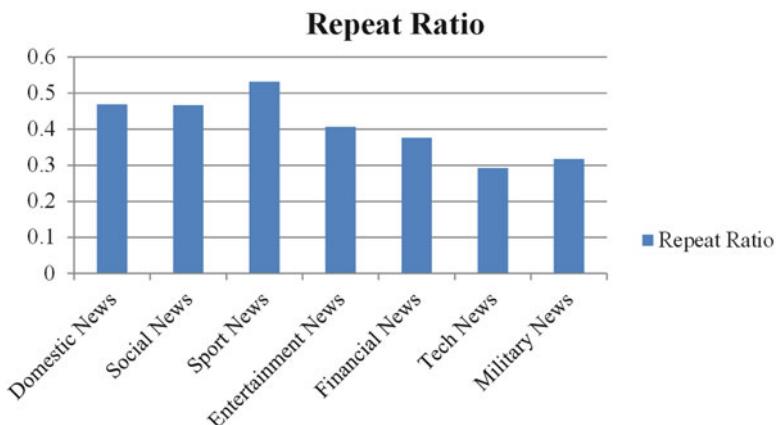


Fig. 3 Redundancy ratio of news for each type of news

3 News Source Analysis

The section mainly centers on news source analysis. Namely, for each type of news, we mainly focus on answering these questions: what is the number of news source? What is the distribution of news source? What is the ratio of each news source in the whole news source dataset of each type of news? We use the answers of these three questions as the empirical evidence of the role of Weibo in news information flow.

Table 3 News source count for each type of news

News category	News source count
Domestic News	217
Social News	174
Sport News	109
Entertainment News	117
Financial News	196
Tech News	128
Military News	95

3.1 News Source Count

We apply statistics to the number of different news source for each type of news, which can be regarded an answer to the first question. Table 3 shows the statistic result. The type of news which has the most news sources is Domestic News, while the least is Military News; their news source counts are 217 and 95, respectively. The average news source count is 148. From the point of view, it is evident that the range of news source is relatively wide.

3.2 News Source Distribution

We explore the distribution of times news source that is referenced in other news media. Through this, we want to learn about the number of news sources whose times referenced are relatively higher and lower, respectively. Figure 4 shows the distribution of news source from each type of news. We can see that except for specific numerical numbers in x-axis and y-axis, styles of curves in seven graphs from Fig. 4 are almost the same. Obviously, times that news source is referenced in other news media have the property of long tail. In other words, the quantity of news sources whose times referenced are relatively higher, however, is comparatively less, while the quantity of news sources whose times referenced are relatively lower is comparatively more.

3.3 News Source Ratio

In this part, we mainly concentrate on exploring the role of Weibo as a news source in different types of news information flow by calculating the ratio which times Weibo is referenced account for in the whole news sources.

More specifically, for each type of news, we first compute ratios of times referenced of each news source and then sort the ratios in descending order. We show ratios ranking graphically and choose the first ten news sources to display in Fig. 5.

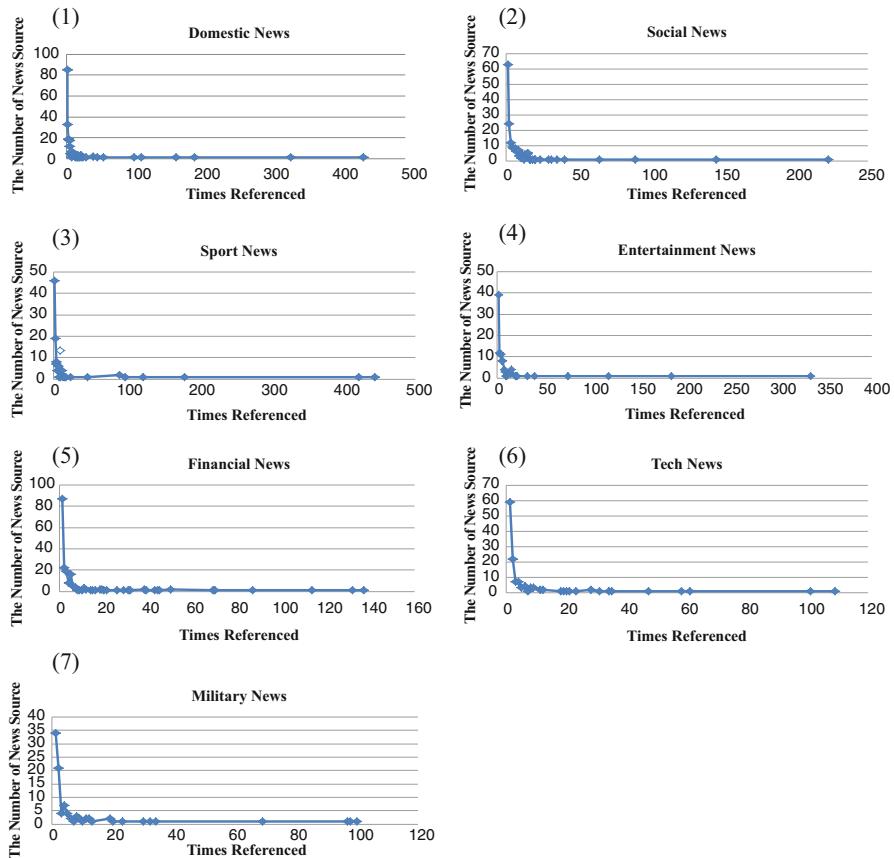


Fig. 4 Distributions of times referenced for each type of news. (1) Domestic News, (2) Social News, (3) Sport News, (4) Entertainment News, (5) Financial News, (6) Tech News, and (7) Military News

Clearly, except for graph 5.7 of Fig. 5 where Weibo is not shown in text section of the graph, the ratio of times that Weibo is referenced in other media is ranked in top 10 in the other graphs. In addition, to further explore the role which Weibo plays in different types of news vividly, we show the ranking of times referenced of Weibo in another way; namely, we generate a table (Table 4) which shows the rank of Weibo in each type of news on the basis of these seven graphs in Fig. 5.

From Table 4, it is manifest that in terms of acting as a news source, Weibo's rank is comparatively high in almost all of types of news, especially in Social News, Entertainment News, and Tech News where Weibo ranks first. Namely, Weibo has become the biggest news source for Social, Entertainment, and Tech News; it is the third biggest news source for Domestic as well as Financial News and the fifth biggest news source for Sport News. From the perspective of the role of Weibo in

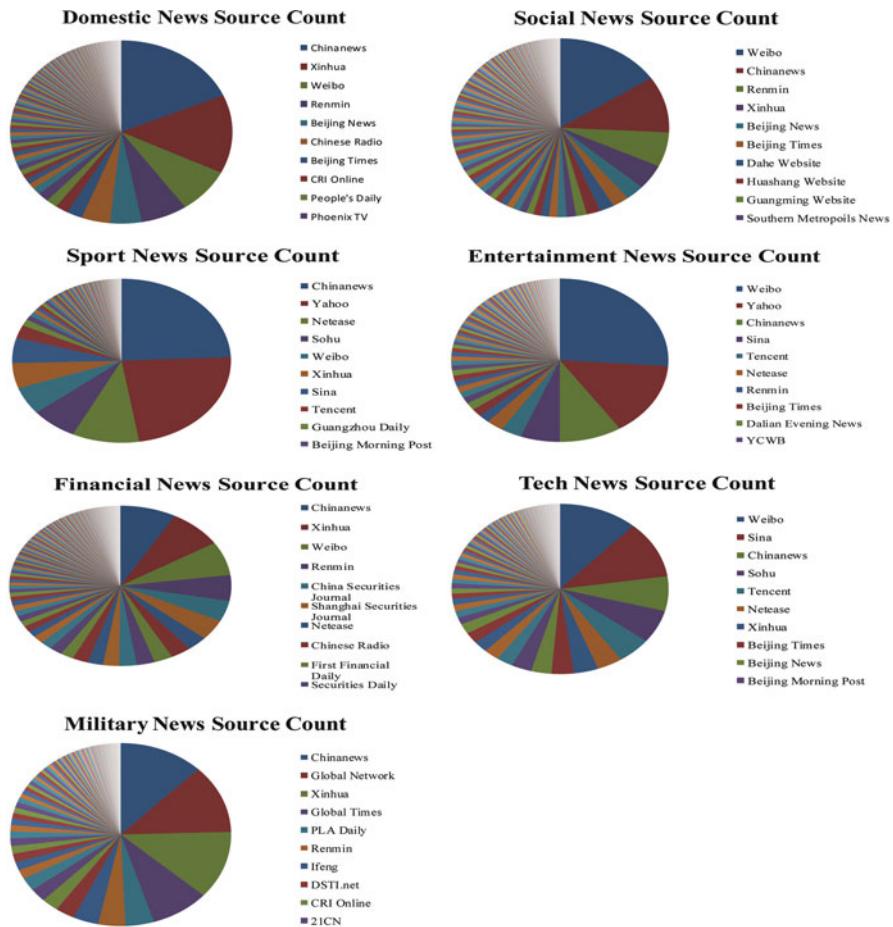


Fig. 5 News source ratio for each type of news. (1) Domestic News, (2) Social News, (3) Sport News, (4) Entertainment News, (5) Financial News, (6) Tech News, and (7) Military News

Table 4 Weibo ranking for each type of news

News category	Weibo ranking
Domestic News	3
Social News	1
Sport News	5
Entertainment News	1
Financial News	3
Tech News	1
Military News	25

different types of news information flow, Weibo plays a very important role in acting as a news source for Social, Entertainment, and Tech News, while the role of Weibo is relatively clear in Domestic, Sport, and Financial News information flow as well. However, in Military News, Weibo's rank is relatively low and only in 25. So, the role of Weibo is not much important for Military News.

Observations from the previous paragraph well validate the two hypotheses we proposed in the beginning of the paper. People tend to post and repost information related to society and entertainment which are closely linked with people's daily life in Chinese microblogging service Weibo. Therefore, Social and Entertainment News information flow is more likely to start from Weibo and flow to traditional news media. However, it is difficult for common people to contact message related to military in comparison with message types mentioned above. This results in people hardly discussing or spreading military information in Weibo. Therefore, flow of Military News information usually comes from professional news media in the aspect of reporting Military News.

4 News Flow Analysis

News information flow analysis involves the question: how does news information flow from one website to another website? In this section, what we take to deal with this question is the way that we show each type of news information flow between websites graphically. Through this way, we can learn about the role of Weibo in news information flow more easily.

Gephi [15] is a powerful and open source graph visualization and manipulation tool. Except for vividly drawing network graph, it can detect community among network and do some statistics, e.g., calculating average degree and network diameter and achieving *PageRank* [16] algorithm. In addition, it can accept as input data source files of multiple formats, such as files with extensions like gexf, gdf, and csv. In our experiment, we use it to display news information flow between websites graphically and use files with extension gexf as our input data files.

In our experiment, we generate gexf file for each type of news and use these seven data files to draw seven network graphs which are shown in Fig. 6. In the figure, text description on one node is news source related to the node. Node's color stands for community which nodes with the same color belong to, while node's size represents node's out-degree, namely, times which node related to news source is referenced by other websites; edge's size means edge's weight, namely, times referenced between nodes which are linked by the edge each other. As to edge's color, if some edge's color is the same color as both nodes which are connected by the edge, it means these two nodes belong to the same community; on the other hand, if some edge's color is different from color of both nodes which are linked by the edge, it indicates that these two nodes come from two different communities. In addition, node's position in each graph indicates some message. In other words, the more centered a node is, the more important the node is.

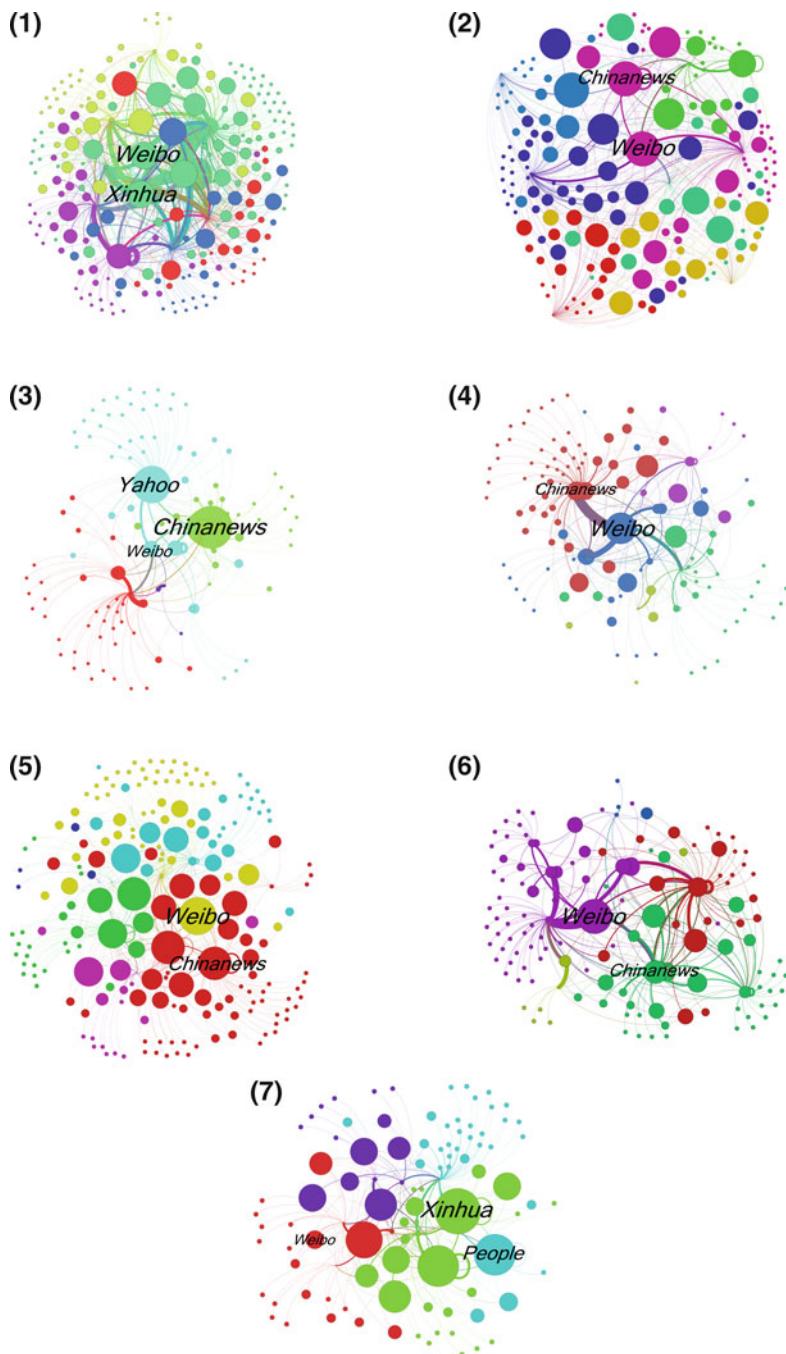


Fig. 6 News information flow for each type of news. Node represents news source, while edge stands for news information flow between two nodes which are linked by it. (1) Domestic News, (2) Social News, (3) Sport News, (4) Entertainment News, (5) Financial News, (6) Tech News, and (7) Military News

Therefore, combining with descriptions stated in the previous paragraph, we can use node's size, node's position in the graph, and color information of edges which connect with the node to evaluate importance of the node in the graph, namely, the role news source related to the node plays in the type of news. Certainly, our main goal is to evaluate the role of Weibo, so we just need to observe the node whose text description is Weibo, which greatly simplifies our work.

Firstly, we analyze size of node Weibo in these seven graphs of Fig. 6. We can see that size of node Weibo is the biggest in graphs of Entertainment News, Social News, and Tech News. In addition, it is comparatively bigger in graphs of Domestic News, Financial News, and Sport News. However, it is relatively smaller in the graph of Military News. This is almost the same result as news source analysis, because size of node in Fig. 6 is based on out-degree of the node.

And then, we analyze position distribution of node Weibo. In accordance with size analysis, node Weibo is located in the center in graphs of Entertainment News, Social News, and Tech News, while it is far away from center in the graph of Military News. However, its size is not the biggest in graphs of Domestic News, Financial News, and Sport News, while its position is the most centered in these three graphs.

Finally, we analyze color information of edges which connect node Weibo. Apparently, it is especially abundant in terms of kinds of color in almost all of graphs, except for in the context of Military News where color information of edges is relatively single.

Through analysis from previous several paragraphs, we can see that role of Weibo is clear in Entertainment News, Social News, Tech News, Domestic News, Financial News, and Sport News information flow. In other words, Weibo plays an important role in these types of news. On the contrary, in Military News, its role is not as clear as in these six types of news. It validates the two hypotheses we mentioned in the abstract section once again. Namely, Entertainment News and Social News tend to flow from Weibo to traditional news media, while Military News tends to flow from traditional news media to Weibo.

5 Conclusion

The goal of the paper is to explore the role of Weibo in different types of news information flow. We first propose two hypotheses that Social and Entertainment News which are closely related to people's life tend to flow from Weibo to traditional news media, while Military News which is relatively serious, professional, and far away from people's life tends to flow from traditional news media to Weibo. Then, data collection and preprocess are involved as well. Especially, duplicate news detection is discussed in detail and we combine *simhash* with *hamming distance* to check duplicated news which we crawl from ten web portals. Moreover, we explore the role of Weibo as a news source in news source analysis. In that section, discussions center on three aspects of news source count, news source distribution, and news

source ratio. Finally, we analyze news flow and show the result graphically. We evaluate importance of node Weibo through its size, its position in each graph, and color information of edges which it connects.

The following is the list of our main findings:

- Weibo is the biggest news source for Social News, Entertainment News, and Tech News.
- For Domestic News, Sport News, Financial News, and Military News, news portals are the biggest news sources.
- Weibo ranks in the top five news sources for Domestic News, Sport News, and Financial News.
- Weibo only places the 25th in news sources of Military News.
- News redundancy in all news categories is much higher than expected, with more than 30% of redundancy ratio. The Sport News category redundancy ratio is even more than 50%. This indicates that most news portals tend to duplicate or refer news reports from other platforms rather than producing novel reports by themselves.

References

1. Li, Y.M., Shiu, Y.L.: A diffusion mechanism for social advertising over microblogs. *Decision Support Systems* (2012)
2. Wan, X., Yang, J.: Learning information diffusion process on the web. In: *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 1173–1174
3. Yang, J., Leskovec, J.: Modeling information diffusion in implicit networks. In: *Proceedings of the 10th IEEE International Conference on Data Mining*, 2010, pp. 599–608
4. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: Learning influence probabilities in social networks. In: *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, 2010, pp. 241–250
5. Romero, D.M., Galuba, W., Asur, S., Huberman, B.A.: Influence and passivity in social media. In: *Proceedings of the 20th International Conference on World Wide Web*, 2011, pp. 113–114
6. Macskassy, S.A., Michelson, M.: Why do people retweet? Anti-homophily wins the day!. In: *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, 2011, pp. 209–216
7. Najar, A., Denoyer, L., Gallinari, P.: Predicting information diffusion on social networks with partial knowledge. In: *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 1197–1204
8. Ratkiewicz, J., Conover, M., Meiss, M., Goncalves, B., Patil, S., Flammini, A., Menczer, F.: Detecting and tracking the spread of astroturf memes in microblog streams. *arXiv preprint arXiv:1011.3768* (2010)
9. Holthoefer, J.B., Rivero, A., Moreno, Y.: Locating privileged spreaders on an online social network. *Physical Review E* 85.6 (2012): 066123
10. Hedley, J.: Jsoup: Java HTML Parsel. <http://jsoup.org/>. (2009)
11. Broder, A.Z.: On the resemblance and containment of documents. In: *Proceedings of Compression and Complexity of Sequences*, 1997, pp. 21–29
12. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, 2002, pp. 380–388

13. Manku, G.S., Jain, A., Sarma, A.D.: Detecting near-duplicates for web crawling. In: Proceedings of the 16th International Conference on World Wide Web, New York, 2007, pp. 141–150
14. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. (1950)
15. Bastian, M: Gephi: The open graph viz platform. <http://gephi.org/>. (2008)
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)

A Two-Step Non-redundant Subspace Clustering Approach

Hai-Tao Zheng, Hao Chen, Yong Jiang, Shu-Tao Xia, and Huiqiu Li

Abstract To detect clusters in high-dimensional database, researchers have proposed several approaches which detect clusters in different subspaces. These approaches are called subspace clustering approaches. However, most of the proposed subspace clustering approaches produce a redundant result which leads to a low clustering quality. Recently, several non-redundant subspace clustering approaches are proposed such as OSCLU (Günnemann et al., Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 1317–1326. ACM, 2009). For the purpose of efficient calculation, these non-redundant subspace clustering approaches, which are mainly based on an NP-hard mathematics model, introduce a lot of parameters. By doing experiments, we find that the existing non-redundant subspace clustering approaches are sensitive to parameter settings.

To solve the parameter setting problem of existing non-redundant subspace clustering, in this paper we propose a subspace clustering approach based on a two-step clustering model. With experiments, we prove that our approach can outperform OSCLU by reducing the parameter's number and sensitivity.

H.-T. Zheng (✉) • H. Chen • Y. Jiang • S.-T. Xia
Tsinghua-Southampton Web Science Laboratory, Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
e-mail: zheng.haitao@sz.tsinghua.edu.cn; jerrychen1990@gmail.com;
jiangy@sz.tsinghua.edu.cn; xiast@sz.tsinghua.edu.cn

H. Li
China Telecom Co., Ltd. Shenzhen Branch
e-mail: 13360097990@189.cn

1 Introduction

Knowledge discovery in database provides data owners with new information and patterns in their data. Clustering is a traditional data mining task for automatically grouping objects [3]. Sometimes, clusters may be hidden in different views of data. In these situations, database researchers have to apply a multiple clustering method. Generally, multiple clustering provides multiple sets of clusters and thus gets more insights than only one solution. There are three goals of multiple clustering: first of all, each object should be grouped in multiple clusters and represents different perspectives on the data. Secondly, the result should consist of many alternative solutions. Users can choose one or use multiple of these solutions. At last, solutions should differ to a high extent and thus each of these solutions provides an additional knowledge.

Some recent researches of multiple clustering focus on subspace (or projection) clustering approaches which detect clusters in a subset of the whole attributes set. They group subspaces and objects to deal with the curse of dimension [6]. A potential problem is that subspace clustering approaches will introduce redundancy and multiple views.

To deal with the redundancy problem, several non-redundant subspace clustering approaches are proposed. These approaches are based on the idea that a pair of subspace clusters which share more than a certain amount of overlapped subspaces and objects should be regarded as redundant to each other. With this redundancy definition, researchers proposed an algorithm which detects all the clusters in different subspaces gradually and only adds non-redundant clusters to the result set. However, this algorithm is proved to be NP hard, so they proposed OSCLU, an approximation algorithm. For efficient calculation, OSCLU introduces a lot of parameters. With experiments we find that OSCLU is sensitive to parameter settings. Besides, the model which adds clusters to the result set gradually makes OSCLU cannot judge clusters' redundancy in a global view which affects redundancy removal quality a lot.

In this paper, we propose a non-redundant subspace clustering approach. Totally different from the existing approaches, we generate a non-redundant subspace clustering result by two steps. First step, we generate a redundant subspace clustering result with some traditional subspace clustering approaches, e.g., SUBCLU [13]. Second step, we consider each subspace cluster as an object, with their subspaces and grouped objects as attributes. We use some traditional clustering approaches, e.g., K-Means [12], to group these clusters generated on the first step. We consider the clusters that are grouped in one group to be similar and thus provide redundant information. So, we pick one cluster to represent this group of clusters. All these picked representor clusters compose the result set. Except the parameters for density-based clustering, we only introduce one parameter K , which determines the result cluster number and can be easily set in a heuristic way. Comparing to a newly proposed non-redundant multiple clustering approach OSCLU, our approach TSCLU mainly have 3 advantages as listed below:

1. TSCLU introduces less parameters and is less sensitive to the parameter settings.
2. TSCLU gets a lower redundancy value than OSCLU and a similar clustering quality measured by traditional evaluating criterion.
3. TSCLU removes redundant clusters with a global which is more reasonable than OSCLU.

To prove that TSCLU can outperform OSCLU in some perspectives, we do a series of experiments to compare TSCLU with OSCLU under different clustering quality evaluation criterions like F1 [5, 15], Entropy [4, 19], Accuracy [8, 20], and Coverage. Since there are no existing redundancy evaluation measures, we also propose a redundancy evaluation criterion.

This paper is structured as follows: in Sect. 2 we review existing multiple clustering approaches and existing subspace clustering evaluation measures. Section 3 will introduce our novel non-redundant subspace clustering algorithm. We will do a series of experiments to prove our approach can outperform OSCLU with less parameters and lower redundancy value in Sect. 4. Finally, we make a conclusion and present our future work in Sect. 5.

2 Related Work

2.1 Subspace Clustering Approaches

Recent research of clustering in high-dimensional data has introduced a number of different approaches summarized in [14, 17]. One approach is to detect clusters in arbitrary subspace projections of the data. Each cluster is associated with a set of relevant dimensions in which this pattern has been discovered. In the view of redundancy of the result set, we divide subspace clustering approaches into two categories: redundant approaches and non-redundant approaches.

2.1.1 Redundant Approaches

We can identify three major paradigms of redundant subspace clustering approaches by the underlying cluster definition and parameterization of the resulting clustering:

Cell-Based Approaches: Cell-based approaches consider clusters as a set of adjacent fixed or variable grid cells that contain more than a certain threshold many objects. A famous cell-based approach is CLIQUE [2], which uses fixed threshold, fixed grid, and prune subspaces by monotonicity property. Based on this idea, researchers also proposed DOC [7], MINECLUS [21], and SCHISM [19].

Density-Based Approaches: Density-based approaches define clusters as dense regions separated by sparse regions. With this definition, we can detect arbitrary shaped clusters in relevant dimensions. The first density-based clustering algorithm

is DBSCAN [9]. SUBCLU [13] applies DBSCAN to subspaces. Density-based subspace clustering approaches all suffer the parameter setting problem and have to deal with dimensionality bias of density while detecting clusters in subspaces.

Clustering-Oriented Approaches: In contrast to the previous approaches, clustering-oriented approaches focus on the clustering result R by directly specifying objective functions like the number of clusters to be detected or the average dimensionality of the clusters, as in PROCLUS [1]. As clustering-oriented approaches directly control the resulting clusters other than an individual cluster, clustering quality is affected by noisy data.

2.1.2 Non-redundant Approaches

Traditional redundant subspace clustering approaches tend to generate a quite large amount of clusters. The result contains a lot of redundant clusters. OSCLU is a recent proposed non-redundant subspace clustering approach. OSCLU is based on the idea that a pair of subspaces which share more than a certain amount of overlapped subspaces and objects should be regarded to be redundant to each other. ASCLU [10] applies this idea to alternative clustering way.

2.2 Subspace Clustering Evaluating Framework

In this section, we review some existing evaluation measures in subspace clustering area. Müller et al. [16] introduced a systematic framework to evaluate subspace clustering results. We mainly divide the existing measures into two categories: object-based measures and object- and subspace-based measure.

We use the classic measures in traditional clustering evaluation such as *Entropy*, *F1*, and *Coverage* to evaluate clustering result in object view.

As we are doing subspace clustering, we should take subspace selection into consideration. So we also get some evaluation measures that divide each object into each dimensions and evaluate the result based on the view of divided objects, e.g., RNIA [18] and CE [18].

As reviewed, we have already got many measures describing the quality of clustering. But we missed an important aspect of subspace clustering: the redundancy of results. Many clustering approaches devote themselves to redundancy removal. However, they just concentrate on the algorithms and use some observation ways to show that the result is not redundant. So we still lack a redundancy-evaluating criterion to measure whether an algorithm is doing good at redundancy removal.

3 TSCLU

Notations: For consistent notations in the following sections we define some notations here. Every cluster C in a subspace projection is defined by a set of objects O that is a subset of the database $DB = \{O_1, O_2, \dots, O_n\}$ and by a set of relevant dimensions S out of the set of all dimensions $D = \{S_1, S_2, \dots, S_m\}$.

Definition 1. A cluster C in a subspace projections S is

$$C = (O, S) \text{ with } O \subseteq DB, S \subseteq D \quad (1)$$

A clustering result is a set of found clusters in the respective subspace projections.

Definition 2. A clustering result RES of k clusters is a set of clusters

$$RES = \{C_1, \dots, C_k\}, C_i = (O_i, S_i) \text{ for } i = 1 \dots k \quad (2)$$

Let All be the set of all possible subspaces of cluster C . Apparently, All contains lots of subspace clusters and can be quite redundant. We need to find an optimal subset of All to minimize redundancy while keeping a certain clustering quality. Generally, we do our job in two steps. First step, we generate as many clusters as possible in all the subspaces. Second step, we remove redundancy of the result by grouping the clusters generated on the first step. As to a group of similar clusters, we consider them to be similar and thus are redundant to each other. So, we retain a most valuable cluster and remove all the other clusters because they bring redundancy. Because of the two-step clustering, we call our approach TSCLU (two-step subspace clustering).

3.1 First-Step Clustering

While doing the first-step clustering, we do not put importance on the result set's redundancy but on the coverage. To compare with OSCLU, we also choose a density-based subspace clustering approach. We choose SUBCLU to do the first-step clustering because it can detect subspace clusters with arbitrary shapes in all subspaces.

3.2 Second-Step Clustering

To remove redundancy, we also adopt the idea in OSCLU that a pair of subspaces which share more than a certain amount of overlapped subspaces and objects should

be regarded as redundant to each other. Instead of setting two subjective overlap rate thresholds, we use traditional clustering algorithms to group similar subspace clusters together. By this way, we can group similar subspace clusters together with a global view. After that, we pick one most interesting subspace cluster from each group and add them to the result set.

3.2.1 Second-Step Clustering Algorithm

To reduce the time consumption and parameter number of the second-step clustering, we choose K-Means to group the subspace clusters generated on the first step. We apply a vector-building algorithm to convert a subspace cluster C to two vectors, $V_o(C)$ and $V_s(C)$.

Definition 3. Considering a subspace cluster $C=\{O, S\}$, we can convert it to a vector $V_o(C)$ in object perspective.

$$V_o(C) = \{V(O_1), V(O_2), \dots, V(O_n)\} \quad V(O_i) = \begin{cases} 1 & : O_i \in O \\ 0 & : \text{otherwise} \end{cases} \quad (3)$$

Definition 4. Considering a subspace cluster $C=\{O, S\}$, we can convert it to a vector $V_s(C)$ in subspace perspective.

$$V_s(C) = \{V(S_1), V(S_2), \dots, V(S_m)\} \quad V(S_i) = \begin{cases} 1 & : S_i \in S \\ 0 & : \text{otherwise} \end{cases} \quad (4)$$

To calculate the distance between two subspace clusters, we should take both grouped objects and relevant dimensions into consideration. So, we plus the cosine distance of object vectors and subspace vectors to be subspace cluster distance.

Definition 5. Given two subspace clusters $C_i = \{O_i, S_i\}$, $C_j = \{O_j, S_j\}$. We define the distance between C_i and C_j to be $Dis(C_i, C_j)$.

$$Dis(C_i, C_j) = \text{Cosine}(V_o(C_i), V_o(C_j)) + \text{Cosine}(V_s(C_i), V_s(C_j)) \quad (5)$$

3.2.2 Cluster Picking

After grouping similar subspace clusters together, we have to choose a representative subspace cluster from each group to represent the group. Considering clusters in one group provide similar information, we want to pick a most interesting one. We make use of the local interesting definition in [16] to define our interesting value:

Definition 6. Given a subspace cluster $C=\{O, S\}$, we define the Interest value of C as below:

$$Interest(C) = |S|^a \cdot |O|^b \cdot \left(\frac{1}{|O|} \sum_{p \in O} density^S(p) \right) \quad (6)$$

$density^S(p)$ stands for the average density of cluster C , which is defined in [16]. By picking the one with the highest interesting value to represent each cluster group and add it to the result set, we get the final non-redundant subspace clustering result.

4 Experiments

4.1 Redundancy Evaluation

As stated in Sect. 2, many clustering approaches devote themselves to non-redundant results. However, they just concentrate on the algorithms and use some observation ways to show that the result is not redundant. So we still lack a redundancy-evaluating criterion to measure whether an algorithm is doing well at redundancy removing or not. In this section we propose a redundancy evaluation criterion and use this criterion to evaluate the redundancy of a clustering result in the following experiments.

4.1.1 Redundancy Definition

Based on the idea in [10] and [11], we consider subspace cluster pairs that share little common subspace or grouped objects to be dissimilar. We can define the redundancy value between two subspace clusters with the product of the subspace overlap rate and object overlap rate.

Definition 7. [Mutual redundancy value] Given two subspace clusters $C_i=\{S_i, O_i\}$ and $C_j=\{S_j, O_j\}$, the mutual redundancy value of C_i and C_j is:

$$Red\{C_i, C_j\} = \frac{|O_i \cap O_j|}{|O_j|} \times \frac{|S_i \cap S_j|}{|S_j|} \quad (7)$$

Based on the mutual redundancy formula above, we can define the total redundancy value of a result set RES. An apparent way is to sum all the mutual redundancy up.

Table 1 Redundancy value of different approaches

ALGORITHM	CLIQUE	SUBCLU	OSCLU
CLUSTER NUM	664	515	12
REDUNDANCY	34554.44	46165.08	17.34

Definition 8 (Total redundancy value). Given a set of subspace clusters $RES = \{C_1, C_2, C_3, \dots, C_n\}$. The total redundancy value $TRed(RES)$ is:

$$TRed(RES) = \sum_{i=1}^n \sum_{j=1}^n Red(C_i, C_j) \times |sign(i - j)| \quad (8)$$

4.1.2 Reasonableness

To prove that our redundancy-evaluating criterion is reasonable, we do experiments on real-world data *glass*, which contains 214 objects in 9 dimensions. We apply three subspace clustering approaches including two redundant approaches CLIQUE and SUBCLU and one non-redundant approach OSCLU. The parameters are adjusted to get the best clustering result. After that we evaluate the redundancy of the results. All the experiments are based on an open-source subspace clustering framework OpenSubspace <http://dme.rwth-aachen.de/en/OpenSubspace>. We implemented OSCLU on this framework and compared clustering result to CLIQUE and SUBCLU, which are already implemented on the framework. We list the experiment result below:

Table 1 shows the total redundancy and cluster number of three different approaches.

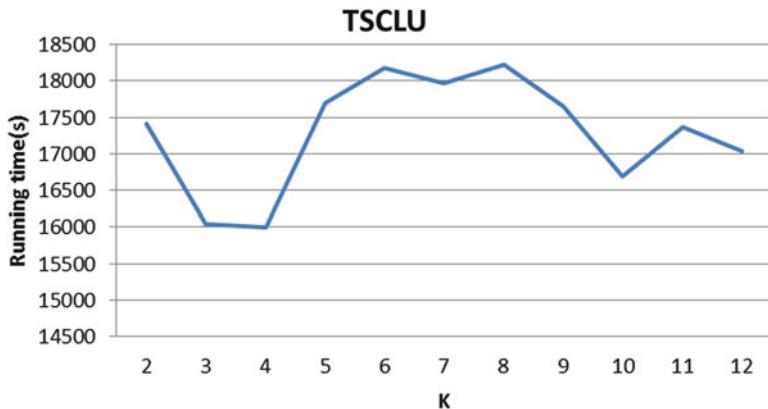
Apparently, OSCLU generated less number of clusters and got a significant reduction of total redundancy value. Although the redundancy value is seemed to be quite biased to cluster sets with smaller cluster number, we have pointed out that a large cluster number itself can be regarded as a kind of redundancy. So this criterion is reasonable and we can apply this criterion to the coming experiments. However, we do admit that the cluster number may play a too important role while evaluating the redundancy of the clustering result. It's better to normalize the redundancy value to be bounded in [0,1], e.g., we can take the answer file's information into consideration. In the future, we will work on this.

4.2 Comparison Between OSCLU and TSCLU

Generally speaking, OSCLU suffers from 3 main drawbacks: (1) OSCLU introduces too many parameters which make the algorithm hard to implement. (2) OSCLU is quite sensitive to the parameter settings and different parameter values may lead to a significant difference of result quality. (3) OSCLU considers the redundancy of a

Table 2 TSCLU clustering quality with different K (cluster number, from 2 to 12)

K	2	3	4	5	6	7	8	9	10	11	12
Cluster number	2	3	4	5	6	7	8	9	10	11	12
Running time (s)	17.4	16.0	15.9	17.7	18.1	17.9	18.2	17.6	16.7	17.3	17.0
Redundancy	0.27	0.39	0.77	1.26	1.65	2.20	2.92	3.55	3.73	4.10	4.58
F1	0.13	0.15	0.15	0.19	0.18	0.18	0.20	0.18	0.24	0.22	0.24
Accuracy	0.36	0.43	0.43	0.44	0.45	0.45	0.47	0.46	0.47	0.47	0.46
Entropy	0.40	0.33	0.37	0.36	0.32	0.33	0.35	0.31	0.32	0.33	0.29
Coverage	0.49	0.73	0.80	0.82	0.94	0.94	0.92	0.97	0.99	0.99	0.99

**Fig. 1** Running time vs K

new coming cluster gradually instead of considering redundancy with a global view. This may make the result not optimal.

On the other hand, the parameters of TSCLU are quite simple and the two-step clustering model can remove redundancy with a global view. To prove that TSCLU can outperform OSCLU in perspectives above, we do a series of experiments on real-world data. All the experiments are based on OpenSubspace framework.

4.2.1 Parameter Sensitivity

The same as the experiments before, we do our experiments on database *glass*, which contains 214 objects and 9 dimensions. First of all, we test the clustering quality of TSCLU with different parameters K . We set parameters $\text{Epsilon} = 8.0$ and $\text{minPt} = 32$ for the first-step SUBCLU clustering.

From Table 2 and Figs. 1, 3, and 5, we can see that TSCLU is not sensitive to the K value. As K 's value grows, the redundancy grows linearly. Coverage grows fast before $k = 6$, after that coverage does not change a lot. Other quality measures like

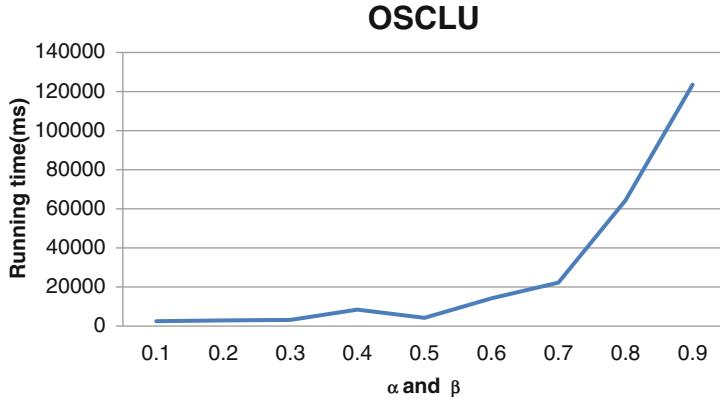


Fig. 2 Running time vs α, β

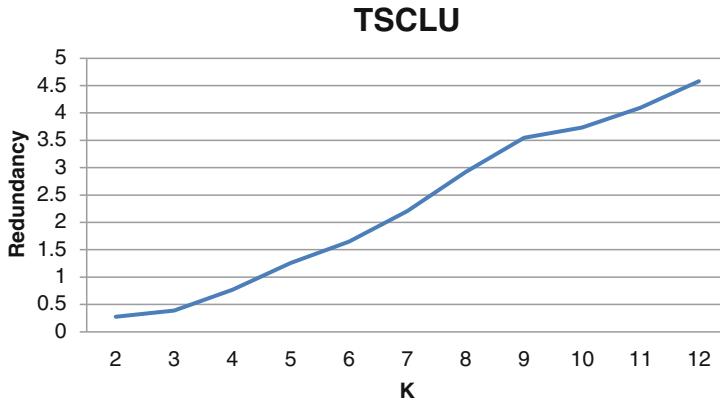
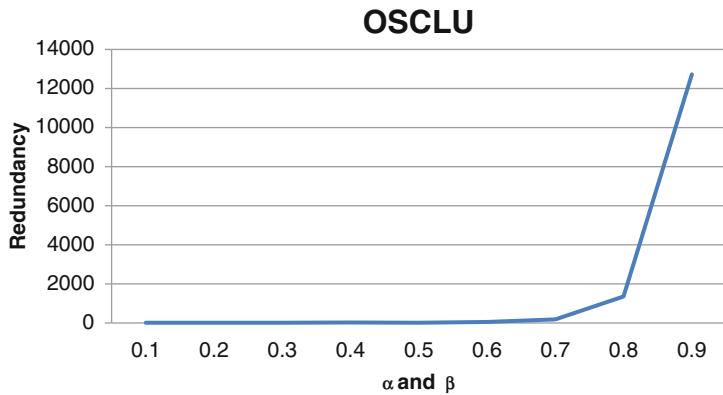
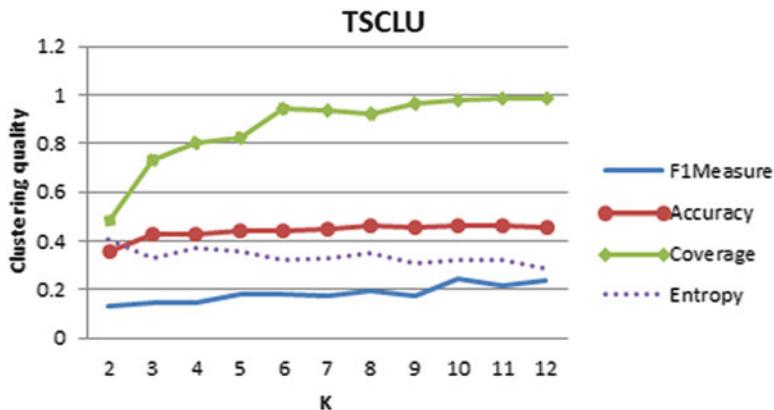


Fig. 3 Redundancy vs K

F1, Entropy, and time consumption are not sensitive to K . So, while implementing TSCLU, we can just set $k = \log(n)$. n is the size of database.

To test OSCLU's parameter sensitivity, we compare clustering quality with different α and β . In fact, we also have to set orthogonal estimation threshold, quality upper bound, quality lower bound, and seed number subjectively to implement OSCLU. To simplify the experiment, we use the best estimation threshold, quality upper bound, lower bound, and seed number value and just compare the quality with different α and β .

From Table 3 and Figs. 2, 4, and 6, we can see that OSCLU is quite sensitive to α and β 's value. As α and β 's value grows, both redundancy and runtime grow exponentially. Clustering quality also changes a lot with the α and β 's value. Because of the paper page number's limitation, we do not show the result of OSCLU with changes of other parameters here.

**Fig. 4** Redundancy vs α, β **Fig. 5** Clustering quality vs K**Table 3** OSCLU clustering quality with different α and β

α and β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Cluster number	15	15	10	12	21	41	59	166	402
Running time (s)	2.5	2.9	3.1	8.4	4.2	14.2	22.2	64.3	8.1
Redundancy	3.36	3.34	5.20	18.73	10.37	42.26	173.17	1344.50	12721.21
F1	0.24	0.28	0.23	0.31	0.39	0.45	0.49	0.51	0.51
Entropy	0.25	0.26	0.32	0.39	0.38	0.44	0.48	0.45	0.49
Accuracy	0.47	0.52	0.48	0.48	0.47	0.51	0.46	0.47	0.47
Coverage	1.0	1.00	0.97	0.97	0.90	0.96	0.92	0.97	0.91

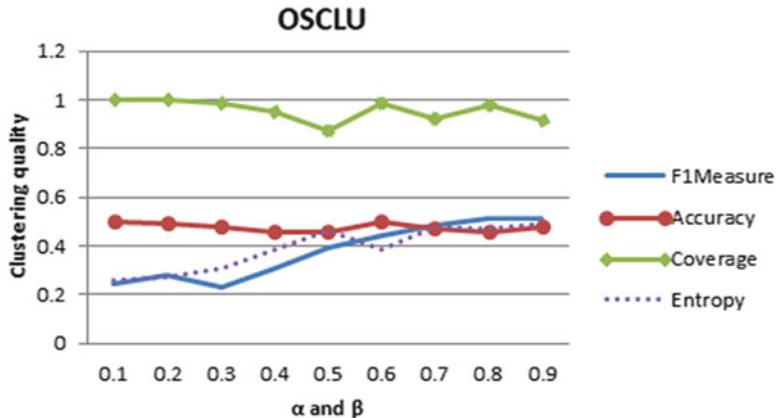


Fig. 6 clustering quality vs α , β

Table 4 Clustering result of TSCLU and OSCLU(1)

Algorithms Parameters	TSCLU		OSCLU	
	$k = 5$	$k = 6$	$\alpha = 0.3$ and $\beta = 0.3$	$\alpha = 0.4$ and $\beta = 0.4$
Cluster number	5	6	10	12
Running time(s)	17.7	18.1	3.1	8.4
Redundancy	1.26	1.65	5.2	18.7
F1 measure	0.19	0.18	0.23	0.31
Accuracy	0.44	0.45	0.48	0.48
Entropy	0.36	0.32	0.32	0.39
Coverage	0.82	0.94	0.97	0.97

4.2.2 Clustering Quality

For fair comparison, we ran massive experiments with various parameter settings and selected the one with the best quality. We set $k = 5$ and 6 for TSCLU, $\alpha = 0.3$, $\beta = 0.3$ and $\alpha = 0.4$, $\beta = 0.4$ for OSCLU.

From Table 4 we can see that TSCLU got a roughly the same clustering quality with OSCLU; however, TSCLU got a significantly smaller redundancy value. As mentioned in Sect. 3, the redundancy is quite relative to cluster number; for a fair comparison, we also did an experiment to compare TSCLU and OSCLU with similar cluster number. We can see that compared to OSCLU, TSCLU still got a significantly smaller redundancy value with similar clustering quality.

With experiments on real-world data, we can see that TSCLU can get a significantly smaller redundancy value than OSCLU with similar clustering quality. Besides, we find that TSCLU is not sensitive to parameter K , while OSCLU contains lots of parameters and is quite sensitive to the parameters.

Table 5 Clustering result of TSCLU and OSCLU(2)

Algorithms Parameters	TSCLU			OSCLU	
	$k = 10$	$k = 11$	$k = 12$	$\alpha = 0.3$ and $\beta = 0.3$	$\alpha = 0.4$ and $\beta = 0.4$
Cluster number	10	11	12	10	12
Running time(s)	16.7	17.3	17.0	3.1	8.4
Redundancy	3.73	4.10	4.58	5.2	18.73
F1Measure	0.24	0.22	0.24	0.23	0.31
Accuracy	0.47	0.47	0.46	0.48	0.48
Entropy	0.32	0.33	0.29	0.32	0.39
Coverage	0.99	0.99	0.99	0.97	0.97

On the other hand, TSCLU suffers a relatively higher running time. Since K -means is not a time-consuming algorithm, the long running time is mainly caused by SUBCLU. We will try some other redundant subspace clustering approaches for the first-step clustering of TSCLU in the future.

5 Conclusion

In this paper, we proposed a two-step non-redundant subspace clustering approach: TSCLU. The approach is based on a two-step clustering model which is never proposed before. Different with traditional non-redundant subspace clustering approaches OSCLU and ASCLU which measure the redundancy of a cluster while adding it to the result set, TSCLU applies a redundant subspace clustering approach which ensures not to miss any clusters firstly and group clusters to remove redundancy in the second step.

With a series of experiments, we find that TSCLU can get a similar clustering quality with OSCLU, while getting a significantly smaller redundancy value. Besides, TSCLU needs less parameters than OSCLU and is not sensitive to parameter settings, which make TSCLU much easier to implement. However, we have to point out that TSCLU's running time is much longer than OSCLU. The long running time is caused by redundant subspace clustering algorithm OSCLU. The experiments have proved that the two-step clustering model is reasonable and easy to implement. In the future we will do some work on trying different redundant subspace clustering algorithms for the first-step clustering to reduce running time and proposing a more reasonable redundancy-evaluating criterion.

Acknowledgements This work is supported by the National Basic Research Program of China (973 Program, Grant No. 2012CB315803), the National High-tech R&D Program of China 863 Program (Grant No. 2011AA01A101), the National Natural Science Foundation of China (Grant No. 61003100 and No.60972011), and Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20100002120018 and No.20100002110033).

References

1. Aggarwal, C.C., Wolf, J.L., Yu, P.S., Procopiuc, C., Park, J.S.: Fast algorithms for projected clustering. ACM SIGMOD Rec. **28**(2), 61–72 (1999)
2. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, vol. 27. ACM, New York (1998)
3. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, pp. 87–499, 1994
4. Assent, I., Krieger, R., Muller, E., Seidl, T.: Dusc: Dimensionality unbiased subspace clustering. In: ICDM 2007. Seventh IEEE International Conference on Data Mining, 2007, pp. 409–414. IEEE, (2007)
5. Assent, I., Krieger, R., Muller, E., Seidl, T.: Inscy: Indexing subspace clusters with in-process-removal of redundancy. In: ICDM 2008. Eighth IEEE International Conference on Data Mining, 2008 pp. 719–724. IEEE, (2008)
6. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is nearest neighbor meaningful? In: Database Theory ICDT'99, pp. 217–235. Springer, (1999)
7. Biersack, J.P., Haggmark, L.G.: A monte carlo computer program for the transport of energetic ions in amorphous targets. Nucl. Instrum. Meth. **174**(1-2), 257–269 (1980)
8. Bringmann, B., Zimmermann, A.: The chosen few: On identifying valuable patterns. In: ICDM 2007. Seventh IEEE International Conference on Data Mining, 2007, pp. 63–72. IEEE, 2007
9. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining, vol. 1996, pp. 226–231. AAAI Press, (1996)
10. Günemann, S., Färber, I., Müller, E., Seidl, T.: Asclu: Alternative subspace clustering. In: In MultiClust at KDD. Citeseer, (2010)
11. Günemann, S., Müller, E., Färber, I., Seidl, T.: Detection of orthogonal concepts in subspaces of high dimensional data. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 1317–1326. ACM, New York (2009)
12. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. J. R. Stat. Soc. C (Applied Statistics) **28**(1), 100–108 (1979)
13. Kailing, K., Kriegel, H.P., Kröger, P.: Density-connected subspace clustering for high-dimensional data. In: Proc. SDM, vol. 4, 2004
14. Kriegel, H.P., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans. Knowl. Discov. Data **3**(1), 1 (2009)
15. Müller, E., Assent, I., Krieger, R., Günemann, S., Seidl, T.: Densest: Density estimation for data mining in high dimensional spaces. In: Proc. SIAM SDM, pp. 173–184, 2009
16. Müller, E., Günemann, S., Assent, I., Seidl, T.: Evaluating clustering in subspace projections of high dimensional data. Proc. VLDB Endowment **2**(1), 1270–1281 (2009)
17. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter **6**(1), 90–105 (2004)
18. Patrikainen, A., Meila, M.: Comparing subspace clusterings. IEEE Trans. Knowl. Data Eng. **18**(7), 902–916 (2006)
19. Sequeira, K., Zaki, M.: Schism: A new approach for interesting subspace mining. In: ICDM'04. Fourth IEEE International Conference on Data Mining, 2004, pp. 186–193. IEEE, (2004)
20. Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proc. Workshop on Self organizing Maps, pp. 225–230, 2003
21. Yiu, M.L., Mamoulis, N.: Frequent-pattern based iterative projected clustering. In: ICDM 2003. Third IEEE International Conference on Data Mining, 2003, pp. 689–692. IEEE, (2003)

Exploiting Ontologies to Rank Relationships Between Patents

Hai-Tao Zheng, Nan Ma, Yong Jiang, Shu-Tao Xia, and Hui-Qiu Li

Abstract Patent relationship analysis plays an important role in patent processing service and research. Since most existing analytical methods utilize word co-occurrence to measure patent relationships without considering the semantics of patents, the importance of patent relationships could not be measured precisely. To address this issue, we propose an ontology-based approach to rank relationships between patents. Our approach takes advantage of both lexical term relatedness and lexical co-occurrence for text analysis, and utilizes International Patent Classification (IPC) as domain ontology to calculate technical classification relatedness. Experiments on patents show that our approach performs well since multiple influential factors are taken into consideration.

1 Introduction

Patent is an important constituent part of intellectual property and the most important kind of technical literature. Fast increasing patent information brings in new opportunity and challenge to patent service and research. In research area of patent processing, analysis of relationship among patents plays an important role. A wide range of applications are based on analyzing relationship between patent documents, such as correlation analysis, clustering, and classification.

H.-T. Zheng • N. Ma (✉) • Y. Jiang • S.-T. Xia
Tsinghua-Southampton Web Science Laboratory, Graduate School at Shenzhen,
Tsinghua University, Shenzhen, China
e-mail: zheng.haitao@sz.tsinghua.edu.cn; nxmanan@gmail.com;
jiangy@sz.tsinghua.edu.cn; xiast@sz.tsinghua.edu.cn

H.-Q. Li
China Telecom Co., Ltd. Shenzhen Branch
e-mail: 13360097990@189.cn

Patent relationship analysis is important to the above applications. However, prior patent relationship analytical methods focus mostly on word co-occurrence analysis; thus, patent with similar concepts but different word usage habits might be lost. Besides, prior methods don't make the most of patent classification information, thus miss influential factor that is effective for relationship analyzing.

We propose a method that calculates relatedness between patent documents according to multiple features, which are document distance (DD), cosine distance (CosD), and IPC distance (IPCD). The relationships between documents are reflected not only by co-occurrence words but also by relationships between words within documents. Thus, we calculate DD taking advantage of Normalized Term Distance (NTD). CosD is complementary to DD. IPC, a patent classification system, is a domain ontology of technology field. IPC ontology can assist computer to better comprehend implicit as well as explicit patent relationships. We define IPCD as distance between IPC codes of two patent documents, which is a utilization of patent classification to obtain patent relatedness. These multiple features are combined to calculate relatedness between patent documents.

The rest of this paper is organized as follows: In Sect. 2, we present related work in the field. Section 3 is the detailed introduction to the proposed method. Section 4 presents the dataset, evaluation metrics, experimental results, and discussion. Finally, conclusions are given in Sect. 5.

2 Related Work

Document and patent similarity measure can be categorized into three types: word co-occurrence-based method, text structure-based method, and citation-based method. Word co-occurrence-based method is based on the idea that similar documents tend to have similar word distributions. Among word co-occurrence-based methods, Vector Space Model (VSM) is the core of information retrieval. Based on VSM, many document similarity calculation methods are proposed, among which the cosine measure is most widely used. Text structure-based method adopts nature language processing method to extract structure characteristic to find patent relationships. Zini et al. [1] and Lin et al. [2] analyze patent text syntactic structure and build a tree for each patent as a functional presentation. By comparing these trees, concepts relatedness of patents can be obtained. Analyzing citation as an interpatent relation analytical method is also studied in some research. Kasravi and Risov [3] leverage patent citations, text mining, and patent classification to improve accuracy of detecting patents similarity. However, citation behaviors in patent and academic literature differ a lot so that patent citation may not be appropriate for analyzing relatedness [4].

3 Our Approach

This study aims to propose a method to solve the problem below.

Problem 1. For a given patent document q and document collection C , the goal is to rank candidate patent documents in C according to each document's relationship with q and return a ranked list L of candidate patent documents.

Our method consists of four steps: (1) documents preprocessing, (2) document distance and cosine distance calculation, (3) IPC distance calculation, and (4) combining multi-evidences to rank.

First, patent document collection is preprocessed. Four substeps are proceeded: (1) target field extraction, useful sections of patent document such as claim field and IPC codes field are extracted from patent documents; (2) Word segmentation, for Chinese patent documents, we use ICTCLAS¹ to segment sentences into Chinese lexical items and add POS tags; (3) term filtering, only lexical terms tagged with noun are reserved for Calculation; and (4) index creation, target text part of documents is indexed with search engine library Lucene.² Besides text field, IPC codes are prepared for IPCD calculation.

The details of the other steps are explained below.

3.1 Document Distance and Cosine Distance Calculation

For calculating semantic distance of words, we adopt Normalized Google Distance (NGD) [5] method and apply it to local document collection. Since this calculation no longer needs Google search engine, we call the word distance Normalized Term Distance (NTD). For a document collection C , the semantic distance between two lexical terms t_1 and t_2 is given by

$$NTD(t_1, t_2) = \frac{\max\{\log(f(t_1)), \log(f(t_2))\} - \log(f(t_1, t_2))}{\log N - \min\{\log(f(t_1)), \log(f(t_2))\}}, \quad (1)$$

where $f(t_1)$ denotes the numbers of documents containing lexical term t_1 in C , $f(t_1, t_2)$ is the number of documents containing both t_1 and t_2 in C , and N is the total number of documents in C .

The document distance between documents d_1 and d_2 in C is given by

$$DD(d_1, d_2) = \frac{\sum_{t_i \in d_1} \sum_{t_j \in d_2} NTD(t_i, t_j)}{n_1 n_2}, \quad (2)$$

where n_1 and n_2 are the numbers of terms in document d_1 and d_2 , respectively.

¹<http://ictclas.org/>.

²<http://lucene.apache.org/>.

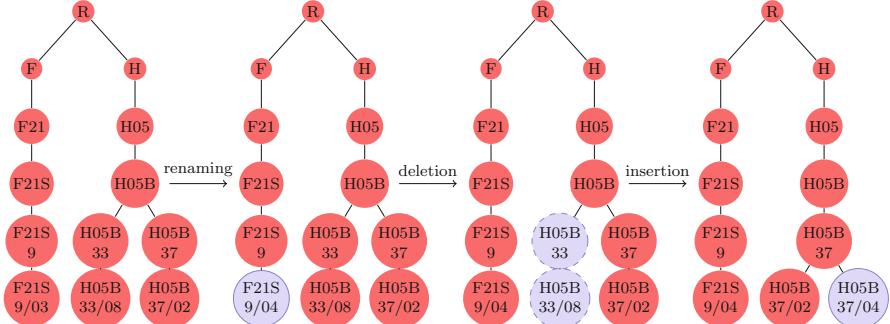


Fig. 1 The process of transforming IPC tree ipc_1 to ipc_2

Formula (2) is effective in most cases, while for certain test documents and answer set it may not achieve better result than cosine measure method. So we also utilize cosine similarity as an influential factor. For a document d_j with feature vector $\overrightarrow{V(d_j)} = (t_{1,j}, w_{1,j}; t_{2,j}, w_{2,j}; \dots; t_{n,j}, w_{n,j})$, $t_{1,j}, t_{2,j}, \dots, t_{n,j}$ are lexical terms, and $w_{1,j}, w_{2,j}, \dots, w_{n,j}$ are term weights. We assign term weights using TF-IDF method. The cosine distance of documents d_1 and d_2 is defined as follows:

$$CosD(d_1, d_2) = 1 - \frac{\overrightarrow{V(d_1)} \cdot \overrightarrow{V(d_2)}}{|\overrightarrow{V(d_1)}| |\overrightarrow{V(d_2)}|} = 1 - \frac{\sum_{k=1}^n w_{k,1} \times w_{k,2}}{\sqrt{\sum_{k=1}^n w_{k,1}^2 \sum_{k=1}^n w_{k,2}^2}} \quad (3)$$

3.2 IPC Distance Calculation

International Patent Classification (IPC) is the classification standard of patent document. IPC contains about 70,000 concepts in technical field and organizes these concepts in hierarchical structure. Every authorized patent is assigned one or multiple IPC codes to describe the patent's technical features. We use IPC as a domain ontology to provide background knowledge of technical concepts and name it IPC ontology. We represent IPC codes of a patent document as tree structure and define this tree structure as IPC tree for patent.

The similarity of IPC trees is an indicative character of patent relatedness. We calculate difference of IPC trees using tree edit distance (TED) algorithm [6, 7]. The TED between two trees is the minimum cost sequence of elementary operations (node deletion, node insertion, node renaming) which transform one tree into the other one. We describe the process of calculating TED between IPC trees below. Consider two IPC trees $ipc_1 = H05B37/02, H05B33/08, F21S9/03$, $ipc_2 = H05B37/02, H05B37/04, F21S9/04$, the process of transforming ipc_1 to ipc_2 is shown in Fig. 1. 1 node renaming, 2 node deletions, and 1 node insertion are carried out during the process; thus, the TED between the two IPC trees is 4.

In order to compare the degree of difference between IPC tree pairs, we define IPC distance (IPCD) between IPC tree ipc_1 and ipc_2 as below:

$$IPCD(ipc_1, ipc_2) = \frac{TED(ipc_1, ipc_2)}{\max(\text{size}(ipc_1), \text{size}(ipc_2))} \quad (4)$$

where $\text{size}(ipc)$ is size of an IPC tree; here, we set it to number of edges of an IPC tree. IPC distance is the TED between two IPC trees which is normalized by the maximal size of the two IPC trees. For the example above, sizes of the IPC trees are 12 and 11; thus, IPC distance between them is $\frac{4}{\max(12, 11)} = \frac{4}{12} \simeq 0.33$.

3.3 Combining Multi-evidences to Rank

For evaluating relationship between patent documents, we calculate the three factors above as ranking evidences. Then these factors are combined with certain weight. We define patent distance between patent p_1 and p_2 as below:

$$\text{PatentDistance}(p_1, p_2) = \alpha DD(p_1, p_2) + \beta CosD(p_1, p_2) + \gamma IPCD(ipc_{p_1}, ipc_{p_2}) \quad (5)$$

The weighting parameters α, β, γ follow a constraint equation: $\alpha + \beta + \gamma = 1$. For a pair of patent documents, the smaller their patent distance, the more semantic related they are.

4 Experiments and Evaluation

To conduct the experiments, we selected 10 technical fields, in each field we selected one document served as query, and 50 documents served as candidate document set for relationships ranking. Human expert made relevance judgment on each query and candidate document pair at five levels, i.e., perfect, excellent, good, fair, and bad. Each relevance label was assigned a corresponding numeric value, 5, 4, 3, 2, and 1. Normalized Discounted Cumulative Gain (NDCG) is a widely used metric to evaluate a ranking model [8]. The idea behind NDCG is giving higher score to the ranked list in which relevant documents are ranked higher.

First we conduct an experiment to determine the parameters' values in formula (5). For simplicity, we set $\beta = \gamma = \frac{1-\alpha}{2}$. We perform ranking experiments on 10 groups of document with different α values. From Fig. 2 we can see that when α is assigned value 0.6, NDCGs in both Top-5 and Top-10 achieve maximum value. So we set $\alpha = 0.6$, $\beta = 0.2$, and $\gamma = 0.2$.

The experimental result in Fig. 3 shows the proposed method has performed better than cosine measure in Top-5 and Top-10 on average. The NDCG in Top-5 and Top-10 obtained from proposed method achieves 19% and 11.5% improvement

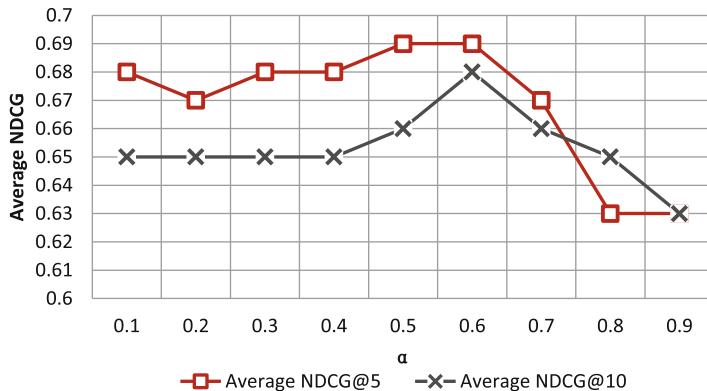


Fig. 2 Average NDCG in Top-5 and Top-10 with different α value

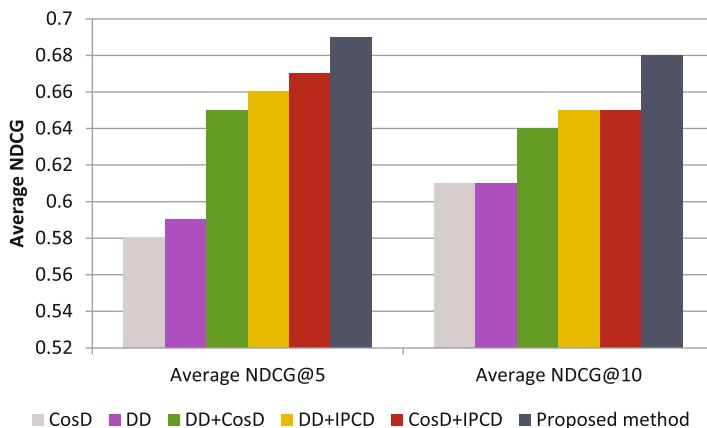


Fig. 3 Comparison of average NDCG in Top-5 and Top-10 of different methods

compared to cosine measure. It can also be perceived that ranking method utilizing DD achieves almost the same result with CosD method, while the combinations of DD+CosD, DD+IPCD, and CosD+IPCD achieve better performance but still perform worse than proposed method.

These outcomes can be explained below. The multiple patent document features we utilize each has an influence on relatedness calculation. Document distance based on NTD is a reflection of relatedness of terms in documents. CosD reflects degree of term distributions' similarity in documents. IPC ontology covers themes of technical field and has a hierarchical structure; thus, IPCD is the distance between concepts of two patent documents. When these influential factors are combined with appropriate weights, the average ranking performance achieves best result compared to methods in which single factor or two factors are adopted since it acquires more evidences of judging relatedness.

5 Conclusion

We propose a method for relationships ranking between patent documents. The method adopts multiple evidences to calculate semantic distance between patents, which are document distance, cosine distance, and IPC distance based on IPC ontology. It utilizes both textual features and hierarchical ontology structure to obtain the semantic features of patents. Experiments show that the proposed method exceeds classical cosine measure on patent relationships ranking.

In the future, we are considering investigating better method of combining influential factors to improve ranking performance. We also intend to reduce computational complexity of the proposed method so that it can be applied in large-scale dataset. Furthermore, we plan to mine the semantics of relationships between patent documents; thus, the related ranking list will be more meaningful.

Acknowledgements This work is supported by the National Basic Research Program of China (973 Program, Grant No.2012CB315803), the National High-tech R&D Program of China (863 Program, Grant No.2011AA01A101), the National Natural Science Foundation of China (Grant No.61003100 and No.60972011), and Research Fund for the Doctoral Program of Higher Education of China (Grant No.2010000212001
8 and No.20100002110033).

Reference

1. Zini, M., Cascini, G.: Measuring patent similarity by comparing inventions functional trees. In: Computer-Aided Innovation (Cai)-IFIP International Federation for Information Processing pp. 31–42 (2008)
2. Lin, F., Huang, F.: The study of patent prior art retrieval using claim structure and link analysis. In: PACIS 2010 Proceedings, pp. 198, 2010
3. Kasravi, K., Risov, M.: Multivariate patent similarity detection. In: Proceedings of the 42nd Hawaii International Conference on System Sciences. HICSS '09, pp. 1–8. IEEE Computer Society, Washington, DC (2009)
4. Joo, S., Kim, Y.: Measuring relatedness between technological fields. *Scientometrics* **83**, 435–454 (2010)
5. Cilibrai, R.L., Vitanyi, P.M.B.: The google similarity distance. *IEEE Trans. Knowl. Data Eng.* **19**, 370–383 (2007)
6. Tai, K.: The tree-to-tree correction problem. *J. ACM* **26**(3), 422–433 (1979)
7. Lakkaraju, P., Gauch, S., Speretta, M.: Document similarity based on concept tree distance. In: Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia. HT '08, pp. 127–132. ACM, New York, NY (2008)
8. Järvelin, K., Kekäläinen, J.: Ir evaluation methods for retrieving highly relevant documents. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 41–48. ACM, New York (2000)

Search Results Diversification Based on Swap Minimal Marginal Contribution

Hai-Tao Zheng, Shaozhou Bai, Shu-Tao Xia, Yong Jiang, and Huiqiu Li

Abstract In this paper we present a method to deal with the problem of diversifying search results for the ambiguous queries. This issue focuses on how to re-rank the result list and get a diverse result set which should contain as much subtopics as possible in the top k results. As the diversity and similarity are incompatible, finding the optimal result set is NP-hard. A feasible approach is to make a trade-off between similarity and diversity and then use a scoring function to maximize the marginal efficiency of the elements to be selected into the result set. Based on this idea we propose *Swap Minimal Marginal Contribution* (SMMC), which is a swap and convergent method instead of an incremental method. SMMC avoids the weakness of being heavily influenced by the “bad” elements in the partial result set. Our experiments show that SMMC outperforms the baseline methods with a low time complexity.

1 Introduction

Ambiguous queries are those with multiple interpretations [1] (e.g., windows, java, eclipse, and apple). When people search something ambiguous by the search engine, they usually want the search engine to supply a comprehensive answer set that contains as much potential answers as possible in the top list of all results, so that they can easily pick up the answer they really want. However, the traditional

H.-T. Zheng (✉) • S. Bai • S.-T. Xia • Y. Jiang
Tsinghua-Southampton Web Science Laboratory, Graduate School at Shenzhen,
Tsinghua University, Shenzhen, China
e-mail: quicklyfly@gmail.com; baishaozhou@gmail.com; xiaст@sz.tsinghua.edu.cn;
jiangy@sz.tsinghua.edu.cn

H. Li
China Telecom Co., Ltd., Shenzhen Branch, China
e-mail: 13360097990@189.cn

IR system retrieves and ranks documents based only on the relevance between documents and the query; as a consequence, the top list of the results only contains highly relevant documents with the same interpretation, which usually degrades the user's searching experience. So far, many diversification methods are proposed; the basic thought is when ranking result documents, the search engine should not only consider relevance but also provide documents that satisfy different interpretations of the query. For example, when searching the ambiguous word “apple” in a search engine, the top list of results should contain the “apple of Steve Jobs” and the “apple of fruit” rather than that the whole pages are all about the products of Steve Jobs.

The methods to diversify search results can be divided into two different categories based on whether extra information is needed or not. The first category needs information such as query logs, taxonomy, and click through rates to construct a probability model or something similarly and then generate diversity by their mathematic model. This kind of algorithms suits for the data which has known specific structure and the extra information gets ready in advance. The second category relies only on the similarity and diversity functions, which can be easily used in any kind of data domains. Many methods of the second kind explore a greedy solution that builds the result set in an incremental way, and the documents are selected one by one into the result set. One famous method of this kind is *Maximal Marginal Relevance* (MMR), proposed by Carbonell and Goldstein [2]. According to MMR, a scoring function mmr is proposed to select documents in the candidate answer set, and at each iteration, the document s_i with the maximal mmr value will be selected into the result set. Vieira et al. [3] optimized MMR and proposed the *Greedy Marginal Contribution* (GMC). Their difference is that the scoring function of MMR considers diversity only to the partial result set while GMC also considers the documents in the candidate set.

We propose the *Swap Minimal Marginal Contribution* based on the idea of marginal efficiency used by MMR and GMC; instead of being an incremental and greedy method like MMR and GMC, SMMC is a swap and convergent method, which avoids the weakness of incremental methods that are being heavily influenced by the previous elements in the partial result set especially the “bad” ones. SMMC firstly initializes the result set with k different documents and then iteratively replaces the “bad” one (with the minimal marginal efficiency) in the result set by the “good” one (with the maximal marginal efficiency) in the candidate set, the convergent process will not stop until the result set has been optimal. We present an experimental evaluation of diversification methods presented in this paper, and the result shows that SMMC has a good precision without sacrificing the time complexity heavily. All the evaluated methods use only the values computed by the scoring function without any external information (e.g., subtopics, categories), so they can be used more widely.

The rest of this paper is organized as follows: in Sect. 2 we review the related works. In Sect. 3 we model the result diversification problem. In Sect. 4 we present the baseline method and our proposed method. In Sect. 5 the experimental evaluation will be shown and Sect. 6 concludes this paper.

2 Related Work

Previous works have investigated results diversification in various methods. Carbonell and Goldstein [2] introduced a preliminary model for diversification, the *Maximal Marginal Relevance*. This model uses the earliest concept of novelty and similarity among documents, and the scoring function makes a trade-off between novelty and similarity to rank documents; at last the marginal efficiency is used to iteratively select documents into the result set. Vieira et al. [3] proposed *Greedy with Marginal Contribution*, which is a variation of MMR; their difference is whether or not to consider the documents outside the partial result set R when computing the diversity; GMC considers more factors by sacrificing the time complexity heavily.

Agrawal et al. [4] presented a systematic approach to diversifying results that aims to minimize the risk of dissatisfaction of the average user. They proposed a greedy algorithm IA-select with the objective function based on the probabilistic model and an existing taxonomy *Open Directory Project*. They assumed that users only consider the top k returned results and also proved that the diversification problem is NP-hard. Zhai et al. [5] proposed a risk minimization framework for information retrieval in which the loss function can be defined by the user. He et al. [6] used the clustering techniques to preprocess the result list and then diversify the filtered results by the traditional diversification methods. Vee et al. [7] and Yu et al. [8] applied the diversification techniques in the recommendation system by topic diversification. Radlinski et al. [9] re-ranked the result list by user's clicking logs. Hu et al. [10] designed a method to predict user's query intent using Wikipedia.

For the evaluation on diversity task, Zhai et al. [11] proposed a simple metric named *S-recall* (subtopic recall), which is defined as the number of unique subtopics covered by the top k results divided by the number of all the subtopics. Clarke et al. [12] proposed the alpha-nDCG to evaluate diversification methods, which is a variation of nDCG by considering novelty. Similarly, Agrawal [4] optimized the MAP to MAP-IA as the metric to evaluate the diversity. Vieira et al. [3] proposed the *F-value* metric which is a trade-off of the similarity and diversity between the query and documents in the result set.

Based on the idea of marginal efficiency, the *Swap Minimal Marginal Contribution* is a variation of MMR. However, SMMC is a swap and convergent method instead of an incremental one, and so it can avoid being heavily influenced by the previous documents in the partial result set, especially the “bad” ones. At each iteration, MMR selects the document with the maximal marginal efficiency but SMMC replaces the document with the minimal marginal efficiency; the iteration of MMR stops when it runs k loops but SMMC stops only when the result set has been optimal. As MMR is cited widely in the field of results diversification and needs no extra information such as query logs and taxonomy, we use it as a baseline method in this experiment.

3 Preliminaries

The result diversification problem can be described as follows: given a candidate answer set S for query q , let $S = \{s_1, \dots, s_k\}$ be a set of n elements, and find the result set $R = \{r_1, \dots, r_k\}$ with k elements, where R is a subset of S , k is an integer, and $k < n$. The elements in R must be as relevant as possible to the query q , and at the same time, the result set should be as diverse as possible. Let the similarity of each element $s_i \in S$ be specified by the function $\delta_{\text{sim}}(q, s_i)$, and the diversity between two elements $s_i, s_j \in S$ is specified by the function $\delta_{\text{div}}(s_i, s_j)$; the larger value implies more similar or diverse, respectively. In our experiment, we present elements in S using the *Vector Space Model*, so it is easy to compute the δ_{sim} and δ_{div} by techniques like *cosine similarity* or *Euclidean distance*.

Vieira et al. [3] think this can be modeled as an optimization problem where there is a trade-off λ between finding relevant elements to q and finding a diverse result set R , and a definition is defined for evaluating the diversity task:

Definition 1. Given a trade-off λ ($0 < \lambda < 1$) between similarity and diversity, the K -similar diversification set R contains k elements of S that:

$$R = \arg \max_{S' \subseteq S, k=|S'|} F(q, S') \quad (1)$$

where

$$F(q, S') = (k - 1)(1 - \lambda) \times \text{sim}(q, S') + 2\lambda \times \text{div}(S') \quad (2)$$

The *F-value* makes a trade-off between the similarity and diversity among the query and the elements in the result set R . The ideal result set R should have the maximal *F-value*. Since the *sim* and *div* have different numbers of elements to compute, the coefficients are used to scale up this equation.

$$\text{sim}(q, S') = \sum_{i=1}^k \delta_{\text{sim}}(q, s_i), \quad s_i \in S' \quad (3)$$

$$\text{div}(S') = \sum_{i=1}^{k-1} \sum_{j=i+1}^k \delta_{\text{div}}(s_i, s_j), \quad s_i, s_j \in S' \quad (4)$$

S' is a subset of S with k elements, $\text{sim}(q, S')$ represents the sum of similarity distances between the query q and all elements in S' , and $\text{div}(q, S')$ represents the sum of diversity distances amongst elements of S' .

4 Methods

4.1 Baseline Methods

4.1.1 Maximal Marginal Relevance

MMR is the earliest method to diversify documents by using the concept of novelty (diversity) and similarity. It iteratively constructs the result set R by selecting a new element in S that maximizes the following function:

$$\text{mmr}(s_i) = (1 - \lambda) \times \delta_{\text{sim}}(s_i, q) - \lambda \times \max_{s_j \in R} \delta_{\text{sim}}(s_i, s_j) \quad (5)$$

The first component computes the similarity between the element s_i and query q , and the second component computes the maximal similarity between s_i and the element s_j in the partial result set. With a minus sign in front, the second component represents the diversity between s_i and the result set R . The coefficient λ makes a trade-off between the relevance and diversity. At each iteration, the mmr function selects the element s_i with the highest mmr value into R until k elements are selected.

Since incrementally inserting new elements relies only on elements already in the partial result set, MMR is heavily influenced by the previous elements and may have low quality if the previous elements are not selected properly.

4.1.2 Greedy Marginal Contribution

Similar to MMR, GMC incrementally builds the result set R by selecting elements with the *maximal marginal contribution* (mmc):

$$\text{mmc}(s_i) = (1 - \lambda) \delta_{\text{sim}}(s_i, q) + \frac{\lambda}{k-1} \sum_{s_j \in R_{p-1}} \delta_{\text{div}}(s_i, s_j) + \frac{\lambda}{k-1} \sum_{\substack{l=1, \\ s_j \in S - R_{p-1}}}^{l \leq k-p} \delta_{\text{div}}^l(s_i, s_j) \quad (6)$$

where R_{p-1} is the partial result set of size $p-1$, $1 \leq p-1 \leq k$, and $\delta_{\text{div}}^l(s_i, s_j)$ gives the l^{th} largest δ_{div} value in $\{\delta_{\text{div}}^l(s_i, s_j) : s_j \in S - R_{p-1} - s_i\}$. This function can compute the maximal contribution of s_i to *F-value* by using the elements already in R_{p-1} and the remaining $k-p$ elements in S which differs with s_i mostly.

Compared to MMR, GMC considers more factors by computing the $\delta_{\text{div}}^l(s_i, s_j)$ for each unselected element, which improves the complexity heavily to $O(kn^2)$.

4.2 Proposed Method

4.2.1 Swap Minimal Marginal Contribution

The basic idea of our proposed method is that the marginal efficiency of one document should be maximized to an entire result set rather than a partial one as in MMR. The result set R is initialized by k different documents, and then using a scoring function mc to iteratively replace the documents with the minimal marginal contribution in R by the document with the maximal marginal contribution in candidate set S , the iteration will not stop until R does not change any more, which is a convergent process. The $\text{mc}(s_i)$ function represents the marginal contribution that one document s_i to a result set R , we define it as follows:

$$\text{mc}(s_i) = (1 - \lambda) \times |R| \times \delta_{\text{sim}}(s_i, q) + 2\lambda \times \sum_{s_j \in R} \delta_{\text{div}}(s_i, s_j) \quad (7)$$

We can deduce the mc function as follows: when a document s_i is added into a result set R , we get a new result set R' , $R' = R \cup s_i$, and then we can get two different F -values, F and F' ; the *marginal contribution* of S_I to R is $F' - F$.

By the definition of F -value from Eq. (2), we get

$$F = (1 - \lambda) \times (|R| - 1) \times \text{sim}(R, q) + 2\lambda \times \text{div}(R) \quad (8)$$

$$F' = (1 - \lambda) \times (|R'| - 1) \times \text{sim}(R', q) + 2\lambda \times \text{div}(R') \quad (9)$$

And from the relation of R and R' , we can also get

$$\text{sim}(R', q) = \text{sim}(R, q) + \delta_{\text{sim}}(s_i, q) \quad (10)$$

$$\text{div}(R') = \text{div}(R) + \sum_{s_j \in R} \delta_{\text{div}}(s_i, s_j) \quad (11)$$

Finally, replace the sim and div in Eq. (9) with Eqs. (10) and (11), respectively, and then we can get the marginal contribution by (9) and (10):

$$\text{mc}(s_i) = (1 - \lambda) \times \text{sim}(R, q) + (1 - \lambda) \times |R| \times \delta_{\text{sim}}(s_i, q) + 2\lambda \times \sum_{s_j \in R} \delta_{\text{div}}(s_i, s_j) \quad (12)$$

Because the first component has no relevance with document s_i , it is a constant to all the documents when computing the marginal contribution to R , so we can ignore this part and simplify the final mc function to what we define in Eq. (7).

The pseudocode of this method is given here:

SMMC Algorithm

```

Input: candidate set  $S$  and result set size  $k$ 
Output: result set  $R$ ,  $|R| = k$ 
Begin
     $R \leftarrow$  selects  $k$  different documents from  $S$ 
     $S \leftarrow S - R$ 
    Loop:
        For each  $s_i$  in  $R$ , find the minimal  $mc(s_i)$  to  $R \setminus s_i$ 1
        Select  $s_i$  with the minimal  $mc$  value
        For each  $s_j$  in  $S$ , find the maximal  $mc(s_j)$  to  $R \setminus s_i$ 
        If  $mc(s_j) > mc(s_i)$  then
             $R \leftarrow (R \setminus s_i) \cup s_j$ 
             $S \leftarrow (S \setminus s_j) \cup s_i$ 
        Else
            Stop loop
    End.

```

Unlike greedy methods such as MMR and GMC, SMMC is a swap and convergent method, initialized by k documents; every time we compute the marginal contribution of one document, the mc value is to the entire result set, not the partial result set; so the marginal efficiency in SMMC is more effective than MMR and GMC. Moreover, as iteratively replacing the “bad” documents in R , SMMC will not be influenced heavily by the improperly selected documents.

As SMMC is a swap and convergent method, the running time is heavily affected by the quality of the initialized k elements in the result set R , and so, how to optimize the initialized R becomes an important problem that needs further study. In order to highlight the advantage of the SMMC algorithm itself, we just use the inefficient *random selection* to initialize R and consider the running time on average in this experiment. The result shows that SMMC outperforms the baseline methods with a low time complexity, which verifies that the idea of SMMC is effective.

4.3 Algorithms Comparison

As greedy methods, both MMR and GMC need to run a certain k iterations to get the result set R . In order to improve the precision, GMC sacrifices the complexity from $O(kn)$ to $O(kn^2)$. As SMMC is a convergent method, its complexity cannot be described in the form of constants, so we use the coefficient c to represent the number of loops and get the approximate complexity $O(cn)$. The coefficient c is an important factor to the complexity; when it is in the same order with k , SMMC will be equal to MMR in terms of time consumed. In fact, the coefficient c keeps a linear relation with k in some degree we find in this experiment, and the time complexity of SMMC is in the same order with MMR (Table 1).

¹ $R \setminus s_i$ means the subset of R that contains all the elements in R except s_i

Table 1 A comparison of the methods evaluated

Algorithms	Construction of R	Complexity	Based on marginal efficiency
MMR	Incremental	$O(kn)$	Yes
GMC	Incremental	$O(kn^2)$	Yes
SMMC	Swap	$O(cn)$	Yes

Note: c represents the number of loops in SMMC; it is a variable, not a constant

5 Experimental Evaluation

5.1 Dataset and Experiment

The dataset we use in our experiment is $dblp$, which is a set of 2,004,817 publication entries extracted from DBLP.² Each entry in the $dblp$ dataset contains the author's name, publication title, publication date, etc. For simplicity, we only extract the title for each entry and create an index of them by Lucene, and then we can search a query by the Lucene search engine and retrieve a list of results. To generate a query dataset, we randomly choose several queries from $dblp$. To obtain the candidate set S for each query, we extract the top- n elements from the result list of Lucene. In our experiment, we employed the tf/idf cosine similarity distance to compute δ_{sim} , and δ_{div} is represented by $1 - \delta_{\text{sim}}$.

In this experiment, we randomly choose three queries from $dblp$, "nearest neighbor," "database systems," and "medical image," and for each query we set 100, 200, and 300 as different sizes of the candidate answer set S ; then for each candidate set, we select the top 5, top 10, and top 20 documents as the result set R ; finally we compute the *F-value* as the metric of precision for each result set. By this method, we can get 27 pairs of correlation data (Table 2). In order to reduce the influence of the randomness, we use the average value to do our evaluation in the following.

5.2 Criteria and Evaluation

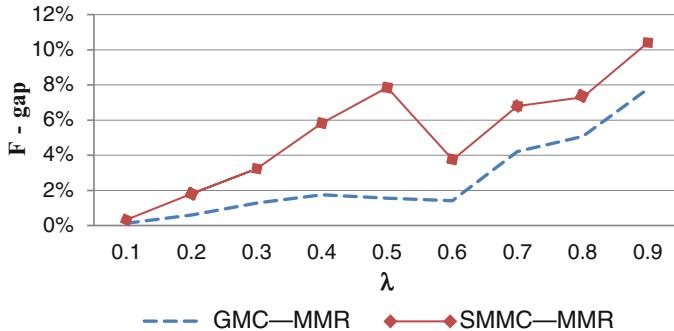
5.2.1 Precision

To measure the precision of the methods, we need a baseline and compare it with the results of our experiment; unfortunately, $dblp$ does not provide any answer set. In this circumstance, an alternative method is using the *F-value* which can be calculated by relevance and diversity functions as the criterion to evaluate the

²<http://www.informatik.uni-trier.de/~ley/db/>.

Table 2 Parameters tested in this experiment

Parameter	Range
Trade-off λ	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Candidate set size $n = S $	100, 200, 300
Result set size $k = R $	5, 10, 20
Sample queries	3

**Fig. 1** Average *F*-value gap vs. trade-off λ

precision. The intuition is that the better algorithm produces the result set closer to be optimal, and the better result set gets the higher *F-value*.

Although there are three methods to be evaluated, we just use two curves in the following graphs, and the MMR is always represented by the horizontal axis.

Figure 1 shows the average *F-value* between *SMMC–MMR* and *GMC–MMR* from the 27 pairs of correlation data we get in the experiment. The gap between F_{value} and F_m is computed by $(F_s - F_m)/F_m$. Obviously, both the *GMC* and *SMMC* have a positive f-gap to *MMR* which means they outperform *MMR* in the experiment in terms of *F-value*, namely, the precision. At the same time, *SMMC* outperforms *GMC* through the range of λ .

Although we get 27 pairs of correlation data, we cannot paste all the graphs in this paper for lacking of space. Because the results of the three queries are similar in the tendency, we choose six of them to analyze here.

In Fig. 2, we find that the f-gap varies with k and n except that when λ closes to zero, which is because all three methods select documents only by the relevance when λ is very small and all get the same *F-value*. For the same k , f-gap varies with n ; when n turns larger, f-gap tends to be larger in the middle range of λ , which means both *SMMC* and *GMC* are more sensitive than *MMR* in a larger candidate set when k is a constant. On the other hand, for the same n , when k turns larger, f-gap tends to be smaller, which means both *SMMC* and *GMC* tends to perform well with a smaller k to a specific candidate. As a summary, *SMMC* and *GMC* tend to perform better than *MMR* when the ratio of k/n turns smaller, and *SMMC* outperforms *GMC*.

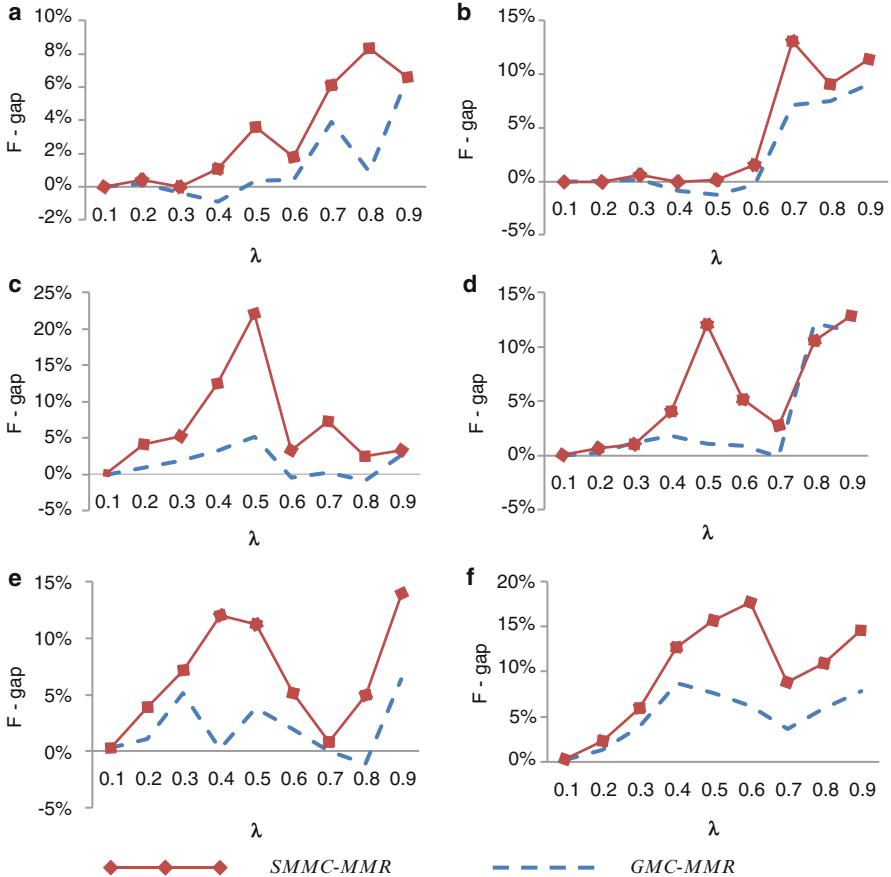


Fig. 2 Average F -value gap vs. variables k and n . **(a)** $k = 5, n = 100$; **(b)** $k = 10, n = 100$; **(c)** $k = 5, n = 200$; **(d)** $k = 20, n = 200$; **(e)** $k = 10, n = 300$; and **(f)** $k = 20, n = 300$

5.2.2 Time Complexity

The next we will test is the running time of each method and then make comparison of them. By this analysis, we want to verify whether the SMMC gets a better precision with sacrificing the time efficiency heavily or not.

Since the running time changes regularly and almost all graphs look like the same, we just choose four graphs to do our analysis.

Figure 3 is a running time comparison of SMMC with MMR and GMC. The horizontal axis represents the trade-off λ , and the vertical axis represents the time ratio. Solid line curves with diamond represent SMMC–MMR and dot line curves represent GMC–MMR. The *time ratio* between T_s and T_m is computed by T_s/T_m . The higher one consumes more time than the lower one.

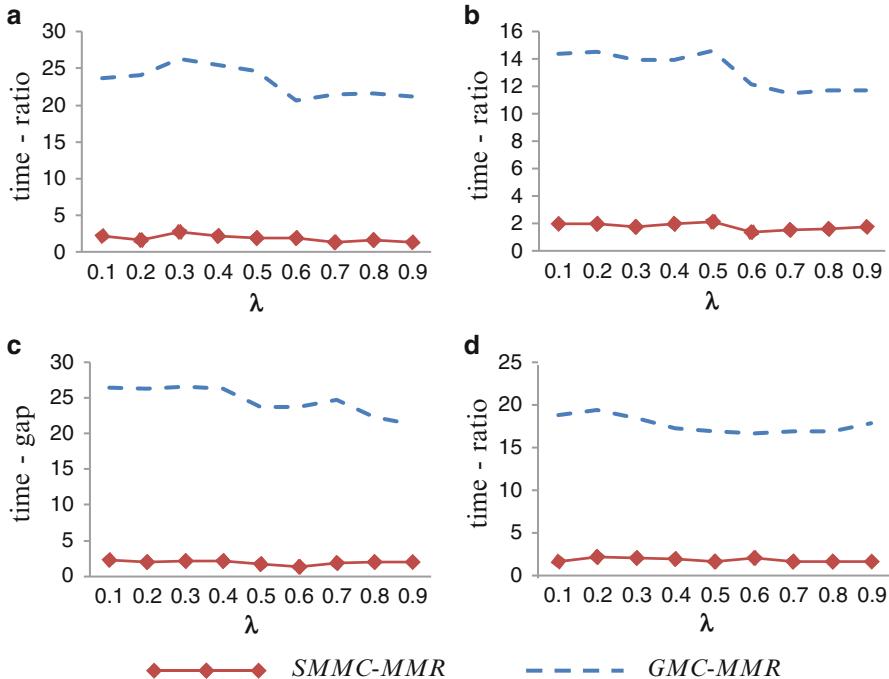


Fig. 3 Average time ratio vs. variables k and n . (a) $k = 5, n = 100$; (b) $k = 10, n = 100$; (c) $k = 10, n = 200$; and (d) $k = 20, n = 300$

In all the tests, the values of the solid line curves hover around 2 and vary little with λ and n , which is the linear relation between c and k mentioned above. In other words, the time complexity of SMMC is in the same order with MMR in our experiment (we will do more experiments to verify this relation in our future work). In fact, from the complexity we can find why this is reasonable, the complexity of SMMC is $O(cn)$ and MMR is $O(kn)$, when the coefficient c approximately equals to k , SMMC and MMR will have the same time complexity. On the other hand, the dot line curves vary with the candidate set size n and the result set size k , but no changes with the trade-off λ . Obviously, for the same k , when n turns larger, the time ratio between GMC and MMR tends to be larger, because GMC's complexity is $O(kn^2)$, n times of MMR. For the same n , when k turns larger, the time ratio tends to be smaller; seemingly this is false because more time will be consumed when k is larger; in fact, with the growing of k , the time consumption of MMR increases quickly than GMC in this experiment; then the ratio decreasing is reasonable.

Through the experiment and evaluation, we prove that our proposed method SMMC can outperform MMR and GMC with a low time complexity. The complexity of SMMC is lower than GMC and approximates to MMR.

6 Conclusion

In this paper, we present some related works on search results diversification and proposed our new method SMMC. As a variation of MMR, SMMC inherits the idea of marginal efficiency; however, instead of being a greedy algorithm like MMR and GMC, SMMC adopts a swap and convergent method. It initializes the result set R with k documents and then iteratively replaces the document with the minimal marginal contribution in R by the documents with the maximal marginal contribution in the candidate set S until the result set R has been optimal, which is a convergent process. SMMC avoids the weakness of MMR and GMC and can reach a better result which has been proven by the experiments. Moreover, as the coefficient c has a linear relation with k in our experiments, the time complexity of SMMC is in the same order with MMR, which is more efficient than GMC. That is, SMMC can outperform both MMR and GMC without sacrificing the time complexity heavily. Our future work will focus on how to optimize the initial result set R and whether a threshold is needed to optimize the convergent process. Moreover, as a preliminary experiment, the size of dataset used in our experiment is relatively small, we need to do further evaluation by larger datasets and evaluate the methods with more criteria in the next step.

Acknowledgements This work is supported by the National Basic Research Program of China (973 Program, Grant No. 2012CB315803), the National High-tech R&D Program of China (863 Program, Grant No. 2011AA01A101), the National Natural Science Foundation of China (Grant Nos. 61003100 and 60972011), and Research Fund for the Doctoral Program of Higher Education of China (Grant Nos. 20100002120018 and 20100002110033).

References

1. Yin, D., Xue, Z., Qi, X., Davison, B.D.: Diversifying search results with popular subtopics. Technical report, DTIC Document (2009)
2. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1998, pp. 335–336
3. Vieira, M.R., Razente, H.L., Barioni, M.C.N., Hadjieleftheriou, M., Srivastava, D., Traina, C., Tsotras, V.J.: On query result diversification. In: 2011 IEEE 27th International Conference on Data Engineering (ICDE), IEEE, 2011, pp. 1163–1174
4. Agrawal, R., Gollapudi, S., Halverson, A., Jeong, S.: Diversifying search results. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, ACM, 2009, pp. 5–14
5. Zhai, C.X., Lafferty, J.: A risk minimization framework for information retrieval. *Inform. Process. Manag.* **42**(1), 31–55 (2006)
6. He, J., Meij, E., de Rijke, M.: Result diversification based on query-specific cluster ranking. *J. Am. Soc. Inform. Sci. Technol.* **62**(3), 550–571 (2011)
7. Vee, E., Srivastava, U., Shanmugasundaram, J., Bhat, P., Yahia, S.A.: Efficient computation of diverse query results. In: 2008 IEEE 24th International Conference on Data Engineering (ICDE 2008), IEEE, 2008, pp. 228–236

8. Yu, C., Lakshmanan, L., Amer-Yahia, S.: It takes variety to make a world: diversification in recommender systems. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, ACM, 2009, pp. 368–378
9. Radlinski, F., Kleinberg, R., Joachims, T.: Learning diverse rankings with multi-armed bandits. In: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 784–791
10. Hu, J., Wang, G., Lochovsky, F., Sun, J., Chen, Z.: Understanding user's query intent with Wikipedia. In: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 471–480
11. Zhai, C.X., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2003, pp. 10–17
12. Clarke, C.L.A., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2008, pp. 659–666

Who Are We Talking About? Identifying Scientific Populations Online

Julie M. Birkholz, Shenghui Wang, Paul Groth, and Sara Magliacane

Abstract In this paper, we begin to address the question of which scientists are online. Prior studies have shown that Web users are only a segmented reflection of the actual off-line population, and thus when studying online behaviors we need to be explicit about the representativeness of the sample under study to accurately relate trends to populations. When studying social phenomena on the Web, the identification of individuals is essential to be able to generalize about specific segments of a population off-line. Specifically, we present a method for assessing the online activity of a known set of actors. The method is tailored to the domain of science. We apply the method to a population of Dutch computer scientists and their coauthors. The results when combined with metadata of the set provide insights into the representativeness of the sample of interest.

The study results show that scientists of above-average tenure and performance are overrepresented online, suggesting that when studying online behaviors of scientists we are commenting specifically on the behaviors of above-average-performing scientists. Given this finding, metrics of Web behaviors of science may provide a key tool for measuring knowledge production and innovation at a faster rate than traditional delayed bibliometric studies.

J.M. Birkholz (✉)

Network Institute, VU University Amsterdam

e-mail: j.m.birkholz@vu.nl

S. Wang

OCLC Research, Leiden, The Netherlands

e-mail: shenghui.wang@oclc.org

P. Groth • S. Magliacane

Computer Science Department, VU University Amsterdam

e-mail: p.t.groth@vu.nl; s.magliacane@vu.nl

1 Introduction

Traditionally, science is assessed using bibliometric techniques - indicators/metrics used to classify scientific output (e.g., publications) by performance and innovation, such as citation scores, or journal impact factors. Such methods rely on publication traditions including citations in a socially regulated environment. Communication and exchange of knowledge is also happening on the Web. The use of the Web as a virtual environment for interaction and exchange provides a ground for assessing impact through the study of traceable behaviors. Shifting research behaviors to the Web in multiple domains exposes more and more diverse processes of knowledge production and communication. Behavior online is traceable. Consequently scientists’ behaviors on the Web provide an additional, arguably complimentary, set of information traces to study science.

With the general rise of Internet use, an increasing portion of the researchers’ work takes place online via e-mail exchange, accessing online bibliographic databases and blogging and collaborating through e-Science tools, as well as general Web usage. A number of studies have begun to explore the online behaviors of science communities (see [16]). These research projects suggest that at face value, similar communication conventions, such as citation of academic articles, occur on both blogs [5] and Twitter [14]. The rise in these online activities suggests that an increasing portion of knowledge production and discussion is occurring on the Web, in parallel to the traditional practices of knowledge dissemination through academic publication, conferences, and the like. These online platforms have consequently been seen as a new terrain for exploring knowledge production, as well as science assessment, through assessing the “total impact” of a scientist’s work article [17].

The validity of these metrics remains debatable. This lack of established validity is an issue for the generation and testing of theories of research behavior and scientific development based on this data. One challenge in the validation of what these metrics represent is to determine who we are actually talking about. We pose the question: in science, who is represented online? And in what manner? We work to answer the level of validity that Web metrics provides in commenting on populations in science. We describe a method based on a combination of social science theory and computer science methods to evaluate the representativeness of a set of actors on the Web, compared to an off-line population. The method focuses on understanding whether a known sample is active on a variety of social platforms (e.g., Twitter, Mendeley, LinkedIn). Using a focused crawling approach, we examine a population of Dutch computer scientists’ and their coauthors’ presence on the Web. We show that

- A relatively low percentage of these scientists are verifiably active online.
- Those who are active on social websites are likely to be active on multiple sites.
- The scientists who are online are largely high performing.

The rest of the paper is organized as follows: We begin with a survey of studies of science online, highlighting the known differences between on- and off-line

communication. We then describe the method itself, which is followed by a description of the results of our study on Dutch computer scientists. Finally, we discuss the results and conclude.

2 Science Online

The practice of science is a practice of communication, an act of dissemination of knowledge to a set of peers. Scientific knowledge is disseminated through scientific publication in journals, conference proceedings, and books. Consequently, these outputs are used as measures of knowledge production in science through bibliometrics. Bibliographic records, through the use of repositories and databases, thus provide a wealth of knowledge on the system of science. These studies shed light not only on the performance or innovation through classification of outputs but also on the collaborative (coauthorship) behaviors of scientists (see work related to [3] and network studies from [12]) undertaken to produce knowledge.

The Web is a platform that expands on these practices of knowledge dissemination. As Wellman [22] suggested online behaviors mimic actual social behaviors. There are a number of outlets for scientists that cater to scientists [16]. An increasing amount of studies has specifically explored scientists online from how online behaviors spark creativity in science [4], differences between online and off-line behaviors of scientists [11], field differences in using computer-mediated communication [21], as well as the function of virtual communities in science [2]. Consequently we know scientists are online and have a variety of options for disseminating knowledge and interacting with peers outside of the traditional/formal publication of knowledge to hard copy text.

These research studies have set a path of inquiry about the effects of knowledge dissemination via the Web. A short list of tools exists that aids in further conceptualizing and understanding these online social behaviors in science, which include methods to track online readership [20], and impact metrics [13]. It is the use of tools that has been explored as way to complement bibliometric assessment.

Where publication trends are applicable for a field or discipline, the use of the Web is not a uniform nor required practice within science. Studies of the Web show that subsegments of populations are more likely to be on the Web, specifically younger ones [7]; correspondingly, certainly not everyone is on the Web, and particularly not everyone in science, but the composition is unknown. It is these behaviors that are of interest for this study, as they are the individual actions of the scientists that have emerged outside of traditional communication system.

The rise of scientists' use of the Web as a platform for sharing knowledge presents a number of methodological questions of validity and reliability to accurately reflect on how Web behaviors, in contrast to publication records, which include all active scientists within fields of practice, these individual behaviors present a case where activities are voluntary, relate to scientists/knowledge production. By validity we refer to the following - how to generalize a sample to a

larger population, how to generalize across settings, or how to generalize both across and within samples. Two aspects of validity need to be considered in science—external and internal. Internal validity relates to the suitability of measures for the population being examined. External validity refers to whether the results can infer causal mechanisms for an entire population [10], giving an indication of the generalizability of the research to a specific population. The external validity is dependent on the population that a study aims to generalize about. In order to accurately comment on scientists' practices online, one must be explicit about the validity of the sample. Specifically the representativeness of studies of these growing online behaviors of scientists on the Web remains a question not only to the description of the Web but also to the implications that we can infer from the sample on the Web. This validity is critical for generalizing and connecting research findings to other knowledge products. In this study, we work to define a method and test the results, developing a marker which defines the reliability of a Web sample in relation to a specific greater population of scientists under study.

3 Methodology

In this study a method is developed to identify individuals on a number of Web platforms. This description is followed by an explanation of the application to a known population. Statistical analysis is done on the results to reflect on the representativeness of the specific sample under study. We begin this section with a general description of the Web crawler method.

3.1 *Method Description*

Identifying scientists online could be achieved in one of the two ways: query a Web platform for users or start from a known sample that you know you want to generalize about and identify online. Both choices have a number of drawbacks. The general query does not ensure we can correctly identify the names of scientist to place them into a population (American scientist, historian, and so forth). The second option is privy to that scientists identify themselves online in a logical/disambiguatable way where we can link scientists through their full names; for example, John Smith is JohnSmith on Twitter and not thewhistler. Thus, both choices of sample select suffer from disambiguation issues. We do not aim to discuss disambiguation techniques in this paper, as that is a field in itself, but rather acknowledge that there are a number of techniques, of which techniques we integrate here.

Since we are interested in identifying scientists on multiple platforms and aim to compare them in some way we choose for the second sample technique.

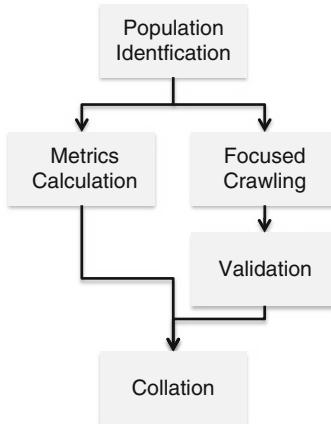


Fig. 1 Method overview

A Web crawler is suited to identify individual online. Although the crawler described could be used for other communities, it is particularly suited for investigating scientists on the Web. The key steps of the method and the data flow between those steps are shown in Fig. 1. We describe each of these steps in turn.

3.1.1 Population Identification

The method starts with identifying an already known population of scientists. Here, we are looking for a precise disambiguated set of individuals. An example list consists of names as well as other metadata such as affiliation. There are a number of mechanisms for gathering such data including using institutional directories and retrieving author lists from bibliographic databases (e.g., the Web of Science) or through membership lists from academic societies. Such lists provide a reasonable picture of off-line membership of the population under question.

3.1.2 Collecting Standard Science Metrics

Once a population list is obtained, the sample needs to be characterized in more detail. One could imagine a number of ways to characterize the population of scientists including age, gender, institutional type (e.g., teaching versus research university), tenure, and so forth. Here, we classify scientists according to the standard metrics used in science studies, namely, the h-index and citation scores, due to the difficulty of automatically collecting traditional variables of individuals from the Web. Regardless, the measures of performance provide information from which we can infer about their likely age/tenure through citations as well as performance and/or value of knowledge to a community through citation score and

h-index. These two standard measures of scientists' knowledge impact provide a representative manner to reflect on the population in terms of activity within the particular scientific community. These statistics also provide a basis for comparison when looking at the population online. Science metrics can be obtained from a number of databases such as Web of Science, Google Scholar, or Scopus.

3.1.3 Focused Crawling

In this study, we investigate the online behaviors of scientists' own enterprise/individual actions. This implies that we are not searching in online bibliographic databases for evidence of publications, or their academic institutional pages, but rather that we are isolating the existence of online activity on the social Web including blogs, microblogging, and activity on social platforms.

In science, blogs are often used as an alternative dissemination space for knowledge, whether presenting new knowledge, ideas, or research or sharing information. Microblogging tools, such as Twitter,¹ are used by academics for sharing academic links [15]. LinkedIn² is a known professional social networking site used by academics as well as other professionals. Mendeley³ is a bibliographic bookmarking service that aids scientists in organizing academic publications and links and in sharing them through profile libraries. SlideShare⁴ is a site used to upload presentations, providing an outlet for scientists to disseminate lectures and presentations. The diverse services offered among these Web platforms provide an outlet to incorporate/consider the multiple forms of knowledge dissemination on the Web.

To obtain information about the online behavior of individuals on these various sites, we developed a focused Web crawler. This crawler takes as input the list of persons from the population identification stage. It automatically performs the following process.

For each scientist, the Web crawler first goes over her/his homepage and searches for evidence of online presence such as links to her/his blog, "follow me" links for LinkedIn, or Twitter, as well as entries in Mendeley and SlideShare. If these activities are not mentioned on the personal homepage, the crawler individually searches LinkedIn, Twitter, Mendeley, and SlideShare to check whether she/he exists on these sites.

The crawler takes the scientist's name as the search string and submits the query to each of these websites, specifically searching for people. If the search returns zero hits, then we consider that the scientist does not have an account on the websites. Very often, the search returns multiple hits. This is due to the way the search strings are handled by the different websites. For example, LinkedIn and Twitter only return

¹<http://twitter.com/>.

²<http://www.linkedin.com/>.

³<http://www.mendeley.com/>.

⁴<http://www.slideshare.net/>.

accounts whose full names match exactly the target scientist, while Mendeley and SlideShare return the accounts whose name contains either the first name or the last name of the target scientist. For the latter situation, we filter out the accounts whose full names are not the same as the target scientist.

Apart from the binary information about whether one scientist is present in the above-mentioned social platforms, data can also be collected on the activity of the scientist. In this case, we focus on Twitter as an example. On Twitter we can also gather information about following and friend (follower) counts to say something additional about the Web behaviors of identified scientists.

The result of this step of the method is a list of possible accounts on these online sites corresponding to a given person in the input population.

3.1.4 Validation

The results of the focused crawl provide some evidence that a particular individual is present online but because multiple hits may occur we need to perform further validation to determine whether indeed an individual is present.

To validate the data a more detailed comparison is carried out based on the metadata of the returned results and the descriptive information of the target scientist. The query results usually contain some metadata of the returned accounts, such as the ID, full name, occupation, and location. We further check whether the location information of the query results matches the location information of the institution where the target scientist works are given in the population list. This is accomplished through the use of the Yahoo PlaceFinder Web service.⁵ This service provides the latitude and longitude as well as the country and city information for both institutions and the returned accounts. If the account is in the same country as the target scientist, we consider that this account belongs to the target scientist. If multiple accounts still are in question after the location check (scientists who share the same country), we consider that the target scientist exists in this Web platform without further distinguishing which accounts belongs to her/him. This is an approach in favor of recall.

3.1.5 Collation

The results of validation are collated together with the information obtained from the metrics calculation step. For each scientist in the population we have information about their membership in a community, their performance, their tenure, and their participation online. Based on this information, the standard statistical measures can then be run to analyze how the online activity relates to the any number of factors within the standard science metrics.

⁵<http://developer.yahoo.com/geo/placefinder/>.

3.2 Method Application

We apply the above method to a population of presently active computer scientists working in nine Dutch academic research institutions and their coauthors.

To perform the population identification step, we obtained a list of Dutch computer scientists from the Dutch NARCIS—National Academic Research and Collaborations Information System database.⁶

We expanded this list by querying the Digital Bibliography and Library Project (DBLP) [9]—an online bibliographic database for the field of computer science with publication streams from a number of top-rated journals, conference proceedings, and books within the field. We queried for all source scientists and coauthors from January 2007 (the year after Twitter’s inception) to March 2011. This query returned in the total of 4,984 individual scientists. They represent a list of all active scientists connected to Dutch computer scientists via coauthorship. This source list is the population we are interested in this study, that we can generalize about.

After the population identification step, traditional science metrics (h-index and citation score) for each scientist were collected. This data was acquired through Arnetminer [19], a search and mining services of computer science researchers which includes semantic data on names, contact information, homepage, and additional traditional scientometric statistics. Arnetminer identified 4,590 scientists (394 less than the first query), providing the metadata to aid in depicting how this sample represents computer scientists online. The use of Arnetminer rather than the Web of Science or Scopus is important for this specific population because of the better coverage of computer science-related publications in broader databases [1].

The population data was put into the focused Web crawler and collated with the data from Arnetminer. The specific results for this population in terms of each website are given in the next section.

4 Results

In this research, we conducted bivariate Pearson’s correlations to explore the external validity of a population of computer scientists. The correlations allow us to determine the relationships between two or more variables. The results from such tests identify both the significant relationships and the direction of relationships (positive or negative), thus providing the researcher with evidence to suggest, but not confirm, a causal relationship. Note that in the results table, a ** denotes statistical significance at the 0.01 level and a * denotes statistical significance at the 0.05 level.

Our findings of the sample of Dutch computer scientists and their coauthors from January 2007 to March 2011 show a relatively low percentage of scientists online.

⁶<http://www.narcis.nl/>.

Table 1 Descriptive statistics of how many author names can be found on the various services and whether those names can be validated

		LinkedIn		Mendeley		SlideShare	
		Frequency	Percent	Frequency	Percent	Frequency	Percent
Valid	0	851	18.5	4,121	89.8	4,522	98.5
	1	3,739	81.5	469	10.2	68	1.5
	Total	4,590	100.0	4,590	100.0	4,590	100.0
		Blog		Twitter			
Valid	0	4,438	96.7	4,503	98.1		
	1	152	3.3	87	1.9		
	Total	4,590	100.0	4,590	100.0		

As displayed in Table 1, each of the Web platforms is analyzed, describing the frequency and percentages represented in the sample under study, with the value ‘1’ representing a positive identification by the Web crawler of individuals and ‘0’ representing a negative identification by the crawler. For LinkedIn 81.5% of the sample can be identified, with 18.5% not found on the platform. In Mendeley, we see 89.8% of scientists not identified on this site and 10.2% confirmed. In SlideShare we find 98.5% not identified and 1.5% identified. Twitter identification is 1.9% and 98.1% not identified. Within this sample only 3.3% of scientists are identified as having a blog and 96.7% not identified. Thus, for the sample of Dutch computer scientists and their coauthors, only a small share is identified on all Web platforms, with the largest shares on LinkedIn and Mendeley.

The results show that computer scientists who are active on the Web are likely to be active on multiple sites (see Table 2, correlations of platforms). Within this population the existence on LinkedIn is related to being identified on Mendeley ($r = 0.124$), SlideShare ($r = 0.054$), and Twitter ($r = 0.058$) and having a blog ($r = 0.057$). The strongest relationship is existence on both LinkedIn and Mendeley. This strong positive correlation holds the same for the relationships of Mendeley to the Web platforms of SlideShare ($r = 0.072$), Twitter ($r = 0.064$), and blogs ($r = 0.090$), with the strongest relationship to Mendeley being the existence of a blog. SlideShare identification also strongly correlates with the existence on other platforms, Twitter ($r = 0.128$) and blog ($r = 0.088$), suggesting the strongest relationship between the use of SlideShare and Twitter. The use of Twitter and a blog also has a strong positive relationship ($r = 0.394$). Overall, those active on Web platforms have the tendencies to be active on multiple sites.

To further contextualize the use of online platforms, we present additional data from Twitter (see Table 3, Correlations to Twitter activity). The correlation results show that those on Twitter have high numbers of both followers and following ($r = 0.777$). We also investigate the relationships to the followers and following and the identification on other Web platforms: there is no significant relationship between LinkedIn (following: $r = 0.025$; followers: $r = 0.026$), although

Table 2 Correlations of platforms

		LinkedIn	Mendeley	SlideShare	Blog	Twitter
LinkedIn	Pearson Correlation	1	.124**	.054**	.057**	.058**
	Sig. (2-tailed)		.000	.000	.000	.000
	N	4,590	4,590	4,590	4,590	4,590
Mendeley	Pearson Correlation	.124**	1	.072**	.090**	.064**
	Sig. (2-tailed)	.000		.000	.000	.000
	N	4,590	4,590	4,590	4,590	4,590
SlideShare	Pearson Correlation	.054**	.072**	1	.088**	.128**
	Sig. (2-tailed)	.000	.000		.000	.000
	N	4,590	4,590	4,590	4,590	4,590
Blog	Pearson Correlation	.057**	.090**	.088**	1	.394**
	Sig. (2-tailed)	.000	.000	.000		.000
	N	4,590	4,590	4,590	4,590	4,590
Twitter	Pearson Correlation	.058**	.064**	.128**	.394**	1
	Sig. (2-tailed)	.000	.000	.000	.000	
	N	4,590	4,590	4,590	4,590	4,590

Table 3 Correlations to Twitter activity

		Twitter	Twitter following	Twitter friends
h-Index	Pearson Correlation	.118**	.083**	.058**
	Sig. (2-tailed)	.000	.000	.000
	N	4,590	4,590	4,590
Total citation	Pearson Correlation	.077**	.065**	.032*
	Sig. (2-tailed)	.000	.000	.031
	N	4,590	4,590	4,590
LinkedIn	Pearson Correlation	.058**	.025	.026
	Sig. (2-tailed)	.000	.092	.073
	N	4,590	4,590	4,590
Mendeley	Pearson Correlation	.064**	.091**	.108**
	Sig. (2-tailed)	.000	.000	.000
	N	4,590	4,590	4,590
SlideShare	Pearson Correlation	.128**	.143**	.121**
	Sig. (2-tailed)	.000	.000	.000
	N	4,590	4,590	4,590
Blog	Pearson Correlation	.394**	.186**	.176**
	Sig. (2-tailed)	.000	.000	.000
	N	4,590	4,590	4,590
Twitter	Pearson Correlation	1	.375**	.399**
	Sig. (2-tailed)		.000	.000
	N	4,590	4,590	4,590
Twitter following	Pearson Correlation	.375**	1	.777**
	Sig. (2-tailed)	.000		.000
	N	4,590	4,590	4,590

Table 4 Performance measures

		LinkedIn	Mendeley	SlideShare	Blog	Twitter
h-Index	Pearson Correlation	.070**	.021	.038*	.154**	.118**
	Sig. (2-tailed)	.000	.159	.011	.000	.000
	N	4,590	4,590	4,590	4,590	4,590
Total citations	Pearson Correlation	.034*	-.009	.004	.103**	.077**
	Sig. (2-tailed)	.020	.528	.805	.000	.000
	N	4,590	4,590	4,590	4,590	4,590

significant positive relationships exist between Mendeley (following: $r = 0.091$; followers: $r = 0.108$), SlideShare (following: $r = 0.143$; followers: $r = 0.121$), and blog identification (following: $r = 0.186$; followers: $r = 0.176$). The strongest relationship between following and followers is with SlideShare and blog use. This suggests a positive relationship between the number of followers and following and activity on other sites.

In order to further understand what is driving the positive relationships observed, it is necessary to also investigate the relationships of traditional performance measures to describe what extent this population online is generalizable (externally valid) to (Dutch) computer science as a field. These results are presented in Table 4. The results show that those online are largely top-ranking scientists; the higher the h-index, the more likely to be found on LinkedIn ($r = 0.070$), SlideShare ($r = 0.038$), and Twitter ($r = 0.118$) and to have a blog ($r = 0.154$). There is no significant relationship between identification on Mendeley and a high h-index score ($r = 0.021$). A number of these relationships are also confirmed in regard to citation score (number of citations), which is used as a measure for performance and as a proxy for tenure. A positive and significant relationship exists between citation score and LinkedIn ($r = 0.034$), Twitter ($r = 0.077$), and identification of a blog ($r = 0.103$). The results suggest that among this community of computer scientists, the measuring of Web behaviors of the scientists' own enterprise is representative of the dynamics of scientists who have both a higher tenure and a higher performance.

5 Discussion

Before discussing the results, it is important to reflect on the limitations of the Web crawler. The implementation of the Web-crawling tool, which takes the names as input and automatically searches the presence of these people on the Web, greatly increases the amount of people who can be analyzed, thus providing a more reliable extension to manual tests of validity of specific communities. The limitations of the Web crawler include disambiguation issues and API constraints and limits. The most common disambiguation issue is the lack of meaningful IDs that match the full names. Search APIs also present some limits to searches for scientists. APIs sometimes use OR instead of AND to increase the recall, which presents a problem

in quickly and reliably locating a name among the results of crawler, thus requiring additional knowledge about individuals. LinkedIn returns entities whose names contain the full string of searched names, while Mendeley and SlideShare return people whose names contain either first name or last name. In the development of the Web crawler, this was overcome through the integration of geolocation data to identify individuals within the returned set. Additionally, some APIs have limits to queries per hour, which constrains the speed of the crawler. The constraints of the Web crawler potentially affect the low identification of the sample online. Further techniques could be developed in the Web crawler to provide more certainty about scientists' presence on these sites. In particular, we are looking at building profiles of scientists based on publications and matching these to profiles produced from websites. Such an approach may help in increasing both the recall and precision of the method [6].

This test showed that the largest percentages of scientists can be identified on LinkedIn and Mendeley, with much lower identification on SlidesShare, Twitter, and blogs. Although this could be related to the techniques of the Web crawler, we suggest that it is rather associated to the services provided on these sites. LinkedIn provides a networking tool for professionals to connect on the Web, which we argue reflects the traditional communication patterns of scientists [8] staying in touch whether through e-mail, phone contact, or face-to-face interaction. Mendeley provides an online bibliographic bookmarking tool that again scientists would be in need of whether on- or off-line to categorize and organize publications. The other three platforms, SlideShare, Twitter, and blogs, are forms of modern communication and thus new ways of disseminating knowledge. They are not innate to the knowledge dissemination practices of the past several hundred years, unlike interacting with others (LinkedIn), as well as reading, reflecting, and reacting to new knowledge in the field (Mendeley).

The significant positive relationships observed in this activity sample on multiple sites give us reason to hypothesize that scientists are using Web platforms in their work, thus providing further support to our previous speculation of tendencies in using specific platforms that facilitate traditional knowledge production. To confirm this, further research should be completed to explore the online presence of other fields and samples of scientists, shedding light on the overall prevalence of online activity in science. Additionally, the results from Twitter bring to light a possible feedback effect of using multiple Web platforms. This effect should be explored further to address not only the mechanism of using such Web platforms but also how visibly increasing on one platform relates to other Web behaviors. Consequently, longitudinal research of scientists' online activities would provide insight into the effect of the use of multiple Web platforms.

This study has shown that when using Web data we oversample the dynamics of top scientists, bringing to light the importance of considering validity questions of Web data to study social phenomena. Thus, when talking about implications of altmetrics [18] or analyzing behavior on these social media sites, we need to be explicit about who we can generalize about and how these reflect to greater patterns in science. If this oversampling of top scientists holds true for other fields, the

use of Web data may provide a reliable, faster tool in measuring, predicting, and understanding trends in science, compared to delayed bibliometric analysis of top scientists.

6 Conclusion

Our results present a depiction of life on the Web for the field of computer science. From an analytics perspective, we have worked to develop a method that provides a tool for reflecting on population as to reliably provide a level of external validity (generalizability) to a greater community of actors. Additionally, it emphasizes the importance and continued need of interdisciplinary research to assess such questions.

In summary, we propose that when measuring the scientific impact and contribution of one's work on the Web, it is necessary to be clear about the level of external validity of these Web activities in order to better infer trends in science. The method described here is a first step to achieving this goal.

References

1. Bar-Ilan, J.: Web of science with the conference proceedings citation indexes: the case of computer science. *Scientometrics* **83**(3), 809–824 (2010)
2. Colazzo, L., Molinari, A., Villa, N.: From e-learning to “co-learning”: the role of virtual communities. In: Kendall, M., Samways, B. (eds.) IFIP International Federation for Information Processing 281, pp. 329–338 (2008)
3. de Solla Price, D.J.: Networks of scientific papers. *Science* **149**, 510–515 (1965)
4. Dunbar, K.: How scientists think: Online creativity and conceptual change in science. In: Ward, T.B., Smith, S.M., Vaid, S. (eds.), *Conceptual Structures and Processes: Emergence, Discovery and Change*, pp. 461–493. American Psychological Association, Washington, DC (1997)
5. Groth, P., Gurney, T.: Studying scientific discourse on the web using bibliometrics: A chemistry blogging case study. In: *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*. Raleigh, NC: US, April 26–27th, 2010
6. Gurney, T., Horlings, E., van den Besselaar, P.: Author disambiguation using multi-aspect similarity indicators. *Scientometrics*, 1–15 (2012)
7. Hampton, K., Sessions-Goulet, L., Rainie, L., Purcell, K.: Social networking sites and our lives. Pew Res. Center (2011)
8. Latour, B., Woolgar, S.: *Laboratory Life: The Social Construction of Scientific Facts*. Sage Publications, Los Angeles (1979)
9. Ley, M.: Dblp - some lessons learned. *PVLDB* **2**(2), 1493–1500 (2009)
10. Lucas, J.W.: Theory-testing, generalization, and the problem of external validity. *Socio. Theor.* **21**, 236–253 (2003)
11. Mika, P.: Social networks and the semantic web. In: *Web Intelligence*, pp. 285–291(2004)
12. Newman, M.E.J.: The structure of scientific collaboration networks. In: *Proceedings of the National Academy of Sciences* 98, pp. 404–409 (2001)
13. Neylon, C., Wu, S.: Article-level metrics and the evolution of scientific impact. *PLoS Biol.* **7**(11: e1000242) (2009)

14. Priem, J., Costello, K.: How and why scholars cite on twitter. In: Proceedings of the 73rd ASIS&T Annual Meeting, Pittsburgh, PA, (2010)
15. Priem, J., Costello, K., Dzuba, T.: Prevalence and use of twitter among scholars. In: Metrics 2011: Symposium on Informetric and Scientometric Research. Poster, New Orleans, LA, October 2011
16. Priem, J., Hemminger, B.M.: Scientometrics 2.0: Toward new metrics of scholarly impact on the social web. First Monday (7) (2010)
17. Priem, J., Parra, C., Piwowar, H., Groth, P., Waagmeester, A.: Uncovering impacts: a case study in using altmetrics tools. In: Workshop on the Semantic Publishing SePublica 2012 at the 9th Extended Semantic Web Conference, pp. 1–5(2012)
18. Priem, J., Taraborelli, D., Groth, P., Neylon, C.: Alt-metrics: A manifesto, (v.1.0). <http://altmetrics.org/manifesto>, 26 October 2010
19. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pp. 990–998. ACM, New York, NY, (2008)
20. Taraborelli, D.: Readermeter: Crowdsourcing research impact. Academic Productivity, 2010. Retrieved April 5, 2011, from: <http://www.academicproductivity.com/2010/readermeter-crowdsourcing-research-impact/>
21. Walsh, J.P., Bayma, T.: Computer networks and scientific work. Soc. Stud. Sci. **26**, 661–703 (1996)
22. Wellman, B., Gulia, M.: Virtual communities as communities. In: Smith, M.A., Kollock, P. (eds.) Communities in Cyberspace.. Routledge, New York (1999)

Study of Ontology Debugging Approaches Based on the Criterion Set BLUEI²CI

Qiu Ji, Zhiqiang Gao, and Zhisheng Huang

Abstract Dealing with logical conflicts is one of the key tasks in the Semantic Web applications, where ontology debugging plays an important role. Debugging an ontology helps users to understand and resolve the conflicts. Although there are some works to compare the debugging approaches, those recently proposed approaches are not included and a relatively systematic and complete criteria set is missing. Thus, it is still difficult for users to choose an appropriate approach during the development of practical application systems. In this paper, we propose the criteria set BLUEI²CI which is used to compare the existing debugging approaches. Especially when comparing the approaches w.r.t. efficiency, the experiments are conducted over various selected data sets. Finally, we conclude a set of recommendations to guide the users for choosing appropriate debugging approaches.

1 Introduction

It is widely believed that ontologies play an important role for the formal representation of knowledge in the Semantic Web, especially when Web Ontology Language (OWL) becomes a W3C recommendation standard. So far, an increasingly large number of OWL ontologies have been developed and ontologies have been used in various information science fields like knowledge engineering [20]. In these ontologies, logical conflicts are always an unavoidable problem as anyone is allowed

Q. Ji (✉) • Z. Gao

School of Computer Science and Engineering, Southeast University, Nanjing, China

Key Laboratory of Computer Network and Information Integration
(Southeast University), Ministry of Education, Nanjing, China

e-mail: jiqiu@seu.edu.cn; zqgao@seu.edu.cn

Z. Huang

Department of Computer Science, Vrije Universiteit Amsterdam, the Netherlands

e-mail: huang@cs.vu.nl

to say anything under the circumstance of the Web. Here, logical conflicts usually consist of inconsistency and incoherence.¹ Particularly, the following cases may cause logical conflicts: First, it is an error-prone effort to build ontologies manually or automatically, especially for those large-scale and expressive ontologies. Second, since ontologies keep evolving, newly added information may be in conflict with the original ontologies [13]. Third, the automatically generated ontology mappings always contain logical conflicts [21].

Dealing with logical conflicts is one of the key tasks in the Semantic Web applications. On the one hand, for an inconsistent ontology O , a standard reasoner suffers explosive problem. Namely, everything can be entailed from O . On the other hand, incoherence is a potential cause of inconsistency. That is, an ontology O' becomes inconsistent if an individual is declared to belong to an unsatisfiable concept in O' . Thus, dealing with both kinds of logical conflicts is a meaningful and important task. In this task, ontology debugging plays an important role to provide help to resolve or understand the conflicts. Currently, various debugging approaches have been proposed. But only a few works study the comparison of the debugging approaches (see [6, 26]). In [6], a survey of existing approaches to dealing with inconsistency is given. In [26], several ontology debugging tools like SWOOP [18] are evaluated with respect to the efficiency.

However, some problems have not been solved by these works, which makes it difficult for users to choose appropriate debugging approaches during the development of practical application system. One problem is that several recent approaches like those in [10, 15] are not included. Another problem is that a relatively systematic and complete criteria set is missing. The third problem is that each selected unsatisfiable concept contains only a few minimal conflict subsets (i.e., no more than 6). Therefore, in this paper, we propose a criteria set BLUEI²CI (see Sect. 3 for more details) for comparing various ontology debugging approaches. The criteria set can be divided into two categories. One category contains the criterion which takes the users' experience and specific data sets into account. We call this category as experimental comparison. Other criteria belong to the other category. Based on the criteria set, we compare and analyze the existing debugging approaches. Note that the involved data sets come from real life or they are developed by using machine learning techniques or ontology mapping. Besides, some selected unsatisfiable concepts contain more than 60 minimal conflict subsets. According to the comparison, we conclude several recommendations for users to select the most suitable debugging approaches.

This paper is organized as follows: We provide background knowledge in Sect. 2. Then we present the criteria set for comparing the debugging approaches in Sect. 3. Sections 4 and 5 compare the existing debugging approaches based on the two categories of the criterion separately. Then a set of recommendations is concluded in Sect. 6. Finally, we conclude this paper in Sect. 7.

¹An inconsistent ontology has no model. An incoherent ontology has at least one concept which is interpreted as an empty set.

2 Background Knowledge

Since OWL is a Description Logic (DL)-based ontology language and OWL DL is an important sublanguage of OWL, we introduce the main notions w.r.t. logical conflicts in an OWL DL ontology. As the space limit, we assume the readers are familiar with DLs (see [1] for more details).

Definition 1 (Inconsistent Ontology). [6] An ontology O is inconsistent iff O has no model.

For inconsistent ontologies, the conclusion obtained by applying a standard DL reasoner may be completely meaningless.

Definition 2 (Unsatisfiable Concept). [11] A named concept C in an ontology O is unsatisfiable iff for each model \mathcal{I} of O , $C^{\mathcal{I}} = \emptyset$. Otherwise, C is satisfiable.

Definition 3 (Incoherent Ontology). [11] An ontology O is incoherent iff there exists at least one unsatisfiable concept in O . Otherwise, O is coherent.

For an incoherent ontology O , the task of a debugging approach is usually to compute a set of minimal unsatisfiability-preserving subsets (MUPS) for an unsatisfiable concept in O .

Definition 4 (MUPS). [24] Assume C is an unsatisfiable concept in a consistent ontology O . A subset $O' \subseteq O$ is a MUPS of C if C is unsatisfiable in O' and satisfiable in every $O'' \subset O'$.

For an inconsistent ontology, the ontology debugging task is to compute a set of minimal inconsistent subsets (MIS).

Definition 5 (MIS). For an inconsistent ontology O , a subset $O' \subseteq O$ is a MIS if O' is inconsistent and every $O'' \subset O'$ is consistent.

Notes: First, since most of the debugging approaches deal with incoherence in TBoxes [6], we also follow this. The inconsistency will be discussed over an entire ontology. Second, the unsatisfiability of a concept C in an OWL DL ontology is a special case of an entailment [17] since C can be represented as an entailment $C \sqsubseteq \perp$. Therefore, those approaches to computing explanations for an entailment ϕ can be applied to compute MUPS in theory. However, in practice their implementations are different and thus a tool to compute explanations cannot be used directly to compute MUPS. Here, an explanation of ϕ means a minimal set of axioms to infer ϕ .

3 The Criteria Set to Compare Ontology Debugging Approaches

The proposed criteria set to compare ontology debugging approaches contains seven criteria. First, we adopt some criteria from the work [22] to compare different revision operators. Among their proposed criterion, three can be used to compare ontology debugging approaches. Specifically, they are implementation (I), language dependence (L), and inconsistency vs. incoherence (I^2). Second, we keep the criteria, efficiency (E), which is frequently used in the works about ontology debugging like [19, 25]. Third, we propose more criteria to constitute a relatively complete and systematic criterion set. They are black-box vs. glass-box (B), completeness (C), and usability (U). For simplicity, we use BLUEI 2 CI to indicate the criterion set. In the following, we will explain these criteria one by one according to the alphabetical order.

Black-box vs. glass-box (B): The black-box approaches do not rely on any reasoner. But a reasoner needs to be invoked many times to check the satisfiability of a concept. The glass-box approaches modify the internal algorithm of a reasoner and usually perform faster than the black-box approaches. There is a trade-off between efficiency and the independence of a reasoner.

Completeness (C): If an approach can find all minimal conflict subsets, we say it is complete. Otherwise, it is incomplete. In practice, some approaches give up the completeness to improve the efficiency. Thus, there is also a trade-off between completeness and efficiency.

Efficiency (E): Efficiency is usually measured by time, namely, the execution time to finish a specific debugging task (e.g., the computation of all MIS). To test the efficiency, various data sets are always needed. Besides, it would be better to select those data sets with different expressivity, domains, numbers of axioms, and numbers of conflicts. This criterion is the most widely used one.

Implementation (I): This criterion mainly concerns whether a debugging approach has been implemented or not. The implementation can be used by users through a graphical user interface (GUI), the application programming interface (API), or command line (CL.). To use the implementation, it is always assumed that the users have the basic knowledge about ontologies. Particularly, we assume the users have the programming skills when using API.

Inconsistency vs. incoherence (I^2): The incoherence is always an implicit factor to cause inconsistency and debugging incoherent ontologies can help to debug inconsistent ontologies. For the two kinds of logical conflicts, different debugging algorithms need to be designed. Among the existing debugging approaches, most of them are dealing with incoherence.

Language dependence (L): Some ontology debugging approaches are designed for a specific DL sublanguage. That is, if the ontology to be tested has the DL

expressivity beyond the given sublanguage, then some conflicts may not be detected. Thus, it is very important to distinguish the supported DL sublanguage for a debugging approach.

Usability (U): This criterion assumes that a tool is available for a specific debugging approach. It considers whether this tool can be easily installed and used and how much background knowledge is needed. In this paper, we use “easy,” medium,” and “hard” to roughly indicate the degree of usability. The usability of a tool often directly influences the tool’s popularity as people always prefer to use the tools with usability “easy”.

4 Comparison of Ontology Debugging Algorithms

In this section, we compare the existing debugging approaches based on the criterion which is not used for experimental comparison. Namely, the criteria black-box vs. glass-box, completeness, implementation, inconsistency vs. incoherence, and language dependence will be discussed.

4.1 *The Approaches to Computing MUPS*

4.1.1 The Glass-Box Approaches

The first debugging approach is originally proposed in [24] and is further developed in [25] for incoherent ontologies. The approach is designed for unfoldable \mathcal{ALC} to find all MUPS for a concept. Here, an unfoldable \mathcal{ALC} TBox T indicates every concept axiom ϕ in T satisfies the following conditions: the concept C_L in the left of ϕ is atomic and the concept in the right of ϕ cannot reference C_L directly or indirectly. When computing MUPS, this approach makes use of the rules as long as possible to construct a Tableau and excludes those irrelevant axioms. Then a MUPS can be reduced. A tool $MUPSTER$ is developed and can be used through command line.

As the approach in [24] restricts the DL language to \mathcal{ALC} , the approach proposed in [19] extends it to $\mathcal{SHIF}(\mathcal{D})$ and provides a GUI called SWOOP.² This approach can compute one MUPS for a concept. After that, the approach in [19] is further extended to $\mathcal{SHOIN}(\mathcal{D})$ (i.e., OWL DL) in [17] and the extended approach satisfies completeness. The extended approach is implemented by modifying the Tableau algorithm in Pellet³ and can be used by API.

²<http://www.mindswap.org/2004/SWOOP/>.

³<http://clarkparsia.com/pellet/>.

In [2], the authors propose a general debugging approach which is not restricted to a specific DL sublanguage. Specifically, they give a general definition of “Tableau algorithm” and then extend a general tableau to a pinpointing algorithm. Here, axiom pinpointing is the task to compute MUPS [24]. Since this approach has some limitations about the tableau-based algorithms, an automata-based approach is proposed in [3] to obtain pinpointing algorithms. Both of the approaches satisfy the completeness. But they are only theoretical work.

4.1.2 The Black-Box Approaches

In [25], a black-box approach based on selection functions is proposed. Here, a selection function is used to select axioms which are responsible for the unsatisfiability of a concept. Once such a set of axioms is found, a MUPS can be computed by removing the redundant axioms. This approach cannot find all MUPS. The corresponding tool is DION and it can be accessed by using the webpage-based GUI.

Another black-box approach is developed in [17]. It can find all MUPS by constructing a hitting set tree [23]. A single explanation can be found by selecting a set of relevant axioms and then pruning the set to find one MUPS. This approach has been implemented with OWL API and can be used by API.

In [4], a simple black-box approach is developed to compute a MUPS by pruning a DL \mathcal{EL}^+ TBox directly. Obviously, the efficiency of this approach is a problem. To improve its efficiency, a binary search algorithm is applied in [5]. Both approaches do not satisfy the completeness. To make these approaches satisfy the completeness, the work in [28] extends them by constructing a hitting set tree. These approaches have been implemented based on CEL reasoner.⁴ But the implementations are not available for public.

Since understanding all MUPS is a tedious effort, the work in [16] proposes a method (marked as MUPS-REL) based on selection functions to compute the explanations of an entailment ϕ according to their relevance degree to ϕ . The approach can be interrupted at any time and still compute those explanations which are relevant to ϕ up to some degree. This work provides an algorithm to compute all explanations for an entailment. All functionalities are implemented with KAON2 API⁵ and can be accessed by command line.

Recently, an efficient approach (marked as MUPS-PAT) to computing MUPS has been proposed in [15]. It designs specific algorithms to instantiate different patterns and thus a set of MUPS can be efficiently returned. Although this approach cannot ensure to return all MUPS for a concept, the experimental results show that almost all MUPS can be found in most cases. It is implemented with OWL API and can be used by command line.

⁴<http://lat.inf.tu-dresden.de/systems/cel/>.

⁵<http://kaon2.semanticweb.org/>.

4.1.3 Other Debugging Approaches

There are some works to compute explanations of the unsatisfiability which may not be MUPS to improve the efficiency. In work [30], the authors provide a set of error patterns to explain the unsatisfiability and then propose some heuristic rules to instantiate the patterns. This approach can only find one explanation which may not always be a MUPS. It has been implemented and provides a plug-in in the ontology editor Protege.⁶ Also, the work in [4] proposes a glass-box approach to computing a superset of a MUPS. The implementation is not available which is based on CEL.

4.2 *The Optimizations of the Approaches to Computing MUPS*

The mentioned debugging approaches are all applied on an ontology directly. In such case, the efficiency is still a problem. To deal with this problem, the modularization-based optimizations are proposed. That is, for an unsatisfiable concept C in an ontology O , we first extract a sub-ontology M from O such that M covers a set of MUPS of C . Then a debugging approach is applied on M .

The work in [4] to compute a superset of a MUPS can be seen as an optimization. But the extracted module only covers one MUPS. In [29], a locality-based modularization method is adopted. This work proves that the extracted module can cover all explanations of an entailment in $\mathcal{SHOIQ}(\mathcal{D})$. It is implemented with OWL API. The work in [5] proposes a reachability-based modularization method and proves that an extracted module can cover all explanations of an entailment in \mathcal{EL}^+ . It is implemented in CEL. This extraction method has been proved to be equal to the locality-based modularization method for \mathcal{EL}^+ ontology in [27]. Since the locality-based modularization method does not consider the super concept in a subsumption entailment, the extracted module may not be minimal. Thus, the authors in [10] propose a goal-directed modularization method (marked as Module_{GOAL}) which is designed for $\mathcal{SHOIQ}(\mathcal{D})$. It is implemented with GNU C++ and can be used by command line.

4.3 *The Approaches to Computing MIS and the Optimization*

Most of the existing debugging approaches are developed to compute MUPS or explanations for an entailment in a TBox, while the works to compute MIS are relatively few. One main reason is that it is more challenging to compute MIS than compute MUPS. Because we have a starting point to select relevant axioms when computing MUPS of a concept C , which does not hold when computing MIS [14].

⁶<http://protege.stanford.edu/>.

Another main reason is that incoherence is a potential cause of inconsistency, and thus, dealing with incoherence can help to deal with inconsistency. One representative approach to computing MIS is the work in [14]. It proposes a divide-and-conquer approach to computing a single MIS and it uses the hitting set tree algorithm to compute all MIS. This debugging approach is a black-box approach and is implemented with OWL API.

Similar to the approaches to computing MUPS, the approaches to computing MIS can also use the modularization methods to improve their efficiency. In [9], a decomposition-based modularization method (marked as Module_{DECO}) is presented which satisfies that the union of the found MIS in each extracted module equals to all MIS in the entire ontology. It is implemented with GNU C++ and can be used by command line.

5 Experimental Comparison

In this section, we compare the existing debugging approaches by considering personal experience and specific data sets with the criterion usability and efficiency. In the experiments, we choose Windows as the platform as it is supported by most of the tools. All experiments were performed on a laptop with 2.13 GHz Intel(R) Core(TM) i3 CPU and 2.00 GB of RAM. Sun's Java 1.6.0 was used for Java-based tools and the maximum heap space was set to 1GB. When performing a specific debugging task, the timeout is set to be 1 h.

5.1 Installation and Usage of the Debugging Tools

All of the available debugging tools can be found in the column of implementation in Table 2. For DION, it requires the installation of the softwares SWI-Prolog,⁷ and XDIG.⁸ Besides, its inputs and outputs are all in DIG syntax⁹ which requires the users to be familiar with DIG. Thus, it is a little hard to use this tool. MUPS_{TER} and CEL accept inputs in KRSS syntax and only support Linux operating system. Therefore, it is not easy to use for users who are familiar with OWL ontologies and Windows operating system. For Module_{GOAL} and Module_{DECO}, the users are required to have some basic knowledge about MySQL. Thus, it is also not very easy to use them. As for other tools, they can be easily installed without having any extra knowledge. Besides, it is quite convenient to use them for performing a specific debugging task.

⁷<http://www.swi-prolog.org>.

⁸<http://wasp.cs.vu.nl/sekt/dig/>.

⁹<http://dig.sourceforge.net/>.

Table 1 Data sets

Ontology ID	Ontology	TBox Size	Classes	Properties	Expressivity	# of UC All	# of MUPS		
							Min	Max	Average
O1	chemical	114	48	20	$\mathcal{ALC}\mathcal{N}(\mathcal{D})$	37	2	26	11
O2	economy	580	339	53	\mathcal{EL}^+	51	1	2	1
O3	miniTambis	173	183	44	$\mathcal{ALC}\mathcal{N}$	30	1	2	1
O4	cmt-ekaw	465	103	92	$\mathcal{SHIN}(\mathcal{D})$	29	1	13	2
O5	confot-ekaw	440	112	69	$\mathcal{SHIN}(\mathcal{D})$	13	12	21	14
O6	confot-sigkdd	323	88	64	$\mathcal{SHIN}(\mathcal{D})$	5	1	1	1
O7	km-4000	4,000	4,022	310	DL-Lite	803	C1: 17, C2: 29 C3: 68, C4: >100		

In this table, “UC” indicates unsatisfiable concepts

5.2 Data Sets

To compare the tools w.r.t. efficiency, we select various data sets (see Table 1). Ontologies O1, O2, and O3 come from real life and are publicly available. O4, O5, and O6 are obtained by ontology mapping. Namely, such an ontology can be constructed by merging a pair of ontologies and the mapping between them by interpreting the mapping as DL axioms. The pairs of ontologies and their mappings come from the conference track in OAEI’2010 and the mappings are generated by the participant AROMA [8]. Ontology O7 is obtained by applying machine learning techniques on a text corpus consisting of abstracts from the “knowledge management” information space of the BT Digital Library. Since O7 contains quite a lot unsatisfiable concepts (i.e., 803), we randomly choose 4 unsatisfiable concepts such that the numbers of their corresponding MUPS are significantly different. The selected concepts are process_innovation (C1), engineering_knowledge_management_system (C2), environment (C3), and law_firm (C4). More details can be found in Table 1, where the expressivity is obtained by using SWOOP. For C4 in ontology O7, we do not know the exact number of its MUPS by using the existing debugging approaches. Within the limit time (i.e., 1 h), the approach MUPS-REL can find more than 100 MUPS.

5.3 Experimental Results

We first introduce the debugging approaches to be compared. First, we do not use any extraction methods for our experiments, because only one extraction method can be used for debugging an incoherent or inconsistent ontology. Specifically, the implementation of Module_{GOAL} does not support the inputs in the form of a concept or $C \sqsubseteq \perp$. CEL only supports Linux. Thus, only the locality-based modularization method can be used for debugging incoherent ontologies. Second, no experiments

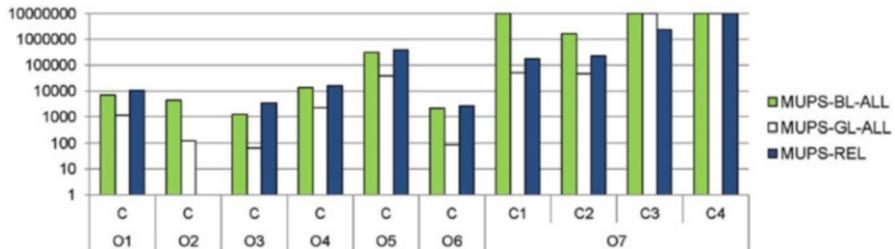


Fig. 1 The average time in million seconds to compute all MUPS of a concept

will be conducted on inconsistent ontologies since only one approach to computing MIS is available. Third, only those approaches to computing all MUPS are evaluated w.r.t. efficiency for fair comparison. Thus, the approaches to be compared here include the glass-box approach (marked as MUPS-GL-ALL) and the black-box approach (marked as MUPS-BL-ALL) in [17] and the relevance-directed approach in [16] (i.e., MUPS-REL).

Figure 1 shows the average time to compute all MUPS of a concept for each selected debugging approach, where C is a dummy concept. It is noted that: First, for concepts C1, C3, and C4, some debugging approaches cannot finish the debugging task within the time limit (i.e., 1 h). For convenience, we use 10000000 instead of 3600000 (i.e., 1 h) in the figure. Second, since ontology O2 cannot be parsed by KAON2 API, MUPS-REL fails to return any results. Third, when applying MUPS-REL, we use the subsumption-based selection function for ontologies O2 and O7 and use the signature-based selection function for others.

We can observe that MUPS-GL-ALL is more efficient than others and MUPS-BL-ALL performs relatively faster than MUPS-REL in most cases. First, MUPS-GL-ALL is more efficient because the glass-box approaches are designed by modifying the internal algorithm of a reasoner and the black-box approaches are developed based on a reasoner. Second, MUPS-BL-ALL is more efficient than MUPS-REL because MUPS-REL spends extra time to extract sub-ontologies by a selection function and then compute MUPS in these sub-ontologies. Thus, when a concept has relatively few MUPS, the extra time becomes obvious. For example, for ontology O5, the average time to compute all MUPS of a concept for MUPS-BL-ALL is 305 s. But it is 395 s for MUPS-REL.

We also observe that MUPS-REL performs better than others when an ontology has more axioms and a concept has many MUPS (e.g., more than 60). Because in such case, the extra time to select sub-ontologies is relatively minor by comparing with the total time to compute MUPS for a concept. Besides, MUPS-GL-ALL becomes inefficient as it works on the entire ontology. Take C1 in ontology O7 as an example. It takes MUPS-REL 175 s to finish the computation of MUPS. But MUPS-BL-ALL cannot finish the task within 1 h. For C4, although all approaches cannot finish the task within 1 h, MUPS-REL can find more MUPS (i.e., 108) than other approaches (i.e., 68 for MUPS-GL-ALL and 60 for MUPS-BL-ALL).

6 The Discussion of Comparison Results

In this section, we conclude a set of recommendations for users to choose appropriate debugging approaches in different use cases. These recommendations can be made according to the previous comparison of the existing debugging approaches in Sects. 4 and 5. Table 2 provides a concise conclusion of the comparison of the debugging approaches based on BLUEI²CI. In this table, all criteria in BLUEI²CI except efficiency have been compared. It is because the efficiency of a debugging

Table 2 The comparison of the existing ontology debugging approaches based on the proposed criterion set

Debugging Approaches	Bl. vs.			Ih. vs. Language		
	Gl.	Completeness	Implementation	Is.	Dependence	Usability
Approaches to computing MUPS						
[25] Schlobach, JAR07						
Gl.	Complete	CL. (MUPSTER)	Ih.	\mathcal{ALC}	Medium	
Bl.	Incomplete	GUI (DION)	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Hard	
[19] Kalyanpur, JWS05	Gl.	Incomplete	GUI (SWOOP)	Ih.	$\mathcal{SHIF}(\mathcal{D})$	Easy
[17] Kalyanpur, ISWC07	Gl.&Bl.	Complete	API (Pellet)	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Easy
[2] Baader, TABLEAUX07	Gl.	Complete	n.a.	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	n.a.
[3] Baader, IJCAR08	Gl.	Complete	n.a.	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	n.a.
[4] Baader, KI07	Bl.	Incomplete	Based on CEL	Ih.	\mathcal{EL}^+	n.a.
[5] Baader, KR-MED08	Bl.	Incomplete	Based on CEL	Ih.	\mathcal{EL}^+	n.a.
[28] Sunti., thesis09	Bl.	Complete	Based on CEL	Ih.	\mathcal{EL}^+	n.a.
[16] Ji, ASWC09	Bl.	Complete	CL. (MUPPS-REL)	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Easy
[15] Ji, JIST11	Bl.	Incomplete	CL. (MUPPS-PAT)	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Easy
Other approaches to computing MUPS						
[4] Baader, KI07	Bl.	Incomplete	Based on CEL	Ih.	\mathcal{EL}^+	n.a.
[30] Wang, ISWC05	Bl.	Incomplete	GUI (Protege)	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Easy
Approaches to computing MIS						
[14] Horridge, SUM09	Bl.	Complete	API (Pellet)	Is.	$\mathcal{SHOIN}(\mathcal{D})$	Easy
Approaches to extracting modules						
[10] Du, ISWC09	n.a.	n.a.	CL. (Module _{GOAL})	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Medium
[12] Cuenca, WWW07	n.a.	n.a.	API (Pellet)	Ih.	$\mathcal{SHOIN}(\mathcal{D})$	Easy
[27] Sunti., ESWC08	n.a.	n.a.	CL.(CEL)	Ih.	\mathcal{EL}^+	Medium
[9] Du, KSEM10	n.a.	n.a.	CL. (Module _{ECO})	Is.	$\mathcal{SHOIN}(\mathcal{D})$	Medium

In the table, “Bl.” and “Gl.” indicate black-box and glass-box, respectively, “Ih.” and “Is.” represent incoherence and inconsistency separately, “CL.” means command line, and “n.a.” indicates not applicable

approach will be varied on different data sets. The specific recommendations will be explained one by one in the following.

If the users prefer to resolve the incoherences one by one or they are interested in obtaining one MUPS for an unsatisfiable concept, SWOOP is a good choice. It is because this tool can be easily installed and used and a MUPS usually can be returned quickly and correctly.

To compute all MUPS, the suggestions will be varied in different cases. Here, the computation of all MUPS is useful to find a repair solution which satisfies a kind of minimal change. By default, we recommend the glass-box approach MUPS-GL-ALL as its efficiency is usually high. If it fails to finish the process to find all MUPS within the limited time, the relevance-directed approach MUPS-REL is recommended. Because MUPS-REL can finish a debugging task relatively fast and it can find more MUPS within the limited time.

If the expressivity of the ontologies to be debugged is \mathcal{EL}^+ , the system CEL is preferred as this system provides various functionalities to support the debugging task. Specifically, it can compute a single MUPS or all MUPS, and it can extract modules by using the reachability-based modularization method or the computation of a superset of a MUPS (see [4]).

Among the optimization methods for computing MUPS, the locality-based modularization method is recommended. One reason is that it can be easily used. The other reason is that an extracted module can cover all MUPS for an unsatisfiable concept. In such case, those debugging approaches which satisfy completeness can still satisfy completeness.

To debug an inconsistent ontology, the users can use the divide-and-conquer approach in [14] directly if the ontology is not very large and does not contain too many MIS. Otherwise, the extraction method Module_{DECO} can be applied.

7 Conclusion and Future Work

In this paper, we proposed the criteria set BLUEI²CI to systematically compare the existing ontology debugging approaches. BLUEI²CI is a firstly proposed criteria set for the debugging approaches and consists of seven criteria like completeness and usability. These criteria can be used to compare the debugging approaches from various dimensions like the designed algorithms, the implementation, the installation, and the efficiency. Based on BLUEI²CI, we discussed the relationships and distinctions among the existing debugging approaches. Particularly, when comparing the approaches w.r.t. efficiency, we selected different kinds of data sets which are constructed differently and contain significantly distinct numbers of MUPS for an unsatisfiable concept (e.g., more than 60). After the discussion, we concluded a set of useful suggestions for users to choose an appropriate debugging approach according to their needs.

Although the existing debugging approaches have shown good performance on some criterion, there is still a big space for improvement, especially in the

aspect of efficiency. First, the efficiency for computing MUPS can be further improved by investigating the characteristics of the data sets and designing domain-specific algorithms. Second, with the rapid growth of Linked Data [7], the scalable algorithms need to be developed for detecting inconsistencies.

Acknowledgements We gratefully acknowledge funding from the National Science Foundation of China under grants 60873153, 60803061, and 61170165.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, Cambridge (2003)
2. Baader, F., Peñaloza, R.: Axiom pinpointing in general tableaux. In: TABLEAUX, pp. 11–27, 2007
3. Baader, F., Peñaloza, R.: Automata-based axiom pinpointing. In: IJCAR, pp. 226–241, 2008
4. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic \mathcal{EL}^+ . In: KI, pp. 52–67, 2007
5. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic \mathcal{EL}^+ . In: KR-MED, 2008
6. Bell, D.A., Qi, G., Liu, W.: Approaches to inconsistency handling in description-logic based ontologies. In: OTM, pp. 1303–1311, 2007
7. Christian, B., Tom, H., Tim, B-L.: Linked data - the story so far. Int. J. Semantic Web Inform. Syst. **5**(3), 1–22 (2009)
8. David, J., Guillet, F., Briand, H.: Matching directories and OWL ontologies with AROMA. In: CIKM, pp. 830–831, 2006
9. Du, J., Qi, G.: Decomposition-based optimization for debugging of inconsistent OWL DL ontologies. In: KSEM, pp. 88–100, 2010
10. Du, J., Qi, G., Ji, Q.: Goal-directed module extraction for explaining OWL DL entailments. In: ISWC, pp. 163–179, 2009
11. Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: AAAI, pp. 1295–1300, 2006
12. Grau, B.C., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: extracting modules from ontologies. In: WWW, pp. 717–726, 2007
13. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: ESWC, pp. 182–197, 2005
14. Horridge, M., Parsia, B., Sattler, U.: Explaining inconsistencies in OWL ontologies. In: SUM, pp. 124–137, 2009
15. Ji, Q., Gao, Z., Huang, Z., Zhu, M.: An efficient approach to debugging ontologies based on patterns. In: JIST, pp. 425–433, 2011
16. Ji, Q., Qi, G., Haase, P.: A relevance-directed algorithm for finding justifications of DL entailments. In: ASWC, pp. 306–320, 2009
17. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: ISWC/ASWC, pp. 267–280, 2007
18. Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C., Hendler, J.A.: Swoop: A web ontology editing browser. J. Web Semant. **4**(2), 144–153 (2006)
19. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.: Debugging unsatisfiable classes in OWL ontologies. J. Web Semantics **3**(4), 268–293 (2005)
20. Li, S., Yin, Q., Hu, Y., Guo, M., Fu, X.: Overview of researches on ontology. J. Comput. Res. Dev. (in Chinese) **41**(7), 1041–1052 (2004)

21. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Repairing ontology mappings. In: AAAI, pp. 1408–1413, 2007
22. Qi, G., Yang, F.: A survey of revision approaches in description logics. In: Description Logics, 2008
23. Reiter, R.: A theory of diagnosis from first principles. *Artif. Intell.* **32**(1), 57–95 (1987)
24. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI, pp. 355–362, 2003
25. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. *J. Automat. Reas.* **39**(3), 317–349 (2007)
26. Stuckenschmidt, H.: Debugging owl ontologies - a reality check. In: EON, 2008
27. Suntisrivaraporn, B.: Module extraction and incremental classification: A pragmatic approach for ontologies. In: ESWC, pp. 230–244, 2008
28. Suntisrivaraporn, B.: Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies. PhD thesis, TU Dresden, Germany, 2009
29. Suntisrivaraporn, B., Qi, G., Ji, Q., Haase, P.: A modularization-based approach to finding all justifications for OWL DL entailments. In: ASWC, pp. 1–15, 2008
30. Wang, H., Horridge, M., Rector, A.L., Drummond, N., Seidenberg, J.: Debugging OWL-DL ontologies: A heuristic approach. In: ISWC, pp. 745–757, 2005

NJVR: The NanJing Vocabulary Repository

Gong Cheng, Min Liu, and Yuzhong Qu

Abstract Widespread deployment of Semantic Web technologies is accompanied with thousands of vocabularies used in practice. The research community has explored various vocabulary-oriented research topics but suffers from the lack of a large freely accessible vocabulary repository for conducting experiments in real-world settings. To meet the challenge, we release the NanJing Vocabulary Repository (NJVR), which has collected 2,996 vocabularies and instantiations thereof distributed in 4.1 billion RDF triples, all crawled from the real Semantic Web. This paper describes its construction and characterization and also, to illustrate its potential applications, presents experimental results obtained on it about the two research topics, namely, vocabulary matching and ranking.

1 Introduction

Vocabularies (aka ontologies), located at the heart of the Semantic Web Stack, enable information exchange among applications at the semantic level. Much effort has been directed to various vocabulary-oriented research topics such as matching and ranking. Evaluations in these areas require not only systematically generated benchmarks but also, more importantly, a large and representative collection of real-world vocabularies. Investigating such a vocabulary repository also helps characterize the deployment of Semantic Web technologies in practice.

Among existing repositories, BioPortal, Schemapedia, and the like are constructed in a *top-down* manner, by receiving submission from a certain community and manipulating registered vocabularies manually; search engines like Swoogle and Watson collect vocabularies in a *bottom-up* manner, by crawling documents

G. Cheng (✉) • M. Liu • Y. Qu

State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing 210023, China

e-mail: gcheng@nju.edu.cn; mf1033015@smail.nju.edu.cn; yzqu@nju.edu.cn

from the Web and identifying vocabularies automatically. The former style is limited by the scope that humans can deal with so that typically only hundreds of vocabularies are maintained by a community; in the latter case, state-of-the-art search engines have indexed thousands of vocabularies, but unfortunately they only provide search services which are inconvenient for researchers to explore and exploit the entire indexes.

To overcome the limitations of both manners, in this work, we release the *NanJing Vocabulary Repository* (NJVR, available at ws.nju.edu.cn/njvr), which is derived from the index of the Falcons search engine (ws.nju.edu.cn/falcons). The latest version of NJVR has the following characteristics:

- It is exported from an index of 4.1 billion RDF triples distributed in 15.9 million RDF documents crawled from 5,805 pay-level domains (PLDs), which is comparable to the indexes maintained by other vocabulary search engines and is even larger than the well-known Billion Triple Challenge Datasets.
- It contains RDF descriptions of 2,996 dereferenceable vocabularies crawled from 261 PLDs and also provides document-level statistical data on their instantiations. All this information is publicly downloadable.

To the best of our knowledge, NJVR is among the largest freely accessible vocabulary repositories. Section 2 will elaborate on its construction and characterization. In Sect. 3 we will illustrate the use of NJVR by experimenting two research tasks. Finally, Sect. 4 concludes this paper with future work.

2 Construction and Characterization

2.1 Crawling

NJVR is derived from a collection of RDF documents crawled by *Falconsbot*, a Semantic Web crawler implemented as part of Falcons. Falconsbot was started in 2007, whose URI pool was initialized with (a) RDF documents indexed by several freely accessible repositories such as pingthesemanticweb.com, (b) sample resources and/or entry points of the data sets in the Linking Open Data cloud, and (c) search results returned by Google (restricted by `filetype:rdf` OR `filetype:owl`) and Swoogle against queries composed of the category names at dmoz.org. Continually, URIs are dereferenced with content negotiation to accept RDF/XML or RDFa formats. After successfully parsing a retrieved document, an index of RDF triples is updated by removing out-of-date triples (if any) and inserting the latest ones, and the URI pool is expanded with new URIs met. Users of Falcons can also submit URIs to Falconsbot via a Web form. After running Falconsbot for years, we paused it in May 2011 and exported a snapshot of our index, from which we derived NJVR.

Fig. 1 Log–log plot of the distribution of the number of vocabularies crawled from a PLD

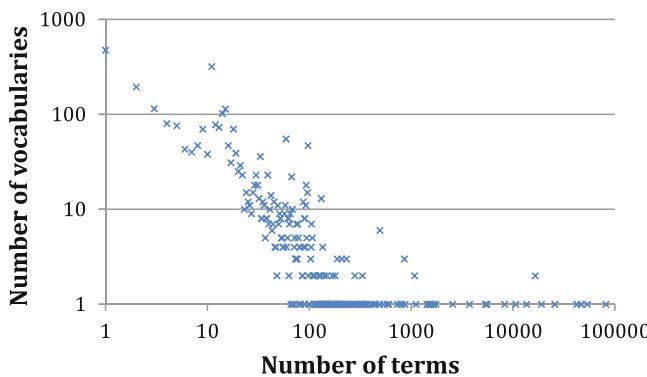
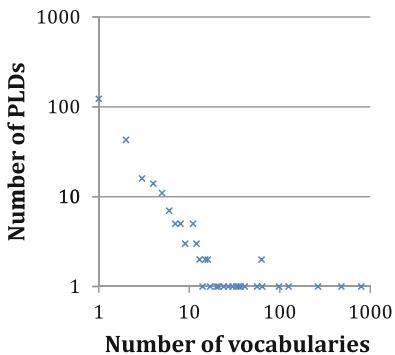


Fig. 2 Log–log plot of the distribution of the number of constituent terms of a vocabulary

2.2 Vocabulary Identification

Given an index of RDF triples and their contexts (i.e., documents where they are found), vocabularies in NJVR are automatically identified based on the following policy: Firstly, terms are distinguished from other URIs; a *term* is a URI that identifies a class or a property according to its RDF description offered by the document retrieved by dereferencing the URI. Then, terms defined in a common namespace are grouped into a *vocabulary*, which is identified by the namespace URI. This policy shows our focus on modern vocabularies that are dereferenceable. However, we are aware that for a vocabulary whose constituent terms are separately defined in multiple documents, we may fail to find all of them, given no practically effective strategy for solving this issue.

By implementing the above policy on our index, we obtained 455,718 terms, comprising 396,023 classes (86.9%) and 59,868 properties (13.1%), having an overlap of 173 which are both classes and properties. All these terms were grouped

into 2,996 vocabularies, which exhibit a great variety. On the one hand, their namespace URIs are distributed in as widely as 261 PLDs. Nevertheless, these PLDs contributed unequally as plotted in Fig. 1. When [columbia.edu](#) and [w3.org](#) offered 794 (26.5%) and 479 vocabularies (16.0%), respectively, 123 other PLDs only contributed a single vocabulary. We chose not to restrict the contribution by a PLD to a limited proportion because such uneven distribution seems to be a universal phenomenon in content creation on the Semantic Web, e.g., as observed in [2], thereby deserving to be noticed by the users of NJVR. On the other hand, Fig. 2 plots the uneven distribution of the number of constituent terms of a vocabulary. In contrast to 941 mini vocabularies (31.4%) comprising five or fewer terms, we see several very large ones containing more than ten thousand terms, e.g., various copies of YAGO and Cyc. Based on these, we conclude that NJVR offers a large and diverse sample of the vocabularies on the Web.

2.3 Vocabulary Instantiation

A class is instantiated in an RDF triple where it is the object and [rdf:type](#) is the predicate; a property is instantiated in an RDF triple where it is the predicate. Such an RDF triple instantiating a term is called an *instantiation* of the term. Instantiations of a term may exist in multiple RDF documents and in each document may be found multiple times. NJVR offers not only vocabulary description but also instantiation, to support broader research tasks such as data mining. Specifically, it provides a “term-document matrix,” each entry of which represents a “term frequency,” namely, the number of instantiations of the corresponding term in the corresponding document. For each document, we also provide its “length,” namely, the total number of RDF triples it contains.

By investigating our index of 15.9 million RDF documents or 4.1 billion RDF triples without reasoning, we found at least one instantiation for 115,707 classes (29.2%) and 25,963 properties (43.4%), which collectively are distributed in 1,874 vocabularies (62.6%). The extent to which the instantiations of a vocabulary are distributed varies greatly as plotted in Figs. 3 and 4. Whereas 96.7% of the instantiated classes and 95.9% of the instantiated properties have instantiations found in only one PLD, terms like [foaf:Person](#) and [dc:creator](#) have been instantiated in more than one thousand PLDs. Based on these, we conclude that NJVR provides a large and diverse sample of the vocabulary instantiations on the Web.

3 Applications

To illustrate the potential use of NJVR and show its value in promoting vocabulary-oriented research, in this section we will perform two sets of experiments over NJVR on two different research tasks, namely, vocabulary matching and ranking.

Fig. 3 Log–log plot of the distribution of the number of PLDs offering instantiations of a class

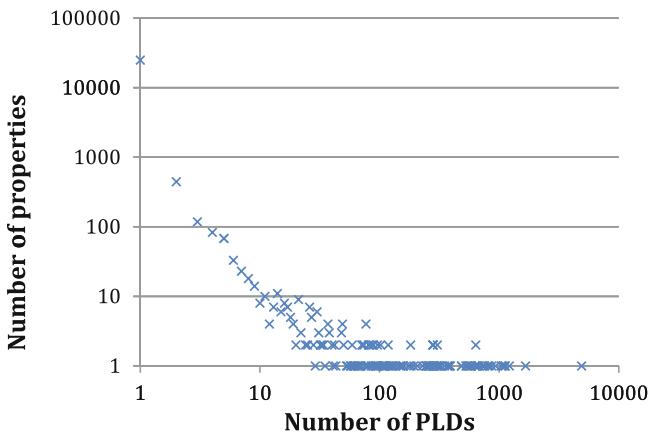
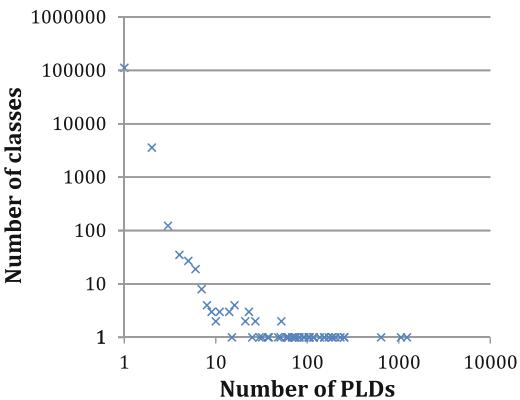


Fig. 4 Log–log plot of the distribution of the number of PLDs offering instantiations of a property

3.1 Vocabulary Matching

To deal with semantic heterogeneity raised by the use of different vocabularies that offer different terms denoting the same concept, matching techniques have attracted many research interests [4]. Test cases widely adopted by this research community are given by OAEI (oaei.ontologymatching.org), consisting of a number of systematically generated vocabularies but only a few real-world ones. More real-world vocabularies can help test and improve the robustness of matching algorithms. To know whether NJVR has great potential as a source of test cases for vocabulary matching, we compute the content similarity [1] between every pair of vocabularies in NJVR, which returns a similarity value inside the interval $[0, 1]$. Then we observe the number of *matchable vocabularies*, i.e., those whose similarity is not lower than a given threshold.

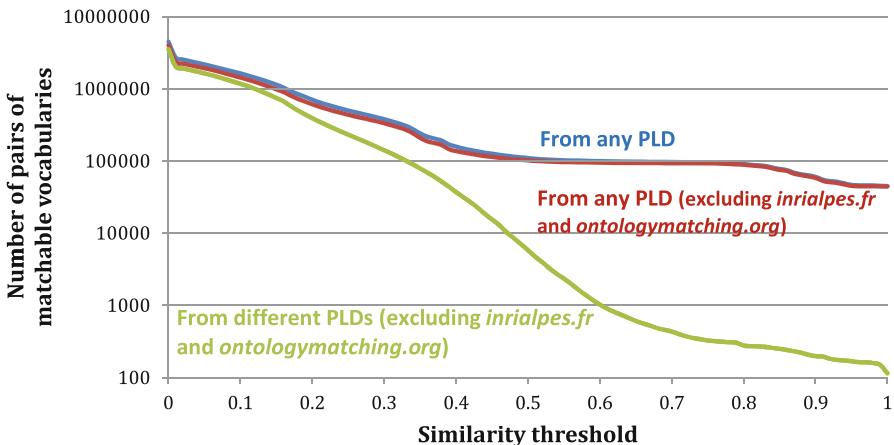


Fig. 5 Log-linear plot of the distribution of the number of pairs of matchable vocabularies under different settings

The results under different thresholds from 0 to 1 with 0.01 increments are depicted in Fig. 5. When setting the threshold to a medium value of 0.5 and a considerably high value of 0.8, we obtained as many as 110,613 and 91,281 pairs of matchable vocabularies, respectively. We found that some vocabularies in NJVR were crawled from [inrialpes.fr](#) and [ontologymatching.org](#), i.e., being part of the OAEI test cases. After excluding them, the distribution in Fig. 5 changes only slightly. That is, NJVR and OAEI share a very small overlap, thereby becoming favorable complements. Further, the research community may be particularly interested in the matchable vocabularies originating from different sources. As shown in Fig. 5, the number of pairs of matchable vocabularies crawled from different PLDs differs widely from the previous two distributions, indicating that most pairs of matchable vocabularies indeed originated from the same source. However, we could still obtain as many as 5,821 and 278 pairs of matchable vocabularies crawled from different PLDs when setting the threshold to 0.5 and 0.8, respectively. To conclude, complementary to OAEI, NJVR has proved to be a rich source of potential test cases for evaluating vocabulary matchers.

3.2 Vocabulary Ranking

Vocabulary ranking is essential to a vocabulary search engine. State-of-the-art approaches, e.g., [3], perform link analysis to assess the importance of vocabularies. The idea is to represent vocabularies and their links by a digraph and then compute the centrality of each vertex within the graph. We intend to experiment with this line of work and compare different centrality measures by using NJVR, namely,

Table 1 Correlation between rankings generated by different measures of centrality

	Indegree	Closeness	Betweenness	Eigenvector	PageRank
Closeness	0.075				
Betweenness	0.709	0.187			
Eigenvector	0.995	0.080	0.698		
PageRank	0.996	0.079	0.700	1.000	
HITS authority	0.996	0.077	0.693	0.991	0.992

indegree, closeness, betweenness, eigenvector, PageRank (with a damping factor of 0.85), and HITS authority. We construct a digraph where each vertex represents a vocabulary (excluding language-level ones such as RDF) and an arc connects vocabulary v_i to v_j when v_i references v_j [1].

In our experiment, the resulting graph comprises 2,993 vertices and 3,904 arcs and is disconnected but contains a large weakly connected component (denoted by G) consisting of 1,332 vertices (44.5%) and 3,276 arcs (83.9%), on which different centralities were computed. Since it is difficult—if not impossible—to manually establish a widely accepted ranking of vocabularies as the gold standard for evaluation, we chose to compare the rankings given by different measures to study to what extent these measures tend to generate similar results. Table 1 presents the Spearman’s rank correlation coefficient between every pair of the rankings. Indegree, eigenvector, PageRank, and HITS authority almost generated the same results. This agrees with the theory that PageRank and HITS are two variants of the eigenvector measure and confirms that they are often highly correlated with indegree. By contrast, basically no correlation was observed between the ranking produced by closeness and the others. A comprehensive evaluation of more measures is left as future work.

4 Conclusions and Future Work

We have released NJVR, one of the largest freely accessible repositories consisting of crawled vocabularies and their instantiations. It is devoted to providing the research community a representative sample of the Semantic Web for conducting experiments and evaluations about all kinds of vocabulary-oriented problems. We have performed two sets of experiments based on NJVR about matching and ranking, showing its great potential. Exploiting it in the future, there are much more analysis and evaluation to be carried out by the research community, leveraging not only vocabularies but also their instantiations, on not only NJVR but also comparatively on multiple repositories. Our major focus will be on the removal of low-quality vocabularies from NJVR.

Acknowledgements This work was supported in part by the NSFC under Grant 61100040, 61223003, and 61170068, and in part by the JSNSF under Grant BK2012723

References

1. Cheng, G., Gong, S., Qu, Y.: An empirical study of vocabulary relatedness and its application to recommender systems. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 98–113. Springer, Berlin/Heidelberg (2011)
2. Ding, L., Finin, T.: Characterizing the semantic web on the web. In: Cruz, I., Decker, S., Allemand, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 242–257. Springer, Berlin/Heidelberg (2006)
3. Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and ranking knowledge on the semantic web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 156–170. Springer, Berlin/Heidelberg (2005)
4. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Berlin/Heidelberg (2007)

Visualizing RDF Data Profile with UML Diagram

Huiying Li and Xiang Zhang

Abstract Various RDF data have been published in the Semantic Web. A key challenge for users to reuse these data is in understanding the large and unfamiliar RDF data, especially when the data schema is absent or when different schemas are mixed. In this paper, we describe a tool that can construct the actual schema, gather corresponding statistics, and can present a visual, UML diagram for RDF data sources, such as SPARQL endpoints or RDF dumps. Moreover, for datasets with complex structure, the tool can mine the class subsumptions and then provide a concise visualization that would serve as a big picture for users. The experimental results compare our approach and ExpLOD using seven datasets (including DBpedia) from the Linked Data cloud. The performance evaluations show that our approach is more efficient than ExpLOD. The concise visualization experiments on complex datasets such as DBpedia show that our approach is feasible.

1 Introduction

In recent years, the Web is being increasingly extended with more RDF data sources and links between objects. The Linking Open Data community project is promoting the emergence of linked open data (LOD) [1] and is fostering the availability of many open RDF datasets, such as DBpedia (extracted from Wikipedia), SwetoDBLP, and LinkedMDB.

The abundance of RDF data brings users with many opportunities to reuse these data. Before an RDF data can be reused, the user must understand the data and then determine whether such data can be easily reused. When handling a large and complex RDF data, users cannot easily obtain the big picture if the RDF dataset uses multiple ontologies or if the ontology cannot be obtained. Facing this challenge,

H. Li (✉) • X. Zhang

School of Computer Science and Engineering, Southeast University, Nanjing 210096, P.R.China
e-mail: huiyingli@seu.edu.cn; x.zhang@seu.edu.cn

firstly, we propose a SPARQL query-based approach for RDF data profiling. For users, data profiling is a cardinal activity when facing an unfamiliar dataset. This process helps in assessing the importance of the dataset as a whole, in finding out whether the dataset or part of the dataset can be easily reused, in improving the user's ability to query or search the dataset, and in detecting irregularities for improving data quality. Then, we visualize the RDF data profile using UML diagram, since UML is the most accepted software engineering standard.

In this paper, we consider the data structure and five kinds of descriptive statistics then visualize them using UML diagram. The contributions of our work are as follows: (1) We propose an approach to obtain the actual schema of a SPARQL endpoint or RDF dump that can be considered as a customized schema for the RDF data, to gather corresponding statistics and to present a UML-based visualization for users; (2) we present a way to deal with large and complex RDF data. An association rule mining algorithm is used to determine the class subsumptions in RDF data. Based on the class subsumptions, a concise visualization is constructed to provide users a big picture of the dataset, and (3) we compare the performance of the proposed approach and ExpLOD which is similar to our approach. The evaluations show that our approach is feasible and more efficient than ExpLOD.

The rest of this article is organized as follows: Sect. 2 introduces the related work. Section 3 introduces the SPARQL-based approach for constructing and visualizing RDF data profiling. Section 4 presents the steps to construct a concise visualization based on the class subsumptions. Section 5 details the experimental results of our approach. Section 6 concludes the study.

2 Related Work

Describing and understanding large RDF data is enabled by statistics and summary. Recently, structural summaries have been proposed in the context of RDF data in [2]. Before this work, some similar work has been carried out for XML data. These prior works have shown the usability of XML path summaries for a variety of scenarios within semi-structured XML data. But most of these works must rely on the tree nature of XML data or on acyclicity assumptions that do not hold for RDF data.

Recently, some researchers are dealing with RDF data statistics and summary, such as semantic sitemaps [3], RDFStats [4], and SCOVOC vocabulary [5]. Among these works, ExpLOD [6] is the most similar work to our approach. It is a tool that supports constructing summaries of RDF usage based on the bisimulation contraction mechanism. Although ExpLOD enables coarse granularity summarization based on the hierarchical bisimulation label, it faces difficulties in handling summarizations grouped by high hierarchy labels, such as namespace, which could be too general sometimes. Moreover, ExpLOD does not care about the relation between instance blocks, which is very important in understanding the data structure.

Compared with ExpLOD, our approach is distinct in three aspects. Firstly, the proposed approach considers the class type instead of predicate usage to divide instances into different blocks. It avoids the situation whereby the block number becomes too large. Secondly, our approach not only considers the data structure but also places importance on the statistics such that users can understand the RDF data. Thirdly, for large and complex datasets, our approach mines the class subsumptions and provides a concise visualization for users.

3 Visualizing RDF Data Profile

The RDF graph can be considered as a set of RDF triples. A triple consists of a subject, a predicate, and an object. Figure 1 shows the RDF graph of a sample snippet of RDF data. The snippet gives the information about two music artists, their names, the records they have made, and the record titles. Such a small snippet uses four ontologies, namely, the FOAF ontology (with prefix label *foaf*), the Dublin Core ontology (with prefix label *dc*), the Music Ontology (with prefix label *mo*), and the RDF meta-ontology (with prefix label *rdf*). When dealing with a very large RDF dataset, users cannot easily understand the data if the ontology is absent. Common profiling methods and tools assume a starting point of relational data with a domain-specific schema. This assumption does not hold for Semantic Web data published on the web.

We consider the data structure and five kinds of descriptive statistics for RDF data profiling. *Data structure* contains the classes and their properties, respectively. *Class instantiation* is the number of distinct instances that are typed as a particular

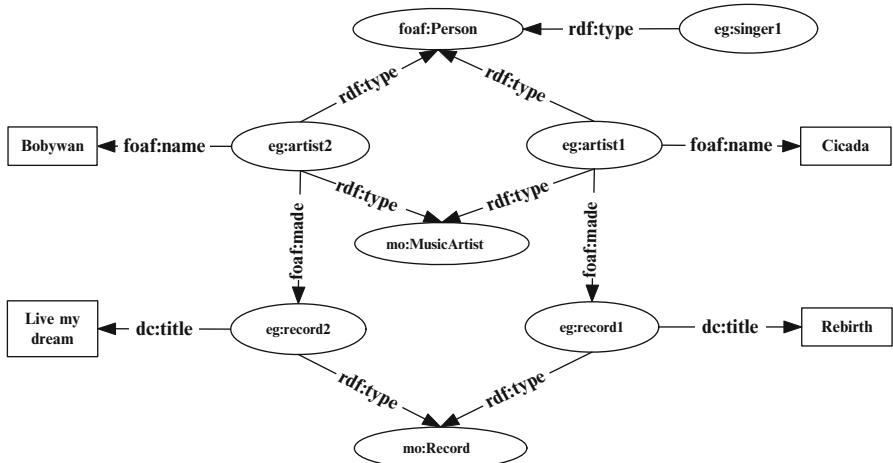


Fig. 1 RDF data snippet

class. *Property instantiation* is the number of distinct triple (s, p, o) for an object property p from class C_1 to C_2 , where the type of s is C_1 and the type of o is C_2 . For a datatype property p of class C , its instantiation is the number of distinct triple (s, p, o) where the type of s is C . *Multiplicity of object* denotes the multiplicity of object for an object property p from class C_1 to class C_2 ; four kinds of multiplicity are used: [0..1] means that every instance I_1 typed as C_1 has at most one relation p to instance I_2 typed as C_2 , [0..*] means that every instance I_1 typed as C_1 has zero or more relation p to instances I_2 typed as C_2 , [1..*] means that every instance I_1 typed as C_1 has at least one relation p to instance I_2 typed as C_2 . [1] means that every instance I_1 typed as C_1 has one and only one relation p to instance I_2 typed as C_2 . The datatype property can also denote the multiplicity of object. *Functional property* denotes whether a property is functional. An object property p from class C_1 to class C_2 is called a functional property if every instance I_1 typed as C_1 has a unique p value I_2 typed as C_2 , i.e., there cannot be two distinct instances I_2 and I_3 typed as C_2 such that there are two triples (I_1, p, I_2) and (I_1, p, I_3) . The definition of “functional” datatype property is similar. *Inverse-functional property* denotes whether a property is inverse-functional. An object property p from class C_1 to class C_2 is called an inverse-functional property if a value I_1 typed as C_2 can only be the value of p for a single instance typed as C_1 , i.e., there cannot be two distinct instances I_2 and I_3 typed as C_1 such that there are two triples (I_2, p, I_1) and (I_3, p, I_1) . The definition of “inverse-functional” datatype property is similar. Note that the *functional property* and *inverse-functional property* in our paper are a little different to OWL functional and inverse-functional properties. We emphasize the domain and range classes of the functional property, while the OWL functional property just requests that it has at most one value for any particular instance. The inverse-functional property is similar.

To compute the RDF data structure, we initially collect all classes used in the RDF data. In RDF data, many classes are not declared as *owl : Class* or *rdfs : Class* explicitly. We use the following SPARQL query to obtain all the named classes: “*SELECT distinct ?c WHERE {?s rdf:type ?c.}*”. The classes with *rdf*, *rdfs*, and *owl* namespaces are considered as metaschema-level classes and are not collected (except for *rdfs : Seq* because many blank nodes are declared to be *rdfs : Seq*). With these named classes, instance in RDF data can be classified into a set of named class according to their type. For each class, we explore all its instances to collect their used properties (including relations and attributes). For example, Fig. 2a queries the used properties of class *mo : Record*. Moreover, for the object property, we determine the range classes by the SPARQL query in Fig. 2b.

Some instances are without type declaration despite many typed instances. In solving this problem, an intuitive solution is to gather all the non-typed instances into one unnamed class. In providing clearer information for users, the second solution is to divide these non-typed instances into different property restriction classes according to their property restrictions. We collect the properties with non-typed

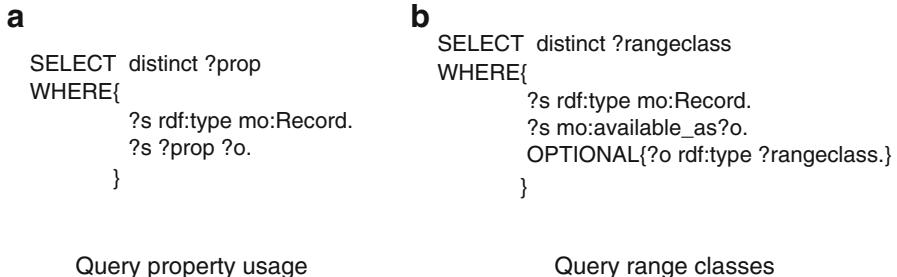


Fig. 2 Query property usage and range classes



Fig. 3 RDF snippet and UML diagrams

instances as subject or object in advance. Then, we use SPARQL query to divide the non-typed instances. In the Jamendo data, the unnamed class is divided into seven restriction classes, as shown in Fig. 4. With the SPARQL queries above, the RDF data structure is constructed. Then, obtaining statistics such as class instantiation and property instantiation using SPARQL becomes easy. Moreover, the statistics of multiplicity of object and functional property are computed.

Since we have induced the actual schema and gather the corresponding statistics, visualizing them for users becomes an important issue. We adopt the notations of properties and classes proposed by [7] as well as the Ontology Definition Metamodel [8]. Figure 3 demonstrates the UML diagrams for classes and properties.

We visualize the data profile of RDF data Jamendo. Figure 4 shows the structure of, and the corresponding statistics on, Jamendo, including 18 classes, of which 7 are restriction classes. For each class, the number of instances belongs to it and the percent of all instances is provided. For every property, the instantiation and the multiplicity of object are provided, and whether the property is functional (denoted by *f*) or inverse-functional (denoted by *inf*) is also provided. Such a UML diagram provides the data structure and corresponding statistics for users. It offers convenience for users to understand the RDF data behind a SPARQL endpoint, to decide whether the data can be reused, to construct suitable SPARQL query, and to retrieve interesting detailed information.

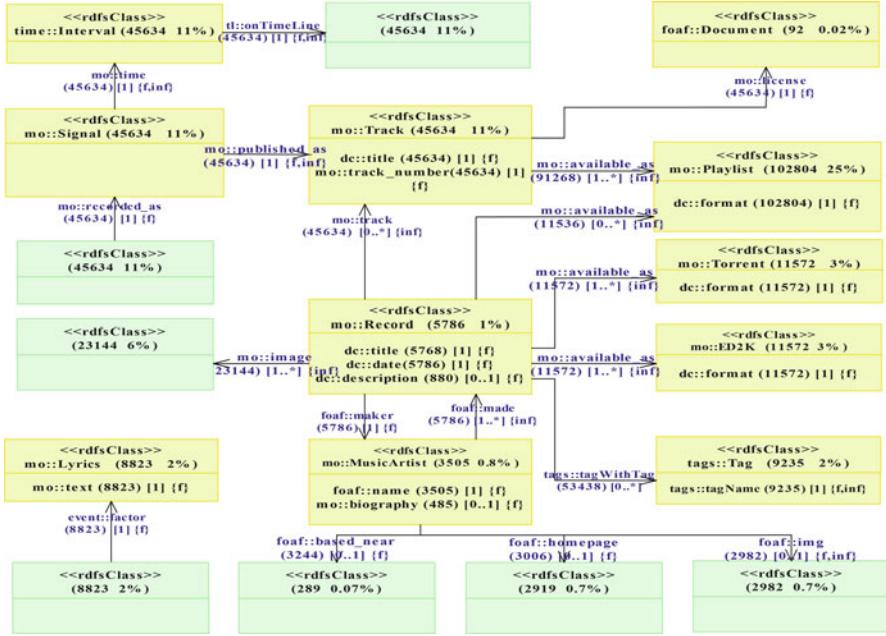


Fig. 4 Visualizing Jamendo data with UML diagram

4 Constructing Concise Visualization

When an SPARQL endpoint has too many classes, visualizing the entire set of classes is inefficient, and meanwhile, it will be difficult for users to understand the visualization. To reduce the number of restriction classes, we gather all the non-typed instances into one unnamed class. However, many classes may persist even after doing this procedure. For example, DBpedia has 326 named classes. We introduce an approach to construct a concise visualization for the dataset with a complex structure. The key problem is to select the classes and properties for constructing the concise visualization. For selecting classes, our purpose is to select a small number of classes that can cover all classes in an RDF dataset, so the class subsumptions are considered. Since the ontology is absent, we have to discover hidden class subsumptions from RDF data. We found that there are many instances typed as different classes in RDF data. So the association rule mining approach is used to extract the class subsumptions.

Association rule mining is a popular and widely researched method for discovering interesting relations between variables in large databases. Considering a transaction table, the rows represent instances, and the columns represent class types. A value of 1 in field (i, j) indicates that instance i is of type j . Association rule mining algorithm can be used to mine the class subsumptions of a large RDF dataset. After creating the transaction table locally using SPARQL query, the classic

algorithm Apriori [9] is used to learn the subclass association rule. Given the class set \mathbb{C} (columns in the transaction table) and the instance set \mathbb{I} (rows in the transaction table), \mathbb{X} is a subset of \mathbb{C} . The support $supp(\mathbb{X})$ is defined as the number of instance that is of $rdf : type$ every $C \in \mathbb{X}$. The Apriori finds the association rules $C_i \implies C_j$ with high confidence value, where C_i and C_j are both classes. The confidence of an association rule is defined as follows:

$$conf(C_i \implies C_j) = \frac{supp(\{C_i, C_j\})}{supp(\{C_i\})}$$

This formula provides evidence for the validity of the subclass relation between C_i and C_j because most instances that are of $rdf : type C_i$ are also of $rdf : type C_j$. Take the RDF data in Fig. 1 for example, the two instances declared as $mo : MusicArtist$ are also of $rdf : type foaf : Person$. The Apriori algorithm can mine the database and show that class $mo : MusicArtist$ is the subclass of $foaf : Person$, with a confidence value of 1. While the Apriori algorithm can also mine that class $foaf : Person$ is the subclass of $mo : MusicArtist$, with a confidence value of $\frac{2}{3}$. Given a set \mathbb{C} of all named classes in an RDF dataset and a nonnegative threshold τ , the mining results of algorithm Apriori can be considered as a relation \mathbb{R} on \mathbb{C} , where $\mathbb{R} = \{< C_i, C_j > | C_i \in \mathbb{C} \wedge C_j \in \mathbb{C} \wedge conf(C_i \implies C_j) \geq \tau\}$; \mathbb{R} is called the **mining relation**. If $C_i \neq C_j$, $< C_i, C_j > \in \mathbb{R}$, and $< C_j, C_i > \in \mathbb{R}$, then C_i is called same as to C_j , and C_j is put into set \mathbb{C}_S .

Definition 1. (Subclass Relation) Given the **mining relation** \mathbb{R} on \mathbb{C} . The set $\mathbb{C}' = \mathbb{C} - \mathbb{C}_S$, \mathbb{R}' is a relation on \mathbb{C}' , where $\mathbb{R}' = \{< C_m, C_n > | < C_m, C_n > \in \mathbb{R} \wedge C_m \in \mathbb{C}' \wedge C_n \in \mathbb{C}'\}$. $rt(\mathbb{R}')$ is the reflexive and transitive closure of \mathbb{R}' . The relation $rt(\mathbb{R}')$ is called the **subclass relation**, and it is denoted as \mathbb{R}_{subC} .

Theorem 1. *The subclass relation is reflexive, asymmetric, and transitive.*

Proof. Given the **subclass relation**, $\mathbb{R}_{subC} = rt(\mathbb{R}')$, where \mathbb{R}' is a relation on \mathbb{C}' . The **subclass relation** is the reflexive and transitive closure of \mathbb{R}' ; therefore, it is **reflexive** and **transitive**. \mathbb{R}' is **asymmetric**; therefore, $\mathbb{R}' = \mathbb{R}'^{-1}$. Because $(\mathbb{R}'^2)^{-1} = (\mathbb{R}' \circ \mathbb{R}')^{-1} = \mathbb{R}'^{-1} \circ \mathbb{R}'^{-1} = \mathbb{R}' \circ \mathbb{R}' = \mathbb{R}^2$, thus \mathbb{R}^2 is **asymmetric**. Hence, \mathbb{R}^n is **asymmetric**, where $|\mathbb{C}'| = n$. $rt(\mathbb{R}') = \Delta \cup \mathbb{R}' \cup \mathbb{R}'^2 \dots \mathbb{R}^m$, where Δ is the diagonal relation on \mathbb{C}' . Hence, $rt(\mathbb{R}')$ is also **asymmetric**. Therefore, the **subclass relation** is **reflexive, asymmetric, and transitive**. ■

Therefore, the **subclass relation** is a *partial order*.

Definition 2. (Maximal Class) Given the **subclass relation** \mathbb{R}_{subC} on set \mathbb{C} , C_{max} is the **maximal class** if $C_{max} \in \mathbb{C}$ and there is no $C_i \in \mathbb{C}$ such that $C_{max} \neq C_i$ and $< C_{max}, C_i > \in \mathbb{R}_{subC}$.

Definition 3. (Minimal Class) Given the **subclass relation** \mathbb{R}_{subC} on set \mathbb{C} , C_{min} is the **minimal class** if $C_{min} \in \mathbb{C}$ and there is no $C_i \in \mathbb{C}$ such that $C_{min} \neq C_i$ and $< C_i, C_{min} > \in \mathbb{R}_{subC}$.

Definition 4. (Class Level) Given the *subclass relation* \mathbb{R}_{subC} on set \mathbb{C} , the level of *minimal class* is 0. The level of class C_i is $\max + 1$; \max is the maximum level of all C_j where $\langle C_j, C_i \rangle \in \mathbb{R}_{\text{subC}}$.

In constructing a concise visualization, we select all maximal classes to represent the classes in an RDF dataset. Given an RDF dataset, \mathbb{C} is the class set, and \mathbb{C}_m is the maximal class set. A partition of set \mathbb{C} can be computed based on the maximal classes. For every maximal class, all its subclasses that are not contained in other blocks construct a partition block. Certainly, set \mathbb{C} can have many different partitions. To obtain the partition with well-distributed element number in every block, we use a heuristic method. Firstly, the maximal class with the lowest level is selected to construct a block; then, the maximal class and selected classes are removed. The process goes on until there is no class left to be selected. Therefore, a class partition is constructed based on the maximal classes; every class belongs to one and only one partition block. Every class belongs to one block because for every class C_i , there exists at least one maximal class C_m that $\langle C_i, C_m \rangle \in \mathbb{R}_{\text{subC}}$. Every class belongs to only one block because we only select the class which is not selected by other blocks when constructing block.

After selecting the maximal classes to construct the concise visualization, another problem is to select the representative properties for every maximal class. We consider two kinds of representative properties: the one is the most popular property, and the other is the most distinctive property.

Definition 5. (Most Popular Property) Given the class C , set \mathbb{P} contains all the properties of C . The *most popular property* of C is the property P_{pop} with the largest instantiation in \mathbb{P} .

Definition 6. (Most Distinctive Property) Given the class C , set \mathbb{P} contains all the properties of C . The *most distinctive property* of C is the property P_{dist} with the smallest number of classes that also have the property P_{dist} .

In the concise visualization, we can select the most popular property (datatype property and object property) or the most distinctive property (datatype property and object property) for every maximal class. For the object property, if it has more than one range class, we only consider the one that has the most number of instances. Figure 5 shows the concise visualization of DBpedia data with the most distinctive property for every maximal class. Among the 326 named classes, only 35 are maximal classes. The integer after the $\ll \text{rdfsClass} \gg$ stereotype denotes the number of subclasses covered by this maximal class. The visualization is concise enough for users to grasp the big picture of the large and complex RDF dataset. If the user wants to know more about one or several classes, she can also use the approach mentioned in Sect. 3 to get the detail statistics.

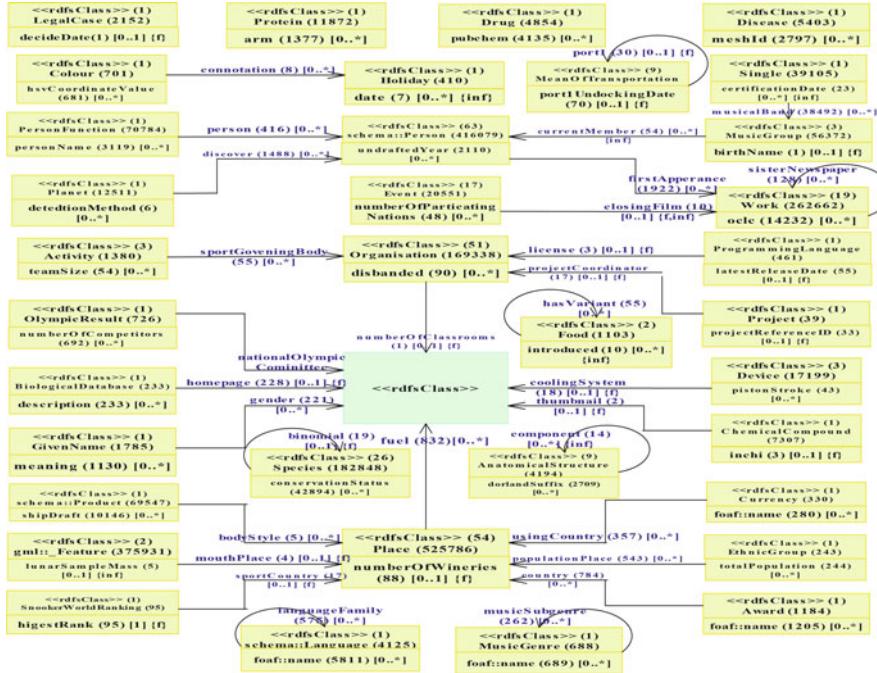


Fig. 5 Concise visualization of DBpedia data with the most distinctive property (default namespace is **dbpedia**)

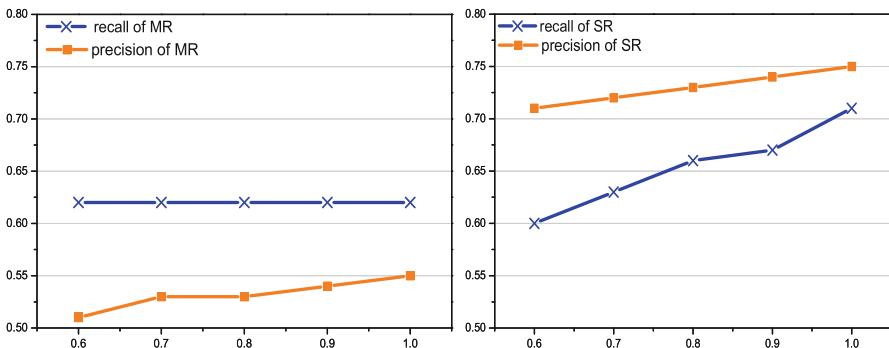
5 Experimental Study

In this section, we compare the performance of our approach with that of ExpLOD. ExpLOD is a tool that supports SPARQL-based summary creation of RDF usage. We use the Jena toolkit (jena.sourceforge.net) to manage the RDF data for ExpLOD and for our approach. All the experiments were developed within the Eclipse environment and on a 64bit ThinkStation with 3.10MHz and 16GB of RAM (of which 14GB was assigned to the JVM).

We chose seven datasets, which are shown in the Table 1. These datasets are selected from the LOD cloud because they vary in the amount and type of information they describe. Data Peel and Jamendo contain information on music artists and their productions. Data GeoSpecies contains information on biological orders, families, species, as well as species occurrence records and related data. Data LinkedCT contains information on linked clinical trials. Data LinkMDB contains linked data on movies. Data SwetoDbp includes information about affiliations, universities, publications, and publishers. Data DBpedia contains extracted data from Wikipedia. We concentrate on the infobox subset of DBpedia 3.7. Table 1 shows the following information about each dataset: the name, the number of triples it contains, as well as the number of properties, classes, and instances.

Table 1 Statistics of RDF datasets

Dataset	# of classes	# of properties	# of instances	# of triples
Peel	9	25	76,894	271,369
Jamendo	11	26	410,893	1,047,950
GeoSpecies	44	168	184,931	2,076,380
LinkedCT	13	90	1,169,905	9,804,652
LinkedMDB	53	222	1,326,001	6,147,995
SwetoDblp	10	145	2,394,479	13,041,580
DBpedia	326	1,378	1,839,009	26,988,054

**Fig. 6** Recall and precision of DBpedia data

For four of the seven datasets (Peel, Jamendo, LinkedCT, and SwetoDblp), we did not obtain any mining result because there are no instances typed as different classes in these four datasets, which is the premise to mine the class subsumptions. For the other three datasets (GeoSpecies, LinkedMDB, and DBpedia), we got the mining relations using Apriori algorithm. To evaluate the effectiveness of mining relation and subclass relation, the recall and precision scores are computed for various thresholds on the confidence values. Figure 6 illustrates the recall and precision of DBpedia data. The results of GeoSpecies and LinkedMDB data are similar and not listed. We evaluated the mining relation and subclass relation by comparing them to the DBpedia ontology (version 3.7). However, as the DBpedia ontology contains different types of axioms, we only considered the class subsumption axioms. Moreover, we considered not only the explicit class subsumption but all the inferable class subsumptions as well. DBpedia ontology contains 276 explicit subsumption axioms, 597 inferable subsumption axioms, and 873 subsumption axioms in total.

Figure 6 illustrates the precision and recall of mining relation (denoted as **MR**) and subclass relation (denoted as **SR**) with varying confidence values. We can observe that the recall of mining relation is 6.2 whatever the threshold on the confidence value is, because we do not find more correct results when the threshold is decreased. We can also observe that the precision of subclass relation is improved largely compared with that of mining relation. We can also observe that the recall

Table 2 Performance (in ms) of ExpLOD, FV, CVpop, and CVdist

	ExpLOD	FV		CVpop		CVdist	
	Time	Time	# of queries	Time	# of queries	Time	# of queries
Peel	62,422	11,359	87	—	—	—	—
Jamendo	202,344	26,906	85	—	—	—	—
GeoSpecies	2,839,031	1,783,593	2,352	67,078	538	35,812	298
LinkedCT	860,719	119,953	172	—	—	—	—
LinkedMDB	645,891	98,968	621	46,829	493	27,922	388
SwetoDbpl	913,594	504,109	566	—	—	—	—
DBpedia	—	249,541,131	152,729	2,945,578	2,574	1,460,828	1,423

of subclass relation is lower than that of mining relation when the threshold is set to 0.6. The reason is that when the threshold on the confidence value is decreased, the Apriori algorithm finds more wrong subsumption axioms. The mining relation contains more results that show C_i as a subclass of C_j and C_j as a subclass of C_i ; although one of these subsumption axioms maybe true, these results are both removed from subclass relation because of **asymmetry**. But with the increase of the threshold, the recall of subclass relation is improved, largely compared with that of mining relation. Finally, based on the subclass relation, we get 35 maximal classes out of all 326 classes when the threshold is set to 1.0. For most datasets, except DBpedia, the time needed to compute the association rules is less than 3 min when the threshold is set to 1.0. For the dataset DBpedia, when the threshold is set to 1.0, the running time is less than 20 min due to the large number of instances.

To show the efficiency of our approach and ExpLOD, we conduct a performance evaluation. In our approach, the threshold on the confidence value is set to 1.0. Performance is measured as the time taken by different approaches. For ExpLOD, the premise of computing the summary graph is to construct a labeled graph. Hence, the time taken by ExpLOD is the sum of the time for constructing a labeled graph and the time for computing the summary graph. The running time of our approach is composed of two parts: the time to obtain the data structure (full or concise) and the time to compute the corresponding statistics. We compare three means of our approach: full visualization (denoted as **FV**), concise visualization with the most popular property for every maximal class (denoted as **CVpop**), and concise visualization with the most distinctive property for every maximal class (denoted as **CVdist**). Peel, Jamendo, LinkedCT, and SwetoDbpl did not yield any mining results. Thus, we list only the full visualization performance for them. For DBpedia, the running time exceeds three days when using ExpLOD; the results are not listed.

Table 2 shows the performance comparison. The running time of ExpLOD is longer than that of our approach (regardless of full visualization or concise visualization). For ExpLOD, before computing the summary graph, a bisimulation label must be assigned to each class, predicate, instance, and literal. This graph will increase the triple number largely. For the Jamendo dataset, for example, ExpLOD records 1,047,950 triples. After adding bisimulation labels, the number of triples increased to 3,940,113, which is about four times more than the original

number. We can observe that the running time for constructing full visualization is much longer than that needed for constructing concise visualization. Moreover, the running time and number of SPARQL queries for constructing **CVdist** are all less than that needed for constructing **CVpop**. The reason is that it does not need to compare the instantiation for every property when constructing **CVdist**. The performance to construct concise visualization is largely improved for the datasets GeoSpecies and DBpedia. One reason is that they have relative complex structure, it reduces a large number of queries when constructing the concise visualization; the other important reason is that they have many mined class subsumptions that help to obtain a small number of maximal classes in the concise visualization.

6 Conclusion

We propose a SPARQL-based tool to construct the actual schema, gather corresponding statistics, and present a UML-based visualization for RDF data sources, such as SPARQL endpoints and RDF dumps. The proposed approach helps users understand the data structure and then construct a suitable SPARQL query. Moreover, for datasets with a complex structure, the Apriori algorithm is used to mine class subsumptions and then create a concise visualization which is convenient for understanding. Experimental results show that our approach does not need to construct a middle graph and is more efficient than ExpLOD. The concise visualization experiments on complex datasets, such as DBpedia, show that our approach is feasible.

Acknowledgements The work is supported by the National Natural Science Foundation of China under grant no. 60973024, no. 61170165, and no. 61003055.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. *IJSWIS* **5**(3), 1–22 (2009)
2. Maduko, A., Anyanwu, K., Sheth, A.P., Schlickelman, P.: Graph summaries for subgraph frequency estimation. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008. LNCS*, vol. 5021, pp. 508–523. Springer, Heidelberg (2008)
3. Cyganiak, R., Stenzhorn, H., Delbru, R., Decker, S., Tummarello, G.: Semantic sitemaps: efficient and flexible access to datasets on the semantic web. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) *ESWC 2008. LNCS*, vol. 5021, pp. 690–704. Springer, Heidelberg (2008)
4. Langegger, A., Woß, W.: RDFStats—an extensible RDF statistics generator and Library. In: 20th International Workshop on Database and Expert Systems Application, pp. 79–83, 2009
5. Hausenblas, M., Halb, W., Raimond, Y., Feigenbaum, L., Ayers, D.: SCODO: Using statistics on the web of data. In: *ESWC 2009*. Springer, Heidelberg (2009)

6. Khatchadourian, S., Consens, M.P.: ExpLOD: exploring interlinking and RDF usage in the linked open data cloud. In: ESWC 2010. LNCS, vol. 6089, pp. 272–287. Springer, Heidelberg (2010)
7. Brockmans, S., Volz, R., Eberhart, A., Loffler, P.: Visual modeling of OWL DL ontologies using UML. In: McIlraith, S.A., Plexousakis, D., Harmelen, F., (eds.) ISWC 2004. LNCS, vol. 3298, pp. 198–213. Springer, Heidelberg (2004)
8. Documents associated with Ontology Definition Metamodel (ODM) Version 1.0, <http://www.omg.org/spec/ODM/1.0/> (2009)
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of 20th International Conference on Very Large Data Bases, pp. 487–499. Morgan Kaufmann, San Francisco (1994)

Empirical Study of POI-Oriented Focused Crawler

Xin Fan, Jun-sheng Zhou, Can-yu Cheng, Yi-chu Zhou, and Di Yin

Abstract Focused crawler is the core of the focused search engine, and the POI-oriented user need is a kind of new focused object which has not been well solved in previous studies. In this paper, we design and realize a POI-oriented focused crawler. The proposed focused crawler adopts classifiers to make relevant judgment and considers both current page's relevance and the URL link information to make the URLs' priority judgment. Experiments were conducted with two kinds of classification algorithms of Naive Bayes (NB) and Support Vector Machines (SVMs) on four sites, respectively. Experimental results show that the focused crawler with NB classifier obtains the average harvest of 95.97%, higher than the one with SVMs by 45.53%, but the focused crawler with SVMs attains the higher recall.

1 Introduction

With the development of Internet technology, the information resources on the web are growing exponentially. In order to retrieve pages of interest to the users, focused search engines have emerged. Focused crawler as the core and foundation of the focused search engine, its performance largely determines the latter's effect. From the perspective of technology combination, existing methods of focused crawler can be divided into the following categories [1-4]: (1) focused crawler based on

X. Fan (✉) • J.-s. Zhou • Y.-c. Zhou • D. Yin

School of Computer Science and Technology, Nanjing Normal University,
Nanjing 210046, China

e-mail: fxhappy1989@126.com; zjsnnu@gmail.com; endymecy@yahoo.cn

C.-y. Cheng

Beijing High Institution Research Center of Engineering Structures and New Materials, Beijing
University of Civil Engineering and Architecture, Beijing 100044, China

e-mail: chengcanyu2007@163.com

intelligent algorithms [5–8], (2) focused crawler based on classification models [9–11], and (3) focused crawler based on semantic concepts [12, 13]. And from the perspective of page analysis, studies of focused crawler also can be summarized as follows: (1) focused crawler based on content analysis [14, 15], (2) focused crawler based on link analysis [16], and (3) focused crawler based on the content and links [17, 18].

The object of focused crawler, which is also called user need, refers to the target pages the user wants to crawl with interest. Existing studies are mostly based on the topic-oriented user needs. Topic keywords are collected manually or in a semiautomatic manner firstly, and then, the relevance between the crawling pages and the topics described with keywords should be calculated to obtain a set of pages related to the given topic. For example, if the topic is limited to “sports,” we can get topic keywords by the given word set such as football, stadium, and sport, which are obtained manually or automatically through ontology or extracted from page contents by feature selection. Then we can calculate the relevance of the crawling page based on the keywords to crawl the pages relative to “sports.”

However, with the development of Internet, people have had a new demand on the focused object: POI-oriented user need, which means the users want to obtain pages which contain special point of interest (POI). Currently, the most typical case of this kind of user need is the comparison of online shopping data. With the advent of a variety of B2C, C2C, and B2B sites, housebound online shopping has become a part of everyone’s life. But it’s also very difficult to find the best and most cost-effective goods from various sites for people expecting to choose best goods by comparison. In order to achieve the goal, we first need to collect the pages that contain the attribute description of specific goods in different sites, extract the attributes information from the pages, and then store the structured information into a database to provide the query and comparison for users. However, faced the mass web pages, how to obtain the pages that contain goods’ basic attribute descriptions becomes the primary problem. Previous studies on focused crawler have not solved this kind of POI-oriented user need. Therefore, in this paper, we proposed a POI-oriented focused crawler on the basis of full analysis of user need and conducted experiments using classification models of Naïve Bayes and Support Vector Machine (SVM) achieving a good harvest rate.

2 Classification Models

2.1 Naïve Bayes

Naïve Bayes classifier [19] is a simplified Bayes algorithm model which introduces the conditional independence assumption based on the Bayes theory, assuming that all the classification features are independent. Let the training sample set be divided into k classes, denoted as $C = \{C_1, C_2, \dots, C_k\}$. The priori probability of each class

C_j can be denoted as $p(C_j)$, $j = 1, 2, \dots, k$. Let P_i denote any web page, represented by its comprising attribute items, i.e., $P_i = (w_1, \dots, w_j, \dots, w_m)$, belonging to class C_j in C . If we want to classify page P_i , the probability of all classification in the case of given P_i , which refers to the posterior probability $p(C_j|P_i)$ of class C_j . It can be calculated as

$$p(C_j|P_i) = \frac{p(P_i|C_j)p(C_j)}{p(P_i)} \quad (1)$$

Bayes classifier is to seek the maximum value of above formula. For a given category, $p(P_i)$ is a constant, and then formula (1) can be converted to formula (2):

$$\max_{C_j \in C} p(C_j|P_i) = \max_{C_j \in C} p(P_i|C_j)p(C_j) \quad (2)$$

According to the conditional independence assumption, the attribute items of P_i are distributed independently. Then the corresponding classification formula is

$$\max_{C_j \in C} p(C_j|P_i) = \max_{C_j \in C} p(C_j) \prod_{i=1}^m p(w_i|C_j) \quad (3)$$

The probability parameters are calculated by maximum likelihood estimation.

2.2 Support Vector Machine

The basic idea of SVM [20] is to map the sample space into a high-dimensional space firstly and then finding an optimal hyperplane in the new space to achieve the best classification results in premise of lowest false rate. Its training set is usually denoted as

$$S = ((x_i, y_i), L(x_i, y_i)) \subset (X \times Y)^I \quad (4)$$

where I is the number of pages, x_i denotes the web page, and y_i is the corresponding mark. X is the input space and Y is the output field. SVM is a model to find a hyperplane by training which can divide all the samples into a certain class correctly and reach the maximum distance between the heterogeneous vectors which are nearest to the plane. Usually it's called the optimal hyperplane and denoted as $(w \times x + b) = 0$, where w is the normal of classification hyperplane, b is the offset, and vector x is in the hyperplane. Then SVM can be turned into an optimization problem:

$$\max \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) \quad (5)$$

$$\text{s.t. } \sum_{i=1}^n y_i \alpha_i = 0 \quad (6)$$

$$\forall i : 0 \leq \alpha_i \leq C, i = 1, 2, \dots, n \quad (7)$$

The above problem is also known as quadratic programming problem under constraints and can be solved by using QP method. But it's often solved by taking advantage of its Lagrangian dual problem and only calculating $f(x) = \sum \alpha_i y_i x_i^T x + b$ during the specific classification; thus, the category of the page can be determined according to the value of $f(x)$. The performance of SVM might be influenced by several factors, including the form of kernel function and its parameters, the complexity degree of the problem itself, noise points near the hyperplane, the parameter selection of input vector, and the samples' number and distribution.

3 Implementation

Focused crawler's goal is to download the pages relative to the specific topic as quick as possible, and minimize the resources consumption at the same time. That is to say, it can use fewer system and network resources to obtain higher page harvest and utilization. So, focused crawler needs to introduce two additional modules: relevance judgment and URL priority judgment modules, in contrast to traditional crawler. Similar to the topic-oriented focused crawler based on the classifiers, the implementation of this POI-oriented focused crawler using classification model also needs feature selection and model training, but there exist great differences on feature selection between them.

3.1 Design of POI Feature Templates

In this paper, we will serve pages that contain the basic attribute descriptions of goods as the focused object to illustrate the design of POI-oriented pages' feature template. Now there are a variety of e-commerce websites, and their partial structures are similar from the perspective of the users' vision, which also refers to the description information in the pages mostly presented in the form of table or list. Two examples of pages including this kind of information are shown in Figs. 1 and 2.

These web pages which contain the information of goods are all described by html tags. Some tags are used to control the layout and the others are used to obtain data or display content. The html codes of goods' descriptions in figures above are shown in Tables 1 and 2 as follows.

Fig. 1 A phone's description in 360buy Mall

品牌	三星(Samsung)
型号	I8250
颜色	金属灰
上市时间	2012年
外观设计	直板
3G视频通话	支持
操作系统	Android 2.3
智能机	是

Fig. 2 A phone's description in Tmall

产品名称: K-Touch/天语 W619
手机价格区间: 1000元以下
网络类型: 联通3G GSM/WCDMA(3G)
机身颜色: 黑色刷安卓4.0.4 纯净版 黑...
是否智能手机: 智能手机
宝贝成色: 全新

Table 1 The html code corresponding to the page in Fig. 1

1.1 Phone's description in 360buy Mall												
<table border="1"> <tr> <td><table></td> </tr> <tr> <td><tr><td>品牌</td></td> </tr> <tr> <td><td>三星(Samsung)</td></tr></td> </tr> <tr> <td><tr><td>型号</td><td>I9008L</td></tr></td> </tr> <tr> <td><tr><td>颜色</td><td>黑色</td></tr></td> </tr> <tr> <td><tr><td>上市时间</td><td>2011年</td></tr></td> </tr> <tr> <td><tr><td>外观设计</td><td>直板</td></tr></td> </tr> <tr> <td><tr><td>3G视频通话</td><td>支持</td></tr></td> </tr> <tr> <td><tr><td>操作系统</td></td> </tr> <tr> <td><td>Android OS v2.2</td></tr></td> </tr> <tr> <td><tr><td>智能机</td><td>是</td></tr></td> </tr> <tr> <td></table></td> </tr> </table>	<table>	<tr><td>品牌</td>	<td>三星(Samsung)</td></tr>	<tr><td>型号</td><td>I9008L</td></tr>	<tr><td>颜色</td><td>黑色</td></tr>	<tr><td>上市时间</td><td>2011年</td></tr>	<tr><td>外观设计</td><td>直板</td></tr>	<tr><td>3G视频通话</td><td>支持</td></tr>	<tr><td>操作系统</td>	<td>Android OS v2.2</td></tr>	<tr><td>智能机</td><td>是</td></tr>	</table>
<table>												
<tr><td>品牌</td>												
<td>三星(Samsung)</td></tr>												
<tr><td>型号</td><td>I9008L</td></tr>												
<tr><td>颜色</td><td>黑色</td></tr>												
<tr><td>上市时间</td><td>2011年</td></tr>												
<tr><td>外观设计</td><td>直板</td></tr>												
<tr><td>3G视频通话</td><td>支持</td></tr>												
<tr><td>操作系统</td>												
<td>Android OS v2.2</td></tr>												
<tr><td>智能机</td><td>是</td></tr>												
</table>												

Through the comprehensive analysis of the POI area and its corresponding html codes, it can be found that the majority of goods' descriptions are in labels "<table>", "", and "" and specific attributes are in labels "<td>" and "". But these tags in one page can be used not only to mark the POI information but also to display other information such as advertisement and copyright. The analysis reveals that different e-commerce websites have the following common characteristics:

Table 2 The html code corresponding to the page in Fig. 2

1.2 Phone's description in Tmall
 产品名称: K-Touch/天语W619 手机价格区间: 1000元以下 网络类型: 联通3G GSM/WCDMA(3G) 机身颜色: 黑色刷安卓4.0.4 纯净版黑色 是否智能手机: 智能手机 宝贝成色: 全新

Table 3 Feature templates for POI

No. Feature description
1. Tag name
2. Total number of cells
3. Ratio of nonempty cells to the total number of cells
4. Number of colons
5. Average length of text in the nonempty cells
6. Domain information in the URL

1. Formats of the pages that contain goods' descriptions within the same site are substantially consistent.
2. Target pages in different sites have great difference on the format.
3. Goods' description fields in different pages are similar and are represented in the form of table or list.

Pages in the same site can be processed with rules to obtain higher accuracy, but different sites need different rules which greatly reduce the system's reusability. However using classification model has no limitation on this type of target sites. Feature selection is the key to build the classifier, and feature templates directly affect the classifier's effect. The topic-oriented focused crawlers usually select features by the following steps: extracting page text, segmentation of text, and word frequency statistics and obtaining features with the methods of document frequency or mutual information. According to the feature templates, we can obtain the page's VSM and judge its relevance with the classifier. However, for the POI-oriented user need, through the above analysis, we can know that text contents in various websites are quite different, and the target pages involve a lot of topics such as clothing, digital, and books. Therefore, the conventional methods of feature selection are not applicable. By analyzing the target pages' characteristics in multiple sites, we proposed the following feature template (Table 3).

Before building the pages' VSM, we need to preprocess the pages' html codes to remove useless label items and prepare for the subsequent processing. Noise labels mainly include “<a> ... ”, “<script> ... </script>”, “<form>

... </form>”, “<style> ... </style>”, “<dl> ... </dl>”, “”, “<link ... />”, “”, “”, “<textarea>”, “</textarea>”, “<!--...-->”, “ ”. Denoising can improve the accuracy of information location, reduce the pages’ redundancy, and optimize the whole analysis. After preprocessing, pages’ VSM should be built according to the feature templates.

Feature 1: Tag name. Difference of labels within the page can be used to filter useless labels such as and <form> and thus reduce the range to judge.

Feature 2: Total number of cells. The statistics of cells in a certain container such as <table>, , or can help to determine whether the information field contains goods’ description. In general, basic attributes of goods are multiple, as shown in Figs. 1 and 2.

Feature 3: Ratio of nonempty cells to the total number of cells. The analysis reveals that the cells are usually filled with text, data, or hyperlinks. After denoising, a cell is either empty or containing static content such as text or data. Most of the goods’ attributes are described with text and data without hyperlinks. Even if there are some hyperlinks, the proportion of nonempty cells will be not less than 50%.

Feature 4: Number of colons. In the field of goods’ description, each attribute is represented in the form of “<property name> delimiters <attribute values>”. One kind of goods has multiple attributes, and the types of attributes are basically consistent, but the corresponding values are quite different. Different attributes have great difference. The attributes of one kind of goods in one site are similar but differ in different sites. Therefore, for multiple goods in different sites, it is difficult to learn rules to recognize the attribute fields with the information of the types and values of attributes. But the delimiters have certain similarities and usually use colons, e.g., “Style: Straight”. The number of colons in the container can help to locate the target information field.

Feature 5: Average length of text in the nonempty cells. In order to let users know goods’ attributes conveniently, websites usually use the form of table or list to display. The attributes and their value are usually brief. Target information of goods can be effectively distinguished from noise by the average length of text in nonempty cells.

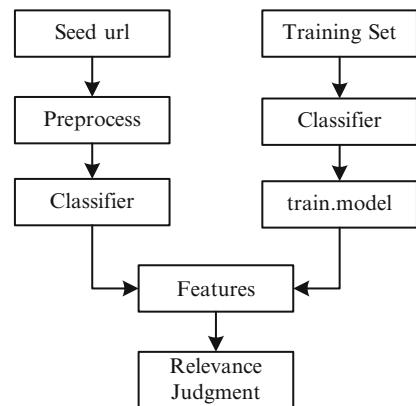
Feature 6: Domain information in the URL. Features described above are all from the target fields and have great difference in different sites. Then the domain information in the URL is introduced to enhance classifier’s performance by using the different characteristics among sites.

Using the above feature templates, it’s effective to distinguish the POI from noise information and locate the description field of goods.

3.2 Relevance Judgment

The goal of relevance judgment is to calculate the similarity of the current page and the target page. For specific POI-oriented user need, whether the current

Fig. 3 The flowchart for POI-oriented focused crawler



page contains goods' description should be judged. Figure 3 shows the process of relevance judgment.

The POI feature templates are designed for the table or list areas in the page to distinguish the target information from noise and locate goods' description field. Then each sample page can be divided into several training instances according to the number of `<table>`, ``, or `` labels it contains. And each instance can be labeled positive or negative according to whether it is the POI information.

Using the designed POI feature templates, we can get the VSM corresponding to the instances in the training set. And then we can obtain the classification model with the classifier algorithms and calculate the relevance of each page to be crawled.

3.3 URL Priority Judgment

Starting from the seed pages, hyperlinks extracted from the current page are in the waiting queue for continuous crawling. The waiting queue of traditional crawlers is sorted according to the hyperlinks' obtaining time. New hyperlinks are pushed into the queue directly from the tail and pulled out from the head when being handled. In order to link to the target page as soon as possible, the focused crawler adopts the method of priority queue and scores for every hyperlink extracted newly. According to the scores, hyperlinks are pushed into the queue orderly to ensure the target pages to be processed earlier and improve the crawling efficiency.

URL priority judgment is to judge the relevance between hyperlinks and target pages. In this paper, we consider two main factors: parent page's relevance and URL itself information. This is based on the following knowledge:

1. The higher the relevance between the current page and target pages, the bigger the probability of its hyperlinks linking to the target page they will have. For example, the product page's navigation bar in taobao site has many hyperlinks,

and the majority of them are pointing to the product pages, while most of hyperlinks in the homepage's navigation bar are pointing to the product list pages or other websites.

2. URLs corresponding to the target pages within the same site have similarity and only differ in a few parameters. For example, the URL corresponding to a product page in taobao is "<http://item.taobao.com/item.htm?id=16158732250>", and its ID number is changing in different products pages. Therefore, for the target pages in different sites, the URL information can be a factor to judge its relevance.

4 Experimental Results and Analysis

In the experiment, we collected 200 pages as training samples from multiple e-commerce websites. These pages contain a total of 1,040 instances in which 200 positive instances and 840 negative instances. We adopted two kinds of training algorithms of Naive Bayes and SVM. Websites "Jiuxian site, Maibaobao site, Yintai site, and Guangjie site" were selected for test with the above algorithms separately. And the homepages were treated as seed URLs. The number of selector channel was 10 and the crawling time was 5 h. For the relevant pages downloaded, their accuracy would be checked manually. The results are shown in Table 4.

In Table 4, ALL denotes the total number of pages downloaded, Good is the number of real relevant pages by manual judgment, and Bad is the number of pages not containing product description. The focused crawlers often adopt harvest and recall as the evaluation indicators. The former refers to the ratio of the number of the obtained relevant pages to the total crawled pages. Recall refers to the ratio of the number of the obtained relevant pages to the total relevant pages. They're calculated as follows:

1. Harvest = Good/ALL
2. Recall = Good/total relevant pages

Harvest can be drawn from Table 4 directly in our experiment. Although we could not get the total number of relevant pages in one website, the number in the same website is certain. Recalls gained by different classification algorithms are

Table 4 Results based on the SVM and NB classifiers, respectively

Site name	Classifier			NB			Crawl time (h)	
	SVM			ALL	Good	Bad		
	ALL	Good	Bad					
Jiuxian	1,950	791	1,159	540	539	1	5	
Maibaobao	1,471	1,357	114	887	877	10	5	
Yintai	2,606	1,425	1,181	1,250	1,247	3	5	
Guangjie	3,822	545	3,277	275	235	40	5	

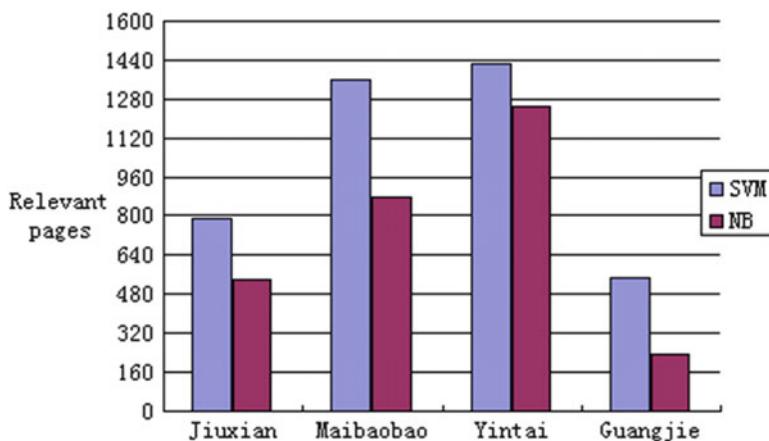


Fig. 4 Contrast of recall among focused crawler achieved by different classifiers

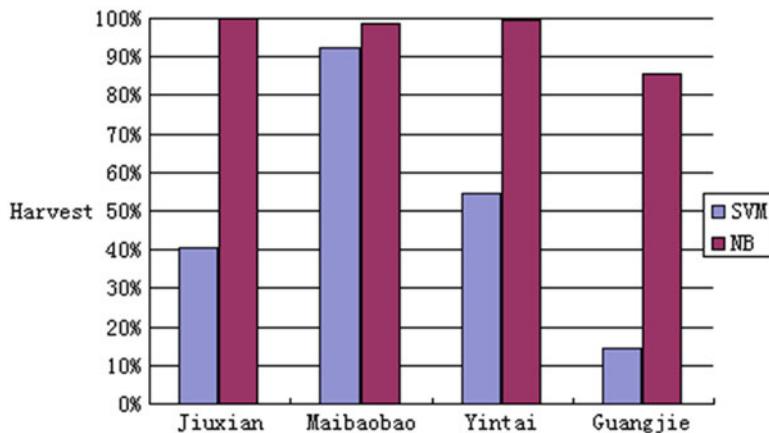


Fig. 5 Contrast of harvest among focused crawler achieved by different classifiers

proportional to the values of Good; thus, they can be compared referring to Good (Figs. 4 and 5).

It can be seen from the above figures that (1) the POI-oriented focused crawler's average harvest is 95.97% by testing on the four sites with Naive Bayes classification model and achieves the highest value of 99.81% on the Jiuxian site. But the average harvest rate is only 50.44% with SVM model. (2) Values of ALL gained by the POI-oriented focused crawler with the NB model are all less than the SVM model. (3) By comparing with recall referring to Good, it can be found that the POI-oriented focused crawler's recall with the NB model is less than the SVM model.

Consequently, the POI-oriented focused crawler with NB model has achieved higher harvest and can filter out unrelated pages to maximum. At the same time,

the focused crawler with SVM model has achieved higher recall and can obtain relevant pages to maximum. By comparing downloaded pages and target pages, we find that most of the misjudgment pages are list pages of products which contain a few of attribute information; therefore, the discrimination between parts of features becomes lower and the classification's effect is affected. The NB model introduces the conditional independence assumption assuming the classification feature items independent with each other and does classification based on the features' overall probability. The feature templates designed in this paper have met the characteristic and the crawler implemented with NB model has achieved the high harvest of 99.81%. The SVM model adopts the mechanism of hyperplane and judges the relevance between support vector samples and current page by calculating their similarities. It ignores most of sample instances and introduces more noise when crawling the relevant pages, but it obtains the overall high recall. The experimental results fully illustrate that harvest and recall are not unrelated and it's difficult to have both. Higher harvest does not always tend to be corresponding to a higher recall.

5 Conclusion

In this paper, we presented a type of POI-oriented user need, treated it as the focused object, and realized the corresponding focused crawler with classification models. Based on the analysis of page structure and target fields, we designed the feature templates suited to the new demand. Relevance of the current page to be crawled was judged with classification algorithms, and URL priority was scored according to the current page's relevance and the URL itself information. We used NB and SVM models on different websites for test, and the results showed that focused crawling pages for the POI-oriented user need are feasible and the corresponding focused crawlers realized with classifiers worked well. However, more in-depth researches are still needed to obtain higher recall and harvest.

Acknowledgements This research is supported by project 61073119 under the National Natural Science Foundation of China and project BK2010547 under the Jiangsu Natural Science Foundation of China.

References

1. Hazman, M.: A survey of focused crawler approaches. *J. Global Res. Comput. Sci.* **3**(4), 68–72 (2012)
2. Zhou, L., Lin, L.: Survey on the research of focused crawling technique. *J. Comput. Appl.* **25**(9), 1965–1969 (2005)
3. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: a new approach to topic-specific Web resource discovery. In: Proceedings of the 8th International World Wide Web Conference, pp. 1623–1640. Elsevier Science, New York (1999)

4. Balaji, S., Sarumathi, S.: TOPCRAWL—community mining in web search engines with emphasize on topical crawling. In: Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering, Salem, Tamilnadu, 2012, pp. 20–24
5. Chen, H., Chung, Y.M., Marshall, R., Yang, C.C.: An intelligent personal spider(agent)for dynamic Internet searching. *Decision Support Syst.* **23**(1), 41–58 (1998)
6. Liu, G., Kang, L., Luo, C.: Focused crawling strategy based on genetic algorithm. *J. Comput. Appl.* **27**(12), 172–174 (2007)
7. Chen, Y., Zhang, Z., Zhang, T.: A searching strategy in topic crawler using ant colony algorithm. *Microcomput. Appl.* **30**(1), 53–56 (2011)
8. Zheng, S.: Genetic and ant algorithms based focused crawler design. In: 2011 Second International Conference on Innovations in Bio-inspired Computing and Applications, Kaohsiung, Taiwan 2011
9. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, Washington, 1998, pp. 307–318
10. Johnson, J., Tsoutsouliklis, K., Giles, C.L.: Evolving strategies for focused web crawling. In: Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, 2003
11. Pant, G., Srinivasan, P.: Link contexts in classifier-guided topical crawlers. *IEEE Trans. Knowl. Data Mining* (2006)
12. Yuvarani, M., Iyengar, N.C.S.N., Kannan, A.: LSCrawler: a framework for an enhanced focused Web crawler based on link semantics. In: The 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06), Hong Kong, 2006
13. Jalilian, O., Khotanlou, H.: A new fuzzy-based method to weigh the related concepts in semantic focused web crawlers. In: 2011 3rd International Conference on Computer Research and Development (ICCRD), Shanghai, 2011, pp. 23–27
14. Peng, H., Wang, Y.: Real-time page classification oriented algorithm on topic extraction. *Comput. Modern.* 8–11 (2008)
15. Taylan, D., Poyraz, M., Akyokuş, S., Ganiz, M.C.: Intelligent focused crawler: learning which links to crawl. In: 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Istanbul, 2011, pp. 504–508
16. Yuan, F.-y., Yin, C.-x., Liu, J.: Improvement of PageRank for focused crawler. In: SNPD 2007: 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007
17. Zhang, X., Li, Z., Hu, C.: Adaptive focused crawler based on tunneling and link analysis. In: 11th International Conference on Advanced Communication Technology, Gangwon-Do, 2009, pp. 2225–2230
18. Batsakis, S., Petrakis, E., Milios, E.: Improving the performance of focused web crawlers. *Data Knowl. Eng.* **68**(10), 1001–1013 (2009)
19. Liu, P., Feng, J.: An improved Naive Bayes text categorization algorithm. *Microcomput. Inform.* **26**(93), 187–188 (2010)
20. Tan, S.: Research on High-Performance Text Categorization. Institute of Computing Technology, Chinese Academy of Sciences, Beijing (2006)

Semantic Word Similarity Learned from Heterogenous Knowledge Bases

Yiling Liu, Yangsheng Ji, Chong Gu, Shouling Cui, and Jiangtao Jia

Abstract Recognition of semantic similarity between words plays an important role in text information management, information retrieval, and natural language processing. There are two major approaches to recognizing the semantic similarity, among which one way is extracting similarity relationships based on a structured semantic dictionary, while the other way is learning the semantic similarity from a large corpus. Building a semantic dictionary is a time-consuming task which also requires much expertise, while the learning method alone cannot extract precise similarity between words. This paper proposes to expand the semantic dictionary by learning the word similarity from heterogenous knowledge bases statistically. This method can not only expand the semantic dictionary from the open knowledge bases but also achieve accurate semantic similarity. In the evaluation of semantic relatedness competition held by CCF, the proposed system ranks the 3rd place according to the macro-average F1 and the 2nd place according to the micro-average F1.

1 Introduction

Natural language has such a diversity that we have multiple choices to describe a single concept. In the domain of text information, the different descriptions of the same concept are often regarded as synonymous words. For example, both of (buy purchase) and (big large) are synonymous word pairs. This means that although the word forms of synonyms are different, they actually have the same semantic meaning.

Y. Liu • Y. Ji • C. Gu (✉) • S. Cui • J. Jia

Section F, Huawei Industrial Base, Bantian Longgang, 518129 Shenzhen, P. R. China

e-mail: liuyiling@huawei.com; jiyangsheng@huawei.com; guchong@huawei.com;

cuishouling@huawei.com; jiajiangtao@huawei.com

The synonymy can give rise to “semantic gap” between words, which makes great challenges for tasks in text mining, information retrieval, and natural language processing. In opinion text, different people tend to comment on the same product attribute in different ways, which are synonymous. In order to get a comprehensive analysis on the reviews of the product, it is essential for review summarization and opinion identification systems to identify the different descriptions of the same attribute [1]. In information retrieval, the query is a kind of descriptions for the concept of users’ interest, while the relevant documents of the concept may employ different kinds of descriptions. The semantic gap which resulted from different descriptions leads to a mismatch between the query and relevant documents [2, 3]. In natural language processing (NLP), the machine translation system after training can be well suited for certain types of descriptions of a concept. However, the system may fail to translate unknown and different descriptions of the concept [4]. The above-mentioned tasks are widely applicable in daily life. However, these applications are faced with severe challenges of semantic gap brought by synonyms. Thus, the identification of semantic word similarity has been an important research topic.

The recent research on approaches to identification of semantic word similarity mainly falls into two categories: extracting similarity relationships based on a structured semantic dictionary and learning the semantic similarity based on a large corpus. The former approaches [5–10] compute the word similarity according to a user-defined similarity measure, which is based on the structure of the dictionary. The dictionary is often created by linguistic experts, who encode the word relationships as the semantic structure in the dictionary. While the dictionary-based approaches are precise for identifying the synonyms, the construction of the dictionary requires much expertise and a huge amount of efforts. As the dictionary is fixed, these approaches are infeasible to extend for handling new words. The second approaches [11–17] compute the word similarity according to a statistical similarity measure based on a large corpus. These approaches can handle more words than the dictionary-based approaches. However, the computed word similarity is not as accurate as the former approaches.

To solve the problems mentioned above, we propose to extract the word similarity from heterogenous knowledge bases. The proposed approach can not only expand the semantic relationships in the dictionary but also achieve accurate results while preserving a higher recall than the dictionary-based approaches. In the evaluation of semantic relatedness competition held by CCF, the proposed system ranks the 2nd place according to the micro-average F1 and the 3rd place according to the macro-average F1.

In the following parts, Sect. 2 introduces the proposed approach, and Sect. 3 performs an extensive evaluation. Finally, we make a conclusion.

2 Identification of Word Similarity from Heterogenous Knowledge Bases

The heterogenous knowledge bases in the proposed approach include a structured synonym dictionary and open resources on the web. The open resources include *Douban*,¹ *Baike*,² and *Baidu-Dictionary*,³ which are heterogenous knowledge bases in the text form. The key of the proposed approach is the fusion of the knowledge bases. First, we mine the synonymous relationship patterns (syn-patterns) from the knowledge bases, respectively. Then, the unstructured information of synonyms on the web can be transformed into structures which comply with the dictionary. Finally, by combining and cleaning the structured information of the former step, the synonym dictionary can get expanded while preserving the high accuracy.

2.1 The Structured Synonym Dictionary

As *Yidian*,⁴ a synonym dictionary, includes a sufficient amount of synonyms, it is selected as the dictionary component of the proposed system. The formal definition of the structured synonym dictionary is as follows:

Definition 1 (The Structured Synonym Dictionary). The dictionary is denoted as $D = D_S \cup D_R$. $D_S = \{(w_i, SID_i, S(w_i))|i = 1, \dots, |D_S|\}$ denotes the synonym item set, with w_i denoting the target word in the synonym item, SID_i denoting the concept ID of w_i , and $S(w_i)$ denoting the set of w_i 's synonyms. $D_R = \{(w_j, RID_j, R(w_j))|j = 1, \dots, |D_R|\}$ denotes the item set of semantic related words, with w_j denoting the target word of the item set, RID_j denoting the related concept ID of w_j , and $R(w_j)$ denoting the set of w_j 's related words.

In Definition 1, the synonym means it shares the same meaning with the target word, while the related word indicates sharing a relevant meaning with the target word. In this paper, the related word is a broader concept than the synonym.

2.2 Extraction of Syn-Patterns from the Knowledge Bases

In *Douban* and *Baike*, person names, organization names, and some other items are explicitly marked with their synonyms in the HTML code of the web pages. These

¹<http://www.douban.com>.

²<http://baike.baidu.com>.

³<http://dict.baidu.com>.

⁴<http://ir.hit.edu.cn>.

Table 1 HTML syn-pattern mining for semantic similarity identification in *Douban*

Input: the target word $w \in D$, and the synonym set $S(w)$

Output: the set of candidate pattern words $pattern(w)$

Steps:

1. type w in the query box of *Douban*, get the content of the return page $page(w)$;
 2. set the candidate pattern words $pattern(w) = \emptyset$;
 3. for each $v \in S(w)$
 4. search for v in $page(w)$; if not found, turn to step 3;
 5. extract the content segment of v in $page(w)$, delete the HTML tags, and get a string $str(v)$;
 6. segment the string $str(v)$ by spaces and tabs, get a group of words, and add them into $pattern(w)$;
 7. end for;
 8. output $pattern(w)$.
-

marks are accurate and reliable for syn-pattern mining. In *Baidu-Dictionary* and *Baike*, the unstructured text also contains much information of synonyms, which can infer the syn-patterns. However, the syn-patterns are not trivial either in the structured HTML code or in the plain text. Thus, we need to design algorithms to mine the patterns in the knowledge bases.

2.2.1 Mining the HTML Syn-Patterns in *Douban*

In order to get the HTML syn-patterns from *Douban*, we need to carry out the following steps: First, we input the target word w in the query box of *Douban* and get the HTML content of the return page. Then, we search for the target word in the HTML content and extract the content segment from the HTML tag before w to the tag after w . At the same time, we record the frequency of words in the content segment. Finally, when the operations for all the target words are carried out, the words with top-ranked frequencies in the content segment can be employed to build the HTML syn-patterns.

It is notable that both Chinese words and other string forms (a word attached with a punctuation) in the segment, which are separated by spaces or tabs, can be indicative for the synonymous relationship. As a result, we need to record all of their frequencies to build the patterns later on. The algorithm for mining the HTML syn-patterns in *Douban* is shown in Table 1.

We gather the candidate pattern words $pattern(w)$ for all $w \in D$ and record the frequency for each word. The words with top-ranked frequencies are selected as the final syn-pattern words $pattern(D)$. In *Douban*, to compute the synonyms of a new word w_{new} , we search for each syn-pattern word $u \in pattern(D)$ in $page(w_{\text{new}})$.

中文名:	关羽	出生日期:	约汉桓帝延熹三年六月
外文名:	GuanYu	逝世日期:	建安二十四年冬（西元219年冬）
别名:	关长生，关云长，关公	职业:	将领
民族:	汉族	主要成就:	阵斩颜良、水淹七军
出生地:	河东郡解县（今山西运城）	官职:	前将军

Fig. 1 An information table from *Baike***Table 2** HTML syn-pattern mining for semantic similarity identification in *Baike*

Input: the target word $w \in D$, and the synonym set $S(w)$

Output: the set of candidate pattern words $pattern(w)$

Steps:

1. type w in the query box of *Baike*, get the content of the return page $page(w)$;
 2. extract the information table $table(w)$ from $page(w)$;
 3. if the extraction fails, turn to step 10;
 4. set the candidate pattern words $pattern(w) = \emptyset$;
 5. for each $v \in S(w)$
 6. search for v in $table(w)$; if not found, turn to step 5;
 7. extract the content segment of v in $table(w)$, delete the HTML tags, and get a string $str(v)$;
 8. segment the string $str(v)$ by spaces and tabs, get a group of words, and add them into $pattern(w)$;
 9. end for;
 10. output $pattern(w)$.
-

If u is found, we extract the content segment from $page(w_{new})$ and remove the tags, syn-pattern words, and punctuations. The remaining string is the synonym.

2.2.2 Mining the HTML Syn-Patterns in *Baike*

Baike contains a lot of items such as person names, location names, and movie names. These items provide much information for synonyms, which are stored in the information tables. The table is shown as in Fig. 1.

Mining the HTML syn-patterns in *Baike* requires in advance extracting the information tables, which could be extracted by matching the regular expression. When the table content is extracted, the syn-pattern mining from the information table is a similar procedure with the algorithm in Table 1. The algorithm for mining the HTML syn-patterns in *Baike* is shown in Table 2.

The above algorithm recognizes the information table in step 2, while other steps are similar with the algorithm for *Douban*. In *Baike*, to compute the synonyms of a new word w_{new} , we search for each syn-pattern word $u \in pattern(D)$ in $table(w_{new})$. If u is found, we extract the content segment from $table(w_{new})$ and remove the tags, syn-pattern words, and punctuations. The remaining string is the synonym.

Table 3 Plain text syn-pattern mining for semantic similarity identification

Input: the synonym dictionary D

Output: the syn-pattern P in the plain text

Steps:

1. initialize the candidate syn-pattern set as $T = \emptyset$;
 2. for each $w \in D$
 3. initialize the candidate syn-pattern set of w as $T(w) = \emptyset$;
 4. type w in the query box, and get the return page content $page(w)$;
 5. extract the plain text $text(w)$ from $page(w)$;
 6. if $text(w)$ is empty, turn to step 2;
 7. for each $v \in S(w)$
 8. search for v in $text(w)$; if not found, turn to step 7;
 9. extract the text segment $context(w, v)$;
 10. perform Chinese word segmentation on $context(w, v)$, and get the syn-pattern $template(w, v)$ for synonym word pair (w, v) ;
 11. $T(w) = T(w) \cup \{ template(w, v) \}$;
 12. end for;
 13. $T = T \cup T(w)$;
 14. end for;
 15. apply frequent pattern mining algorithm, and get $P = FrequentPattern(T)$.
 16. output P .
-

2.2.3 Mining the Syn-Patterns in Unstructured Text

In the open knowledge bases on the web, e.g., *Baidu-Dictionary* and *Baike*, the synonym patterns are more often contained in the plain text than in the structured HTML code. If the syn-patterns can be mined from the unstructured text, the knowledge base of synonyms can be largely expanded, especially the synonym dictionary.

However, the syn-patterns in the plain text are not trivial. So we need to design algorithms to discover the patterns or rules. The syn-patterns in the plain text have similar properties with the HTML syn-patterns. Along with the occurrence of the synonym, there are explicit or implicit marks for indicating the synonymous relationship. The major difference between the syn-patterns is that the syn-patterns in the plain text employ free natural language to describe the synonymous relationship, while the syn-patterns in HTML use limited descriptions. Thus, the syn-patterns in the plain text are more difficult to extract and less reliable than the HTML syn-patterns. The extraction algorithm of syn-patterns in the plain text is shown in Table 3.

The algorithm in Table 3 describes the steps to build the plain text syn-patterns. The core idea is to extract the context template $template(w, v)$ for each synonym word pair (w, v) in the dictionary. We assume that $template(w, v)$ is a set of words without the order information. The words in $template(w, v)$ are extracted from $context(w, v)$, which is the context of the words w and v . $context(w, v)$ denotes the shorter one of (1) the segment from the target word w to the synonym v and

Table 4 Weights for different syn-patterns

Syn-Pattern				Weight	Type
[近义词]	又名:	简称:	别名:	1.0	HTML Syn-Pattern
也说	亦称	俗称	的别名	0.9	Plain Text Syn-Pattern
谓	泛称	亦称(之称	0.8	Plain Text Syn-Pattern
犹				0.7	Plain Text Syn-Pattern
指	的对称			0.6	Plain Text Syn-Pattern

(2) the text segment separated by the punctuations with containing the synonym v . The templates for all synonym word pairs build up a pattern database T . By applying the frequent pattern mining algorithm to T , we can get the most representative syn-patterns from the plain text. The synonym extraction of new words is the same as previous algorithms.

2.3 Evaluation on the Reliability of Syn-Patterns

As mentioned above, the HTML syn-patterns are more reliable than the plain text syn-patterns. Thus, reliable patterns should be assigned higher weights so that they can make more contributions. In order to evaluate the reliability of different patterns, we perform the pattern mining based on a small subset of the dictionary and extract some new synonyms by applying the patterns. By human annotating the new synonyms, we can get the accuracy of each pattern. The higher accuracy of the syn-pattern, the larger weight the syn-pattern will get. We excerpt some results shown in Table 4.

2.4 Synonym Information Fusion from Heterogenous Knowledge Bases

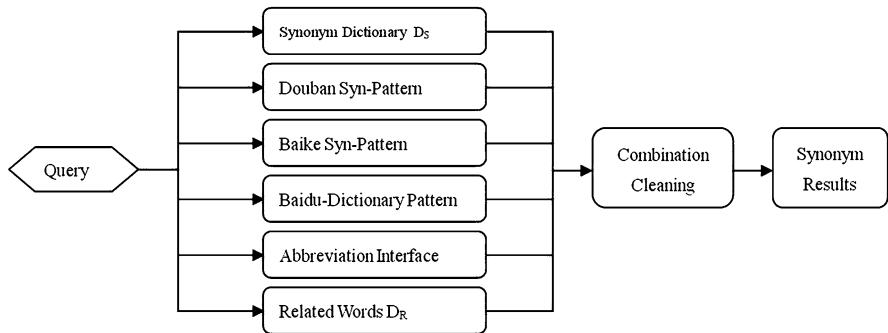
When the syn-patterns are extracted, we can apply them to the web pages and build an open synonym dictionary $D_{\text{open}} = \{S_i^o = \{w_{i,1}, \dots, w_{i,t_i}\} | i = 1, \dots, k\}$. The rank j for $w_{i,j}$ is determined according to the weight of the syn-pattern, which extracts $w_{i,j}$ from the text. To expand the existing synonym dictionary D , we need to fuse D and D_{open} together. The fusion algorithm is shown in Table 5.

2.5 Synonym Computation Based on Heterogenous Knowledge Bases

When the syn-patterns are extracted and the synonym dictionary D has been expanded, the proposed system can compute the synonyms of a query word effectively,

Table 5 Synonym dictionary fusion from heterogenous knowledge bases

Input : D and D_{open}
Output: fused dictionary D
Steps :
1. for each $S_i^o \in D_{\text{open}}$
2. initialize $R = \emptyset$;
3. for each $w_{i,j} \in S_i^o$
4. for each $(w_k, SID_k, S(w_k)) \in D$
5. if $w_{i,j} \in S(w_k)$, then $R = R \cup \{(w_k, SID_k, S(w_k))\}$
6. end for
7. end for
8. if $ R == 0$, update $D = D \cup \{(w_{i,1}, D.\text{totol_id} + 1, S_i^o)\}$;
9. if $ R == 1$, update $(w_k, SID_k, S(w_k)) = (w_k, SID_k, S(w_k) \cup S_i^o)$;
10. if $ R > 1$, the set S_i^o contains multiple concepts, the fusion of R and D needs human annotation;
11. end for
12. output D .

**Fig. 2** Synonym computation based on heterogenous knowledge bases

which is shown in Fig. 2. As abbreviations are a sort of synonyms with complicated forms, the knowledge bases can only include a few of them [19, 20]. We implement a component to deal with the problem [21] so that our system can handle more complicated scenarios.

3 Experiment

3.1 Data Set

The heterogenous knowledge bases include *Yidian* and other open resources on the Internet, e.g., *Baidu-Dictionary*, *Baike*, and *Douban*.

Table 6 Patterns for semantic similarity

HTML Syn-Pattern	“中文名 : ”, “别名 : ”, “本名 : ”, “别名昵称 : ”, “封爵 : ”, “谥号 : ”, “爵位 : ”, “别称 : ”, “公司名称 : ”, “庙号 : ”, “中文名称 : ”, “其它译名 : ”, “粤语名 : ”, “简称 : ”, “外文名 : ”, “近义词 : ”, “同音词 : ”, “中文学名 : ”, “定义 : ”
Plain Text Syn-Pattern	简称[“《为 : 》(.*)?["]) (, 、 ; 。] 简称(.*)?[.,]) (, 、 ; 。]

Table 7 Statistics of expanded synonym dictionary

#Synonym	#Synonym Concept	#Related Word	#Related Concept
79960	28201	38096	3907

The **evaluation data set** for the proposed system employs the data provided by CCF Conference on Natural Language Processing and Chinese Computing 2012. We use NLPCC2012 to denote the data set. NLPCC2012 includes 9455 query items.

3.2 *Syn-Patterns and an Expanded Synonym Dictionary*

Synonymous relationship patterns (syn-patterns) are categorized into the HTML syn-patterns and the plain text syn-patterns. The HTML syn-patterns are extracted from the content of web pages based on the HTML structures. The plain text syn-patterns are extracted from the unstructured text based on the frequent pattern mining algorithm. Table 6 lists some of the syn-patterns in each category.

The statistics of the expanded synonym dictionary are listed in Table 7. It can be seen that the expanded dictionary has a large scale, which is important for real applications.

3.3 *System Evaluation*

We implement two systems for evaluating the proposed approach in this paper. These two systems compute the synonyms for the words in NLPCC2012, and we evaluate the systems according to accuracy, recall, and F1 measure. The systems work as the following procedures:

1. *System1*: if the components of dictionary D_S , *Douban* syn-patterns, or the abbreviation interface produce results, then the system outputs the union of above three components; else if *Baidu-Dictionary* produces results, then the system outputs them; otherwise, the system outputs the results from the component of dictionary D_R .

Table 8 Evaluation of participant systems on the data set NLPCC2012

	Macro	Macro	Macro	Micro	Micro	Micro
	accuracy	recall	F1	accuracy	recall	F1
System1	0.3641	0.5176	0.3664	0.2754	0.5829	0.3740
System2	0.3305	0.5506	0.3635	0.2615	0.6102	0.3662
NNU	0.3588	0.6041	0.3968	0.3025	0.6358	0.4100
ZZU1	0.2975	0.6395	0.3588	0.2530	0.6762	0.3682
ZZU2	0.3256	0.6930	0.3919	0.2540	0.7040	0.3734
CAS	0.1328	0.1034	0.1033	0.4737	0.0687	0.1199
BIT	0.1999	0.2441	0.1874	0.2115	0.2299	0.2203
BJTU	0.2878	0.3394	0.2733	0.3088	0.3737	0.3382
HQU	0.0382	0.0111	0.0151	0.2996	0.0115	0.0221
HIT	0.3225	0.3885	0.2842	0.2303	0.3676	0.2832

2. *System2*: if the components of dictionary D_S , *Douban* syn-patterns, *Baidu-Dictionary*, *Baike*, or the abbreviation interface produce results, the system outputs the union of above components; otherwise, the system outputs the results from the component of dictionary D_R .

The major difference is that system1 doesn't include the results of *Baike*. *Baike* extracts plain text syn-patterns, which can cover a lot of synonym relationships and improve the coverage or recall of the system. The performance evaluation results are listed in Table 8. It can be seen that system1 ranks the 3rd place according to the macro-average F1 and the 2nd place according to the micro-average F1. Although system2 performs a little bit worse than system1, system2 does improve the recall. This means including the *Baike* component can improve the coverage of the synonym relationships.

4 Conclusion

The synonymy can give rise to “semantic gap” between words, which makes great challenges for tasks in text mining, information retrieval, and natural language processing. The identification of semantic word similarity has been an important research topic.

This paper proposes to expand the semantic dictionary by learning word similarity from heterogenous knowledge bases statistically. This method can not only expand the semantic dictionary from the open knowledge bases but also achieve accurate semantic similarity. In the evaluation of semantic relatedness competition held by CCF, the proposed system ranks the 3rd place according to the macro-average F1 and the 2nd place according to the micro-average F1.

References

1. Wang, S.G., Li, D.Y., Wei, Y.J., Song, X.L.: A synonym-based word sentiment orientation discriminating approach. *J. Chin. Inform. Process.* (2009)
2. Cao, J.: Synonyms recognition and application research in concept-based information retrieval system. Master Thesis, Northeastern University (2006)
3. Lu, Y., Hou, H.Q.: Automatic recognition of synonyms for iformation retrieval and its trend. *J. Nanjing Agriculture University (Social Science Edition)* **3**, (2004)
4. Li, Z.F., Yarowsky, D.: Unsupervised Translation Induction for Chinese Abbreviations using Monolingual Corpora. In: *Proceedings of the Association for Computational Linguistics*, 425–433, 2008
5. Budanitsky, A., Hirst, G.: Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures. In: *Workshop on WordNet and Other Lexical Resources of the 2nd North American Chapter of the Association for Computational Linguistics*. Pittsburgh, USA (2001)
6. Mei, L.J., Zhou, Q., Zang, L., Chen, Z.S.: Merge information in HowNet and TongYiCi CiLin. *J. Chin. Inform. Process.* **1**, (2005)
7. Zhang, C.Z.: Research On synonyms dictionary - based on recognition of synonyms. *J. Huaiyin Instit. Tech.* **1**, (2004)
8. Tian, J.L., Zhao, W.: Words similarity algorithm based on Tongyici Cilin in semantic web adaptive learning system. *J. Jilin University (Information Science Edition)* **6**, (2010)
9. Li, Y.H., Bandar, Z.A., Mclean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.* Piscataway NJ USA **4**, 871–882 (2003)
10. Lin, D.K.: An information-theoretic definition of similarity. *International Conference on Machine Learning*, pp. 296–304. Morgan Kaufmann, San Francisco (1998)
11. Salahli, M.A.: An approach for measuring semantic relatedness between words via related terms. *Math. Comput. Appl.* **1**, 55–63 (2009)
12. Pantel, P., Lin, D.K.: Discovering word senses from text. In: *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 613–619. New York, NY (2002)
13. Bollegala, D., Matsuo, Y., Ishizuka, M.: Measuring semantic similarity between words via search engines. In: *Proceedings of the 16th International Conference on World Wide Web*, pp. 757–766. New York, NY (2007)
14. Turney, P.D.: Mining the web for synonyms: PMI-IR vs LSA on TOEFL. In: *Proceedings of the 12th European Conference on Machine Learning*, pp. 491–502, 2001
15. Lu, Y., Hou, H.Q.: Automatic recognition of Chinese synonyms based on PageRank algorithm. *J. Xihua University (Natural Science)* **2**, (2008)
16. Song, Y.X.: The research and implementation of synonyms mining method based on the search log and click log. Master Thesis, Beijing Jiaotong University (2011)
17. Shi, J., Qiu, L.K., Wang, F., Wu, Y.F.: Ensemble methods for similar words extraction. *Research Advances (2009–2011) of Computational Linguistics in China* (2011)
18. Lu, Y., Zhang, C.Z., Hou, H.Q.: Using multiple hybrid strategies to extract chinese synonyms from encyclopedia resources. *J. Library Sci. China* **1**, (2010)
19. Xie, L.X., Sun, M.S., Tong, Z.J., Wang, C.H.: Identification of Chinese abbreviations using query log and anchor text. *Research Advances (2007–2009) of Computational Linguistics in China* (2009)
20. Wang, H.F.: Survey: abbreviation processing in Chinese text. *J. Chinese Inform. Process.* **5**, (2011)
21. Chang, J.S.: A preliminary study on probabilistic models for Chinese abbreviations. In: *Proceedings of the 3rd SIGHAN Workshop on Chinese Language Learning* (2004)

A Workload-Based Partitioning Scheme for Parallel RDF Data Processing

Mengdong Yang and Gang Wu

Abstract Distributed RDF database has been proved to be an effective solution for the Semantic Web big data challenge. Existing distributed RDF data management approaches mostly employ simple and straightforward data partitioning methods, and RDF data characteristics are not considered. The only related work in graph-based RDF data partitioning declares high complexity, which hinders its application in large-scale RDF data management. In this paper, we proposed a parallel RDF data processing approach by partitioning the dataset by pattern graphs. Our partitioning approach takes both graph data characteristics and workload information into consideration. It is advisable in an online transaction processing (OLTP) scenario. Our approach speeds up SPARQL query processing by over an order of magnitude and shows well scalability. The evaluation result under three mainstream benchmarks illustrates the correctness and effectiveness of our approach.

1 Introduction

The Semantic Web is showing its superiority in knowledge representation and interoperability. The Semantic Web has become a web of data and is assembling massive RDF data. The quantity of RDF data on the Semantic Web is therefore increasing. Hence, storage and management of such big RDF data becomes a major challenge. Under such circumstances, several distributed RDF processing approaches were proposed. They partially solve the problem of RDF data processing scalability. However, they still have several bottlenecks in the design. These bottlenecks hinder

M. Yang
EMC Corporation, Beijing, China
e-mail: mengdong.yang@emc.com

G. Wu (✉)
Northeastern University, Shenyang, China
e-mail: wugang@ise.neu.edu.cn

the application of these distributed RDF data processing approaches in the ever-growing RDF data. In this paper, we proposed a new parallel processing technique for distributed RDF data management. Our approach speeds up the reference system by over an order of magnitude and shows impressive scalability in a distributed context.

1.1 Related Work

The very initial RDF data management approaches are centralized and have foreseeable scalability overhead. Numerous distributed RDF data management projects have been proposed under this circumstance. YARS2,¹ Clustered TDB [6], OpenLink Virtuoso [9], and BigOWLIM [9] are distributed evolvements for some of the most well-known centralized RDF databases.

However, existing works on distributed RDF database are mostly simple and straightforward approaches, which mainly concentrate on the indexing of RDF data and SPARQL query optimization.

The reason why we address these methods simple and straightforward is the granularity they choose to partition the global dataset. To summarize, the granularities they applied mainly include RDF document and RDF triple.

RDF document is a collection of physically aggregated triples. RDF documents are usually stored in terms of files on disk. Hence, RDF document is the most straightforward granularity to partition the global dataset. Edutella [5] is a distributed RDF data store based on a P2P topology. In Edutella, RDF documents are partitioned based on the ontology information associated with them. The documents distributed to the same peer are independent from each other, and therefore, the system has reasonable scalability. The cons with these document-based partitioning approaches are obvious as well: the numbers of documents associated with different ontologies may vary a lot. This may cause load skew in the system.

RDF triple is the smallest granularity to partitioning the global RDF dataset. Partitioning by RDF triple is another most widely applied partitioning method in distributed RDF databases. Most well-known distributed RDF databases, YARS2, Clustered TDB, OpenLink Virtuoso, BigOWLIM, RDFPeers [2], Bigdata [9], and AllegroGraph [9], all use triple-based horizontal partitioning. Partitioning by triple requires a lot of joins to be issued during query evaluation, which can be a performance bottleneck. SHARD [7] aggregates triples with the same subject value, thus assembles triples describing the same entity and eliminates intra-entity joins, but is not applicable in multi-entity query context.

As for vertical partitioning in RDF data management, Abadi et al. [1] first proposed a vertical RDF data partitioning scheme, which partitions the global RDF dataset by predicate value and organizes triples in predicate tables. Tanimura

¹<http://sw.deri.org/2007/02/swsepaper/iswc2007.pdf>.

et al. [8] proposed scalable distributed RDF data processing approaches with vertical partitioning based on MapReduce. Since most SPARQL queries contain triple patterns with specific predicate values rather than variables, partitioning by predicate speeds up triple lookup and improves performance. But triples are not distributed uniformly across different predicate values, and as a result, partitioning by predicate may cause distribution skew.

Other topics include workload-based partitioning and graph-based RDF data partitioning. As for workload-based partitioning, Curino et al. proposed Schism [3], a workload-based partitioning scheme in distributed relational database. Schism eliminates most cross-node transactions by analyzing sample query sets. As RDF uses SPARQL, a graph pattern matching-based query language, there would be a lot of join operations in the evaluation process of a query. And when triples are distributed across nodes, a join operation may involve two or more nodes. Since workload-based partitioning can reduce internode operation in a distributed relational database, it could be useful for distributed RDF databases as well. As for graph-based RDF data partitioning, only Huang et al. [4] proposed a data-driven RDF graph partitioning scheme. This RDF data partitioning scheme declares high complexity, which hinders its application in large-scale RDF data management.

1.2 Contributions

In this paper, we proposed a distributed RDF data processing approach based on pattern graph partitioning. Our approach consists of mainly two technical points:

1. A highly efficient RDF data partitioning method based on pattern graph partitioning which takes both RDF data characteristics and workload information into consideration
2. A corresponding SPARQL query processing algorithm

Our partitioning method is advisable in an online transaction processing (OLTP) application context, where the queries have several relatively fixed patterns to follow.

We implemented our approach in Sesame, an open-source RDF data processing framework. The evaluation result under three mainstream benchmarks illustrates the correctness and effectiveness of our approach.

2 Pattern Graph-Based Partitioning

In this section, we will present the concepts and principles of our pattern graph-based partitioning. In Sect. 2.1, terminologies that will be frequently used in our future introduction are defined. In Sect. 2.2, we will explain why our pattern

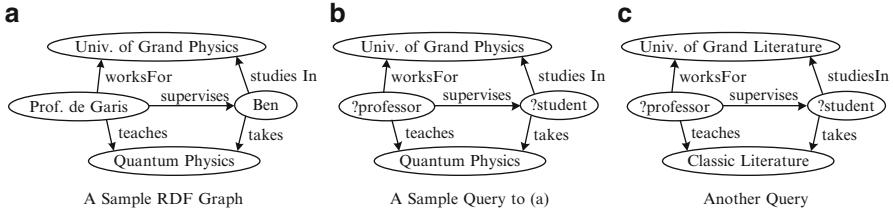


Fig. 1 RDF graph and SPARQL query

graph-based partitioning could speed up SPARQL query processing. Based on this preliminary knowledge, our partitioning approach is introduced in Sect. 2.3.

2.1 Concepts

Definition 1 (Triple). Let \mathcal{U} be the infinite set of URIs, \mathcal{B} the infinite set of blank nodes, and \mathcal{L} the infinite set of literals; then we define $t = (s, p, o) \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ a triple.

Definition 2 (Triple Pattern). Let \mathcal{V} be the infinite set of variables; then we define $t_p = (s_p, p_p, o_p) \in (\mathcal{U} \cup \mathcal{B} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L} \cup \mathcal{V})$ a triple pattern (TP).

Definition 3 (PQT and PQG). We define $t_{qp} = (s_{qp}, p_{qp}, o_{qp}) \in \mathcal{V} \times \mathcal{U} \times \mathcal{V}$ a patternized query triple (PQT). A graph pattern which consists of only PQTs is defined as a patternized query graph (PQG). A graph that matches a specific PQG is defined as a matched pattern graph (we will call it pattern graph for short in the coming text).

2.2 Principle

SPARQL is a graph pattern matching-based query language for RDF. In a *basic graph pattern* (BGP) SPARQL query, only triple patterns are included. Figure 1 shows an example of an RDF graph and two SPARQL queries. Figure 1a describes a fact that student Ben and Professor de Garis do their jobs in University of Grand Physics on Quantum Physics. Ben and Professor de Garis have a student-supervisor relationship.

In query Fig. 1b, there are five triple patterns, and four join operations need to be issued during query evaluation. Figure 1b presents a BGP SPARQL query that queries $?professor$ and $?student$, whose values can be easily answered with the RDF graph in Fig. 1a.

Since join operation can be expensive when data are in very large scale, our first inspiration comes from the possibility to eliminate these join operations. The easiest way is to prefetch the result of query Fig. 1b. It will effectively increase the evaluation performance of query Fig. 1b after the first time. But in real applications, such scenario isn't likely to happen. The successive query is more likely to be a query like Fig. 1c. So simply prefetching the result of query Fig. 1b is only effective in some minor scenarios, which doesn't contribute obvious performance increase.

2.2.1 Patternization

Fortunately, there are still some common patterns to follow. Although query Fig. 1b and query Fig. 1c are two distinct queries that specify different criteria, they are all about querying ?student and ?professor, where ?student and ?professor have a student-supervisor relationship and ?student and ?professor work in the same university on the same course. In real OLTP applications, such scenario is virtually very commonly seen. Take an e-commerce application, for example, customer A first searched a book that has some specific name. Then customer B logged in and searched a cell phone with another name. To cover all these possible queries, an abstraction of existing queries needs to be performed. We call this abstraction *patternization*. In our system, before the system is loaded with RDF triples, a set of SPARQL queries are input and to which the patternization are performed. Basically, the patternization includes the following tasks:

1. BGPs are extracted from these SPARQL queries.
2. For each BGP, the subjects and objects are replaced with variables (if they are not). Thus, all triples in the BGP become PQTs and the BGP itself becomes a PQG.
3. The subject/object positions that have specific values in the BGP are recorded as a reference to build search indexes for the pattern graphs. The pattern graphs discovered under this PQG in the future are all indexed in the same positions.

2.2.2 Pattern Graph Discovering

By the end of patternization, a set of SPARQL queries are transformed into a set of PQGs. The PQG can be used to discover pattern graphs in the global RDF dataset. After patternization, the global RDF dataset is loaded triple by triple. Each time a triple is inserted, a pattern graph discovering operation is triggered. There are three possible conditions for new pattern graph discovery. Figure 2 shows the three conditions: growing, assembling, and branching.

In the *growing* condition, the pattern graph grows from a smaller pattern graph. Growth may start from any position of the PQG. The growing condition happens when a new triple is added. In the *assembling* condition, two sub-

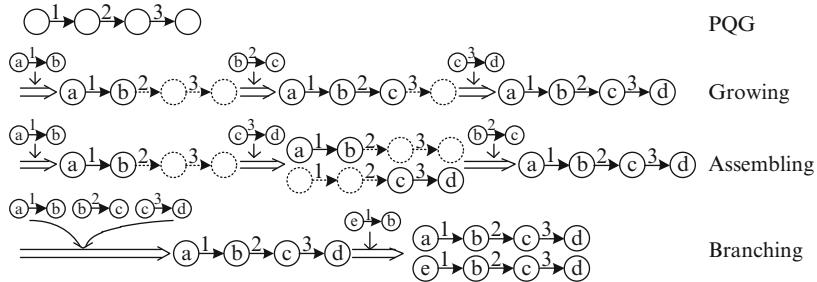


Fig. 2 Three conditions

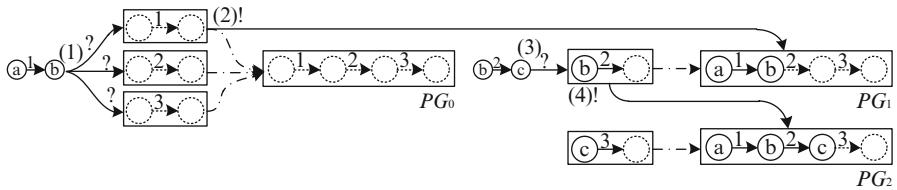


Fig. 3 A fragment of the whole growing process

pattern graphs are connected to form a new larger pattern graph. The *assembling* condition happens between two existing sub-pattern graphs. The *branching* condition is actually not a stand-alone condition. In the *branching* condition illustrated in Fig. 2, when $\langle e, 1, b \rangle$ is added, there is an implicit sub-pattern graph $\{ \langle b, 2, c \rangle, \langle c, 3, d \rangle \}$, and this sub-pattern graph grows with $\langle e, 1, b \rangle$ and hence $\{ \langle e, 1, b \rangle, \langle b, 2, c \rangle, \langle c, 3, d \rangle \}$ is produced. In fact, $\{ \langle b, 2, c \rangle, \langle c, 3, d \rangle \}$ is an intermediate result in the process of generating the previous pattern graph, $\{ \langle a, 1, b \rangle, \langle b, 2, c \rangle, \langle c, 3, d \rangle \}$.

2.2.3 The Growing Condition

The growing condition is what is first to be done when a triple is added. The system needs to search its *sub-pattern graph store* (SPGS) and locate those sub-pattern graphs that can grow with the newly added triple. To do this, we apply the *inverted index* to index the sub-pattern graphs. Figure 3 shows a fragment of the whole growing process with inverted index.

Initially, there is only an ESPG PG_0 , which has three keys. In step (1), a triple $\langle a, 1, b \rangle$ is added. This triple matches key $\langle ?, 1, ? \rangle$ and sub-pattern graph $PG_1 = \langle a, 1, b \rangle$ is produced in step (2). PG_1 is a nonempty sub-pattern graph and accepts triples that match the neighbor keys of existing triples. In PG_1 , the existing triple is $\langle a, 1, b \rangle$ and the neighbor key is $\langle b, 2, ? \rangle$. In step (3), another triple $\langle b, 2, c \rangle$ matches key $\langle b, 2, ? \rangle$ of PG_1 and PG_2 is produced in step (4). PG_2 has a key $\langle c, 3, ? \rangle$.

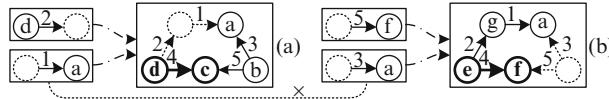


Fig. 4 An example of overlap

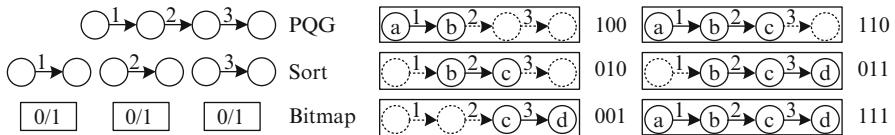


Fig. 5 Bitmap principles

2.2.4 The Assembling Condition

To discover the assembling condition, one extra operation is added. When a new sub-pattern graph is produced, the inverted index is searched to find sub-graph patterns that could connect to the new sub-pattern graph. There is possibility to make errors when discovering pattern graphs of the assembling condition. Two matching sub-pattern graphs may overlap, as is shown in Fig. 4.

To avoid such errors, we introduced *bitmap index* to index the triple presence of each sub-pattern graph. Triple positions are sorted by their predicate values, and each triple position uses a bit to indicate whether a triple exists in this triple position. When two matching sub-pattern graphs under the same PQG are about to be connected, just issue an AND operation with the two bitmap index values. If the result is zero, that means they don't overlap. Otherwise they overlap and can't be connected. Figure 5 shows the principles of this bitmap design.

Take the two pattern graphs in Fig. 4 as an example. (a) has a bitmap index value of 00111 and (b) has a bitmap index value of 11010; the result of the AND operation is 00010, a nonzero value. Thus, (a) and (b) can't be connected, though they have matching inverted index keys: $\langle ?, 1, \alpha \rangle$ to (a) and $\langle ?, 3, \alpha \rangle$ to (b).

2.2.5 The Branching Condition

When a new sub-pattern graph or pattern graph is produced, the previous smaller sub pattern graph is not deleted to ensure the correctness of discovering pattern graphs of the branching condition.

By now, the pattern graph discovering method is completed.

2.3 Partitioning

In distributed environment, when a new pattern graph is produced, the triple is sorted by predicate, and the subject value of the first triple is used to partition the pattern graph. Given slave nodes $S_0, S_1 \dots S_{n-1}$, for a pattern graph \mathcal{G} and the subject value v of its first triple, \mathcal{G} is partitioned to S_m , where

$$m = \mathcal{H}(v) \text{ MOD } n \quad (1)$$

\mathcal{H} is a hash function that generates an integer with the subject value as a string.

3 SPARQL Query Processing

3.1 BGP Query Processing

With these pattern graphs, our system is able to process SPARQL query efficiently without issuing a lot of join operations. Figure 6 shows the flow of BGP SPARQL query evaluation with pattern graphs.

In steps (1)–(2), a BGP SPARQL query is patternized, generating a PQG and the positions where query criteria exist. In step (3), pattern graphs are discovered as triples are added to the system. The discovered pattern graphs are indexed in the positions determined during query patternization. When a new query with the same pattern comes in (4), a lookup step (5) in the index is performed. In step (6), one pattern graph is hit and is returned in step (7). Finally the pattern graph is projected with the variable list and query result is given.

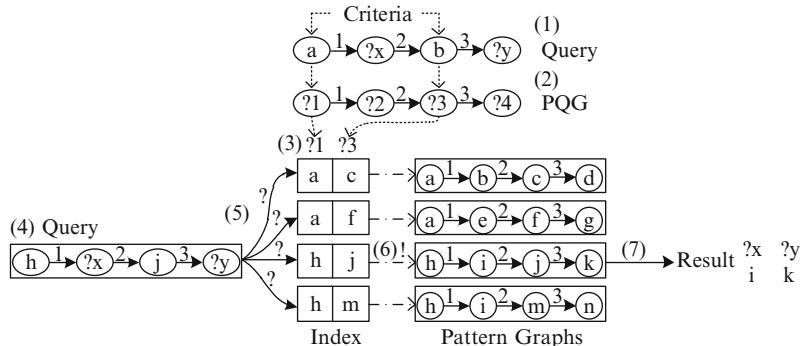


Fig. 6 SPARQL query evaluation with pattern graphs

3.2 Complex Query Processing

3.2.1 Filter

is an operator that could be appended to a graph pattern. Our system first extracts the BGP out of the SPARQL queries with Filter clauses, answers the BGP with the methods presented in Sect. 3.1, and processes the evaluation result with the Boolean expressions specified in the Filters clauses. If one BGP result fails to pass any of the given Boolean expression, it is removed from the final query results.

3.2.2 Optional

is an operator that has the same semantics as the outer join operation in relational database. In our system, the Optional clause is first removed to create a BGP query. After that the BGP query results are involved in an ordinary outer join processing, and final query results are produced.

3.2.3 Union

is an operator that has the same semantics as the union operation in the set theory. In our system, a SPARQL query with Union is first split by Union into multiple sub-queries, each of which has only one graph pattern. Each of these sub-queries is evaluated separately. After all these sub-queries are processed, the result sets are involved in a set union calculation and final query results are produced.

4 Implementation and Evaluation

We implement our scheme with Sesame, an open-source RDF database written in Java. We use a single-master-multiple-slave distributed architecture to implement the distributed version of our system.

Three mainstream benchmarks are employed in our evaluation. LUBM is a benchmark that generates RDF data based on a university ontology. LUBM accepts the number of university as the input parameter to control the scale of output data. SP²Bench is a benchmark based on an ontology derived from DBLP schema and directly uses triple number to control generated data scale. Both LUBM and SP²Bench provide a fixed set of SPARQL queries. BSBM is a benchmark based on an e-commerce application ontology and accepts the number of goods items as scale control parameter. BSBM uses dynamically generated SPARQL queries during benchmark runtime.

Table 1 LUBM datasets and BSBM datasets

	LUBM(10)	LUBM(100)	LUBM(1000)	BSBM(1K)	BSBM(10K)	BSBM(100K)
Triples	1.3M	13.8M	138M	0.3M	3.5M	35M

M=million

Table 2 Evaluation steps

Load dataset \mathcal{DS} in environment \mathcal{E}	
1	Total response time $\mathcal{T} \leftarrow 0$
2	Repeat 50 times
3	Execute a query mix, $t \leftarrow$ time cost
4	$\mathcal{T} \leftarrow \mathcal{T} + t$
5	Output $\mathcal{T}/50$

$\mathcal{E} = \{\text{Centralized, 1M2S, 1M4S, 1M8S}\}$, M=Master, S=Slave
 $\mathcal{DS} = \{\text{LUBM}(10), \text{LUBM}(100), \text{LUBM}(1000), \text{SP}^2\text{Bench}(2.5M), \text{SP}^2\text{Bench}(10M), \text{SP}^2\text{Bench}(40M), \text{BSBM}(1K), \text{BSBM}(10K), \text{BSBM}(100K)\}$

Table 1 shows the scale of the three LUBM datasets and BSBM datasets. Since SP²Bench directly uses the number of triples to control generated data scale, there is no need to specially present the scale of SP²Bench datasets.

Table 2 shows the evaluation steps.

4.1 Evaluation Metrics

4.1.1 Evaluation Time

is a metric that evaluates the performance of the system. Less time means higher system performance.

4.1.2 Speedup

is a metric that evaluates the scalability of our system. Assume the time of a query mix evaluation under centralized environment is \mathcal{T} and the time of a query mix evaluation under distributed environment of n slaves is \mathcal{T}_n ; then the speedup under this n -slave distributed environment, \mathcal{S}_n , is defined as $\mathcal{S}_n = \frac{\mathcal{T}}{\mathcal{T}_n}$.

The ideal value of \mathcal{S}_n is n and \mathcal{S}_n is less than its ideal value in real-world environment. So a bigger \mathcal{S}_n means higher performance and scalability.

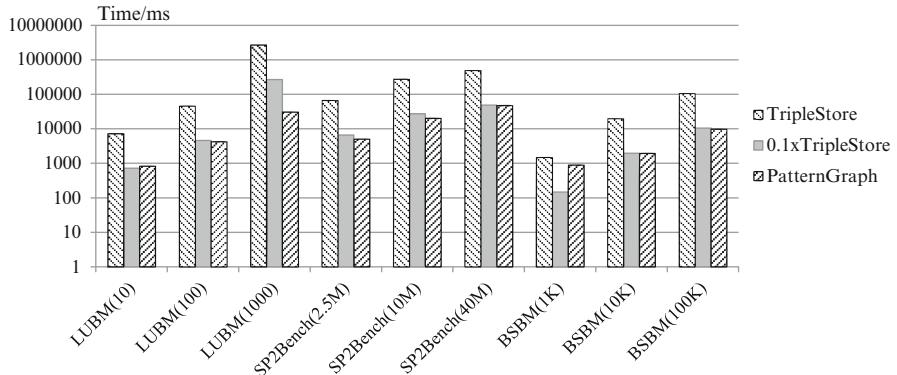


Fig. 7 Evaluation time under centralized environment

4.1.3 Load Balance

is a metric that evaluates the uniformity of load distribution. In a well-designed distributed system, the load distribution should be uniform across all slave nodes to achieve the best performance. In the evaluation of load balance in our system, we trace every request sent to slave nodes and do statistics of the number of requests for each node.

4.2 Evaluation Results

4.2.1 Evaluation Time

Figure 7 presents the evaluation time under centralized environment. There are three data series in Fig. 7. The first series is the evaluation time when using the original version of Sesame. The second series is a reference series, which is 1/10 the time of the first series. The third series is the evaluation time when using pattern graph-based store. We can see from Fig. 7 that, when using large-scale datasets, pattern graph-based store has a performance higher than 10-time-speed triple-based RDF store. Figure 7 illustrates that our pattern graph-based approach can effectively increase the performance of an RDF database by over an order of magnitude.

Figure 8 shows the evaluation time under distributed environment. It can be seen that when using large-scale datasets, the evaluation time is reduced significantly as the number of nodes increases. The evaluation result in Figs. 7 and 8 demonstrates that our pattern graph-based partitioning is effective under both centralized environment and distributed environment.

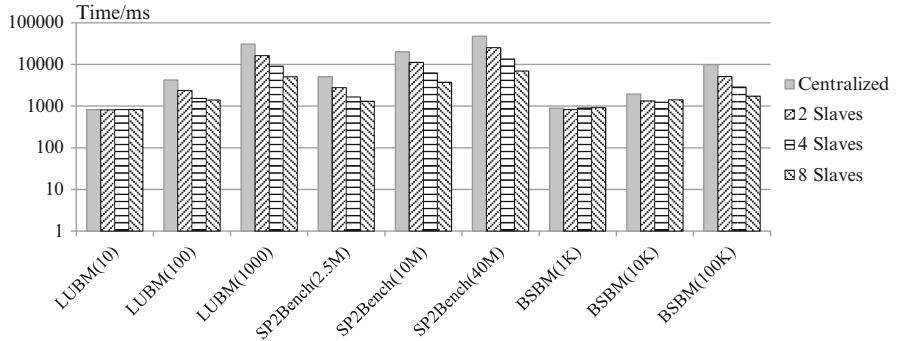


Fig. 8 Evaluation time under distributed environment

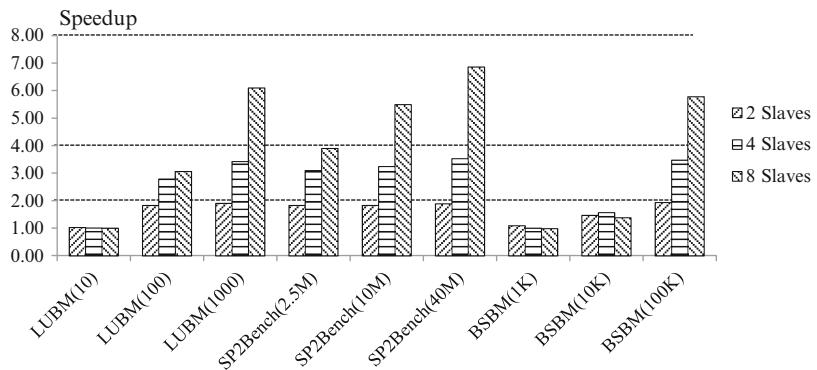


Fig. 9 Speedup

4.2.2 Speedup

Figure 9 presents the speedup when using different datasets under different distributed settings. Three reference lines are added: speedup=2, speedup=4, and speedup=8. Reference line speedup=2 is added for the first series, the 2-slave configuration. Reference line speedup=4 is for the 4-slave and reference line speedup=8 is for the 8-slave. It can be seen from Fig. 9 that our system has an impressive speedup when processing large-scale datasets under distributed environment. The 2-slave configuration has a speedup of nearly 2 on large datasets, while the 4-slave has around 3 and 8-slave has around 6.

4.2.3 Load Balance

Figure 10 presents the load distribution. We can see from Fig. 10 that our system has a very uniform load distribution under distributed environment. The reason why the per-node requests are not reduced as the number of slaves increases is that the

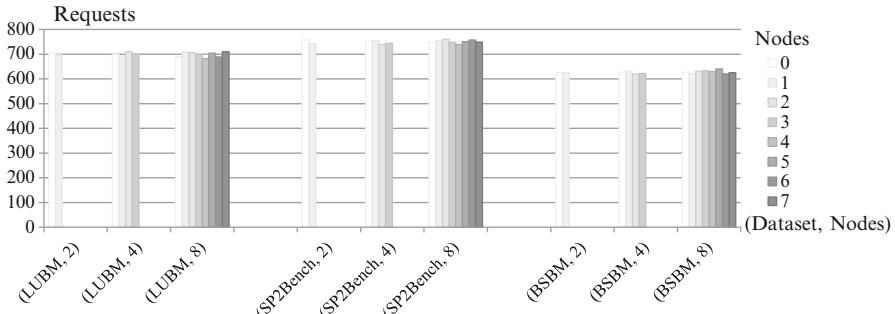


Fig. 10 Load distribution

lookup keys in the SPARQL queries are different from the partition key in our pattern graphs. As a result, every query needs to be broadcasted to all slaves to get the integrated query results.

5 Conclusion and Future Work

In this paper, we presented a workload-based partitioning scheme for parallel RDF data processing. To summarize, this method does the following three tasks:

1. Extracts PQGs from a given set of SPARQL queries
2. Discovers pattern graphs with the PQGs, stores them, and indexes them
3. Partitions the pattern graphs across multiple processing nodes

Since our method is a workload-based approach, it shows high efficiency in RDF data processing which runs a SPARQL query set with a relatively fixed pattern. This characteristic makes our approach a good solution for large-scale OLTP applications that have an RDF data back end. We also provide the evaluation data of our approach and a Sesame-based implementation. The evaluation results illustrate the correctness and effectiveness of our approach.

Future Work

Although our approach shows great advantage in large-scale OLTP Semantic Web applications, it still has several shortcomings, which point out our future directions:

1. *Improvement of current PQG identification.* Existing method identifies PQG by sorting the PQTs in it by their predicates and generating a serialization of the sorted PQTs. Provided that there could be multiple PQTs that have same predicate value in one single PQG, this method may have ambiguity. The future work in this direction is to figure out a way to effectively identify PQGs without ambiguity.

2. *Compression of intermediate results in pattern graph discovering.* Our current method records all intermediate results in the process of pattern graph discovering. The objective of this design is to correctly discover all three conditions of new pattern graphs. Many intermediate results are not effectively utilized. This could be an impact on space utilization when large-scale data are processed and complex queries are issued. Our future work in this direction is to find a solution of compressing this intermediate result to increase space utilization.

Acknowledgements Project supported by the National Natural Science Foundation of China (Nos. 60903010, 61025007, and 60933001), the National Basic Research Program (973) of China (No. 2011CB302200-G), the Natural Science Foundation of Jiangsu Province, China (No. BK2009268), the Fundamental Research Funds for the Central Universities (No. N110404013), and the Key Laboratory of Advanced Information Science and Network Technology of Beijing (No. XDXX1011).

References

1. Abadi, D.J., et al.: Scalable semantic web data management using vertical partitioning. In: Proceedings of the 33rd International Conference on Very Large Data Bases, 2007
2. Cai, M., et al.: RDFPeers: a scalable distributed RDF repository based on a structured peer-to-peer network. In: Proceedings of the 13th International Conference on World Wide Web, 2004
3. Curino, C., et al.: Schism: a workload-driven approach to database replication and partitioning. In: Proceedings of the VLDB Endowment, vol. 3(1–2), pp. 48–57, 2010
4. Huang, J., et al.: Scalable SPARQL querying of large RDF graphs. In: Proceeding of the VLDB Endowment, vol. 4(11), 2011
5. Nejdl, W., et al.: EDUTELLA: a P2P networking infrastructure based on RDF. In: Proceedings of the 11th International Conference on World Wide Web, 2002
6. Owens, A., et al.: Clustered TDB: A Clustered Triple Store for Jena: Electronics and Computer Science, University of Southampton, 2008
7. Rohloff, K., et al.: High-performance, massively scalable distributed systems using the MapReduce software framework: the SHARD triple-store. In: Programming Support Innovations for Emerging Distributed Applications, 2010
8. Tanimura, Y., et al.: Extensions to the Pig data processing platform for scalable RDF data processing using Hadoop. In: ICDE Workshops, pp. 251–256, 2010
9. W3C. Large Triple Stores. <http://www.w3.org/wiki/LargeTripleStores>

Chinese Microblog Sentiment Analysis Based on Semi-supervised Learning

Shaojie Zhu, Bing Xu, Dequan Zheng, and Tiejun Zhao

Abstract This paper adopts a semi-supervised method which is based on bootstrapping to analyze Sina microblog data which size is about 269 M. The Support Vector Machine (SVM) method is used in subjective and objective classification and polarity classification. Our method can extend the size of seed samples by learning automatically with a small size of labeled corpus. It can improve the ability of sentiment classification of SVM by using the iteration method. A weighted factor to control the weight of new seed samples during the following training process can improve classification performance. The experiment results show that sentiment analysis of Chinese microblog based on bootstrapping not only saves much time of manual annotation but also can get better performance. The results of subjective and objective classification achieve the best accuracy rate of 62.9 %, and the best accuracy rate of sentiment polarity classification is 57 %.

1 Introduction

With the emergences of microblog, a lot of users are organized into social network, which satisfies the personalization publication of users' information, sociality transmission, and social communication needs. Microblog has characteristics with social media as well as instant messaging. Plenty of users express their personal views and emotions freely on various hot events, characters, products, etc., and this information has great commercial value and useful value. Facing the challenges of information explosion, people need to acquire this information much faster and

S. Zhu • B. Xu (✉) • D. Zheng • T. Zhao

Harbin Institute of Technology, School of Computer Science and Technology,
150001 Harbin, China

e-mail: sjzhu@mtlab.hit.edu.cn; xb@mtlab.hit.edu.cn; dqzheng@mtlab.hit.edu.cn;
tjzhao@mtlab.hit.edu.cn

more effectively. Microblog sentiment analysis is generated in this context and becomes a hot research problem in recent years.

Microblog sentiment analysis mainly refers to analyze microblog subjective information with the existing sentiment analysis technology. At present mature research results mainly focus on Twitter sentiment analysis, and research contents include subjective and objective classification and sentiment polarity classification. It is just a beginning of Chinese microblog sentiment analysis. This paper will do research on Chinese microblog sentiment analysis.

Chinese microblog content, whose size is 140 characters, is more abundant in content than Twitter, and its expression is more random, which increases the processing difficulty of Chinese microblog sentiment analysis. Therefore the text sentiment analysis technology with supervised learning, unsupervised learning, and semi-supervised learning has developed rapidly [1].

In supervised learning, it needs a lot of labeled training samples. But with the rapid development of data collection and storage technology, it becomes much easier to collect a large number of unlabeled samples, while it is relatively more difficult to collect amounts of labeled samples, which limits the scale of samples. And the model usually has dependence of specific field [2]. Unsupervised learning saves labor and time. But without the guidance of labeled samples, it leads to lower accuracy. In this case, semi-supervised learning draws more attention. Semi-supervised learning not only makes full use of amounts of low-cost unlabeled data, conserves resources while saving a lot of time and effort, but also gains the classifier with strong generalization ability. This is why semi-supervised learning attracts more and more attention [1, 2].

To the state of Chinese microblog sentiment analysis, this paper proposes a bootstrapping semi-supervised learning approach to process Chinese microblog sentiment analysis. This method can achieve a comparable result to supervised learning, which can save much labor and time at the same time.

2 Related Work

Text sentiment analysis is also known as opinion mining, namely, the procedure that analyzes, processes, generalizes, and reasons the subjective text with emotional color. Through extensive survey, we have found that existing research of sentiment analysis based on semi-supervised learning is still focused on sentiment information classification and sentiment information extraction.

Davidov [3] proposed robust SASI (semi-supervised sarcasm identification) algorithm to process subjective and objective information classification and opinion sentence extraction. Riloff [4] proposed recognition subjective nouns based on semi-supervised learning algorithm to process subjective and objective information classification. Wan [5] proposed co-training algorithm for cross-lingual sentiment classification, which mainly concerns making full use of labeled English corpus to process the problem of Chinese sentiment classification. Li [6] proposed co-training

semi-supervised learning algorithm to process the polarity problems with imbalanced sentiment classification. Jin [7] proposed a machine learning framework using lexicalized HMMs. Using improved L-HMM based on bootstrapping and a little labeled data into automatic learning can improve the performance of the system. Wang [8] proposed cross-training algorithm to recognize both product properties and opinion words. The main idea is that using a much small labeled corpus trains NNBopword classifier and NNBproperty classifier based on Naive Bayesian method and then selects the highest confidence data of each iteration into train corpus. This paper learns from this idea in selecting seed sets.

So far, the research of microblog sentiment analysis mainly focuses on Twitter. The main content is microblog sentiment classification. Jiang [9] proposed target-dependent strategy. Experimental results show this approach can greatly improve the performance of target-dependent sentiment classification. Wang [10] proposed a graph-based hashtag sentiment classification approach. Xie [11] investigated Chinese microblog sentiment classification based on Support Vector Machine (SVM).

3 Bootstrapping Approach

This paper [12] indicated that bootstrapping is an automatic classification learning approach. This approach only takes a much small labeled samples as seed corpus. Firstly, seed corpus is used as train corpus to train the classifier. Secondly, the unlabeled samples are labeled by the classifier. Finally, part of those samples is integrated into train corpus as new seeds. This process is repeated until the end of algorithm. In this way, bootstrapping can achieve a better classifier which only uses small labeled seeds and a lot of unlabeled corpus.

L indicates labeled seed corpus, and U indicates unlabeled corpus; the main process of bootstrapping learning is as follows:

1. Train SVM classifier with labeled seed corpus L .
2. Classify test corpus with the classifier, and treat the accuracy of test corpus before iteration as baseline.
3. Classify unlabeled corpus U with the classifier.
4. Sort the classification results, and then take part of the highest confidence data as new seed corpus L' ; do it as follows in the condition to keep balance in L :

$$L + L' \rightarrow L$$

$$U - L' \rightarrow U$$

5. Use the newest L to train SVM classifier.
6. Classify test corpus with the classifier, and get the accuracy of test corpus.

If there is enough unlabeled samples, we set terminal condition $n = 50$. If it meets, it's over; else, return to step 3.

4 Experiment

Feature selection in our experiments includes effective features, such as the word and part of speech, and some specific symbol in microblog, like microblog emoticon sets and sentiment dictionaries.

In order to minimize the adverse impact of the new seeds in the training process, weighted factor is introduced to decrease the weighted factor value during the train process. Formula (1) shows the definition of weighted factor δ :

$$\delta_{(s,c)} = \begin{cases} \beta(s \in U) \\ 1(s \in L) \end{cases}. \quad (1)$$

4.1 Sentiment Classification of Microblog

All parameters of SVM^{light} toolkit are default in the sentiment classification task. The number of iterations sets 50, and the size of subjective blogs is 500, which is equal to the size of objective blogs. Formula (2) gives the definition of accuracy:

$$\text{accuracy} = \frac{\text{System.Correct}}{\text{System.Total}}. \quad (2)$$

Table 1 lists the best accuracy values of different feature sets in bootstrapping.

As it can be seen in Table 1, the accuracy of the first feature set is the worst one, while the accuracy of the third feature set is the best, which is 0.626. The second result is quite similar to the third one, both of which are higher than the first one 1 % at least. It indicates emoticon and Hownet features are effective to increase the accuracy value.

The best weighted factor selection experiments depend on the optimal feature set. In our experiments we set the weighted factor value of 0.2, 0.4, 0.6, 0.8, and 1.0. Table 2 lists the best accuracy values of different weighted factors in bootstrapping.

The results show that $\delta = 0.8$ is the best, the accuracy of which is 0.629. The result suggests the weighted factor can improve the performance of Chinese microblog sentiment classification task. All results prove the sentiment classification method based on bootstrapping is effective.

Table 1 The best accuracy values of different feature sets

Feature set	Accuracy value
POS + TF-IDF	0.611
POS + TF-IDF + emoticon	0.621
POS + TF-IDF + emoticon + Hownet	0.626

The values provided in italic are the best values

Table 2 The best accuracy values of different weighted factors

δ	Accuracy value
0.2	0.627
0.4	0.624
0.6	0.626
0.8	0.629
1.0	0.626

The values provided in italic are the best values

Table 3 Compare the best accuracy value in different feature sets of two experiments

Feature set	The best accuracy value of related experiment	The best accuracy value of unrelated experiment
POS + TF-IDF	0.547	0.545
POS + TF-IDF + emoticon	0.546	<i>0.570</i>
POS + TF-IDF + emoticon + Hownet	<i>0.551</i>	0.551

The values provided in italic are the best values

4.2 Polarity Classification of Microblog

In order to increase the accuracy value of Chinese microblog sentiment polarity, we divide the emoticon set into positive and negative emoticon list. For each blog, calculate its positive emoticon probability and negative emoticon probability. At the same time, we process the Hownet as well as emoticon set.

All parameters of SVM^{light} toolkit are default in the polarity classification task. The number of iterations sets 50, and the size of positive blogs is 100, which is equal to the size of negative blogs.

Table 3 lists the best accuracy value in different feature sets of subjective and objective related experiment and unrelated experiment. The former experiment refers that train corpus of polarity classification comes from the results of sentiment classification, while the unrelated experiment refers that train corpus comes from the original data corpus.

Table 3 shows that the accuracy value of second feature set is the best in subjective and objective unrelated experiment. It is higher than the results in other conditions 2.5 % at least, and the best value is 0.570.

The best weighted factor selection of polarity classification depends on the optimal feature sets and corpus of subjective and objective unrelated experiment. Table 4 lists the best accuracy values of different weighted factors in bootstrapping.

As can be seen in Table 4, the results of using weighted factor decrease on account of unlabeled corpus orientation, in which each sample contains emoticon symbol.

Table 4 The best accuracy value of different weighted factors

δ	Accuracy value
0.2	0.554
0.4	0.555
0.6	0.553
0.8	0.552
1.0	0.570

The values provided in italic are the best values

The experimental results of polarity classification indicate that the best accuracy value is 0.570 using corpus of subjective and objective unrelated, weighted factor $\delta = 1.0$ and integrate word, TF-IDF, and emoticon set features.

5 Conclusion and Future Work

This paper adopts a semi-supervised method which is based on bootstrapping and integrates the SVM method to research on sentiment classification and polarity classification of Sina microblog data. Various feature combinations and weighted factors are introduced to improve the performance. The results of sentiment classification achieve the best accuracy rate of 62.9 %, and the best accuracy rate of sentiment polarity classification is 57 %. Our contributions are using semi-supervised learning and SVM classifier in Chinese microblog sentiment analysis and introducing weighted factor in bootstrapping to reduce the impact of adverse factors. In future work, we will improve the sentiment analysis performance in the following two ways: (1) build the corpus tagging platform and extend the size of training corpus and (2) think of the context semantic feature, social relationships, and relationships between microblogs to improve the performance.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Nos. 61073130 and 61173073) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A207).

References

- Li, S.S., Huang, C.R., Zhou, G.D., et al.: Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, 2010, pp. 414–423
- Niu, G., Luo, A.B., Shang, L.: A survey of semi-supervised text categorization. J. Front. Comput. Sci. Technol. 5(4), 313–323 (2011)
- Davidov, D., Tsur, O., Rappoport, A.: Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. In: AAAI, 2010

4. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: Proceedings of the Seventh CoNLL Conference Held at HLT-NAACL2003, Edmonton, 2003, pp. 25–32
5. Wan, X.J.: Co-training for cross-lingual sentiment classification. In: Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, Suntec, Singapore, 2009, pp. 235–243
6. Li, S.S., Wang, Z.Q., Zhou, G.D., et al.: Semi-supervised learning for imbalanced sentiment classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp. 1826–1831
7. Jin, W., Ho, H.H.: A novel lexicalized hmm-based learning framework for Web opinion mining. In: Proceedings of the 26th International Conference on Machine Learning, Montreal, 2009, pp. 465–472
8. Wang, B., Wang, H.F.: Bootstrapping both product properties and opinion words from Chinese reviews with cross-training. In: 2007 IEEE/WIC/ACM International Conference on Web Intelligence, 2007, pp. 259–262
9. Jiang, L., Yu, M., Zhou, M., et al.: Target-dependent Twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, 2011, pp. 151–160
10. Wang, X.L., Wei, F.R., Liu, X.H., et al.: Topic sentiment analysis in Twitter: a graph-based Hashtag sentiment classification approach. In: CIKM, 2011, pp. 1031–1040
11. Xie, L.X.: Sentiment analysis of Chinese micro blog using SVM. Master Thesis. Tsinghua University, Beijing (2011)
12. Chen, W.L., Zhu, M.H., Zhu, J.B., Yao, T.S.: Semi-supervised text categorization using bootstrapping. *J. Chin. Inform. Process.* **19**(2), 86–92 (2005)

Lexicon-Based Sentiment Analysis on Topical Chinese Microblog Messages

Anqi Cui, Haochen Zhang, Yiqun Liu, Min Zhang, and Shaoping Ma*

Abstract Microblogging is a popular social media where people express their opinions and sentiment on social topics. The Chinese microblogging service, called *Weibo*, has become a remarkable media in the Chinese society. People are eager to know others' attitudes towards social events; thus sentiment analysis on those topical microblog messages is important. In this paper we introduce a lexicon-based sentiment analysis method. We construct a *Weibo Lexicon* with representative topical words and out-of-vocabulary (OOV) words, which are usually informal and are not existing in formal dictionaries. In addition, we use a propagation algorithm to automatically assign sentiment polarity scores to the discovered words. These scores are more closely reflecting the *Weibo* context since words may have new or opposite polarities instead of their formal meanings. Evaluations on the classification tasks show that our method is effective on recognizing the subjectivity and sentiment of *Weibo* sentences. The *Weibo* lexicon increases the performance of the classifications.

* This work was supported by Natural Science Foundation (60903107, 61073071), National High Technology Research and Development (863) Program (2011AA01A207) and the Research Fund for the Doctoral Program of Higher Education of China (20090002120005). This work has been done at the NUS-Tsinghua EXtreme search centre (NExT).

A. Cui (✉) • H. Zhang • Y. Liu • M. Zhang • S. Ma

State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

e-mail: cuiyanqi@gmail.com; rukycz@outlook.com; yiqunliu@tsinghua.edu.cn;

z-m@tsinghua.edu.cn; msp@tsinghua.edu.cn

1 Introduction

Microblogging is a popular User-Generated Content (UGC) service in the Web 2.0 era. Different from a traditional web article, a microblog message has a limited length of content, typically 140 characters. This results in a faster composition; thus users participate in microblogging more often. Microblogging has become a significant media and a rich corpus of users' emotions and opinions, especially on hot topics and events. Since then, researchers have paid much attention to microblog messages.

The textual content of a microblog message is different from traditional web texts, mainly because of its length limit. For example, the topic in a shorter piece of text is usually more focused, and the expressed emotion (or opinion) is more straight forward. This makes it easier for lexicon-based analysis. However, words and expressions used in microblog messages are less formal. They contain abbreviations and out-of-vocabulary (OOV) words which make it more difficult to understand the content. Thus, microblog messages as informal short texts have become a new interest to the researchers.

Currently the most popular microblogging service in the world is *Twitter*¹ which has more than 500 million users as of April 2012.² The Twitter messages (called *tweets*) are mostly in English; hence most studies focus on English tweets analysis. Unfortunately, Twitter is not accessible in mainland China. As an alternative, Chinese microblogging services (called *Weibo*) have attracted the Internet users in China. The most popular Weibo services include *Sina Weibo*,³ *Tencent Weibo*,⁴ etc. As of June 2012, 274 million (50.9%) Internet users in China have used Weibo services [2], within only three years since the sites' opening to the public. Though Weibo is new in China's Internet services, it has now become an influential media to the society. Many companies, celebrities, and governments are eager to know what people are talking about in Weibo, especially people's opinion against some specific topic. To the opposite, research on Weibo analysis is still preliminary. Besides the fact that Weibo gets popular much later than Twitter, some characteristics of Chinese microblogs different from English microblogs make it more difficult on textual analysis:

1. Chinese as a character-based language contains more information than English with a same length. In Weibo, each message is limited to 140 Chinese characters, but it contains more words and sentences than an English message with 140 alphabets. Longer contents lead to the fact that sentences in one message may or may not be coherent; sometimes they express different topics or even different opinions. Figure 1 shows an example Weibo message on the topic of "Teachers in

¹<http://twitter.com/>.

²http://www.mediabistro.com/alltwitter/500-million-registered-users_b18842.

³<http://weibo.com/>.

⁴<http://t.qq.com/>.

S#	Original Text	English Translation	P
1	#中国教师收入全球几垫底# 没有几个。	#Teachers in China get almost the lowest salary in the world# Not so many.	ϕ
2	妈妈今天还说当老师很不容易。	Today mom said it isn't easy to be a teacher.	+
3	不过老师虽然累，但有一群她的学生们。	A teacher works hard and is demanding, but she has her students.	-
4	好老师的心态永远、每天都会是：累并快乐着！	A good teacher should always keep happy though demanding!	+

Fig. 1 An example Weibo message with four sentences (S#: Sentence number. P: Sentiment polarity, ϕ for neutral, + for positive, and - for negative). Different sentences have different sentiment polarities

M#	Original Text	English Translation
1	#疯狂的大葱#天葱它肿么了？	#Crazy green onions# What happened to the green onions?
2	#疯狂的大葱#什么都涨，就工资不涨。等工资涨了，什么就都又涨了。 <u>鸭梨</u> ，，，	#Crazy green onions# Every price is going up except salary. When salary increases, every others is rising again. Pressure...

Fig. 2 Example Weibo messages with new words or meanings (M#: Weibo message number. New words and their translations are underlined). The OOV word “肿么了” pronounces similar as “怎么了” (what happens to). The word “鸭梨” (pear) sounds like “压力” (pressure), so it is often used in Weibo as self-deprecation

China get almost the lowest salary in the world.” The message has four sentences with a total of 71 Chinese characters, while its English translation has more than 240 characters. The sentences have different sentiment polarities, implying that a finer granularity, i.e., sentence level of analysis, is necessary.

- Chinese word segmentation is a big challenge in Chinese text analysis. Existing Chinese word segmentation tools and part-of-speech (POS) taggers usually work well on formal texts. However, Internet texts are often informal with many OOV words. Moreover, formal words in Internet contexts may have opposite meanings against their original meanings. Existing algorithms usually fail under these situations. For sentiment analysis, current sentiment lexicons contain only the formal words and their formal polarities. In Internet texts, we need to extend the existing lexicons to catch more sentiment words and expressions and discover their “new” meanings. Figure 2 shows two examples of a topic “Crazy green onions” where people complain about the high prices of green onions.

Facing these problems, on one hand, in this paper we restrict our Weibo sentiment analysis problem to the sentence level analysis on topical Weibo messages. This treatment removes the diverse irrelevant topics in Weibo and is more useful in practice. On the other hand, our methodology especially discover words from the Weibo corpus. These words are then assigned Weibo-based polarity scores, which represent the sentiment polarities in the context of Weibo instead of formal texts.

The paper is organized as follows: Sect. 2 gives a formal definition of our problems. Sections 3 and 4 introduce the algorithm of constructing the Weibo lexicon and classification with them. Section 5 shows the experiments and evaluations. Section 6 introduces related work. Finally, Sect. 7 concludes the paper and raises possible future work.

2 Problem Definition

In this paper, we conduct a two-stage sentiment analysis on topical Chinese Weibo messages. The messages are collected with hashtags (topical words or phrases quoted by a pair of # symbol). These messages contain less spam and are relevant to the specified topic. The messages are then divided into sentences for analysis.

Task 1: Given a sentence in a Weibo message, we find out whether or not it is an *opinionated* sentence. Opinionated sentences are the ones that have comments (opinions) on some specific objects, exclusive of personal feelings, wishes, and moods. For example, “I am happy” is not opinionated; “I love iPhone’s screen effects” is opinionated.

Task 2: Given an *opinionated* sentence, we find out its sentiment *polarity*, i.e., if the opinion is positive, negative, or others. Figure 1 shows some examples of the polarities of Weibo messages.

Note *Task 1* is served as the first stage of *Task 2*. Our evaluations of both tasks are based on the ground truth of opinionated sentences.

3 Lexicon Construction

As mentioned before, Chinese Weibo messages have many OOV words and words that have new meanings. Thus we deploy algorithms to automatically discover them and their sentiment polarities, which form the *Weibo lexicon*.

We first pick out some words as the entries of the lexicon. These entries come from the mid-frequency words and OOV words in our corpus. Then we use a label propagation algorithm to assign polarity scores to them.

3.1 Representative Topical Mid-Frequency Words

We collect a background corpus from Tencent Weibo. A total of 849,783 Weibo messages (of 2,264,464 sentences) are collected, covering three months prior to the 22 topics in our training and test datasets. The words in these messages are not concentrated on those topics.

We also collect an extended set of the 22 topics in Tencent Weibo. A total of 44,603 Weibo messages (of 108,113 sentences) are collected. The distribution of words in these messages is much closer to the one in the training and test datasets.

After the messages are cut into sentences, word segmentation (with the ICT-CLAS tool [10]) is applied to generate a preliminary segmentation result, together with their POS tags. To overcome the incorrect segmentation, we also generate bi-words and tri-words in addition to the uni-words.

The top-50 n -gram words in each set (uni-words, bi-words, and tri-words) from the background corpus are considered as high-frequency words, mainly stop words. The n -gram words with frequency lower than three in the extended topic corpus are considered as low-frequency words, mainly username or proper nouns. From the extended set, we remove the high- and low-frequency words and get the mid-frequency words, which represent the corresponding topic.

3.2 OOV Words

The OOV words are discovered with context entropy gain and mutual information [5]:

$$\text{Entropy}(W) = - \sum_{i \in \text{Next}(W)} p_i \ln p_i \quad (1)$$

where $\text{Next}(W)$ is the adjacent word (character) of the original word W , and p_i is the probability of that word.

These OOV words are grown from characters in the extended topic corpus; hence, they are independent to the segmentation results which are not so reliable in Weibo texts. The OOV words serve as a complement to the word segmentation results.

Note the OOV words may contain a shorter word that is a substring of another longer word. When generating the features, we match the longer words first. If a longer word matches in the sentence, all of its substrings are not considered.

3.3 Sentiment Polarity Assignment

We construct a co-occurrence graph to propagate polarity scores to the words in the lexicon. The mid-frequency words and the OOV words are nodes, while their co-occurrences in the Weibo sentences are edges. The weights of the edges are the numbers of co-occurrences between the two nodes.

A public sentiment dictionary [11], containing 728 positive words and 933 negative words (all are formal words), is used for seeds for label propagation. The propagations are conducted twice, one for positive scores (starts with the positive seeds) and one for negative scores (starts from the negative seeds). The label propagation step is similar to [3], which assigns a score to each word by:

$$x^{(n+1)} = x^{(n)} \times W / \left(\sum \sum W_{ij} \right) \quad (2)$$

Table 1 Example non-seed words with polarity scores

Word	Translation	Positive score	Negative score
蛊惑人心	Demagogic	0.521	0.759
嚣张跋扈	Typo of 嚣张“跋”扈	0.493	0.728
嚣张跋扈	Arrogant and domineering	0.572	0.759
宋祖德	A celebrity who always criticize others in public thus has a low reputation	0.584	0.757
/心碎	Icon of a breaking heart	0.544	0.728
性感	Sexy	0.792	0.626
不拘一格	Not restricted to rigid rules	0.757	0.588
李娜	A top-10 tennis player, the first from China to win a Grand Slam in singles	0.744	0.579
/给力	Icon of 给力, a new word in Internet meaning “awesome”	0.763	0.627

where x is the node vector and W is the adjacent matrix of the co-occurrence graph. Starting with the vector $x^{(0)}$ where only the seeds have a score of one, all the other nodes are propagated with scores after iterations. The scores of seeds are reset to one again between two iterations. Table 1 lists some example words (exclusive of the seeds) and their scores.

From the table we see that the algorithm automatically recognizes and assigns polarity scores to typos, new words, as well as celebrities. Positive words have a higher positive score than its negative score, and vice versa. Typos and new words are common in Internet texts, especially in Weibo; thus this methodology is helpful for Network Informal Language analysis.

The polarity scores are within the range of [0, 1]. Words with scores higher than a given threshold are considered as positive (or negative) words. In our work we use multiple thresholds to generate multiple lexicons. For example, a threshold of 0.5 excludes “嚣张跋扈” from the negative lexicon, while a lower threshold may keep it.

4 Subjectivity and Sentiment Classification

In this section we introduce our supervised classification method.

4.1 Feature Candidates

Three categories of features are chosen as candidate features for the classifier:

1. Non-semantic features: Length (number of characters) of the sentence. Word count (by segmentation).

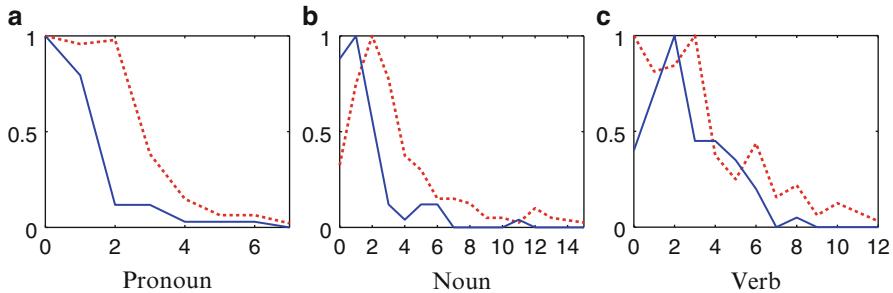


Fig. 3 Normalized number of opinionated (dashed red) sentences and non-opinionated (blue) sentences to the number of words of a specified POS in each sentence. Distributions are computed from the training set. **(a)** Pronoun, **(b)** Noun, **(c)** Verb

Table 2 Number of entries of the sentiment dictionaries

Dictionaries	Positive count	Negative count
Hownet	836	1,254
Student Dictionary	728	933
NTUSD	2,810	8,276

2. POS features: Word count of each POS tag. Some POS tags are more predictive to the subjectivity such as pronouns, nouns, and verbs, as shown in Fig. 3. For example, sentences with no noun are more likely to be opinionated compared to sentences with two nouns.

One set of the POS features are directly from the ICTCLAS output, i.e., all words are counted into the POS frequencies. The other set is from the Weibo lexicon. In this set we only consider the words that occur in this lexicon. Their POS (or bi-words and tri-words) frequencies are contributed to the features. In this way, the high- (and low-) frequency words do not influence the POS count.

3. Sentiment dictionaries: We collect three public sentiment dictionaries from Hownet,⁵ a Student Dictionary [11], and NTUSD [4]. Table 2 lists the statistics of these dictionaries. Features are the number of positive (and negative) words appeared in the dictionaries. Hence $3 \times 2 = 6$ features are generated.

In addition, we use our Weibo lexicon to generate more sentiment features. As mentioned before, different thresholds generate different word lists; hence, the word counts are different. We use them as individual features.

A feature selection algorithm is later used to remove some redundant features from the three categories of candidates.

⁵http://www.keenage.com/html/c_bulletin_2007.htm.

4.2 Classification Method

A provided training set is used to train the classifier. In this work we use libSVM [1] with its default settings as the classifier.

We first classify the subjectivity of each sentence (Task 1). The subjective (opinionated) sentences are used for sentiment classification (Task 2). A challenge is that most of the topics are social events where people tend to criticize on them. The training sets for Task 2 is imbalanced. One commercial topic has 41 positive and 59 negative sentences (out of all the opinionated sentences), while the other social topic have negative sentences only (170 sentences). We combine the two topics together as a full training set to decrease the imbalance.

5 Experiments and Evaluations

5.1 Corpus Details

The corpus is provided from *Tencent Weibo* by the Weibo Sentiment Analysis Evaluation Tasks of the First Conference on Natural Language Processing & Chinese Computing (NLP&CC 2012). The training set contains two topics, with a total of 205 Weibo messages with 464 sentences. The test set contains 20 topics, with 17,518 messages of 31,675 sentences. Among them, 1,908 sentences in 10 topics have been annotated. The labels are annotated by experts and are considered as gold standards. Table 3 lists the proportions of classes in each topic.

5.2 Feature Selection

To avoid overfitting, we first remove redundant features—features that do not bring information gain in the training set.

$$\text{InfoGain}(C, F) = \text{Entropy}(C) - \text{Entropy}(C|F) \quad (3)$$

where C is the class and F is the feature (attribute).

The selected features are:

1. Non-semantic features: Length of the sentence. Word count.
2. POS features: Adjectives, attributive words, adverbs, numerals, nouns, quantifiers (measure words), auxiliary words, and verbs from both the ICTCLAS POS tags and the Weibo lexicon. ICTCLAS's pronouns and punctuations, as well as Weibo lexicon's bigrams and trigrams, are also included.

Table 3 Proportions of classes in topics

Dataset	topic#	Opinionated		sentiment polarity		
		Yes	Total	Pos	Neg	Others
Training	1	170	242	0	170	0
	2	101	222	41	59	1
	Total	271	464	41	229	1
Test	1	126	220	5	121	0
	2	128	176	21	106	1
	3	132	222	21	111	0
	4	115	147	8	107	0
	5	123	135	110	13	0
	6	133	217	25	105	3
	7	112	147	2	110	0
	8	122	201	8	110	4
	9	142	230	10	129	3
	10	135	213	24	109	2
	Total	1,268	1,908	234	1,021	13

3. Sentiment dictionaries: Negative word count of all the dictionaries, both public resources and our sentiment lexicons. The negative words are important here due to the imbalanced training set.

5.3 Classification Results

Within the test set of more than 30,000 sentences, experts have annotated 1,761 labels. These gold standards are used to evaluate our performance.

Evaluation measurements are precision, recall, and F-measure:

$$\text{Precision} = \frac{\#\text{system_correct}(\text{opinion} = Y)}{\#\text{system_proposed}(\text{opinion} = Y)} \quad (4)$$

$$\text{Recall} = \frac{\#\text{system_correct}(\text{opinion} = Y)}{\#\text{gold}(\text{opinion} = Y)} \quad (5)$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

We compute the micro-average (performance on the whole dataset) and macro-average (average of performances on each topic). Moreover, we compare the results with and without the features of our Weibo lexicons. Figures 4 and 5 illustrate the classification performances on subjectivity and sentiment polarity. The figures show that adding our Weibo lexicon features increases the classification performances.

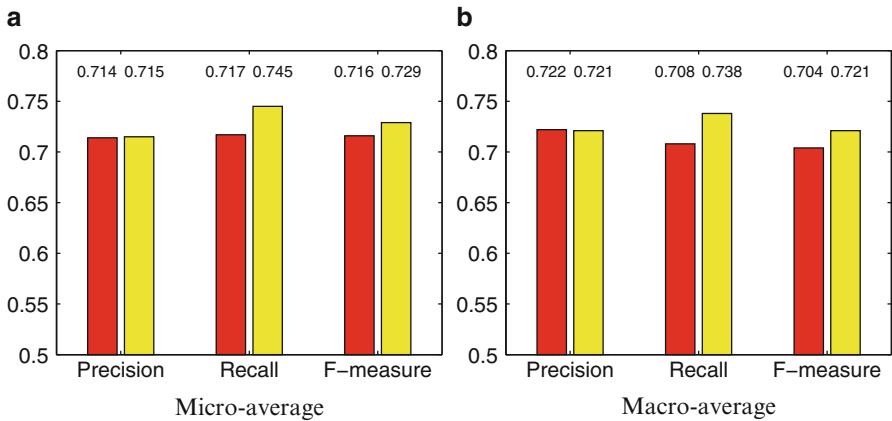


Fig. 4 Comparison of without (left red bar) and with (right yellow bar) the Weibo lexicon features on subjectivity classification. (a) Micro-average, (b) Macro-average

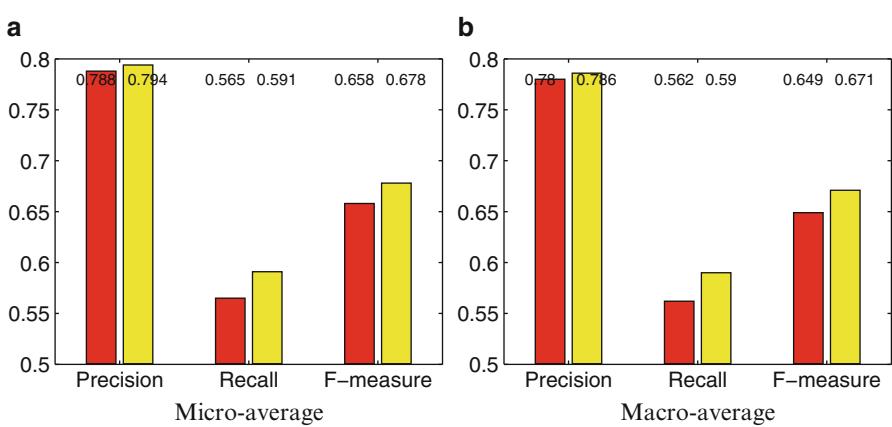


Fig. 5 Comparison of without (left red bar) and with (right yellow bar) the Weibo lexicon features on sentiment classification. (a) Micro-average, (b) Macro-average

The method on subjectivity classification produces similar precision and recall rates. For sentiment classification, however, precisions are higher than recalls. The reason is similar as the one causing errors in the subjectivity task, where the misclassified sentences are usually with positive sentiment. Due to the imbalanced training set, positive information is not learned correctly; thus these sentences are often recognized as non-opinionated.

6 Related Work

Sentiment analysis is important in Web text analysis. Traditional methods include building statistical models with machine learning techniques [6, 9]. Part-of-speech tags are also used for rule-based approaches [7]. These traditional studies usually focus on formal or longer texts (such as product reviews), where words are formal and their POS tags are reliable.

In Twitter sentiment analysis, researchers have been using some specific features such as emoticons [8] or irregular spellings [3]. However, Chinese is not a spelling language. People do not use English emoticons very often, since it requires them to switch their input methods from typing Chinese characters to English alphabets. In addition, irregular spellings (such as repeating letters) are less common. These characteristics restrict us from these methods.

In Chinese microblogs, Weibo, people are using emotional icons provided by the Weibo websites. People choose icons on the web interface to insert them into their posting messages. Thus, these icons are clues for sentiment analysis [12]. However the icons are not widely used in most Weibo messages; we still need to find a way to analyze the text-only messages.

7 Conclusion and Future Work

In this paper, we propose an algorithm for sentiment analysis on topical Chinese microblog (Weibo) messages. The method is based on the Weibo lexicon, which is automatically generated from the microblog corpus. It contains many new words and OOV words which help recognize the sentiment of informal texts.

Weibo messages have many typos and new words. Formal words may also have different meanings from their original meanings. Therefore, we build a lexicon from a background Weibo corpus and a topical Weibo corpus to collect potential sentiment words or n -grams that reflect the topics. A label propagation algorithm is applied to assign sentiment polarity scores to the words we have discovered. In this way, the actual polarities of the words with respect to the Weibo context are assigned.

Topical Weibo messages usually have imbalanced sentiment polarities. Many social events lead to a huge number of negative opinions. Therefore, training data is highly imbalanced. This also limits our supervised algorithm. In future work, we will try to find some balanced corpus to construct the lexicon. Some other models for imbalanced training, such as one-class SVM, can also be considered. Moreover, we would like to evaluate the lexicon itself, for how precise the scores are assigned to the new words.

Acknowledgements The NExT search center is supported by the Singapore National Research Foundation and Interactive Digital Media R&D Program Office, MDA, under research grant (WBS: R-252-300-001-490).

References

1. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**(3), 27:1–27:27 (2011)
2. CNNIC: The 30th china internet development report. Tech. rep., China Internet Information Center (2012)
3. Cui, A., Zhang, M., Liu, Y., Ma, S.: Emotion tokens: bridging the gap among multilingual twitter sentiment analysis. In: Proceedings of the 7th Asia conference on Information Retrieval Technology, pp. 238–249. AIRS’11, Springer, Berlin, Heidelberg (2011)
4. Ku, L.W., Chen, H.H.: Mining opinions from the web: Beyond relevance retrieval. *J. Am. Soc. Inf. Sci. Technol.* **58**(12), 1838–1850 (2007)
5. Li, Z., Zhang, M., Ma, S., Zhou, B., Sun, Y.: Automatic extraction for product feature words from comments on the web. In: Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology, pp. 112–123. AIRS ’09, Springer, Berlin, Heidelberg (2009)
6. Liu, B.: Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2nd edn. In: Indurkhy, N., Damerau, FJ. (eds.) pp. 627–666 (2010)
7. Neviarouskaya, A., Prendinger, H., Ishizuka, M.: SentiFul: A lexicon for sentiment analysis. *IEEE Trans. Affect. Comput.* **2**(1), 22–36 (2011)
8. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of LREC, vol. 2010 (2010)
9. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations Trends Inform. Retrieval* **2**(1–2), 1–135 (2008)
10. Zhang, H.P., Yu, H.K., Xiong, D.Y., Liu, Q.: Hhmm-based chinese lexical analyzer icclas. In: Proceedings of the second SIGHAN workshop on Chinese language processing - vol. 17, pp. 184–187. SIGHAN ’03, Association for Computational Linguistics, Stroudsburg, PA (2003)
11. Zhang, W., Liu, J., Guo, X.: Positive and Negative Words Dictionary for Students (First Edition). Beijing, China: Encyclopedia of China Publishing House, 75–77 (2004)
12. Zhao, J., Dong, L., Wu, J., Xu, K.: Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1528–1531. ACM, New York (2012)

Research on Indexing Page Collection Selection Method for Search Engine

Liyun Ru, Zhichao Li, Yingying Wu, and Shaoping Ma

Abstract With the rapid development of the Internet, the number of web pages has grown explosively. There are also many pages with similar content and low-quality pages. In terms of search engine, indexing such pages is no significant effect for retrieval results but increases the search engine's indexing and retrieval burden. This paper presents a page selection algorithm, building indexing page collection from massive web data for search engine. On the one hand, a web signature-based clustering algorithm is used to filter the similar pages to compress the size of the indexing page collection; on the other hand, it combines a variety of features of the page dimensions and user dimensions, to ensure the quality of the collection. Experiments show that the size of indexing page collection selected by the proposed algorithm is only one-third of the entire page collection, and can meet the vast majority of user click needs, with a strong practical.

L. Ru (✉) • S. Ma

State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

e-mail: lyru@vip.sohu.com; msp@tsinghua.edu.cn

Z. Li

Sogou Corporation, Beijing 100084, China

e-mail: lizhichao@sogou-inc.com

Y. Wu

Graduate School of Arts and Sciences, Harvard University, Cambridge, MA 02138, USA

e-mail: theresewu@gmail.com

1 Introduction

With the rapid development of the Internet, the number of pages has grown explosively. This presents a huge challenge for search engines, which provide web page search services. On the one hand, the indexing capacity of search engines is limited in keeping up with the proliferating number of pages, and on another hand, there are many pages which have similar or even the exact same content; therefore, indexing these pages is not very useful to search results. The huge numbers of low-quality pages that exist also lead to resource wastage in indexing and searching [1]. Hence, in the face of such massive numbers of pages, search engines need to select pages with valuable content, and not indiscriminately index all pages. The process of selecting pages to index from the entire collection of pages is called page selection, the aim of which is to satisfy the majority of search queries in terms of relevance and diversity through indexing of a smaller collection of documents [2, 3].

There are two main strategies for document selection: one, filtering pages so as to choose only one to index amongst repeated pages and two, assessing the quality of pages, selecting the higher-quality ones to index. Both strategies help to lower the number of pages indexed, with a different focus: the first focuses on retaining the amount of information in the page collection, while the latter focuses on ensuring the quality of the pages by eliminating meaningless pages in the indexing process.

There are currently two main research methodologies in filtering pages. One way is to calculate the extent of similarities between pages judging from their text content [4–7]. The advantage of this method is that it can fully take in consideration the similarity in semantics in the page content, but the disadvantage is that it is too complicated to come up with a formula to calculate the similarity of any two pages [8]. Another method is to use a hash function to cluster repeated pages into groups [9, 10], which involves quickly calculating a hash value for each page based on its content, clustering pages with the same hash values together, and choosing one page from each group to achieve the effect of filtering. However, this method does not tend to cluster pages with similar content into the same groups. This paper will combine the advantages of both methods and propose a clustering algorithm based on web signatures, which involves selecting important phrases or sentences from a page that can represent its content and using them as the page signature, which is then used to calculate a hash value for the page. This method is not only able to quickly cluster pages into groups but also able to cluster pages based on similar important phrases and keywords.

In evaluating page quality, the relatively more mature algorithm used currently is PageRank [11]. PageRank uses hyperlinks between pages to obtain an important parameter to evaluate page quality. Traditionally the consensus has been that the greater the number of hyperlinks toward a page, the higher its PageRank value and hence the higher its quality. In recent research, there are many improvements on this traditional PageRank algorithm, such as TrustRank [12–14]. Furthermore, search engines introduce measures such as anti-cheating [15] to filter out low-quality and spam pages during the document selection process. The current processes of

high-quality page selection are all based on the page collection itself, including hyperlinks between pages and content and structure within pages. This paper builds upon this to introduce features of user dimensions and then generalize this to site-level features to help to select indexing page collection. Experiments show that the selection of page collections with the incorporation of user dimension features is able to better satisfy search requirements of users.

This paper will introduce a page selection algorithm that combines page filtering and a method based on page quality, using genuine online datasets to select indexing page collections. The data collection process in the experiment and evaluation method will be discussed in Part 2. Part 3 presents the signature-based algorithm which is based on page content. Part 4 covers the page and user dimension features that are used to calculate page quality and presents the entire algorithm for selecting pages. Part 5 evaluates this algorithm and concludes and discusses future research.

2 Data Collection and Evaluation Method

In order to more accurately reflect the effect of this algorithm in the true Internet circumstance, this paper uses SogouT, which is published by Sogou laboratory, in experiments. SogouT includes a real Internet web data collection, hyperlink graph between pages, and a month's records of the search queries and click results of users in the search engine.

The web data collection in SogouT includes 130 million different real pages, the total size of which exceeds 5 TB. Records of search engine statistics include the click actions of users. User information has been removed from the data for privacy purposes, so every record only includes click time, user's search query, the URL that was clicked, and the rank that URL had in the search results. These records total more than 50 million user clicks.

This paper's algorithm will select indexing page collections from SogouT's web data collection and use compression and coverage ratios as criteria to evaluate the effect of the selection algorithm on indexing page collections.

The compression ratio is denoted as compress and is the ratio of the number of pages in the original web collection (S) to the number of pages in the selected collection (I):

$$\text{Compress} = \frac{|I|}{|S|}$$

The larger the compression ratio, the smaller the number of pages selected and hence the lower the indexing and searching burden on search engines. When there is no selection of pages, all pages in the original web collection will be indexed, so the compression ratio will equal 1.

The coverage ratio measures the extent to which the users' needs are satisfied by the indexing page collection. Records of 10 days of search engine user queries and click logs are selected as the evaluation collection of the coverage ratio, and the remaining records are used as training data. User-clicked URLs are extracted and used as evaluation collection $E = \{\langle \text{url}, \text{count} \rangle\}$. Every element in this collection includes a URL and count of that URL appeared in the click logs. The coverage ratio is

$$\text{Coverage} = \frac{\sum_{u \in E \& u.\text{url} \in P} u.\text{count}}{\sum_{u \in E} u.\text{count}}$$

where u denotes an element in E , the denominator is the total number of clicks of all URLs in the evaluation collection E , and the numerator only calculates the total number of clicks of URLs in the indexed page collection. Higher coverage ratios mean that the indexing page collection is better able to satisfy user query click needs and implies a more effective algorithm. When there is no selection of pages, all URLs clicked by users will be indexed, so the coverage ratio will equal 1.

The goal of page selection is to increase the compression ratio while preserving a high coverage ratio, reducing the number of pages indexed by the search engine. Another way to put it is to select user-clicked pages while preserving a high compression ratio, in order to satisfy user search requirements.

3 Signature-Based Algorithm Based on Document Content

In creating signatures for pages, if we create a hash value based on page content features and use it as a signature for the page, pages with same signatures would be recognized as having similar or same content and will be grouped together. It is important to ensure that the chosen features must reflect the content semantics of pages in order to group pages with repetitive content together, resulting in more effective filtering. We can separate page content into different domains, such as "title," "body title," "body content," and "linked text", to select page features.

We can use the fact that text lengths vary across different domains to choose terms or sentences as content features to calculate the signature. In choosing terms, the weight of every term t in the domain is calculated:

$$W(t) = \frac{\sum \text{length}(s) + k}{\sqrt{f(t)}}$$

where s denotes the sentences that contain the term t , $\text{length}(s)$ represents the length of the sentence s , $f(t)$ represents the number of times the term t appears in the domain, and k represents a constant related to the domain. When a term t appears more in longer sentences, it means that this term is more important. Dividing by the

frequency of the term t is to reduce the weight of stop words. To calculate the weight of each sentence,

$$W(s) = \frac{\text{length}(s)}{\sqrt{1 + \frac{\text{rank}(s)}{|S|}}} \times \frac{|\text{bigram}(s)|}{\text{bigramcount}(s)}$$

where $\text{length}(s)$ represents the length of the sentence s and $\text{rank}(s)$ represents the position of the sentence in the text of the domain. The earlier the sentence appears, the greater the weights, implying that the more important the sentence is. $|S|$ represents the number of sentences in the domain, $\text{bigram}(s)$ denotes the collection of binary groups in a sentence (a binary group is a string constructed by two neighboring words), and bigramcount represents the total number of binary groups in a sentence. The sentence and N terms with the largest weights are chosen as page features, which are used to calculate the signature of the page. The hash algorithm used in the calculation of signature is the traditional MD5 algorithm [16], which is able to largely avoid collision.

Such a signature-based algorithm that is based on page-specific terms and sentences will result in the same signature for pages with similar content, grouping pages that copy content from other pages with small changes with the original pages. Together with within-group selection, this will keep high-quality original page, achieving the effect of page filtering.

4 Selection Algorithm for Indexing Page Collection

Page selection should be based on quality of pages, and the features of page collection, hyperlink relationships, etc., are important in calculating page quality. User action features are also important [17, 18]. We introduce user features in search engine records to calculate page quality, to achieve better effects of page selection.

Since page selection should guarantee coverage ratios of user-clicked pages, we use the probability of user clicks for a particular page to determine its quality. This probability is calculated using a linear regression model. Page features include that of page dimensions and user dimension, and the linear regression model is obtained through a machine learning training. As for the user data not used in evaluating coverage ratios, we separated them equally into two groups to obtain two web collections of user-clicked pages: collection A is used to summarize user dimension features, while collection B consists of the positive examples that are used in the linear regression model.

We use a vector to describe a page $P = \langle x_1, x_2, \dots, x_n, y \rangle$, where x_i denote page features and y denotes whether or not the page is in collection B (i.e., whether the page has been clicked by users). Five million pages randomly chosen from SogouT's web collection and collection B are merged to form a training collection. The linear

regression model can be described by the vector $M = \langle c_1, c_2, \dots, c_n \rangle$, where c_i denotes the weights of the respective x_i . The probability of a page being clicked, P , is

$$\text{prob}(P) = \sum_{i=1}^n c_i x_i$$

The larger the $\text{prob}(P)$, the higher the probability of a page to be clicked by a user and the greater the need for this page to be indexed in search results.

4.1 Page Dimension Features

Page dimension features are calculated based on the page collection, the two key factors being features of the page itself and features of the site on which the page sits. Page features include PageRank, length of the page URL, number of parameters in the URL, and rank of page according to PageRank within a group after being clustered. In using the features of the site on which the page sits, on the one hand, pages on the same site may be similar to an important extent; on another hand, this can increase the generalizability of the model. Site features include SiteRank and number of pages in the site.

The calculation of SiteRank for a site is similar to that of PageRank for a page. When constructing a hyperlink graph, depict sites as nodes and hyperlinks between sites as the edges of the graph. The weight of the link between site A and site B is

$$W(A, B) = |\{\langle a, b \rangle | a \in A, b \in B, \text{link}(a, b)\}|$$

where a represents a page in site A , b represents a page in site B , and $\text{link}(a, b)$ indicates that there are hyperlinks in a that link to b .

In this hyperlink graph, the rank value for each site calculated using the traditional PageRank algorithm is the SiteRank of the site. The SiteRank value can also reflect to a certain extent the quality of the page that sits on the site.

4.2 User Dimension Features

However, pages that are clicked by users are not necessarily pages with high PageRank values. Figure 1 depicts PageRank distributions in the original web collection and the user-clicked web collection. We can see that there are pages with low PageRank yet high user clicks, and about 10% of user-clicked pages have a PageRank value of 3 and below. Therefore, it is important to introduce user dimension features in page selection.

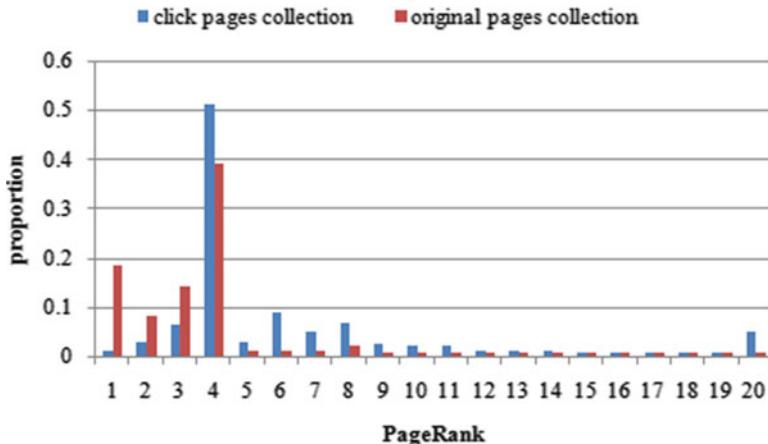


Fig. 1 PageRank distribution of original page collection and click page collection

First, we can obtain the count of clicks by users for every page of each day in collection A (which is taken from search engine records). The UserRank value for each page P can be calculated from this data:

$$\text{UserRank}(P) = \sum_{i=1}^n \frac{uv_i}{\sigma}$$

$$\sigma = \frac{1}{n} \sum_{i=1}^n (uv_i - \bar{uv})^2 \quad \bar{uv} = \frac{1}{n} \sum_{i=1}^n uv_i$$

where n denotes total number of days of the records and uv_i represents the number of user clicks on page P on the i th day. In order to decrease the weight of pages with repeated clicks in a short amount of time, the calculation takes into account click times and distributions of a page.

Similar to page dimension features, user dimension features can also take into consideration site features. The average UserRank of all pages in a site and the number of user-clicked pages within a site can be added to user dimension features.

As the data points from the search engine records are limited, we expanded user dimension features. We chose search queries with higher frequencies from our search engine records and used these queries in actual online searches, extracting the first ten search results to be incorporated into calculating user dimension features. This is because the first ten search results are extremely likely to be clicked by users. In our calculation, we appropriately decreased the importance of pages that were never clicked by users in our data.

Table 1 Page selection algorithm

-
- Input data: page collection $P = \{p_1, p_2, \dots, p_m\}$
1. Calculate signature for each page p_i in P , $p_i \cdot \text{sig} = \text{Sig}(p_i)$;
 2. Cluster p_i into groups based on their signatures to obtain group collections $C = \{c_1, c_2, \dots, c_k\}$; where $c_j = \{p_i | p_i \cdot \text{sig} = c_j \cdot \text{sig}\}$;
 3. Calculate page dimension features and user dimension features, adding them to the feature vector $p_i \cdot V$; rank pages in each c_j using PageRank, and add the ranks to the feature vector $p_i \cdot V$;
 4. Use the linear regression model to calculate the click probability of every page p_i in P ;
 5. Rank pages in each c_j according to click probability, and add the top-ranked page p_i to selected collection S ;
- Output data: selected collection S .
-

4.3 Page Selection Algorithm

In page selection, we first cluster pages into groups based on their page signatures, and then the page with the highest click probability within each group is to be selected into the indexing collection. Table 1 describes the page selection algorithm. The input data is the entire page collection, and output data is an indexing page collection selected from the input data.

In the process, the algorithm eliminates pages with the same or similar content, which will not negatively affect the effect and diversity of search results. The algorithm also selects pages with the highest click probability, which prevents pages that are important to users from being filtered out.

5 Algorithm Evaluation

5.1 Compression

Using the algorithm described in this paper, we randomly extracted different sizes of data from the SogouT web collection to undergo experiments. The compression ratios obtained are presented in Fig. 2, which shows that the larger the page collection, the higher the compression ratio in the page selection algorithm, in turn implying there is more saved space. We could obtain a compression ratio of 3.12 on the whole SogouT page collection. This is because a massive web collection consists of many similar pages and information tends to saturate. However, if we use the entire page content as a signature, we can only obtain a compression ratio of 1.09, which does not make much of a contribution to the page filtering process since the proportion of pages with exact same content is small. When we randomly selected 100 pairs of pages out of the grouped pages, 18 pairs were found to be grouped together erroneously under human inspection—the accuracy of clustering reaches 82%.

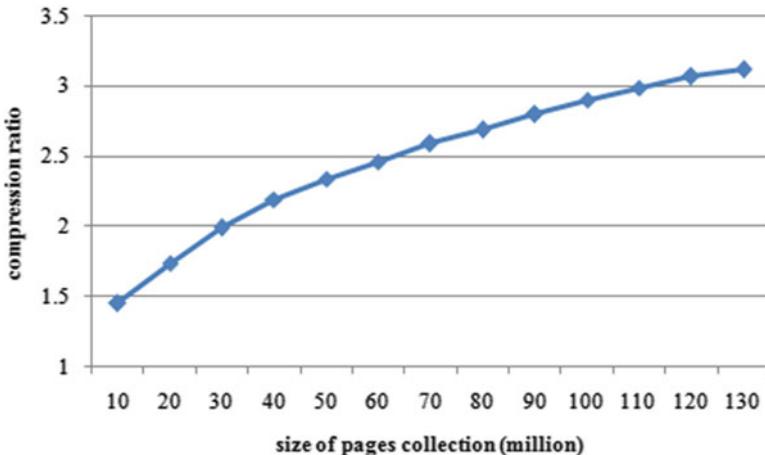


Fig. 2 Compression ratio of selection algorithm on different sizes of page collection

Table 2 Coverage ratios of different algorithms without cluster

Algorithm	Coverage ratio
No selection	1.000
Random selection	0.322
Selection based on PageRank	0.778
Selection based on PageRank + UserRank	0.802
Selection based on click probability	0.836

5.2 Coverage

We first discuss the coverage ratios of algorithms that do not cluster pages into groups but directly select pages based on page quality, under the same compression ratio of 3.12. Table 2 shows the coverage ratios of different algorithms when pages are not grouped.

We can see that PageRank is able to represent the extent of importance of a page during page selection, resulting in a coverage ratio of 0.778 that is more than double than that of random selection. After incorporating UserRank, which takes into account user dimension features, the coverage ratio reaches 0.802. After incorporating more page and user dimension features and using a linear regression model to calculate click probability (which better represent user tendencies in searching and clicking), the coverage ratio reaches 0.836.

Table 3 presents the coverage ratios of different algorithms that are used in selecting a page from every group after data is clustered into groups. We can see a significant increase in coverage ratios overall compared to the previous table where pages are not clustered into groups. This shows that even pages with high click probability have a large amount of repeated contents, and indexing these repeated contents is not only unable to provide users with more information but also unable

Table 3 Coverage ratios of different algorithms based on content signature cluster

Algorithm	Coverage ratio
Randomly selecting one page from each group	0.532
Selecting the page with largest PageRank from each group	0.861
Selecting the page with largest PR + UR from each group	0.891
Selecting the page with highest click probability	0.938

to increase the coverage ratios of user clicks. After pages with similar content have been clustered together, a large amount of repeated pages can be filtered out, and we can select pages with relatively lower click probability, which can increase the amount of information of the selected page collection, satisfying more needs of users searching and clicking. This will result in an even higher coverage ratio, showing that page filtering is extremely important in the page selection process.

Whether or not pages are clustered into groups, incorporating user dimension features has a significant positive impact on coverage ratios. The effect of click probability is better than when only using UserRank and PageRank. This is because UserRank focuses on page-level features, and if a page were not clicked in the training set, this page would not have been assigned a UserRank value and will not have any advantage in becoming selected. However, the algorithm to calculate click probability incorporates site-level features; therefore, even if a page has not been clicked on before, but its site has many other pages that had been clicked by users, indicating that the quality of this site is relatively high, and then this otherwise abandoned page now has a higher click probability and hence increases the coverage ratio. Hence, selecting the page with the highest click probability within each group (after clustering pages into groups) is able to achieve the highest coverage ratio of 0.938. This implies that the algorithm presented in this paper is extremely effective and is able to only select one-third of the original web collection but yet satisfies more than 90% of user needs.

6 Conclusion and Future Works

This paper presents a selection algorithm for search engines indexing page collection. This algorithm represents page content via selecting specific terms and sentences in different domains and calculates the signature of each page and clusters pages into groups. After clustering the pages, user dimension features, which are also generalized to site-level features, are used in a linear regression model to calculate the click probability, and the page with the highest click probability in a group is taken to construct the indexing page collection. This algorithm is not only able to quickly cluster and select pages but also achieve a higher compression ratio while still preserving the amount of information present in the indexing page collection. Our experiments with actual page collections can prove the effectiveness of this method. In particular, better compression ratios are achieved when dealing

with massive page collections. Future research can consider selecting more than one high-quality page from each clustered group of pages and improve the coverage ratios on the consideration that compression ratios will not be much affected.

Acknowledgements This work was supported by Natural Science Foundation (60903107, 61073071) and National High Technology Research and Development (863) Program (2011AA01A205) of China.

References

1. Agrawal, A., Husain, M., Tiwari, R.G., et al.: A novel technique for database selection and document selection. *Int. J. Comput. Appl.* **17**(8), 22–26 (2011)
2. Lin, H., Zhang, Y., Davis, J.: Best document selection based on approximate utility optimization. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1215–1216. ACM, New York (2011)
3. Welch, M.J., Cho, J., Olston, C.: Search result diversity for informational queries. In: Proceedings of the 20th International Conference on World Wide Web, pp. 237–246. ACM, New York (2011)
4. Broder, A.Z.: Identifying and filtering near-duplicate documents. In: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching, 2000, pp. 1–10
5. Broder, A.Z., Classman, S.C., Manasse, M.S.: Syntactic clustering of the Web. In: Proceedings of the 6th International Web Conference, 1997, pp. 11–20
6. Mathew, M., Shine, N.D., Lakshmi, T.R., et al.: A novel approach for near-duplicate detection of Web pages using TDW matrix. *Int. J. Comput. Appl.* **19**(7), 16–21 (2011)
7. Dubes, R.C., Jain, A.K.: Algorithms for Clustering Data. Prentice Hall, New York (1988)
8. Salloum, M., Tsotras, V.J., Srivastava, D., et al.: Selection and ordering of candidate documents for effective query answering in XML databases. In: Fifth International Workshop on Ranking in Databases, pp. 201–207. ACM, New York (2011)
9. Ding, Z., Wu, B., Xin, Y.: Research of large-scale URL filter based on bloom filter. *New Technol. Lib. Inform. Serv.*, **3**, 45–50 (2008)
10. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, pp. 388–397. ACM, New York (2002)
11. Page, L., Brin, S., Motwani, R., et al.: The PageRank Citation Ranking: Bringing Order to the Web. Stanford Digital Libraries, Stanford (1998)
12. Wei, C., Chen, F., Xu, D., et al.: A framework for web page quality evaluation. *J. Chin. Inform. Process. (AD of Publication, Beijing, China)* **25**(5), 3–8 (2011)
13. Wang, C., Liu, Y., Zhang, M., et al.: Topic-independent web high-quality page selection based on K-means clustering. *Lect. Notes Comput. Sci.* **3689**, 516–521 (2005)
14. Spirin, N., Han, J.: Survey on web spam detection: principles and algorithms. *ACM SIGKDD Explorations Newslett.*, **13**(2), 50–64 (2012)
15. Gyngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with TrustRank. In: Proceedings of the 30th International Conference on Very Large Databases (VLDB), pp. 576–587. ACM, New York (2004)
16. Rivest, R.: MIT Laboratory for Computer Science and RSA Data Security Inc. The MD5 message-digest algorithm[J], (1992)
17. Singh, D.: Improving web search ranking through user behavior information. *Int. J. Inform. Technol. Knowl. Manag (Serials Publications, New Delhi, India)* **4**(2), 635–638 (2011)
18. Chen, M., Yamada, S., Takama, Y.: Investigating user behavior in document similarity judgment for interactive clustering-based search engines. *J. Emerg. Technol. Web Intell. (Academy Publisher, Oulu, Finland)* **3**(1), 3–10 (2011)

The Chinese Bag-of-Opinions Method for Hot-Topic-Oriented Sentiment Analysis on Weibo

Jingang Wang, Dandan Song, Lejian Liao, Wei Zou,
Xiaoqing Yan, and Yi Su

Abstract With the rapid growth of Weibo, sentiment analysis on the hot topics which are spotlighted suddenly, spread rapidly, and influence widely during a short period becomes crucial. However, because of the urgent analysis requirement and diversity of the hot topics, the state-of-the-art supervised methods would fail due to the lack of annotated training data. To address this problem, we first propose a Chinese bag-of-opinions model based on dependency grammar representing Weibo sentences. Then, we calculate sentiment polarity score for every opinion and get a weighted summation sentiment evaluation for each sentence. A confidence value of a sentence's polarity score is also defined. With it, we can extract sentences with high confidences as annotated data which can guide further analysis. We applied our model with the summation evaluation and semi-supervised methods. Experiments conducted on the NLP&CC 2012 dataset for Chinese sentiment analysis validate the effectiveness of our method.

1 Introduction

Akin to a hybrid of Twitter and Facebook, Weibo is one kind of the most popular websites in China.¹ Users could post a message, which is called tweet in this paper, of up to 140 Chinese characters to share with their followers.

¹http://en.wikipedia.org/wiki/Sina_Weibo.

J. Wang • D. Song (✉) • L. Liao • W. Zou • X. Yan • Y. Su
Lab of High Volume Language Information Processing & Cloud Computing, School of Computer Science, Beijing Institute of Technology, Beijing, China
e-mail: bitwjg@gmail.com; sdd@bit.edu.cn; liaolj@bit.edu.cn; 1120101922@bit.edu.cn;
yanxiaoqing1987@126.com; be2n2me@gmail.com

Some topics, especially some emergencies, would be spotlighted suddenly, spread rapidly, and influence widely during a short period. These topics emerge as hot topics. For example, after a report saying that the salary of teachers in China is lowest in the world (中国教师收入全球几垫底) was published, this topic attracted people's attention and aroused wide discussion instantly. So the amount of tweets about this topic grew rapidly and made it a hot topic during the following days.

It is meaningful to mine the public opinions on the hot topics of Weibo. We define the problem as hot-topic-oriented sentiment analysis. Due to the length restriction, most tweets just contain one or two sentences; we focus our work on the sentence level. The state-of-the-art approaches for sentiment analysis basically follow Pang and Lee's work [10], who utilize supervised methods to classify the movie reviews. However, the urgent analysis requirement of hot topics can't tolerate the time-consuming manual annotation. And because of the various diversities of hot topics, transfer learning from cross domains is also a difficult task. In this paper, inspired by the previous bag-of-opinions work by [9, 11, 16], we propose a Chinese bag-of-opinions (CBoO) model based on dependency grammar representing Weibo sentences. Based on the CBoO model, we represent every sentence as several opinions and assign a sentimental polarity score for every opinion. Then, we calculate a weighted summation sentiment evaluation for each sentence. In addition, a confidence value of a sentence's polarity score is defined, which can be used to extract sentences with high confidence as annotated data for further analysis.

Our sentiment analysis experiments at NLP&CC 2012 consist of two parts: subjectivity classification and polarity classification. We applied our model with the summation evaluation and semi-supervised clustering methods. Experimental results validate the effectiveness of our method.

The rest of this paper is organized as follows. Section 2 summarizes related work. In Sect. 3, we briefly introduce our Chinese bag-of-opinion model, including the dependency grammar representing Weibo sentences, calculation rules of opinion's sentiment polarity score, and the confidence value definition. Section 4 depicts the experimental dataset, methodologies, results, and its analysis. At last, we conclude this paper in Sect. 5 with some future work.

2 Related Work

Inspired by the prior work [6, 10, 14] on movie or Amazon product reviews, sentiment analysis, a.k.a. opinion mining, has attracted much attentions both in NLP and data mining research communities.

Pang and Lee consider the sentiment classification of movie reviews as a kind of topic-based text classification [10]. They conduct three machine learning methods: naive Bayes, maximum entropy classification, and support vector machine (SVM). The results show that SVM outperforms the others. Correspondingly, for Twitter data, these methods are also applied to classify the sentiment of Twitter messages [4, 5]. To avoid laborious annotation, they both extract tweets with

emoticons and hashtags as training set. Obviously, training data plays a vital role in supervised methods, but they are not easy to acquire in reality. Besides, domain dependency is another shortage of supervised methods in sentiment classification.

Basically, supervised methods are the mainstream to address the task when abundant training data is available; otherwise, unsupervised or semi-supervised methods are more appropriate. An unsupervised method called SO-PMI [14] calculates phrase's semantic orientation score according to the mutual information values between the phrase and two predefined seed words, so a review could be classified based on the average semantic orientation of the sentimental phrases in it. Zagibalov et al. describe an unsupervised seed word selection method for sentiment classification of product reviews in Chinese [17]. They analyze the underlying features of Chinese text and utilize an iteration process to enhance the classification accuracy. However, the performance of unsupervised methods is still limited.

Hot-topic-oriented sentiment analysis in social network websites, like Twitter and Weibo, is a newly appearing problem. The hot topics are essential resource for some practical applications. Some researchers make use of political sentimental tweets to monitor political and predict election results [2,13]. C-Feel-it [8], a system which can detect opinion content in tweets, categorizes tweets pertaining to a topic as positive, negative, or objective and gives an aggregate sentiment score which represents a sentiment snapshot for a topic. Barbosa and Feng consider the effect of noisy and biased data in their work, in which they explore some characteristics of how tweets are written and meta-information of the words that compose the tweets [1].

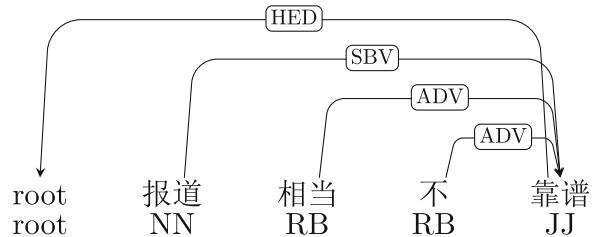
Target-dependent Twitter sentiment classification problem is addressed in [7]. Given a query as the target of the sentiment analysis, they incorporate target-dependent features and related tweets to reinforce classification accuracy. Wang et al. focus topic sentiment analysis task in Twitter through classifying the hashtags as positive or negative, which is called as hashtag-level sentiment classification [15]. However, hashtag-level sentiment can't completely cover the topic's sentiment.

3 Approach Overview

3.1 Dependency Grammar

Dependency grammar [12], deems all but one word, depends on other words in a sentence. The one word that does not depend on others is called the root. Based on dependency grammar, we could parse a sentence into a dependency tree. Figure 1 shows an example, in which 靠譜 is the root, and all the other three words depend on it. ADV denotes adverbial-core relationship, SBV denotes subject-verb, and HED denotes that root word depends on a dummy super node. The notations of these relationships could be found in [3].

Fig. 1 Dependency tree
of 报道相当不靠谱



3.2 Chinese Bag-of-Opinions Model

We define opinion as the minimum independent sentimental unit in a sentence, which is composed of a sentimental indicator, a set of modifiers, and dependency relationships between them. Negation words are also treated as modifiers in the model. Therefore, an opinion could be represented as a set of quadruples:

$$\text{Opinion} = (\text{Modifier}, \text{Indicator}, \text{Relationship}, \text{Offset}) \quad (1)$$

or

$$\text{Opinion} = (\text{Modifier}_1, \text{Modifier}_2, \text{Relationship}, \text{Offset}) \quad (2)$$

- *Sentimental Indicator* is the word which possesses sentimental polarity in an opinion and belongs to the sentiment lexicon. Adjectives, verbs, and nouns could become sentimental indicators. Here are some instances: 美丽 (*beautiful*)-adjective, 鄙视 (*disdain*)-verb, 混蛋 (*bastard*)-noun.
- *Modifier* is the word modifying the sentimental indicator or other modifiers, usually tagged as adverb. In the dependency tree, modifiers are the child nodes of sentimental indicator or other modifiers with specific relationship.
- *Relationship* represents the dependency relationship between the indicator and the modifier or between modifiers.
- *Offset* is a signed integer which denotes the offset of the modifier from the indicator, positive if the modifier lies in the indicator's left, otherwise negative. We take offset into account due to the situation that several modifiers modify a same indicator together, in which we need to know their order for calculation.

Consider the example in Fig. 1. There is only one opinion in the sentence, where “靠谱” (reliable) is a sentimental indicator, with three child nodes depending on it. Both “相当” (rather) and “不” (not) are modifiers modifying the indicator with ADV relationship. So the opinion is composed of two quadruples: (“相当”, “靠谱”, ADV, -2) and (“不”, “靠谱”, ADV, -1). The offset -1 denotes that the modifier “不” is in the left of the indicator and their dependency distance is 1.

3.3 Sentimental Polarity Score

3.3.1 Sentiment Polarity Strength of Indicators

According to the sentimental polarity, the indicators can be divided into three sets, {positive}, {negative}, and {neutral}. Each sentimental indicator could accordingly be assigned a sentiment polarity value. Similar to English, different Chinese word has different sentiment polarity strength. For example, both 赞 (amazing) and 不错 (good) are positive words, but their polarity strength differs with each other. However, an acknowledged fine-grained Chinese sentiment lexicon with polarity strength does not exist yet, so we treat all the words equally. The positive words are assigned with +1, negative words are assigned with -1, and other words are assigned with 0, as Eq. (3) shows.

$$score_{\text{indicator}} = \begin{cases} +1, & \text{if indicator} \in \{\text{positive}\} \\ -1, & \text{if indicator} \in \{\text{negative}\} \\ 0, & \text{if indicator} \in \{\text{neutral}\} \end{cases} \quad (3)$$

This disposal is really simple. The performance of sentiment classification would be improved significantly if more accurate values are given.

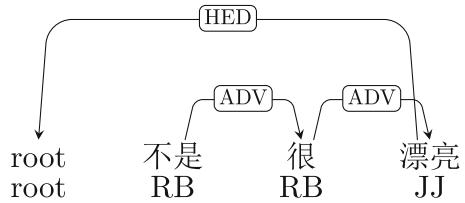
3.3.2 Sentiment Factor of Modifiers

We categorize modifiers into three types: {intensifier}, {reducer}, and {negation}. Intensifier would intensify the sentimental strength of indicator or other modifiers, while reducer has the opposite effect. We assign a value larger than 1 called intensify factor for every intensifier and a value less than 1 called reducing factor for every reducer.

$$factor_{\text{modifier}} = \begin{cases} > 1, & \text{if modifier} \in \{\text{intensifier}\} \\ < 1, & \text{if modifier} \in \{\text{reducer}\} \end{cases} \quad (4)$$

Negation word is one special type of modifiers. The order of the modifiers needs to be paid attention to when negation words appear, whose polarity could not be simply reversed. Both Liu [9] and Qu [11] have pointed out this linguistic phenomenon in English, which is also common in Chinese. For example, 不是很漂亮 (*not very beautiful*) would be assigned with a strong negative polarity if we just reverse the polarity of 很漂亮 (*very beautiful*), although 不是很漂亮 is actually a weaker positive phrase than 很漂亮 (*very beautiful*). The difference is recognized by the role of the negation played in the dependency tree. When its dependent word is an indicator, the polarity should be negatively changed while it inverts the modifier's factor if it depends on a modifier. Equation (5) shows how to calculate the factor score when negation words appear.

Fig. 2 Dependency tree of 不是很漂亮



$$factor_{\text{negation}} = \begin{cases} -1, & \text{if negation depends on an indicator} \\ 1/factor(\text{modifier}), & \text{if negation depends on a modifier} \end{cases} \quad (5)$$

Finally, multiplying indicator's polarity score by the modifier's factor would produce an aggregate polarity score, as Eq. (6) shows:

$$score_{\text{opinion}} = factor_{\text{modifier}} \times score_{\text{indicator}} \quad (6)$$

Take the above example; suppose score (漂亮) = +1, factor(很) = 1.5, 不是 is a negation modifier and their dependency tree is shown in Fig. 2. We calculate the whole polarity score from left to right. Because 不是 modifies intensifier 很, it inverts the intensifier's factor to 1/1.5, then we multiply the factor by the polarity score of 漂亮, at last obtaining the aggregate score (不是很漂亮) = to 2/3.

3.3.3 Sentiment Calculation of a Sentence

After all the opinions' polarity scores are calculated, we could evaluate the whole sentence's polarity score based on their positions in the dependency tree. It's an intuitive assumption that the opinions near the root in the dependency tree are more weighted than the opinions far away from the root. Taking the position of sentimental indicator as the opinion's position, we construct the calculation method in Eq. (7):

$$\begin{aligned} pos(op_i) &= dis(indicator_i, root) \\ score_{\text{sentence}} &= \sum_{i=1}^N \frac{\sum_{i=1}^N pos(op_i)}{pos(op_i)} * score(op_i) \end{aligned} \quad (7)$$

where $pos(op_i)$ denotes the position of the i -th opinion, which is calculated as the dependency distance between its indicator and root in the dependency tree $dis(indicator_i, root)$; N is the total number of opinions in the sentence; and $score(op_i)$ denotes the polarity score of the i -th opinion. Therefore, the sentiment polarity of a sentence can be classified as positive, negative or neutral, with respect to the signal of the calculated polarity score.

3.4 Confidence Value

To extract the annotated data for further analysis, a confidence value between 0 and 1 is defined for each sentence. Here we have two assumptions: (1) the sentence with single opinion is more confident than those with multiple opinions, and (2) if the sentence has multiple opinions, the greater the difference between a sentence's positive scores and its negative scores, the more confident the sentence score. The calculation method is listed in Eq. (8):

$$\text{confidence} = \begin{cases} \frac{|\text{score}_{\text{sentence}}|}{\left| \sum_{i=1}^{N^+} \text{score}(\text{op}_i) \right| + \left| \sum_{j=1}^{N^-} \text{score}(\text{op}_j) \right|}, & \text{if } \text{score}_{\text{sentence}} \neq 0 \\ 1, & \text{if } \text{score}_{\text{sentence}} = 0 \end{cases} \quad (8)$$

where N^+ and N^- represent the number of positive opinions and the number of negative opinions, respectively.

4 Experiments and Discussions

4.1 Preliminary

The experimental dataset is the evaluation set for Chinese sentiment analysis at NLP&CC 2012.² The dataset contains 20 hot topics from various domains, including over 1,500 sentences (10% annotated) in every topic.

We utilize HowNet Chinese sentiment lexicon³ as the sentiment lexicon in our experiments. Besides, we manually append some newly emerged sentimental words such as *给力(awesome)* and *脑残(foolish)* to the lexicon. LTP⁴ is a language processing platform containing the functions of Chinese word segment, POS tagger, dependency parsing, etc [3]. In our experiments, LTP is employed to process the raw dataset and generate the dependency trees for further process.

4.2 Methodologies

We adopt three methods to test our CBoO model. (1) The baseline is Sentimental Word Count (SWC)-based method, which classifies a sentence subjective if any

²http://tcci.ccf.org.cn/conference/2012/pages/page04_eva.html(In Chinese).

³<http://www.keenage.com>.

⁴<http://ir.hit.edu.cn/ltp/>.

sentimental word exists in the sentence, otherwise objective. SWC accomplishes polarity classification through comparing the numbers of positive and negative words, and the major side would dominate the sentence's polarity. (2) CBoO-based method, which classifies a sentence according to its aggregated sentimental polarity score, is an unsupervised method. (3) We also incorporate the CBoO model in a clustering method, which becomes a semi-supervised method named as CBoOC. We represent sentences as vectors including unigram words, punctuation and emoticons as content features, and the number of sentimental words as sentimental features. We extract a certain amount of sentences that belong to each category with the confidence value larger than a threshold (0.7 in the experiment) as three initial clusters (i.e., positive, negative and objective). For an unclassified sentence, we calculate its Euclid distance with the mean vector of every clusters, then classify it into the proper cluster. The calculating and clustering process iterates until the clusters do not change anymore.

4.3 Results

We use the same measurements with NLP&CC 2012 evaluation, including precision, recall, and F1-measure. Because there are 20 topics in the dataset, we calculate *micro-average* and *macro-average* for all the measurements. The calculation method is shown in Eq. (9), where N denotes the number of topics ($N = 20$ in our experiment).

$$\begin{aligned} \text{Micro_Precision} &= \frac{\sum_{j=1}^N \text{correct}(\text{polarity} = \text{POS}, \text{NEG})}{\sum_{j=1}^N \text{proposed}(\text{polarity} = \text{POS}, \text{NEG})} \\ \text{Micro_Recall} &= \frac{\sum_{j=1}^N \text{correct}(\text{polarity} = \text{POS}, \text{NEG})}{\sum_{j=1}^N \text{annotated}(\text{polarity} = \text{POS}, \text{NEG})} \\ \text{Macro_Precision} &= \frac{1}{N} \sum_{i=1}^N \text{Precision}_i, \quad \text{Macro_Recall} = \frac{1}{N} \sum_{i=1}^N \text{Recall}_i \quad (9) \end{aligned}$$

The final results of subjectivity classification and polarity classification are shown in Tables 1 and 2, respectively.

4.4 Discussions

Subjectivity classification results are shown in Table 1. CBoO-based method outperforms SWC in all the measurement notably, which proves the effectiveness of our CBoO model. Especially in precision, CBoO improves the performance by

Table 1 Subjectivity classification result

Method	Micro-average			Macro-average		
	Precision	Recall	F1	Precision	Recall	F1
SWC	52.0%	48.6%	50.2%	51.3%	47.2%	49.2%
CBoO	74.0%	55.7%	63.6%	73.3%	54.1%	61.8%
CBoOC	73.4%	52.8%	61.4%	72.4%	51.4%	59.9%

Table 2 Polarity classification result

Method	Micro-average			Macro-average		
	Precision	Recall	F1	Precision	Recall	F1
SWC	46.1%	48.6%	47.3%	45.5%	47.0%	46.2%
CBoO	72.4%	40.3%	51.8%	70.8%	38.7%	49.6%
CBoOC	71.8%	37.9%	49.6%	70.3%	36.5%	47.7%

over 20 points. Nevertheless, the recall is improved by less than 10 points. The possible reason could be that we restrict the part of speech(POS) of sentimental indicators as verb, noun, and adjective, which result in missing some sentimental words with other POSs. Because many Chinese words possess more than one POS, incorrect POS tagging would cause a bad influence. CBoOC isn't comparable with CBoO, it may result from the features employed in the method. We mainly utilize common features like unigram words and emoticons, which couldn't represent some implicit characteristics of sentences. Moreover, because of the short length of Weibo sentences, the vectors are quite sparse. How to select and compose suitable features to represent Weibo sentences is still an acknowledged hard problem.

For polarity classification, according to Table 2, although the F1 measurements of both CBoO and CBoOC are still better than SWC, the recalls decline. Except the reasons mentioned above, the misclassification of special sentences could be the cause. Composite sentences with adversative conjunctions and special structures like subjunctive mood can't be handled precisely. These syntactic features deserve more attention to improve the classification performance.

5 Conclusions and Future Work

In this paper, we propose a Chinese bag-of-opinions model and calculation methods to address hot-topic-oriented sentence-level sentiment analysis on Weibo. According to the experiment results, the CBoO-based method overtly improves the subjectivity classification and polarity classification. We also extract annotated data automatically for semi-supervised methods via calculating a confidence value for each sentence and get an effective result.

In the future, we would explore more fine-grained bag-of-opinions model incorporating syntactic structures. Furthermore, the calculation rules of confidence

value are rather roughly. We just consider characteristics of subjective sentences, while all the objective sentences assigned a consistent confidence value, which is not reasonable. Taking objective sentences' characteristics into consideration would improve the reliability of confidence value.

Acknowledgements This work is funded by the National Program on Key Basic Research Project(973 Program, Grant No.2013CB329605), Natural Science Foundation of China (NSFC, Grant Nos. 60873237 and 61003168), Natural Science Foundation of Beijing (Grant No.4092037), Outstanding Young Teacher Foundation, and Basic Research Foundation of Beijing Institute of Technology and partially supported by Beijing Key Discipline Program.

References

1. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 36–44. COLING '10, ACL, Stroudsburg, PA (2010)
2. Bermingham, A., Smeaton, A.F.: On using twitter to monitor political sentiment and predict election results. In: Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP), IJCNLP 2011, pp. 2–10, 2011
3. Che, W., Li, Z., Liu, T.: Ltp: a chinese language technology platform. In: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, pp. 13–16. COLING '10, ACL, Stroudsburg, PA (2010)
4. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 241–249. COLING '10, ACL, Stroudsburg, PA (2010)
5. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing **150**(12), 1–6 (2009)
6. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168–177. KDD '04, ACM, New York, NY (2004)
7. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - vol. 1, pp. 151–160. ACL, Stroudsburg, PA (2011)
8. Joshi, A., Balamurali, A.R., Bhattacharyya, P., Mohanty, R.: C-feel-it: a sentiment analyzer for micro-blogs. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations, pp. 127–132. HLT '11, ACL, Stroudsburg, PA (2011)
9. Liu, J., Seneff, S.: Review sentiment scoring via a parse-and-paraphrase paradigm. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: vol. 1 - Volume 1, pp. 161–169. EMNLP '09, ACL, Stroudsburg, PA (2009)
10. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing - vol. 10, pp. 79–86. EMNLP '02, ACL, Stroudsburg, PA (2002)
11. Qu, L., Ifrim, G., Weikum, G.: The bag-of-opinions method for review rating prediction from sparse text patterns. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 913–921. ACL, Stroudsburg, PA (2010)
12. Tesnière, L.: *Eléments de Syntaxe Structurale*. Klincksieck, Paris, FRA (1959)
13. Tumasjan, A., Sprenger, T., Sandner, P., Welpe, I.: Predicting elections with twitter: What 140 characters reveal about political sentiment. In: Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, pp. 178–185 (2010)

14. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. ACL '02, ACL, Stroudsburg, PA (2002)
15. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1031–1040. CIKM '11, ACM, New York, NY (2011)
16. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 625–631. CIKM '05, ACM, New York (2005)
17. Zagibalov, T., Carroll, J.: Automatic seed word selection for unsupervised sentiment classification of chinese text. In: Proceedings of the 22nd International Conference on Computational Linguistics - vol. 1, pp. 1073–1080. COLING '08, ACL, Stroudsburg, PA (2008)

A Semantic-Driven Music Recommendation Model for Digital Photo Albums

Jiansong Chao, Haofen Wang, Wenlei Zhou, Weinan Zhang, and Yong Yu

Abstract Digital photo album softwares like iPhoto¹ have enjoyed great popularity for years. These years, online photo album services (e.g., Flickr² and Picasa³) have been becoming more and more popular with the development of social Web. In this paper, we present a semantic-driven model to recommend music for photo albums automatically. In particular, we exploit semantic data to represent both images and music. Furthermore, we leverage mining techniques to capture semantic relatedness between these different types of multimedia data, which is the essential step for recommendation. In the experiment, our method achieved a performance of about 68% satisfaction measured by participants' feedback.

1 Introduction

With the development of Web 2.0, digital photo album services have been becoming an indispensable part of social network sites. Representative examples such as Flickr and Facebook⁴. Photo albums are more and more popular among people. In September 2010, it was reported that Flickr was hosting more than 5 billion images.⁵

¹<http://www.apple.com/ilife/iphoto/>.

²<http://www.flickr.com/>.

³<https://picasaweb.google.com/>.

⁴<http://www.facebook.com/>.

⁵<http://en.wikipedia.org/wiki/Flickr>.

J. Chao (✉) • H. Wang • W. Zhou • W. Zhang • Y. Yu

Department of Computer Science and Engineering, APEX Data & Knowledge Management Lab,
Shanghai Jiao Tong University, Shanghai, P.R.China

e-mail: jiansong.chao@apex.sjtu.edu.cn; whfcarter@apex.sjtu.edu.cn;

wenlei.zhouwl@apex.sjtu.edu.cn; wnzhang@apex.sjtu.edu.cn; yyu@apex.sjtu.edu.cn

A lot of social network services, such as Facebook, even provide a larger-scale photo album service. By July 2011, Facebook had more than 750 million active users⁶ and 100 billion photos.⁷ Moreover, mobile devices develop rapidly in recent years. More and more people use their mobile phones or iPads to take photos and create albums. It makes the scale of digital photos increase more and more rapidly. In a word, digital photo album services are more and more popular for personal computers, Web services, and mobile devices. It is a meaningful work to provide more functions and enhance user experience for digital photo album services.

Besides photo publishing and sharing, some softwares, like iPhoto, even allow users to assign background music for some specified album. When browsing the photo album, it might be a fantastic experience if the background music matches the photos. For example, it will be a nice experience if there are some pieces of romantic background music for the wedding photo album and some rock ones for a boxing match photo album. However, the manual assignment limits the wide usage of such an attractive feature: (1) it will make a user exhausted if he or she has lots of albums and (2) it is hard for a user to select the suitable music if he has little related knowledge. Therefore, automatic background music recommendation for photo albums can greatly improve the user experience.

In this paper, we present a semantic-driven model which is the first effort trying to recommend suitable music for the given photo album automatically. From the technical perspective, the main challenge of automatic recommendation lies in calculating the relatedness between music and images indicating whether they share the common artistic conception or express the similar emotion. In order to solve the above challenge, we have to find a way to represent the very different data (i.e., image and music) in a unified semantic manner. With the advance of social Web, more and more multimedia data is annotated with tags. On the other hand, relatedness computing in the textual space has been well studied for years. Therefore, we take advantage of image annotations in form of tags to represent both images and music. More precisely, we make use of Flickr as the high-quality source to prepare for a large image database annotated with tags. AllMusic⁸ is used to associate mood tags to music. Further, we leverage WordNet⁹ [8, 18] ontology to enrich these tags and disambiguate them into synsets so that we can easily connect images with their suitable music according to their emotional semantic relatedness. In order to recommend music for the input images, we exploit the visual similarity between images so that the input images can be represented by several most similar images in our image database. The technical details can be found in the following sections. The snapshot of our demonstration is shown in Fig. 1. Photos of the album are arranged in a slide view at the bottom of the screen, and the selected photo is shown at the center. Users can browse the album while listening to the recommended

⁶<http://en.wikipedia.org/wiki/Facebook>.

⁷<http://www.photoweeklyonline.com/>.

⁸<http://allmusic.com/>.

⁹<http://wordnet.princeton.edu/>.

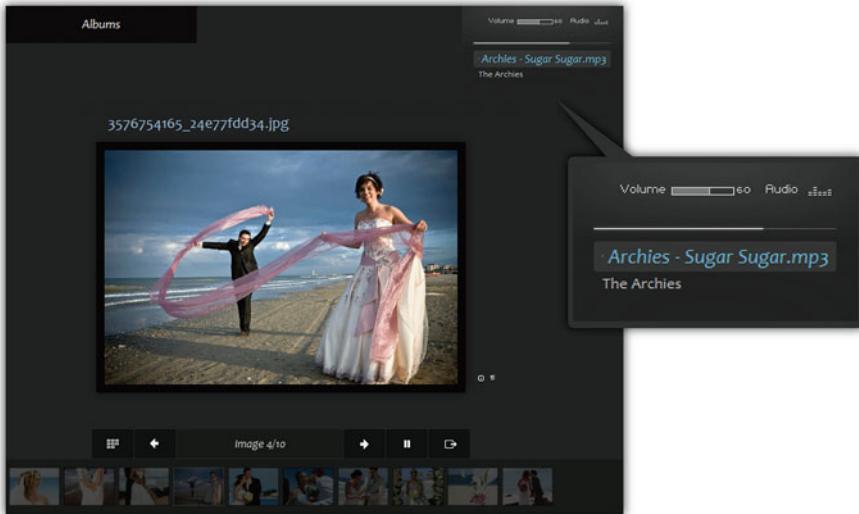


Fig. 1 The snapshot of our demonstration user interface

music in which details are presented at the top right of the screen. In our example, given a photo album of wedding dress, the playing background music recommended is *Sugar Sugar* from The *The Archies*.

The contributions of this paper are threefold. First, we present a cross-media semantic relevance model and apply the model to our system of music recommendation for photo albums. Second, we study the relations between photos and music in respect of how they will affect our feelings. Last, we provide a novel and interesting way to search music.

The rest of this paper is organized as follows. In Sect. 2, related work is presented. Sections 3 and 4 describe our method and evaluation, and in Sect. 5, we conclude our work and discuss the future work.

2 Related Work

There are two kinds of work related to our work. They are semantic relatedness measuring and image annotating. In the following subsections, we will discuss the related work in these two areas respectively.

2.1 Semantic Relatedness Measuring

Much research work has been done in the area of text semantic relatedness. There are currently three major lines of work. One line focuses on the use of ontology.

References [12, 13, 15, 21, 27] investigated path-based approach on ontology. In [22], information content-based approach for WordNet was investigated to calculate the relatedness between concepts. Gloss-based approach was investigated in [1] and [11]. Patwardhan and Pedersen [20] talked about vector based method for WordNet. The second line takes advantage of knowledge base like Wikipedia.¹⁰ Gabrilovich and Markovitch [9] represented each concept as a vector whose dimension is relevant to each Wikipedia article. The relatedness of two concepts is evaluated by the cosine distance of such two vectors. Path-based and information content-based methods are also used for Wikipedia [24]. The third line is statistical method, where much work is based on the Web. Bollegala et al. [3] used search engine to get statistic data and also took advantages of patterns. “Bag of Words” representation further considered in [23] based on [3]. Moreover, Gracia [10] proposed word relatedness measure based on normalized Google distance [5]. Our task is based on measuring semantic relatedness between photos and music, which has little previous work and is different from the existing semantic relatedness problems. First, our goal is to calculate the semantic relatedness between images and music, which are two heterogeneous spaces. But traditional methods discussed above are for text. Second, we try to represent images as nouns and represent music as adjectives so we can describe images with music. The existing methods put most focus on relatedness between noun concepts and are not suitable for relatedness between nouns and adjectives. Last, our concepts are domain specific, so we choose Flickr as a data source for statistics to better describe the relatedness between photos and moods.

2.2 *Image Annotation*

Another related research area is image annotation. Word co-occurrence model for image annotation was investigated in [19]. Li and Wang [16, 26] discussed the statistical models. Image annotation was regarded as a machine translating process in [7]. Some other researchers modeled the joint probability of image regions and annotations. Barnard et al. [2] investigated image annotation under probabilistic framework and put forward a number of models for the joint distribution of image blobs and words. In [4], image annotation was posed as classification problems where each class was defined by images sharing a common semantic label. Coherent language model was investigated in [14]. For music recommendation of photo albums, one possible solution is to leverage some image annotation techniques to find the relations between images and image tags we choose. There are some particularities of the image annotations in our work, since what we want to catch from the target image is its feeling, which makes a bridge to music. Thus, for the labeled images, we need to choose some images with typical features that bring

¹⁰<http://en.wikipedia.org/>.

people different kinds of feelings, and for input images, we assign them to several categories to get the semantic labels. Because we care more about the feelings of images, so we choose low-level image features like color and texture for image annotation.

3 Methodology

First, we formulize the problem of music recommendation for photo albums. Then we introduce our relevance model in detail.

3.1 Problem Formulation

We define $P = (p_1, p_2, \dots, p_n)$ as the input photo album where each $p_i \in P$ is one photo belongs to the album and n is the number of photos in it. Define $\mathcal{S} = (S_1, S_2, \dots, S_m)$ as the image semantic label space where each $S_i \in \mathcal{S}$ is a synset in WordNet and m is the number of synsets. Each S_i is represented by $S_i = (s_i^1, s_i^2, \dots, s_i^v)$ where s_i^j is a semantic tag of synset S_i and v is the number of semantic tags in it. Let $\mathcal{M} = (M_1, M_2, \dots, M_k)$ be the music space where $M_i \in \mathcal{M}$ is one track for recommendation and k is the number of candidate tracks. Each $M_i \in \mathcal{M}$ is represented by $M_i = (F_i^1, F_i^2, \dots, F_i^w)$ where F_i^j is one mood tag vector for music M_i and w is the number of mood tag vectors for it. Each F_i^j is represented by $F_i^j = (f_{ij}^1, f_{ij}^2, \dots, f_{ij}^u)$ where f_{ij}^k is a music mood tag and u is the number of music mood tags in vector F_i^j .

3.2 Relevance Model

In this section, we give a detailed description of the relevance model. Here the task is to evaluate the semantic relatedness between photo album P and each track $M_i \in \mathcal{M}$. For a target photo album, the most relevant tracks will be recommended. We define $Rel(\alpha, \beta)$ as the relatedness between α and β , where α and β can be spaces, vectors, or tags. The semantic relatedness of a photo album P and a track $M \in \mathcal{M}$ can be computed using

$$Rel(P, M) = Rel(P, \mathcal{S})^T \cdot Rel(\mathcal{S}, M), \quad (1)$$

where vector $Rel(P, \mathcal{S})$ represents the relatedness between album P and each image semantic label in \mathcal{S} , vector $Rel(\mathcal{S}, M)$ is the relatedness between each image semantic label in \mathcal{S} and music M .

$Rel(P, \mathcal{S})$ is computed as

$$\begin{aligned} Rel(P, \mathcal{S}) &= \sum_{i=1}^n Rel(p_i, \mathcal{S}) \\ &= \left(\frac{1}{n} \sum_{i=1}^n Rel(p_i, S_1), \frac{1}{n} \sum_{i=1}^n Rel(p_i, S_2), \dots, \frac{1}{n} \sum_{i=1}^n Rel(p_i, S_m) \right), \quad (2) \end{aligned}$$

where $Rel(p_i, S_j)$ is the relatedness between image p_i and synset S_j . We collect a certain number of images for each synset which is similar to ImageNet¹¹ [6]. Define Q as the image set representing S_j . $Rel(p_i, S_j)$ is computed as the sum of similarity values of p_i and all the images $q \in Q$.

$$Rel(p_i, S_j) = \sum_{q \in Q} (Sim(p_i, q)), \quad (3)$$

Similarly, $Rel(\mathcal{S}, M)$ is computed as

$$\begin{aligned} Rel(\mathcal{S}, M) &= (Rel(S_1, M), Rel(S_2, M), \dots, Rel(S_m, M))^T \\ &= \left(\frac{1}{w} \sum_{i=1}^w Rel(S_1, F_i), \frac{1}{w} \sum_{i=1}^w Rel(S_2, F_i), \dots, \frac{1}{w} \sum_{i=1}^w Rel(S_m, F_i) \right)^T, \quad (4) \end{aligned}$$

where $Rel(S_i, F_j)$ represents the relatedness between synset S_i and music mood tag vector F_j which can be computed using

$$Rel(S_i, F_j) = \frac{1}{uv} \sum_{x=1}^v \sum_{y=1}^u (Rel(s_i^x, f_j^y)), \quad (5)$$

where $Rel(s_i^x, f_j^y)$ estimates the image semantic tag s_i^x and music mood tag f_j^y . It is a well-learned problem in semantic relatedness area. In our model, $Rel(s_i^x, f_j^y)$ is calculated by the statistic method with data on Flickr as

$$Rel(s_i^x, f_j^y) = \frac{n(s_i^x, f_j^y)}{Weight(n(f_j^y))}, \quad (6)$$

where $n(s_i^x, f_j^y)$ is the number of occurrences of image semantic tag s_i^x and music mood tag f_j^y in the same photo description on Flickr. $n(f_j^y)$ is the number of occurrences of music mood tag f_j^y on Flickr. Because we want to choose the most related music mood tags for image semantic tags, we must consider the frequency of

¹¹<http://www.image-net.org/>.

mood tags and reduce the weight of these with high frequency with function weight. We normalize $Rel(s_i^x, f_j^y)$ to make sure too big or too small tag relatedness values won't affect the result.

4 Experiments and Results

4.1 Data Set

For music, we crawl 4,547 tracks from AllMusic Web site with their mood tags and metadatas like artist, album, and genre. There are totally 130 different mood tags for these tracks, each track is labeled with one or several mood tags. Then we use WordNet to expand each mood tag to a bunch of tags as a more robust representation of tracks. For the 4,547 tracks we crawled, each one is labeled by 2.04 AllMusic mood tags on average. After WordNet expansion, the average number of mood tags for one track is 15.97 which means the expansion step greatly improves the robustness of mood tag representations for tracks. The crawled data is not complete; we use MusicBrainz [25] to complement the metadata for each track. For the data of image semantic label, we choose 25 synsets which we think will lead people to different kinds of feelings. We select 20 typical images for each synset as its visual representation. For statistics of Flickr, we use Flickr API¹² to get the global statistic data for all photos on Flickr. For our experiments, we choose 30 photo albums from Flickr that vary widely from holidays to scenery, from ceremony to art. Each album contains 10 photos.

4.2 Experimental Setup

We name our music recommendation system TuneSensor. For an input photo album, TuneSensor analyzes the album images, searches for the most related music in its music database, and finally recommends it for the input album as output. Here we introduce TuneSensor via offline and online modules as shown in Fig. 2.

4.2.1 Construct Cross-Media Semantic Relatedness Graph

The offline module constructs a cross-media semantic relatedness graph, which contains both image and music mood tag synsets as vertices, and the semantic relatedness between these synsets as edges.

¹²<http://www.flickr.com/services/api/>.

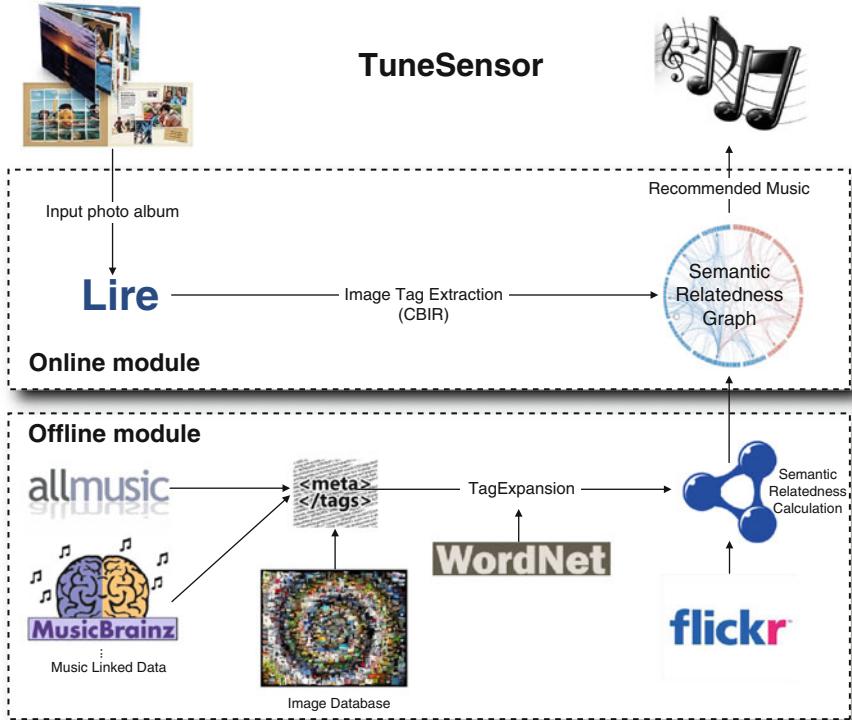


Fig. 2 The architecture of our music recommendation system for photo albums

For music tag synset construction, we crawl music metadatas from AllMusic Web site, including name, mood tags, artist, genre, and album of each track, and then we use service provided by MusicBrainz to complement the information for each track. The reason we choose AllMusic as one of our data source is that it has more than 100 types of mood tags for music. The mood tags are essential to our work. MusicBrainz is one of the semantic data sources about music; we can choose more semantic data sources like DBTune¹³ and last.fm¹⁴ to expand the information of music. We download music that we have metadatas from music download services like Google Music¹⁵. In this way, our music database and corresponding metadata database are prepared. We use WordNet to expand each music mood tag to a music tag vector F , as discussed in Eq. (4). Because mood tags are adjectives, we use WordNet synonym relations to expand each tag. The benefit of using WordNet is that

¹³<http://dbtune.org/>.

¹⁴<http://www.last.fm/>.

¹⁵<http://www.google.cn/music/>.

the errors caused by ambiguous words and users' preference of words are avoided effectively. In this way, each track is represented by several mood tag vectors.

For image tag synset construction, we choose WordNet synsets with typical subjective feelings and select certain number of images for these synsets which is similar to ImageNet. The difference is that we choose synsets based on our specific application requirements. Each synset can be converted as a semantic tag vector S discussed in Eq. (4). We choose WordNet synsets to organize our database of visual content because it is comprehensive and we can scale up the database conveniently according to the structure of WordNet. Furthermore, we can take advantage of the ontology to build a better relevance model. Similar with music synset expansion, each image tag synset can be converted into a semantic tag vector.

With the vector representation of image and music synsets, we can get the relatedness between them by computing the relatedness between two vectors. We use Eqs. (5) and (6) to complete the task. The number of images and corresponding tags on Flickr is quite large so the statistics of Flickr is reliable. The mood tags we use have an average of 89193.3 occurrences on Flickr. On the other hand, users naturally use both objective tags to describe the content of photos and subjective tags to describe the feelings of photos. So the relatedness of image content tags and music mood tags can be achieved by taking advantage of the statistics of Flickr. In our work, we use Gauss normalization for normalize function in Eq. (5) and square root as weight function in Eq. (6) which performs well in practice. In this way, our cross-media semantic relatedness graph is constructed.

4.2.2 Music Recommendation Using Relatedness Graph

For a photo album as input, we use LIRE (Lucene Image Retrieval) [17] to compute the visual similarity between input images in the album and the indexed synset images. In our work, we use default image features of LIRE to build the index of images. In this way, we get the relatedness between user album and each semantic tag vector S . Using the constructed semantic relatedness graph, each track is assigned with a relatedness score with the input album. Finally, the track with the highest match score is recommended. At the same time, we show the information of the recommended track so that users can find more similar tracks such as the ones sung by the same singer or in the same album. So the system also presents a novel and interesting way for searching music based on photos.

4.3 *Compared Algorithms*

Since little existing work investigates the problem of music recommendation for photo albums to our best knowledge, we compare our algorithm with two baselines as below:

Table 1 Satisfaction of three recommendation methods

	Lower Bound	TuneSensor	Manually Select
Satisfaction	30.67%	68.67%	77.33%

- *Lower Bound (LS)*. For this method, tracks are randomly selected for each target photo album. This is the lower-bound performance of our problem. We refer it as *Lower Bound* method.
- *Manually Selection (MS)*. For this method, tracks are manually selected for each photo album based on subjects’ preferences. We refer it as *Manually Selection* method.

Because the problem of music recommendation is relatively subjective, so these two baselines are very important to show the performance of our method. For the same reason, different people may prefer different genres of music, so we recommend several tracks with different genres to a photo album; each track is the most related one in its genre class. And user can choose his or her favorite genre from the recommendation tracks for each photo album.

4.4 Evaluation Measure

We invited 10 participants to do the test. For each recommended album–music pair, all the 10 participants will their labels as below:

- *Relevant*. The recommended music is considered relevant or suitable to be a background music to the target album, labeled with score 1.
- *Irrelevant*. Otherwise, labeled with 0.

For each algorithm, the satisfaction r is computed as the proportion of relevant album–music pairs in the whole recommended pairs:

$$r = \frac{\sum_i^t \tau_i}{t} \quad (7)$$

where τ_i is the average labeling score for the i th test case and t is the number of test cases.

4.5 Results and Analysis

Table 1 shows the satisfaction of the *Lower Bound* method, our method TuneSensor, and *Manually Selection* method. We can see that our method achieves a performance of 68% satisfaction which is much better than the random recommendation and

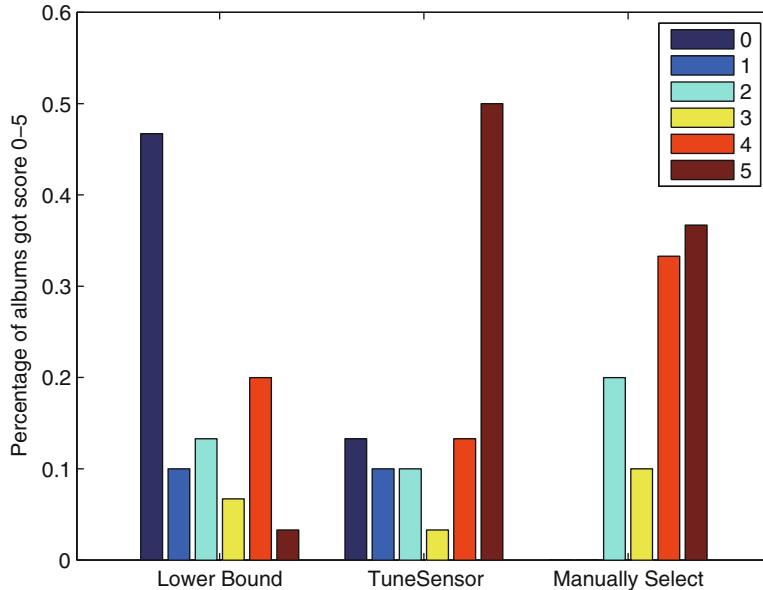


Fig. 3 Statistics of scores for albums. 0–5 means the percentage of people who are satisfied with the music recommended

is close to the performance of manual method. We can come to conclusion that our method indeed improves the performance of music recommendation for photo albums. For testing runtime performance, we use 30 photo albums, each contains 10 photos. The test was done on a desktop computer with an Intel Core 2 Quad CPU with two 2.66 GHz cores and 8 GB RAM running Ubuntu 10.10. The average time using TuneSensor for recommending music for one photo album is 0.8 s which is very efficient and much faster than the manual way.

We have 10 participants to evaluate our results. Each album will get 1 score if one participant think the music recommended is suitable for it. So each album will get a score between 0 and 5. We gather statistics of the percentage of albums with score 0 to 5 which is shown in Fig. 3. We can see that compared to the random recommendation, we recommend much more music in which most users are satisfied. Compared to the manual method, we still have some albums that nobody is satisfied, but we recommend more music in which all users are satisfied. This means our method is able to recommend really great music for albums. Table 2 shows part of the results using our method for music recommendation. In this table, images in the first column are the photo albums asking for background music. The related music mood tags found using our model are shown in the second column. The last column is the music we recommend and some information about it. More recommendation cases can be experienced in our online demo <http://tunesensor.apexlab.org/>. From the cases, we can see that our method can indeed find suitable music for photo albums.

Table 2 Results of our music recommendation for photo albums

Photo Album	Mood Tags Matched	Music Recommended	
	romantic, peaceful, dramatic, etc.		“Because You Loved Me” by Celine Dion
	sensual, fun, sexy, etc.		“Great Ball Of Fire” by Jerry Lee Lewis
	calm, elegant, mellow, etc.		“Gentle On My Mind” by Lisa Ono

5 Conclusion and Future Work

In this paper, we proposed a novel method to recommend music for photo albums. We presented a cross-media semantic relatedness model and introduced the architecture of our system. We compared the performance of our method with the random method and the manual method, which showed our method was able to recommend great music for albums. For the future work, we plan to scale up the number of synsets and the number of images for each synset. In this way, we can describe more visual content and feelings exactly. At the same time, we must build a more robust model for image annotation part of our model. We can also provide TuneSensor as applications for photo album services like Flickr and Facebook album in the future. In addition, music search engine queried by images can be developed based on our model.

References

1. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using WordNet. *Lect. Note Comput. Sci.* **2276**, 136–145 (2001)
2. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. *J. Mach. Learn. Res.* **3**, 1107–1135 (2003)
3. Bollegala, D., Matsuo, Y., Ishizuka, M.: Measuring semantic similarity between words using web search engines. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) *WWW*. pp. 757–766. ACM, New York (2007)
4. Carneiro, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(3), 394–410 (2007)

5. Cilibraši, R., Vitanyi, P.M.B.: The google similarity distance. *IEEE Trans. Knowl. Data Eng.* (3), 370–383 (2007)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei, L.F.: Imagenet: A large-scale hierarchical image database. In: *CVPR*, pp. 248–255 (2009)
7. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: *ECCV*, p. IV: 97 ff. (2002)
8. Fellbaum, C.: *WordNet: An Electronic Lexical Database*. MIT Press (1998)
9. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Veloso, M.M. (ed.) *IJCAI*, pp. 1606–1611 (2007)
10. Gracia, J., Mena, E.: Web-based measure of semantic relatedness. In: *The 9th International Conference on Web Information Systems Engineering*, pp. 136–150. Springer, Berlin, Heidelberg (2008)
11. Gurevych, I.: Using the structure of a conceptual network in computing semantic relatedness. In: Dale, R., Wong, K.F., Su, J., Kwong, O.Y. (eds.) *IJCNLP. Lecture Notes in Computer Science*, vol. 3651, pp. 767–778. Springer, New York (2005)
12. Hirst, G., St-Onge, D.: Lexical chains as representations of context for detection and correction of malapropisms. In: Fellbaum, C. (ed.) *WordNet: An Electronic Lexical Database*, pp. 305–332. MIT Press, Cambridge, MA (1998)
13. Jarmasz, M., Szpakowicz, S.: Roget's thesaurus and semantic similarity. In: Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R. (eds.) *RANLP. Current Issues in Linguistic Theory (CILT)*, vol. 260, pp. 111–120. John Benjamins, Amsterdam/Philadelphia (2003)
14. Jin, R., Chai, J.Y., Si, L.: Effective automatic image annotation via a coherent language model and active learning. In: Schulzrinne, H., Dimitrova, N., Sasse, M.A., Moon, S.B., Lienhart, R. (eds.) *ACM Multimedia*, pp. 892–899. ACM, New York (2004)
15. Leacock, C., Chodorow, M.: Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (ed.) *WordNet: An Electronic Lexical Database*, pp. 265–283. MIT Press, Cambridge, MA (1998)
16. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(6), 985–1002 (2008)
17. Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java CBIR library. In: El-Saddik, A., Vuong, S., Griwodz, C., Bimbo, A.D., Candan, K.S., Jaimes, A. (eds.) *ACM Multimedia*, pp. 1085–1088. ACM, New York (2008)
18. Miller, G.A.: Wordnet: A lexical database for english. *Commun. ACM* **38**(11), 39–41 (1995)
19. Mori, Y., Takahashi, H.: Image-to-word transformation based on dividing and vector quantizing images with words (Oct 27 1999)
20. Patwardhan, S., Pedersen, T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: *Proceedings of the EACL 2006 Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together*, pp. 1–8. Trento, Italy (April 2006)
21. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. *IEEE Trans. Syst. Man Cybern.* **19**(1), 17–30 (1989)
22. Seco, N., Veale, T., Hayes, J.: An intrinsic information content metric for semantic similarity in wordnet. In: de Mántaras, R.L., Saitta, L. (eds.) *ECAI*, pp. 1089–1090. IOS Press (2004)
23. Spanakis, G., Siolas, G., Stafragopatis, A.: A hybrid web-based measure for computing semantic relatedness between words. In: *The 2009 21st IEEE International Conference on Tools with Artificial Intelligence*, pp. 441–448. Washington, DC (2009)
24. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: *21. AAAI/18. IAAI 2006*. AAAI Press, Atlanta, Georgia (2006)
25. Swartz, A.: Musicbrainz: A semantic web service. *IEEE Intell. Syst.* **17**(1), 76–77 (2002)
26. Wang, J.Z., Li, J.: Learning-based linguistic indexing of pictures with 2-D MHMMs. In: *Proceedings of the Tenth ACM International Conference on Multimedia (MM-02)*, pp. 436–445. ACM Press, New York (2002)
27. Wu, Z., Palmer, M.S.: Verb semantics and lexical selection. In: *ACL*, pp. 133–138 (1994)

The DReW System for Nonmonotonic DL-Programs

Guohui Xiao, Thomas Eiter, and Stijn Heymans

Abstract Nonmonotonic DL-programs provide a loose integration of Description Logic (DL) ontologies and Logic Programming (LP) rules with negation, where a rule engine can query an ontology with a native DL reasoner. However, in most systems for DL-programs, the overhead of an external DL reasoner might be considerable. *Datalog-rewritable* DL ontologies, such as most fragments of OWL 2 RL, OWL 2 EL, and OWL 2 QL, can be rewritten to *Datalog* programs, so that DL-programs can be reduced to *Datalog*[¬], i.e., *Datalog* with negation, under both well-founded and answer set semantics. We developed the reasoner **DReW** that uses the *Datalog*-rewriting technique. In addition to DL-programs, **DReW** can also answer conjunctive queries under DL-safeness conditions over *Datalog*-rewritable ontologies as well as reason on terminological default logics over such ontologies.

1 Introduction

Nonmonotonic DL-programs [5] provide a loose integration of Description Logic (DL) ontologies and Logic Programming (LP) rules with negation, where a rule engine can query an ontology using a native DL reasoner. For DL-programs over tractable DL ontologies under well-founded semantics, the reasoning problem is tractable [4]. However, even for tractable DL-programs, the overhead of an external DL reasoner might be considerable.

G. Xiao (✉) • T. Eiter
Institute of Information Systems 184/3 Vienna University of Technology Favoritenstraße 9–11,
A–1040 Vienna, Austria
e-mail: xiao@kr.tuwien.ac.at; eiter@kr.tuwien.ac.at

S. Heymans
Artificial Intelligence Center, SRI International Menlo Park, CA 94025, USA
e-mail: stijn.heymans@sri.com

To remedy the overload of calling external DL reasoners, we proposed the notion of **Datalog**-rewritability in [8]. Intuitively, a **Datalog**-rewritable ontology can be rewritten to a **Datalog** program in a modular way with respect to data access. Moreover, DL-programs over such **Datalog**-rewritable ontologies can then be reduced to **Datalog**[−] programs, i.e., to **Datalog** with negation. A particular DL that is polynomially **Datalog**-rewritable is $\mathcal{L}\mathcal{D}\mathcal{L}^+$, which is essentially an extension of OWL 2 RL and was also proposed in [8]. Reasoning in $\mathcal{L}\mathcal{D}\mathcal{L}^+$ is tractable, under both data and combined complexity. Based on [9], it was easily established that OWL 2 EL ontologies (modulo data types) are also polynomial **Datalog**-rewritable [7]. OWL 2 QL is even FO rewritable [1] and thus **Datalog**-rewritable.

Based on the concept of **Datalog**-rewriting, we developed a reasoner **DReW** (**Datalog ReWriter**)¹ [7, 12], which rewrites DL-programs over **Datalog**-rewritable ontologies to **Datalog**[−] programs and calls an underlying rule-based reasoner, currently **DLV**, to perform the actual reasoning. DL-programs are a very expressive language. Several formalisms, e.g., conjunctive query (CQ) answering under DL-safeness restriction [10] and terminological default reasoning [5], can be rewritten to DL-programs. We support these two reasoning services directly in the **DReW** system.

2 DL-Programs

Informally, a DL-program (Σ, P) consists of a DL knowledge base (or ontology) Σ over predicates \mathcal{P}_o and a **Datalog**[−] program P over predicates \mathcal{P}_p , distinct from \mathcal{P}_o , where P may contain queries to Σ via the so-called DL-atoms. Due to space constraints, we refer to [4, 5] for the formal syntax and semantics of DL-programs and confine here to illustrate the intuition behind on an example from [3].

Example 1. Suppose that an existing network must be extended by new nodes. The knowledge base Σ contains information about existing nodes (n_1, \dots, n_5) and their interconnections as well as a definition of “overloaded” nodes (concept *HighTrafficNode*), which are nodes with more than three connections:

$$\begin{aligned} & \geq 1.wired \sqsubseteq Node; \quad \top \sqsubseteq \forall wired.Node; \quad wired = wired^-; \\ & \geq 4.wired \sqsubseteq HighTrafficNode; \quad n_1 \neq n_2 \neq n_3 \neq n_4 \neq n_5; \\ & Node(n_1); \quad Node(n_2); \quad Node(n_3); \quad Node(n_4); \quad Node(n_5); \\ & \quad wired(n_1, n_2); \quad wired(n_2, n_3); \quad wired(n_2, n_4); \\ & \quad wired(n_2, n_5); \quad wired(n_3, n_4); \quad wired(n_3, n_5). \end{aligned}$$

¹<http://www.kr.tuwien.ac.at/research/systems/drew>.

The following program P evaluates possible combinations of connecting the new nodes:

$$\text{newnode}(x_1). \quad (1)$$

$$\text{newnode}(x_2). \quad (2)$$

$$\text{overloaded}(X) \leftarrow \text{DL}[\text{wired} \uplus \text{connect}; \text{HighTrafficNode}](X). \quad (3)$$

$$\begin{aligned} \text{connect}(X, Y) &\leftarrow \text{newnode}(X), \text{DL}[\text{Node}](Y), \text{not overloaded}(Y), \\ &\quad \text{not excl}(X, Y). \end{aligned} \quad (4)$$

$$\text{excl}(X, Y) \leftarrow \text{connect}(X, Z), \text{DL}[\text{Node}](Y), Y \neq Z. \quad (5)$$

$$\text{excl}(X, Y) \leftarrow \text{connect}(Z, Y), \text{newnode}(Z), \text{newnode}(X), Z \neq X. \quad (6)$$

$$\text{excl}(x_1, n_4). \quad (7)$$

The facts (1)–(2) (bodyless rules) define the new nodes to be added. Rule (3) imports knowledge about overloaded nodes in the existing network, taking new connections already into account. Rule (4) connects a new node to an existing one, provided the latter is not overloaded and the connection is not to be disallowed, which is specified by Rule (5) (there must not be more than one connection for each new node) and Rule (6) (two new nodes cannot be connected to the same existing one). Rule (7) states a specific condition: node x_1 must not be connected with n_4 .

The meaning of DL-programs is given by formal semantics, among which, answer set semantics [5] and well-founded semantics [4] are widely used (see [11] for a survey). The DL-program (Σ, P) in Example 1 has four strong answer sets : $M_1 = \{\text{connect}(x_1, n_1), \text{connect}(x_2, n_4), \dots\}$, $M_2 = \{\text{connect}(x_1, n_1), \text{connect}(x_2, n_5), \dots\}$, $M_3 = \{\text{connect}(x_1, n_5), \text{connect}(x_2, n_1), \dots\}$, and $M_4 = \{\text{connect}(x_1, n_5), \text{connect}(x_2, n_4), \dots\}$. Note that the ground DL-atom

$$\text{DL}[\text{wired} \uplus \text{connect}; \text{HighTrafficNode}](n_2)$$

from Rule (3) is true in any partial interpretation of P . According to the proposed well-founded semantics for DL-programs in [4], the atom $\text{overloaded}(n_2)$ is thus true in the well-founded model.

3 Reasoning with DL-Programs by Datalog[¬] Rewriting

We present the rewriting approach in DReW by means of Example 1. This is achieved by carefully rewriting different components of DL-programs into Datalog[¬] rules.

1. Rewriting Ontology into Datalog. For Datalog-rewritable DLs, the instance query problem can be reduced to the query in Datalog. The DL component Σ in Example 1 is in OWL 2 RL, which is Datalog-rewritable. We transform Σ to Datalog program $\Phi_{\text{RL}}(\Sigma)$:

For TBox axiom $\geq 1.\text{wired} \sqsubseteq \text{Node}$, we add the following rule to $\Phi_{\text{RL}}(\Sigma)$:

$$\text{Node}(X) \leftarrow \text{wired}(X, Y).$$

For TBox axiom $\top \sqsubseteq \forall \text{wired}.\text{Node}$, we add the following rule:

$$\text{Node}(Y) \leftarrow \text{wired}(X, Y).$$

For TBox axiom $\text{wired} = \text{wired}^-$, we have

$$\text{wired}(X, Y) \leftarrow \text{wired}(Y, X).$$

For TBox axiom $\geq 4\text{wired} \sqsubseteq \text{HighTrafficNode}$, we have

$$\begin{aligned} \text{HighTrafficNode}(X) &\leftarrow \text{wired}(X, Y_1), \text{wired}(X, Y_2), \text{wired}(X, Y_3), \text{wired}(X, Y_4), \\ Y_1 &\neq Y_2, Y_1 \neq Y_3, Y_1 \neq Y_4, Y_2 \neq Y_3, Y_2 \neq Y_4, Y_3 \neq Y_4. \end{aligned}$$

Finally, the ABox assertions in Σ (e.g., $\text{Node}(n_1)$) are transformed to Datalog facts directly. Note that after transformation, $n_i \neq n_j$, $1 \leq i < j \leq 5$, is dropped because of the Unique Name Assumption (UNA) adopted by Datalog.

2. Duplicating Rewritten Ontologies According to the DL-Inputs. Note that each DL-atom sends up a different input to Σ and that entailments for each different input might be different. To this purpose, we copy $\Phi_{\text{RL}}(\Sigma)$ to new disjoint equivalent versions for each DL-input, i.e., for each distinct DL-input λ , we define a new program $\Phi_{\text{RL}, \lambda}(\Sigma)$ that results from replacing all the predicates by a λ -subscripted version.

Thus, for the set $\Lambda_P = \{\lambda_1 = \emptyset, \lambda_2 = \text{wired} \uplus \text{connect}\}$ of DL-atoms, we have $\Phi_{\text{RL}, \lambda_1}(\Sigma) = \{\text{Node}_{\lambda_1}(X) \leftarrow \text{wired}_{\lambda_1}(X, Y), \dots\}$ and $\Phi_{\text{RL}, \lambda_2}(\Sigma) = \{\text{Node}_{\lambda_2}(X) \leftarrow \text{wired}_{\lambda_2}(X, Y), \dots\}$.

3. Rewriting DL-Rules to Normal Rules. To rewrite DL-rules P into normal rules P^{ord} , we simply replace each DL-atom $DL[\lambda; Q](\mathbf{t})$ by a new atom $Q_\lambda(\mathbf{t})$. For example, Rule (3) is replaced by

$$\text{overloaded}(X) \leftarrow \text{HighTrafficNode}_{\lambda_2}(X).$$

4. Rewriting DL-Atoms to Datalog Rules. The inputs in the DL-atoms Λ_P can then be encoded as rules $\rho(\Lambda_P)$:

$$\text{wired}_{\lambda_2}(X, Y) \leftarrow \text{connect}(X, Y).$$

5. Calling Datalog Reasoner. Now we have transformed all the components into a Datalog^- program $\Psi(\Sigma, P) = \Phi_{RL,\lambda_1}(\Sigma) \cup \Phi_{RL,\lambda_2}(\Sigma) \cup P^{\text{ord}} \cup \rho(\Lambda_P)$. We can send it to a datalog engine, e.g., DLV, and compute the answer set or well-founded models.

4 Reasoning with Conjunctive Queries and Terminological Default Logics

DL-programs are a very expressive language. Several formalisms, e.g., conjunctive query (CQ) answering under DL-safeness restriction [10] and terminological default reasoning [5], can be captured by DL-programs. We support these two reasoning tasks directly in the DReW system.

4.1 Conjunctive Query Answering Under DL-Safeness Condition

A conjunctive query q is a rule of the following form:

$$ans(X_1, \dots, X_n) \leftarrow C_1(Y_1), \dots, C_m(Y_m), r_1(Z_{11}, Z_{12}), \dots, r_k(Z_{k1}, Z_{k2}),$$

where C_i 's and r_i 's are concepts and roles in the ontology, respectively, and ans is a fresh predicate name.

When applying the DL-safe condition [10], every such query can be equivalently converted to a DL-rule by replacing every atom $Q(X)$ with the DL-atom $\text{DL}[Q](X)$ having empty input list.

$$\begin{aligned} ans(X_1, \dots, X_n) &\leftarrow \text{DL}[C_1](Y_1), \dots, \text{DL}[C_m](Y_m), \\ &\quad \text{DL}[r_1](Z_{11}, Z_{12}), \dots, \text{DL}[r_k](Z_{k1}, Z_{k2}), \end{aligned}$$

Example 2. The following CQ retrieves pairs of wired *HighTrafficNode* X and Y :

$$ans(X, Y) \leftarrow \text{HighTrafficNode}(X), \text{wired}(X, Y), \text{HighTrafficNode}(Y)$$

It can be converted to a DL-rule:

$$ans(X, Y) \leftarrow \text{DL}[\text{HighTrafficNode}](X), \text{DL}[\text{wired}](X, Y), \text{DL}[\text{HighTrafficNode}](Y)$$

4.2 Reasoning with Terminological Default

The bidirectional flow of knowledge between a DL ontology and a logic program enables a variety of possibilities. One application for DL-programs is terminological default theory [2,5]. Here, Reiter-style default rules are applied to named individuals explicitly occurring in the knowledge base; the classic birds&penguins example can be captured by rule $Bird(X) : Flier(X)/Flier(X)$ (informally, birds fly by default).

The intuition of reduction of terminological default logics to DL-programs is to use in/out predicates to guess the default extension and to check the default extension by different DL-inputs; see [7] for more details.

5 System Architecture and Usage

Figure 1 shows a schematic overview of the components of DReW in charge of reasoning with DL-programs by Datalog rewriting. DReW is written in Java using OWL API² for parsing ontologies. The underlying Datalog engine we use is DLV³ which supports both answer set semantics and well-founded semantics. For the ontology component, the current version DReW supports OWL 2 RL and OWL 2 EL (modulo data types). At present, DReW implements DL-programs under both well-founded semantics and answer set semantics.

DReW is distributed as several jars and scripts. It can be used both as a Java library and from the command line. Some of the command line options are listed as follows:

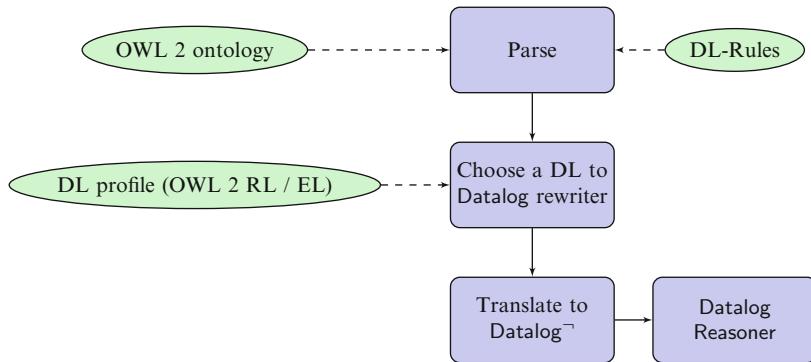


Fig. 1 DReW control flow of DReW with DL-programs

²<http://owlapi.sourceforge.net/>.

³<http://www.dlvsystem.com/dlv/>.

```
% drew [Rew] <Ontology.owl> [Rule] [-dlv /path/to/dlv]
```

Option [Rew] is the datalog rewriter for DL:

-rl	Using OWL 2 RL rewriting (default)
-el	Using OWL 2 EL rewriting

Option [Rule] specifies which rule formalism is used:

-dlp	<rule.dlp>	Reasoning with DL-programs
-sparql	<query.sparql>	Conjunctive query answering
-df	<default.df>	Terminological defaults

6 Conclusions and Outlook

Interesting classes of DLs are Datalog-rewritable, and reasoning with DL-programs over such DLs can be reduced to Datalog[¬] under well-founded semantics and well-founded semantics. This reduction is implemented in DReW system which avoids calling external DL reasoner and runs (sometimes significantly) faster than the traditional hybrid reasoner over Datalog-rewritable ontologies (cf. [7, 12]). Conjunctive query answering (under DL-safeness) and terminological default reasoning can be reduced to DL-programs, and both tasks are directly supported by DReW system. Future work is planned in two directions. One direction is to support further DLs, e.g., OWL 2 QL and Horn-SH_{IQ} [6]; the other is to support tailored nonmonotonic reasoning modalities, e.g., closed world reasoning [5].

Acknowledgements This work has been partially supported by the Austrian Science Fund (FWF) project P20840 and EU Project OntoRule (FP7 231875).

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. *J. Artif. Intell. Res.* **36**, 1–69 (2009)
2. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reason.* **14**(1), 149–180 (1995)
3. Drabent, W., Eiter, T., Ianni, G., Krennwallner, T., Lukasiewicz, T., Maluszynski, J.: Hybrid reasoning with rules and ontologies. In: Bry, F., Maluszynski, J. (eds.) REWERSE, vol. 5500 of Lecture Notes in Computer Science, pp. 1–49. Springer, New York (2009)
4. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R.: Well-founded semantics for description logic programs in the Semantic Web. *ACM Trans. Comput. Log.* **12**(2), 11 (2011)
5. Eiter, T., Ianni, G., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Combining answer set programming with description logics for the semantic web. *Artif. Intell.* **172**(12–13), 1495–1539 (2008)
6. Eiter, T., Ortiz, M., Simkus, M., Tran, T., Xiao, G.: Query rewriting for Horn-SHIQ plus rules. In: Proc. of AAAI 2012. AAAI, Toronto, Canada (2012)

7. Eiter, T., Krennwallner, T., Schneider, P., Xiao, G.: Uniform evaluation of nonmonotonic DL-programs. In: FoIKS'12, pp. 1–22. Springer, New York (2012)
8. Heymans, S., Eiter, T., Xiao, G.: Tractable reasoning with DL-programs over datalog-rewritable description logics. In: Proc. of ECAI 2010. IOS Press, Lisbon, Portugal (2010)
9. Krötzsch, M.: Efficient inferencing for OWL EL. In: JELIA, LNCS, vol. 6341, pp. 234–246, 2010
10. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. *J. Web Semant.* **3**(1), 41–60 (2005)
11. Wang, Y., You, J.-H., Yuan, L.-Y., Shen, Y.-D., Zhang, M.: The loop formula based semantics of description logic programs. *Theor. Comput. Sci.* **415**, 60–85 (2012)
12. Xiao, G., Heymans, S., Eiter, T.: DReW: a reasoner for datalog-rewritable description logics and dl-programs. In: Informal Proc. 1st Int'l Workshop on Business Models, Business Rules and Ontologies (BuRO 2010), 2010

Accessing Information About Linked Data Vocabularies with vocab.cc

Steffen Stadtmüller, Andreas Harth, and Marko Grobelnik

Abstract Linked Data vocabulary designers and application developers need means to easily identify relevant vocabularies, to allow them to reuse existing vocabularies and to develop applications making use of Linked Data. We describe a system that provides information about the popularity of classes and properties based on the Billion Triple Challenge data set. The information about classes and properties can be accessed via a web portal or via Linked API resources. We describe both the data analysis process and the architecture of the web portal.

1 Introduction

Providing data in a machine understandable manner—for example, as Linked Data—significantly improves access and integration of such data. Vocabularies provide schema information for Linked Data, i.e., allowing to talk about classes and properties that are used to describe instances. The fourth Linked Data principle¹ implies the reuse of existing vocabulary URIs. Reusing existing URIs improves the interlinkage of hitherto disparate pieces of data. Thus, data publishers should reuse existing vocabulary URIs, rather than minting new URIs, if possible and appropriate [1,4]. However, it is currently not easy to find out which domain existing

¹<http://www.w3.org/DesignIssues/LinkedData.html>.

S. Stadtmüller (✉) • A. Harth
Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
e-mail: steffen.stadtmueller@kit.edu; andreas.harth@kit.edu

M. Grobelnik
Jožef Stefan Institute, Slovenia
e-mail: marko.grobelnik@ijs.si

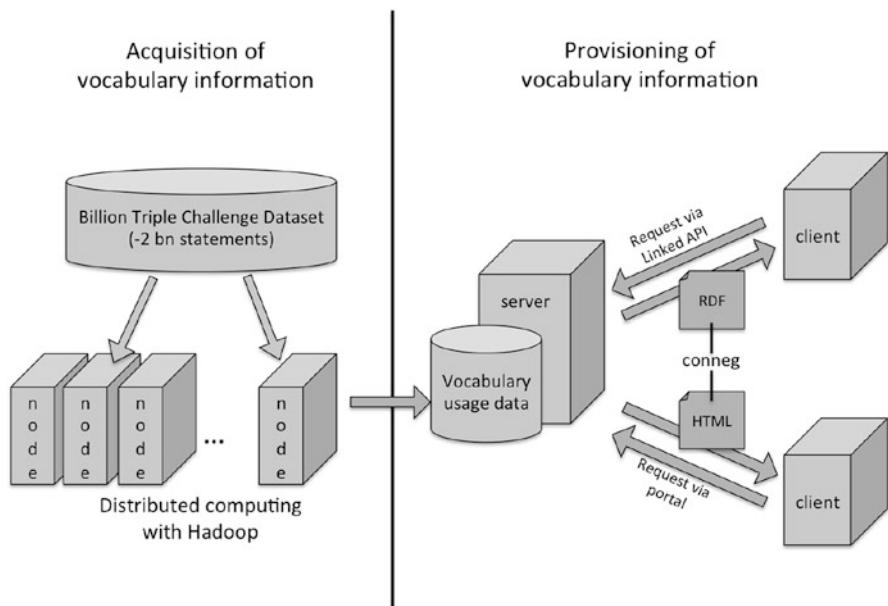


Fig. 1 Architecture overview

vocabularies cover or how relevant existing vocabularies are. Therefore, we see the need for a system that enables data publishers to swiftly acquire information about already available vocabularies and their relevance.

We devise a web portal, called *vocab.cc*,² where data publishers and developers can access information about popular classes and properties.

Rather than requiring manual effort, our notion of popularity stems from an analysis of a crawled data set. In other words, *vocab.cc* focuses on the ex post acquisition and provision of information about already existing ontologies. Information about the real use of class URIs and property URIs in the web of data provides an indicator for the relevance of a specific vocabulary URI. In addition to a web interface targeted at human users, *vocab.cc* offers additionally a Linked API, which allows for an easy integration of the data into other applications. The code of *vocab.cc* is available as open source.³

Our demonstration will show how to acquire useful information from a large Linked Data corpus and how the information acquired for *vocab.cc* can be accessed via the portal and via the Linked API. Figure 1 illustrates an overview of the architecture. We describe related approaches in Chap. 2. In Chap. 3 we describe how we acquire the necessary statistical information. Next, we explain how the information can be accessed in Chap. 4. Finally, we conclude with Chap. 5.

²<http://vocab.cc/>.

³<http://code.google.com/p/vocab/>.

2 Related Work

LODStats offers information according to 32 statistical criteria about data sets published in the CKAN repository.⁴ To do so it accesses dump files and the SPARQL end points of the registered data sets. In its current release, the tool covers 226 data sets with a volume of 1,211,878,106 triples. The analysis of the underlying ontologies covers 14,433 unique vocabulary elements. *SchemaCache*⁵ and *Linked Open Vocabularies (LOV)*⁶ operate as registries for ontologies used by Linked Data publishers. The documentation of these ontologies is provided by the developers or submitted by (registered) users. *Schema-Cache* covers 9,489 unique vocabulary elements. With a more limited coverage (3,714 vocabulary elements), *LOV* focuses on the classification of vocabularies and the provisioning of detailed metadata information about them.

Cupboard is an approach to support ontology engineers to publish ontologies in a way that users can assess and reuse ontologies [2].

vocab.cc offers with 261,119 unique vocabulary elements a significantly larger coverage of existing vocabularies than previous approaches.

3 Analysis of Existing Linked Data Vocabularies

Our demonstration will provide an introduction in the methods used to extract information from a large data set.

As basis for our analysis, we use the Billion Triple Challenge 2011 data set,⁷ which contains over 2.1 bn statements in N-Quads⁸ format, collected from 7.4 m documents. We extract all URIs from the BTCD that are used as predicates (a total of 47,681) and all URIs that represent a class (a total of 213,438), thus covering 261,119 unique vocabulary elements. URIs are identified as classes if they are in object position in a triple with *rdf:type* as predicate.

Considering the size of the corpus, we use Apache Hadoop⁹ to analyse the data. Hadoop allows for the parallel and distributed processing of large data sets across clusters of computers. We run the analysis on the KIT OpenCirrus¹⁰ Hadoop cluster. OpenCirrus is a collaboration of several organizations to provide an open cloud-computing research test bed designed to support research. For our analysis we used

⁴<http://stats.lod2.eu/>.

⁵<http://schemacache.com/>.

⁶<http://labs.mondeca.com/dataset/lov/index.html>.

⁷<http://km.aifb.kit.edu/projects/btc-2011/>.

⁸<http://sw.deri.org/2008/07/n-quads/>.

⁹<http://hadoop.apache.org/>.

¹⁰<https://opencirrus.org/>.

Table 1 Top vocabulary URIs by overall occurrence

(a) Top 10 classes		(b) Top 10 properties			
#	URI	Overall frequency	#	URI	Overall frequency
1	foaf:Person	365 623 021	1	rdf:type	579 095 292
2	cube:Observation	6 783 306	2	rdfs:seeAlso	369 286 912
3	rdf:Statement	5 767 380	3	foaf:nick	366 167 925
4	mo:MusicArtist	3 979 450	4	foaf:knows	365 522 760
5	cc:Work	3 055 547	5	rdfs:label	25 755 421
6	foaf:OnlineAccount	2 930 600	6	foaf:weblog	21 814 705
7	foaf:PersonalProfileDocument	2 593 101	7	foaf:member_name	19 146 708
8	foaf:Agent	2 535 723	8	foaf:tagLine	19 146 699
9	owl:Class	2 096 025	9	foaf:image	18 133 652
10	swrc:Person	1 850 559	10	owl:sameAs	8 552 727

Table 2 Top vocabulary URIs by count of documents

(a) Top 10 classes		(b) Top 10 properties			
#	URI	Document frequency	#	URI	Document frequency
1	foaf:Person	1 633 434	1	rdf:type	6 694 991
2	foaf:Document	814 800	2	rdfs:label	2 867 107
3	freebase:common.topic	572 382	3	rdfs:seeAlso	2 381 790
4	owl:Thing	468 387	4	foaf:primaryTopic	2 099 555
5	mo:MusicArtist	346 728	5	owl:sameAs	1 778 210
6	dc:IMT	330 971	6	foaf:weblog	1 590 806
7	frbr:Manifestation	330 946	7	foaf:nick	1 496 280
8	frbr:Expression	330 943	8	foaf:knows	1 469 700
9	metalex:BibliographicManifestation	330 943	9	foaf:img	1 341 111
10	metalex:BibliographicExpression	330 943	10	foaf:page	1 194 188

54 work nodes, each with a 2.27 GHz 4-Core CPU and 100GB RAM, a setup which completes a scan over the entire corpus in about 15 min.

During a scan over the data, for each identified class URI and property URI, we derive two frequency measures (results in Tables 1 and 2):

- We count how often each identified class and property is used in the BTCD overall. An overall count regards classes and properties more important even if they appear often in just a few large documents.
- We also count for every identified URI how many of the original data sources (i.e., documents) make use of the URI. A count per document regards vocabularies more important that are used by many different documents, even if they are small.

Furthermore we extract all labels of class URI and property URI to allow for keyword search functionality. We also extract the local names of identified URIs and add them to the set of labels. A web application, described next, provides the statistics and the keyword search functionality to users.

4 Access to the Information

We provide access to the data derived from the BTC corpus via *vocab.cc*. The portal provides a minimal interface with a central input field, where users can specify a URI or type in a keyword query. Users can input URIs with their common namespace rather than their fully qualified name. *vocab.cc* makes use of *prefix.cc*¹¹ which also inspired name and layout of the web portal. Figures 2 and 3 show the portal and how results are represented.

Users can define an arbitrary query (i.e., a string of words) for their domain of interest to search for existing vocabularies. *vocab.cc* matches the words in a query with the labels found for the URIs. The response details the classes and properties, which labels contain all of the specified words. Words in the query are disregarded, if they do not appear in any label, thus increasing the number of potential result sets.

vocab.cc also allows users to specify a URI directly. Returned information includes the number of overall appearances in the BTC data set as well as the number of documents the URI appeared in. Additionally *vocab.cc* returns the positions in the rankings.

A Linked API allows access to the information, beyond the human readable way to access *vocab.cc*. The Linked API allows for an easy integration of the functionalities in other applications, fostering the Linked Data principles. Linked

Search Results			
maybe these URIs represent what you are looking for:			
URI	Occured Overall	Type	
http://openresearch.org/wiki/Special:URIResolver/Property-3AHas_program_chair ↗ 1589	1589	Property	
http://semanticweb.org/id/Property-3AHas_program_chair ↗ 448	448	Property	
http://semanticweb.org/id/Property-3AHas_area_program_chair ↗ 45	45	Property	
http://semanticweb.org/id/Property-3AHas_program_committee_chair ↗ 4	4	Property	

Fig. 2 Query results

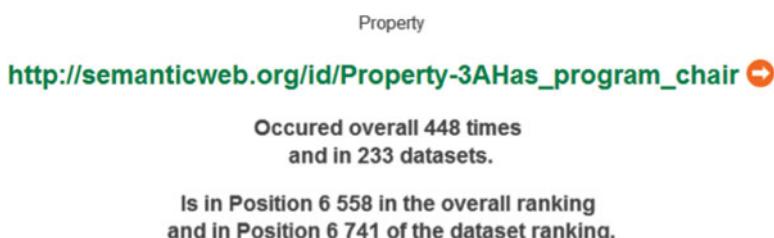


Fig. 3 Usage information for a URI

¹¹<http://prefix.cc/>.

APIS [5, 6] offer web service functionalities as RDF prosumers by combining LD technologies with RESTful services [3].

The demonstration will illustrate the different methods to make use of *vocab.cc*.

The resources of the Linked API allow to submit queries to *vocab.cc* in an HTTP POST request. The HTTP response contains RDF data, detailing the usage information of the found URIs. Accessing the output RDF as resources is also possible directly via content negotiation: A Client can perform an HTTP GET on the corresponding URI of the portal asking for an RDF content type. This direct access adheres to a RESTful architecture style.

5 Conclusion and Outlook

vocab.cc provides the means to search for RDF vocabularies based on labels and URIs and decides on the relevance of the vocabularies based on usage information.

To improve the *vocab.cc*, accounting for subclass and subproperty hierarchies could lead to a refined definition of the popularity of a URI. Furthermore, aggregating the usage information can lead to an understanding of the relevance of a vocabulary itself, rather than just of the individual classes and properties. Possible synergies can be achieved by linking *vocab.cc* with other vocabulary catalogues. Finally, we intend to allow users to contribute data about vocabularies.

Acknowledgements The research leading to this paper was partially supported by the Network of Excellence PlanetData,¹² funded by the European Community's Seventh Framework Programme FP7/2007-2013 under contract 257641.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* **5**(3), 122 (2009)
2. d'Aquin, M., Lewen, H.: Cupboard - a place to expose your ontologies to applications and the community. In: ESWC. LNCS, vol. 5554, pp. 913–918. Springer, New York (2009)
3. Fielding, R.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine (2000)
4. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, 1st edn. Synthesis Lectures on the Semantic Web, Morgan & Claypool (2011)
5. Krummenacher, R., Norton, B., Marte, A.: Towards linked open services. In: 3rd Future Internet Symposium, September 2010
6. Speiser, S., Harth, A.: Integrating linked data and services with linked data services. In: Proceedings of 8th Extended Semantic Web Conference, ESWC 2011. pp. 170–184 (2011)

¹²<http://planet-data.eu/>.

Qualitative Cognition for Uncertainty Knowledge Using Cloud Model

Yuchao Liu, Lin Li, and Juanzi Li

Abstract Concepts are basic elements of natural language processing, studying on concept representation and transformation between connotation and extension become more and more important. Multi-granularity concept extraction is still a difficult problem in uncertainty knowledge representation. Cloud model is an uncertainty cognition model, which realizes the bidirectional transformation between a qualitative concept and quantitative data by Gaussian cloud algorithm. Gaussian cloud transformation provides a method to transform a group of data in problem domain to multiple concepts in different granularities in cognition domain. This paper introduces cloud model and Gaussian cloud transformation algorithm to describe the multi-granularity concepts. A case study is also given to prove the effectiveness of the proposed method.

1 Introduction

The uncertainty knowledge representation and processing, including qualitative and quantitative transformation, soft computing, and granularity changing computing, are becoming hot issues in the Internet age. At present, most uncertainty knowledge representation methods study on set division, operation, and reduction by the equivalence relation or fuzzy equivalence relation [1]. However, rather than using the data sets or much more complex logical computation, people would like to use the qualitative concepts in a cognition process. Human cognition process is a qualitative and quantitative bidirectional transformation.

Y. Liu (✉) • J. Li

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084 China
e-mail: yuchao_liu@163.com; lijuanzi2008@gmail.com

L. Li

Astronaut Center of China, Beijing, 100193 China
e-mail: eleven8011@126.com

How to transform original data set to multiple qualitative concepts on different granularities and simulate human cognition process is still a puzzle to most uncertainty representation methods. Probability theory [2] is an important branch of mathematics research in randomness. Uncertain theories and methods based on probability statistics, which have been existed over a hundred of years, are currently widely used and universally accepted uncertainty representations. One of the main contributions of the 2011 annual Turing Award winner, Professor Judea Pearl, is that he took the Bayesian networks and probabilistic methods into artificial intelligence.

The purpose of this paper is to research the denotation of concept connotation, the qualitative and quantitative transformation, and the variable granular computing between multi-granularity concepts.

2 Cloud Model: An Uncertainty Cognition Model

Cloud model is a cognitive model which can realize the bidirectional transformation between a qualitative concept and quantitative data [3]. Cloud model employs three characters to represent the concept connotation. The expected value Ex is the point that is most representative of the qualitative concept, or the most classical sample while quantifying the concept. The entropy En is the uncertainty measurement of the concept extension. The hyperentropy He is the uncertainty measurement of the entropy.

2.1 *Cloud Drop's Contribution to the Concept*

Cloud drops of Gaussian cloud model can be classified by their contribution to the concept. The contribution can be computed by mathematical integral. Within the one-dimensional universal domain, the cloud drop cluster Δx in any small region will make contribution to the qualitative concept A by ΔC , which satisfies

$$\Delta C \approx \mu_A(x) \times \Delta x / \sqrt{2\pi} En$$

Obviously, the total contribution C to the concept A by all the elements in the universal domain is

$$C = \frac{\int_{-\infty}^{+\infty} \mu_T(x) dx}{\sqrt{2\pi} En} = \frac{\int_{-\infty}^{+\infty} e^{-(x-Ex)^2/2En^2} dx}{\sqrt{2\pi} En} = 1$$

Because

$$\frac{1}{\sqrt{2\pi} En} \int_{Ex-3En}^{Ex+3En} \mu_T(x) dx = 99.74\%$$

the contributive cloud drops to the concept A in the universal domain U lie in the domain $[Ex - 3En, En + 3En]$.

As a result, we can ignore the contribution to the concept C by the cloud drops out of the domain $[Ex - 3En, Ex + 3En]$. This is the “ $3En$ ” rule of the Gaussian cloud.

According to calculation, the cloud drops within $[Ex - 0.67En, Ex + 0.67En]$ take up 50% of the overall contribution, and they are named as “key elements.” The cloud drops within $[Ex - En, Ex + En]$ take up 68.26% of the overall contribution, and they are named as “basic elements.” The cloud drops within $[Ex - 2En, Ex - En]$ and $[Ex + En, Ex + 2En]$ are called “peripheral elements,” which contribute 27.18% to the concept. The cloud drops within $[Ex - 3En, Ex - 2En]$ and $[Ex + 2En, Ex + 3En]$ are called “weak peripheral elements” because they make only 4.3% of the whole contribution.

2.2 Concept Confusion Measurement

In a Gaussian cloud, He/En can be used to measure the confusion degree of a concept. When $He/En = 0$, cloud drops obey a Gaussian distribution. With the increase of He/En , cloud drops move towards center and peripheral at the same time. According to the FGC, 99.7% cloud drops are located in between two Gaussian distribution curves: $y1 = \exp(-((x - Ex)^2 / 2(En + 3He)^2))$ and $y1 = \exp(-((x - Ex)^2 / 2(En - 3He)^2))$, which just like type-2 fuzzy sets. As shown in Fig. 1, when $He/En = 1/3$, $y2$ is not existed and cloud is atomized, that is to say, cloud drops are too discrete to form a commonsense concept [4].

Gaussian cloud model provides a basic method to realize the transformation between the connotation and extension of a concept, and He/En in Gaussian cloud provides a way to measure the confusion degree of a concept. However, it still cannot solve multi-granularities concepts formation and switch problems in granular computing.

3 A Multi-Granularity Concepts Extraction Method

How to realize the qualitative and quantitative transformation on different granularities is still a difficult problem. Gaussian cloud transformation(GCT)is a method based on Gaussian mixture model (GMM) and transfers a data set in problem domain into multiple concepts of different granularities in cognition domain. He/En is used to measure the overlap extent between concepts.

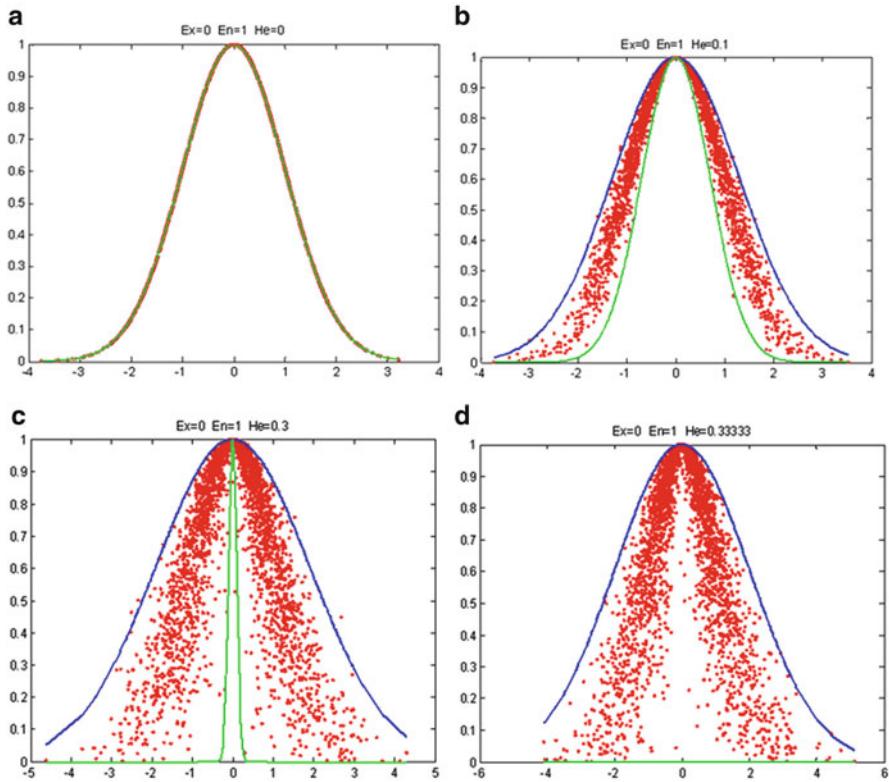


Fig. 1 Effect of He/En in Gaussian cloud. (a) $He/En = 0$. (b) $He/En = 0.1$. (c) $He/En = 0.3En$. (d) $He/En = 0.333$

3.1 From GMM to GCT

Normally, the elements with close relation often exist in a same class, and elements in different classes have loose relation. We try to use the overlap extent of a Gaussian distribution with its neighbors to compute En and He of its relevant concepts.

In a Gaussian distribution $G(\mu_k, \sigma_k)$, if its weak peripheral element region is separated with that of other Gaussian distribution, its relevant concept parameter is $En_k = \sigma_k$, $He_k = 0$. Otherwise, zoom their standard variance at the same scale to guarantee their weak peripheral element region does not overlap. Then two scale parameters α_1, α_2 can be got by the formula $\mu_{k-1} + 3 \times \alpha_1 \times \sigma_{k-1} = \mu_k - 3 \times \alpha_1 \times \sigma_k$ and $\mu_k + 3 \times \alpha_2 \times \sigma_k = \mu_{k+1} - 3 \times \alpha_2 \times \sigma_{k+1}$. The standard variance range of $G(\mu_k, \sigma_k)$ is $[\alpha \times \sigma_k, \sigma_k]$, $\alpha = \min(\alpha_1, \alpha_2)$. According to the definition of Gaussian cloud, En is the expected value of standard variance, and He is the standard variance of standard variance. The Gaussian cloud $C(Ex_k, En_k, He_k)$ parameters can

be calculated: $Ex_k = \mu_k$, $En_k = \sigma_k - (1 - \alpha) \times \sigma_k / 2$, $He_k = (1 - \alpha) \times \sigma_k / 6$. $He/En = (1 - \alpha) / 3(1 + \alpha)$ can be used to measure the clarity of an extracted concept. Using the above method, we can get a cognition relation table for each Gaussian distribution in GMM.

3.2 Adaptive Gaussian Cloud Algorithm

Adaptive Gaussian cloud transformation (A-GCT) can transfer real data sample set to multiple concepts on different granularities automatically without a pre-specified number of concepts. In common sense, relative to the low-frequency data, high-frequency data values have more contribution to the qualitative concept. So if the wave number of data sample frequency distribution can be taken as the initial concept quantity M, then call the GMM to generate M Gaussian distribution and transform them to Gaussian clouds. According to the concept confusion degree He/En Gaussian cloud transform strategy could be set. A-GCT can continuously iterate convergence and form multi-concepts on different granularities finally.

Algorithm A-GCT (Adaptive Gaussian cloud algorithm)

Input : Data set $X \{x_i | i = 1, 2, \dots, N\}$, Concept confusion limitation β

Output : Gaussian clouds $C(Ex_k, En_k, He_k) | k = 1, \dots, m$

step1 : Count the wave number of data sample frequency distribution, as an initial concepts quantity M

step2 : Using GMM to transfer X to M Gaussian distributions:

$G(\mu_k, \sigma_k) | k = 1, \dots, M$;

Step3 : for each $G(\mu_k, \sigma_k)$, compute α_k , and parameters of Gaussian cloud

: $Ex_k = \mu_k$, $En_k = \sigma_k - (1 - \alpha_k) \times \sigma_k / 2$, $He_k = (1 - \alpha_k) \times \sigma_k / 6$,

$He_k / En_k = (1 - \alpha_k) / 3(1 + \alpha_k)$;

Step4: Using H_GCT to transfer X to M Gaussian Clouds:

$C(Ex_k, En_k, He_k)$, $k = 1, \dots, M$

Step5 : for each $C(Ex_k, En_k, He_k)$

if $He_k / En_k > \beta$

$M=M-1$ Loop execution steps 2

else

output m Gaussian clouds in which $He_k / En_k \leq \beta$, $k = 1, \dots, m$

From the algorithm, we can find that the time complexity of the algorithm A-GCT is decided by the loop times, in each loop the complexity of algorithm execution is equal to GMM:

$$\mathcal{O}(m \times N) + \mathcal{O}((m - 1) \times N) + \dots + \mathcal{O}(M \times N) = \mathcal{O}(m^2 \times N)$$

where m is the wave number of data sample frequency distribution; M is the final concept quantity. Compared with GMM, A-GCG can extract concepts on different granularities from original data.

4 A Case Study: Image Gray Concept Extraction on Different Granularities

If only considering the image gray value attribute, then each pixel of an image can be a value in interval [0, 255]. In statistics, the pixel quantity of each gray can get an image gray histogram, i.e., image gray frequency distribution (shown as Fig. 2). Using A-GCT can extract image gray concept represented by Gaussian clouds.

Set basic element independence. A-GCT can obtain four Gaussian clouds: $C_1(13.2, 2.7, 0.40)$, $C_2(39.1, 12.9, 1.91)$, $C_3(122.7, 14.2, 2.29)$, and $C_4(164.6, 12.9, 2.09)$; their He/En , respectively, is 0.149, 0.149, 0.161, and 0.161, as shown in Fig. 3.

Obviously, we also can extract concepts from this image on a coarser granularity. Set peripheral element independence. A-GCT can obtain two Gaussian clouds: $C_1(17.4, 9.9, 0)$ and $C_2(149.7, 31.1, 0)$, as shown in Fig. 4.

Image gray is a one-dimensional attribution data; we also can design multidimensional Gaussian cloud transform to consider multidimensional attributes. This case study shows that cloud model provides a simple knowledge connotation

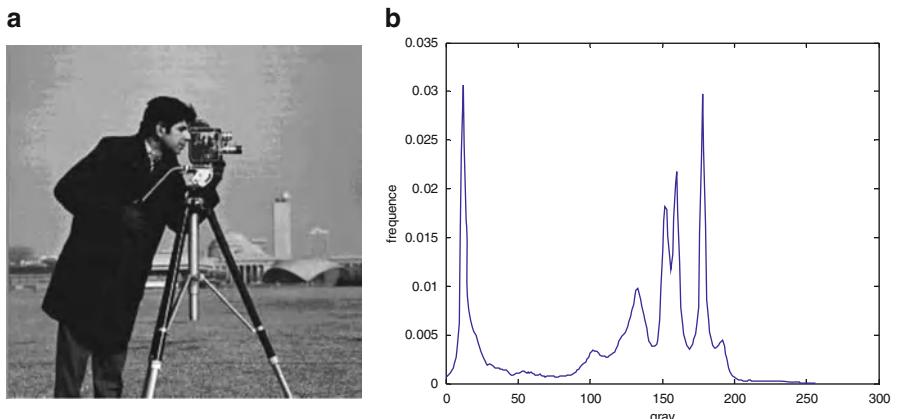


Fig. 2 Image and its gray histogram. (a) Original image. (b) Gray histogram

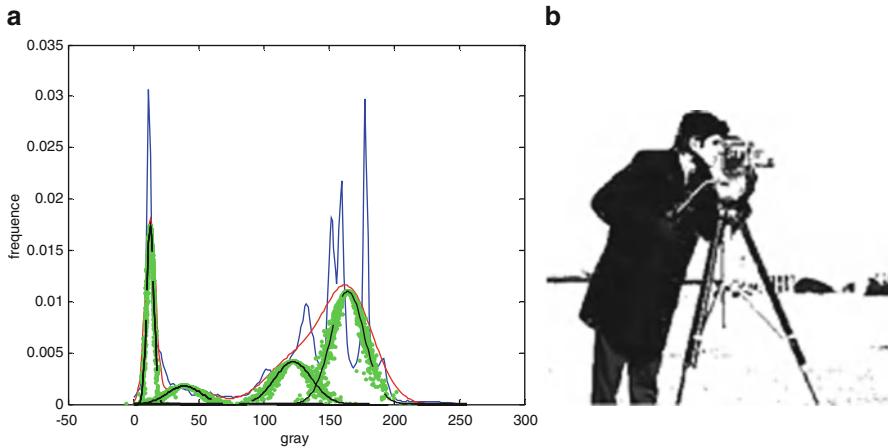


Fig. 3 Four concepts generated by A-GCT with a thinner granularity. (a) Four Gaussian clouds. (b) Pixel image of concept C1

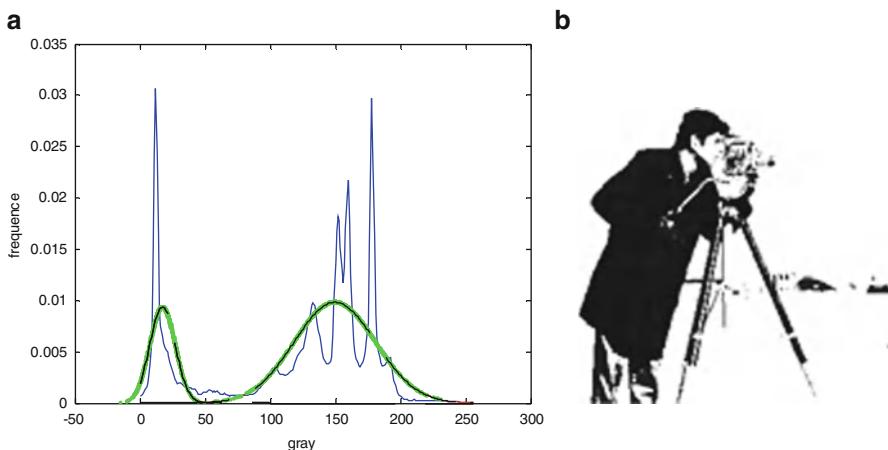


Fig. 4 Two concepts generated by A-GCT with a coarser granularity. (a) Two Gaussian clouds. (b) Pixel image of concept C1

representation method and a way to produce simulation data for incomplete data sets. In the network age, people have been taken as intelligent computing factors and uncertainty computing will be the kernel character of cloud computing; this will provide a great chance for intelligent science researchers.

Acknowledgements This work is supported by the Key Program of the National Natural Science Foundation of China under Grant Nos. 61035004 and 91120306.

References

1. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning—1 [J]. *Info. Sci.* **8**, 199–249 (1975)
2. Wang, Z.: Probability theory and its applications. Beijing Normal University Press, Beijing (1995)
3. Li, D., Du, Y.: Artificial intelligent with uncertainty [M]. Chapman & Hall/CRC, London (2007)
4. Liu, Y., Deyi, L., Guangwei, Z.: Atomized feature in cloud based evolution algorithm. *Acta Electronics Sinica* **37**(8), 1651–1658 (2009)