UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

DIR 2011: Dutch_Belgian Information Retrieval Workshop Amsterdam

Boscarino, C.; Hofmann, K.; Jijkoun, V.; Meij, E.; de Rijke, M.; Weerkamp, W.

Publication date
2011

Document Version
Final published version

Link to publication

Citation for published version (APA):
Boscarino, C., Hofmann, K., Jijkoun, V., Meij, E., de Rijke, M., & Weerkamp, W. (2011). *DIR 2011: Dutch_Belgian Information Retrieval Workshop Amsterdam*. University of Amsterdam, Information and Language Processing group. http://edgar.meij.pro/wp-content/papercite-data/pdf/dir-2011.pdf

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:09 Mar 2023

DIR

2011 DUTCH_BELGIAN INFORMATION RETRIEVAL WORKSHOP AMSTERDAM

Supported by:

TOMTOM _textkernel ru | STEVIN Q-GO RELEVANT ONLINE ANSWERS BETTER CUSTOMER SERVICE TEEZIR Google

Preface

DIR 2011, the 11th Dutch-Belgian Information Retrieval Workshop, was organized by the Information and Language Processing group (ILPS) of the University of Amsterdam in collaboration with the Centrum Wiskunde en Informatica (CWI). Two types of submissions were accepted for the workshop: research papers describing original research, compressed contributions presenting a summary of previously published work, and demonstrations.

There were many people who helped organize DIR 2011, making it a success. We would like to thank them all. In particular, we are gratefull to our keynote speakers, Nick Belkin (Rutgers University) and Gabriella Kazai (Microsoft Research).

Organizing committee

Corrado Boscarino, CWI Amsterdam
Katja Hofmann, University of Amsterdam
Valentin Jijkoun, University of Amsterdam
Edgar Meij, University of Amsterdam
Maarten de Rijke, University of Amsterdam
Wouter Weerkamp, University of Amsterdam

Program committee

Robin Aly, Universiteit Twente
Avi Arampatzis, Democritus University of Thrace
Leif Azzopardi, University of Glasgow
Toine Bogers, RSLIS Copenhagen
Gosse Bouma, University of Groningen
Marc Bron, University of Amsterdam
Walter Daelemans, University of Antwerp
Martine De Cock, Ghent University
Guy De Tre, Ghent University
Arjen De Vries, Delft University of Technology and CWI Amsterdam
Anne Diekema, Utah State University
Erik Duval, KU Leuven
Khairun Nisa Fachry, University of Amsterdam
Claudia Hauff, Universiteit Twente
Djoerd Hiemstra, Universiteit Twente
Eduard Hoenkamp, University of Maastricht
Veronique Hoste, Hogeschool Gent
Theo Huibers, Universiteit Twente
Tamas Jambor, University College London
Jeannette Janssen, Dalhousie University
Jaap Kamps, University of Amsterdam

Rianne Kaptein, University of Amsterdam
Marijn Koolen, University of Amsterdam
Maarten Marx, University of Amsterdam
Marie-Francine Moens, University of Leuven
Henning Rode, TextKernel BV
Dolf Trieschnigg, Universiteit Twente
Manos Tsagias, University of Amsterdam
Theodora Tsikrika, CWI Amsterdam
Antal Van Den Bosch, Tilburg University
Maarten Van Der Heijden, Radboud University Nijmegen
Paul Van Der Vet, Universiteit Twente
Theo Van Der Weide, Radboud University Nijmegen
Marieke Van Erp, Vrije Universiteit Amsterdam
Remco Veltkamp, Utrecht University
Suzan Verberne, Radboud University Nijmegen
Werner Verhelst, Vrije Universiteit Brussel
Junte Zhang, University of Amsterdam
Jianhan Zhu, University College London
Jeannette Janssen, Dalhousie University

Contents

Demonstrations **65**

Keynote talks

Usefulness as the Criterion for Evaluation of Interactive Information Retrieval Systems

Nicholas J. Belkin (School of Communication and Information, Rutgers University)

Relevance has been the classic criterion for evaluation of the effectiveness of information retrieval (IR) systems since the earliest days of IR system evaluation. This criterion has been understood as the ability of an IR system to recognize documents relevant to a person's "information need", and understood as the ability of the system to provide to the person all of the documents in an information resource relevant to that need, and only those documents relevant to the need. The measures of effectiveness of the system have thus been understood as recall and precision. These measures have been applied in the evaluation of the performance of an IR system as referring to the system's ability to maximize these measures in its response to a single query (representation of the information need) put to the system. This criterion, these measures, and the application of the measures depend crucially on both a specific model of IR, and a specific model of the user's desired results, both of which are based on the example of the special purpose bibliography of a topic constructed on demand by documentalists and science librarians in the early and middle 20th century. In this presentation, I argue that the criterion, measures, and application of those measures based on this example are inappropriate for the general interactive IR situation and evaluation of interactive IR systems, and propose that the usefulness of the IR system in supporting the goal or task which led the person to engage in information seeking should be the basic criterion according to which an IR system is evaluated. In particular, I argue that the relevance criterion and its associated measures cannot be used alone to evaluate the performance of an IR system over an information seeking episode, and that usefulness is a criterion which can be used to evaluate both the effectiveness of an IR system over an entire information seeking episode, and the constituent parts of that episode.

About the speaker

Nicholas Belkin has been Professor of Information Science in the School of Communication and Information at Rutgers University since 1985. Nick has been president of the American Society for Information Science and Technology (ASIST), and was Chair of the ACM Special Interest Group on Information Retrieval (SIGIR) from 1995 to 1999. He is the recipient of the ASIST Teaching Award, Research in Information Science Award, and Award of Merit, as well as the New Jersey ASIST Distinguished Lectureship.

Nick's research over the past 25 years has focused on understanding why people engage in interactions with information, the nature of such interactions, and the problems that people face in engaging with information systems. He is one of the founders of the so-called "cognitive view" of information science, which has led to his being one of the most highly cited information scientists in the world. His current research is concerned with personalizing people's interactions with information systems, particularly in the context of information seeking in the Internet environment.

Crowdsourcing for Search Evaluation

Gabriella Kazai (Microsoft Research in Cambridge, UK)

Crowdsourcing has become a widely popular mechanism for solving a range of human intelligence tasks. Such tasks include the labelling of images or search results, a job where humans still outperform machines. As a result, crowdsourcing is increasingly relied upon as a feasible alternative to traditional methods of gathering relevance labels for the evaluation of search engines. However, crowdsourcing raises a range of questions regarding the quality of the resulting data. What indeed can be said about the quality of the data that is contributed by anonymous workers who are only paid cents for their efforts?

In this talk, I will provide an introduction into the world of crowdsourcing for search engine evaluation with specific focus on considerations for quality control within the design of crowdsourcing experiments. I will then discuss the findings of a recent large scale crowdsourcing experiment to gather relevance labels for the evaluation of the INEX Book Track. The experiments offer insights that can aid in the design of HITs for improved output quality.

About the speaker

Gabriella Kazai is a research consultant, working for Microsoft Research in Cambridge, UK. Her research interests include crowdsourcing, social information retrieval, IR evaluation measures, test collection building, book search and active reading, and personal digital libraries. She is founder and organiser of the INEX Book Track since 2007, in the context of which she developed a crowdsourcing system for collecting relevance judgements for digitized books as part of a social game. She is also currently working on a book on Crowdsourcing for Search Engine Evaluation with Omar Alonso and Stefano Mizzaro. Gabriella holds a PhD in computer science from Queen Mary University of London. She published over 40 papers and organised several IR conferences and workshops.

Research papers

Semi-Supervised Priors for Microblog Language Identification

Simon Carter
ISLA, University of Amsterdam
s.c.carter@uva.nl

Manos Tsagkias
ISLA, University of Amsterdam
e.tsagkias@uva.nl

Wouter Weerkamp
ISLA, University of Amsterdam
w.weerkamp@uva.nl

ABSTRACT

Offering access to information in microblog posts requires successful language identification. Language identification on sparse and noisy data can be challenging. In this paper we explore the performance of a state-of-the-art n-gram-based language identifier, and we introduce two semi-supervised priors to enhance performance at microblog post level: (i) blogger-based prior, using previous posts by the same blogger, and (ii) link-based prior, using the pages linked to from the post. We test our models on five languages (Dutch, English, French, German, and Spanish), and a set of 1,000 tweets per language. Results show that our priors improve accuracy, but that there is still room for improvement.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing

General Terms

Algorithms, Theory, Experimentation, Measurement

Keywords

Language identification, microblogs, semi-supervised priors

1. INTRODUCTION

Microblogging platforms such as Twitter have become important real-time information resources [4], with a broad range of uses and applications, including event detection [8, 10], media analysis [1], and mining consumer and political opinions [6, 9]. Microbloggers participate from all around the world contributing content, usually, in their own native language. Language plurality can potentially affect the outcomes of content analysis, and we therefore aim for a monolingual content set for analysis. To facilitate this, *language identification* becomes an important and integrated part of content analysis. In this work, we address the task of language identification in microblog posts.

Language identification has been studied in the past (see Section 2 for previous work in this field), showing successful results on structured and edited documents. Here, we focus on an other type of documents: user generated content, in the form of microblog posts. Microblog posts ("tweets," "status updates," etc.) are a special type of user generated content, mainly due to their limited size, which has interesting effects. People, for example, use word abbreviations or change word spelling so their message can fit in the allotted space, giving rise to a rather idiomatic language that is difficult to match with statistics from external corpora.

To address this effect, we use language models trained on microblog posts. To account for very short ambiguous (in terms of what language) microblog posts, we go a step further and introduce two *semi-supervised priors*, and explore the effects on accuracy of (i) a blogger-based prior, using previous microblog posts by the same blogger, and (ii) a link-based prior, using content from the web page hyperlinks within the post.

In particular, we aim at answering the following research questions: (i) What is the performance of state-of-the-art language identification for microblogs posts? (ii) What is the effect on identification accuracy of using language models trained on microblog posts? (iii) What is the effect on accuracy of using blogger-based and link-based priors? This paper makes several contributions: (i) it explores the performance of state-of-the-art language identification on microblog posts, (ii) it proposes a method to help identification accuracy in sparse and noisy data, and (iii) it makes available a dataset of microblog posts in for others to experiment.

The remainder of the paper is organized as follows: in Section 2 we explore previous work in this area. In Section 3 we introduce our baseline model, and the semi-supervised priors. We test our models using the setup detailed in Section 4, and in Section 5 we present and analyze the results. Finally, we conclude in Section 6.

2. RELATED WORK

Language identification can be seen as a subproblem in text categorization. Cavnar and Trenkle [3] propose a simple, yet effective n-gram-based approach to solving text categorization in general, and test it on language identification. Their approach compares a document "profile" to category profiles, and assigns to the document the category with the smallest distance. Profiles are constructed by ranking n-grams in the training set (or the document) based on their frequency. These ranked lists are then compared using a rank-order statistic, resulting in a distance measure between document and category. Tested on a set of Usenet documents, it achieves an accuracy of 99.8% for language identification.

In [2] the authors compare a neural network approach for language identification to the simple n-gram approach of Cavnar and Trenkle [3] . Although the paper is aimed at comparing performance in terms of processing time, they show that the n-gram approach achieves better accuracy than the neural network approach, reaching up to 98.8%. Accuracy is often very high when looking at structured and well-written documents. Language identification on web pages already seems more difficult [7]: an n-gram-based approach with web-related enhancement has an accuracy between 80% and 99%, depending on the language.

Most language identification work is done on full documents. In our case, however, documents are comparatively (very) short to

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.
.

web documents and are more like queries with regard to length. Interesting work in that respect is done by Gottron and Lipka [5]. The authors explore performance of language identification approaches on (short) queries. They compare a Naive Bayes approach (using n-grams as features) to a Markov approach (such as one found in [11]) and the frequency-ranking approach described above. They conclude that Naive Bayes is the best performing, reaching an accuracy of 99.4% using 5-grams. Both the Markov and frequency-ranking approach perform substantially less, possibly due to the very short length of "documents" (on average, the queries are 45.1 characters long).

Based on previous work, we opt for using an n-gram approach to language identification. More precisely, we use the implementation of the approach by Cavnar and Trenkle [3] as in TextCat.[1]

3. MODELING

In the previous section we explained how TextCat works to identify a document's language. We use the TextCat algorithm for language identification on our microblog post set and study the effect on TextCat accuracy of language models trained on different data sets. We consider three types of language models for: (i) **out-of-the-box**, which uses the training data supplied by TextCat and we set this as our baseline, (ii) **microblog**, for which we use a training set of posts from our target platform to re-train TextCat, and (iii) **combined**, that merges n-grams from both other models.

Let n be the total number of languages for which we have trained language models and $i \in \{1, \ldots, n\}$ denote the corresponding model for a language. For each post p we define a language vector

$$\boldsymbol{\lambda_p} = \langle \lambda_p^1, \lambda_p^2, \ldots, \lambda_p^n \rangle \qquad (1)$$

where λ_p^i is a score denoting the distance between p and language i (the smaller the distance the more likely is p to be written in language i). TextCat scores are not normalized by default and therefore we normalize $\boldsymbol{\lambda_p}$ using the z-scores: $\hat{\boldsymbol{\lambda}}_{\boldsymbol{p}} = \langle \hat{\lambda}_p^1, \hat{\lambda}_p^2, \ldots, \hat{\lambda}_p^n \rangle$. We call vectors constructed from the microblog post itself *content-based identification vectors* and for post p we write ${}_C\hat{\boldsymbol{\lambda}}_{\boldsymbol{p}}$.

3.1 Semi-supervised priors

On top of the language identification on the actual post, we use two semi-supervised priors to overcome problems due to sparseness or noise. Our priors are (i) semi-supervised, because they exploit classifications of the supervised language identifier on unlabeled data, for which we do not know beforehand the true language, to improve the accuracy of our baseline classifiers, and (ii) priors, because they allow us to identify the language of a post without the content-based identification. We propose the use of two priors:

Blogger-based prior: behind each post is a blogger who wrote it, and probably the current post is not her first; there is a post history for each blogger the content of which can be beneficial for our purposes. By identifying (or guessing) the language for previous posts by the same blogger, we construct a blogger-based prior for the current post.

Let $P = \{p_1, \ldots, p_k\}$ be a set of posts predating p from blogger u. For each $p_i \in P$, we use the *microblog* language models, and construct $\hat{\boldsymbol{\lambda}}_{\boldsymbol{p_i}}$, as explained before. We then derive a blogger-prior from the average of content-based identification vectors of previous posts:

$${}_B\hat{\boldsymbol{\lambda}}_{\boldsymbol{p}} = \frac{1}{|P|} \sum_{i=1}^{k} {}_C\hat{\boldsymbol{\lambda}}_{\boldsymbol{p_i}}. \qquad (2)$$

Link-based prior: posts in microblogs often contain features like links or tags. Links refer to content elsewhere on the web, and this content is often of longer text length that the post itself. We identify the language of the linked web page, and use this as link-based prior for the post that contains the link.

Let $L = \{l_1, \ldots, l_j\}$ be a set of links found in post p. For each web page $l_i \in L$ we apply the *out-of-the-box* model to its content, and construct a link-based prior vector from the average of content-based identification vectors of web pages found in p:

$${}_L\hat{\boldsymbol{\lambda}}_{\boldsymbol{p}} = \frac{1}{|L|} \sum_{i=1}^{j} {}_C\hat{\boldsymbol{\lambda}}_{\boldsymbol{l_i}}. \qquad (3)$$

Having constructed three vectors (content, blogger and link-based) with scores for each language, we combine the three vectors using a weighted linear combination. More formally, we identify the most probable language for post p as follows:

$$lang(p) = \text{argmin}\,\frac{1}{|v|} \cdot \sum^{v} w_v {}_v\hat{\boldsymbol{\lambda}}_{\boldsymbol{p}}, \qquad (4)$$

where $v = \{C, B, L\}$, and $\sum^{v} w_v = 1$. Finally, language λ^i that is closest to the language profile (i.e., has the lowest score) is selected as language for post p.

4. EXPERIMENTAL SETUP

For testing our models we need a collection of microblog posts. We collect these posts from one particular microblog platform, Twitter.[2] We test our models on a set of five languages, Dutch, English, French, German, and Spanish, and gather an initial set of *tweets* (Twitter posts) by selecting tweets on their location. From this initial sample, we manually select 1,000 tweets in the appropriate language. In case of a multilingual tweet, we assign the language that is most "content-bearing" for that post. For training purposes, we split each set in a training set of 500 tweets and a test set of 500 tweets.[3] We construct test and training sets by taking one every other tweet so both sets contain approximately the same language.

TextCat allows us to select the number of n-grams we want to use for profiling our language and documents. Preliminary experimentation with this parameter revealed that the standard value (top 400 n-grams) works best, and we use this value for the remainder of the experiments. In our experiments we use fixed weights for the three language vectors; our intuition is that the content-based identification should be leading, supported by the blogger-based prior. Since people can link to pages in other languages as well, we assign least weight to the link-based prior. The actual weights are given in Table 2.

Run	w_C	w_B	w_L
microblog + blogger-based prior	0.66	0.33	-
microblog + link-based prior	0.75	-	0.25
microblog + both priors	0.50	0.33	0.17

Table 2: Weights for runs, results are shown in Table 3.

We report on accuracy (the percentage of tweets for which the language is identified correctly) for each language, and overall. In total we look at six runs: the out-of-the-box language model, the

[1] http://www.let.rug.nl/~vannoord/TextCat/

[2] http://www.twitter.com

[3] The actual dataset will be made available online

Language		Content of microblog post
Assessed	Classified	
Fluent multilingual posts		
Dutch	Spanish	french viel uit. god loves me. i love god. x
Dutch	English	Sunshine and soul music... Heerlijk.
French	English	What about France Celina? On t'aime!!! :)
French	English	Blagues ta mère: une application surtaxée // Good to know.
Spanish	English	asi tipo emmm happy bday cody! maybe this is not the best present but it's spanish so it rocks! o algo asi xd
Posts containing named entities		
Dutch	English	Moon Patrol op Atari 2600. Uit de oude doos gevist... Beter dan WoW.
French	English	Okay Facebook est devenu un terrain de foot et Twitter un plateau télé gokillyourself 0_0
Spanish	English	He marcado un vídeo como favorito en YouTube. – Friendly Fires - Your Love (EP Version)
Automatically generated posts		
French	English	Le Sacré Coeur la Nuit: ADRIEN has added a photo to the pool: Photoreporter de la mairie de Paris pour la nu
Spanish	English	I uploaded a YouTube video – Centro Quiropractico Nilsson pgm 9 ciatica.divx
Language ambiguous posts		
French	English	♥♥
German	Dutch	Morgen!
German	Dutch	aha. ok. danke :)
Spanish	Dutch	Hoolaaa :)

Table 1: Examples of misclassified tweets, along with the languages assigned, broken down by error type.

microblog language model, the combined language model, the microblog model with each prior separately, and the microblog model with both priors.

5. RESULTS AND ANALYSIS

In Table 3 we present the accuracy of our runs for all languages. The results show that language identification on short posts in microblogs is not as straightforward as it is on longer pieces of text. Training the n-gram-based approach on the target corpus obviously gives much better results, but accuracy is still limited. Incorporating the semi-supervised priors does lead to an increase in accuracy for all languages, and especially the combination of the blogger-based and link-based priors outperforms other approaches.

Run	Dutch	English	French	German	Spanish	Overall
Content-based identification						
Out-of-the-box	90.6%	85.0%	86.0%	93.6%	82.2%	87.5%
Microblog	90.4%	91.6%	92.2%	95.4%	85.2%	91.0%
Combined	92.2%	89.0%	91.6%	92.2%	83.2%	89.6%
Microblog content-based identification + priors						
Blogger-based	**94.6%**	93.8%	**94.8%**	96.4%	84.6%	92.8%
Link-based	92.0%	90.6%	92.6%	92.8%	83.0%	90.2%
Both priors	94.4%	**95.0%**	94.0%	**97.2%**	**85.4%**	**93.2%**

Table 3: Results for baseline content-based identification runs and the combination with the priors.

We notice differences in accuracy between languages: for German, English, French, and Dutch, accuracy is high (although there is room for improvement), for Spanish accuracy is quite low. In the next section we briefly touch on this with some examples of errors made in the identification process.

5.1 Error analysis

In analyzing the posts misclassified by our final classifier using all priors, we group them into four distinct categories: fluent multilingual posts, those containing named entities, automatically generated, and language ambiguous. We give examples in Table 1, and explain each type of error in turn.

Fluent multilingual posts: These are posts which are a grammatical sentence with words written in two or more languages. Usually these take the form of a sentence split into two, with both halves in different languages.

Named entity errors: These posts are misclassified because they contain a reference to a foreign language named entity, such as a company or product name, song title, etc. The named entities contained in the post outweigh the correct language tokens in the post in scoring, leading to the misclassification.

Automatically generated posts: These posts are automatically generated by external applications and software, which insert phrases into the post foreign to the language of the user.

Language ambiguous: These posts are misclassified because they only contain a few tokens which could belong to a number of different languages.

6. CONCLUSION

In this paper we explore the performance of an n-gram-based approach to language identification on microblog posts. Given the short nature of the posts, the rather idiomatic language in these (due to abbreviations, spelling variants, etc.), and mixed language usage, we expect language identification to be a difficult task. To overcome the challenges of microblogs, we introduce two semi-supervised priors: (i) a blogger-based prior, using the previous posts of a blogger, and (ii) a link-based prior, using the pages a post links to. Results show that accuracy for 3 out of 5 languages is the best using both priors, and the remaining 2 languages benefit most from the blogger-based prior alone.

Analysis reveals four main categories of errors: fluent multilingual posts, named entity errors, automatically generated posts, and language ambiguous posts. All of these types of errors could, in principle, be overcome using different relative weighting of the priors to the content-based identification.

Although accuracy for most languages is high, we feel that there is room for improvement. Microblogs (and possibly other social media as well) offer several other priors that we have not yet discussed or explored. Bloggers often write posts in reply to a previous post by another blogger; we can take use the language profile of this other blogger as a prior on the current post, e.g., as a *reply-based prior*. In the current setup we did not use tags attached to posts (besides keeping them for identification purposes); a future direction could involve collecting posts with the same tag, and construct a language profile for this tag. We can then use this score as a *tag-based prior* for language identification. Finally, in our experiments we used fixed weights for combining priors and content-based identification, but we are interested in investigating how weights affect accuracy. We believe weights should be dependent on the individual post: when content-based identification results are close for multiple languages, we might want to lower its weight, and rely more on our priors. Future work aims at finding a proper way of estimating these post-dependent weights.

Acknowledgements

Tsagkias is supported by the Netherlands Organisation for Scientific Research (NWO) under project number 612.061.815.

References

[1] D. L. Altheide. *Qualitative Media Analysis (Qualitative Research Methods).* Sage Pubn Inc, 1996.

[2] A. Babu and P. Kumar. Comparing Neural Network Approach with N-Gram Approach for Text Categorization. *International Journal on Computer Science and Engineering*, 2(1): 80–83, 2010.

[3] W. Cavnar and J. Trenkle. N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[4] G. Golovchinsky and M. Efron. Making sense of twitter search, 2010.

[5] T. Gottron and N. Lipka. A comparison of language identification approaches on short, query-style texts. In *Advances in Information Retrieval, 32nd European Conference on IR Research (ECIR 2010)*, pages 611–614, 2010.

[6] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *J. Am. Soc. Inf. Sci. Technol.*, 60(11):2169–2188, 2009.

[7] B. Martins and M. Silva. Language identification in web pages. In *Proceedings of the 2005 ACM symposium on Applied Computing*, pages 764–768, 2005.

[8] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 851–860, New York, NY, USA, 2010. ACM.

[9] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *International AAAI Conference on Weblogs and Social Media (ICWSM 2010)*, 2010.

[10] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 1079–1088, New York, NY, USA, 2010. ACM.

[11] P. Vojtek and M. Bieliková. Comparing natural language identification methods based on markov processes. In *International Seminar on Computer Treatment of Slavic and East European Languages*, pages 271–282, 2007.

Scope of Negation Detection in Sentiment Analysis

Maral Dadvar
Human Media Interaction Group
University of Twente
Enschede, Netherlands
m.dadvar@ewi.utwente.nl

Claudia Hauff
Human Media Interaction Group
University of Twente
Enschede, Netherlands
c.hauff@ewi.utwente.nl

Franciska de Jong
Human Media Interaction Group
University of Twente
Enschede, Netherlands
f.m.g.dejong@ewi.utwente.nl

ABSTRACT

An important part of information-gathering behaviour has always been to find out what other people think and whether they have favourable (positive) or unfavourable (negative) opinions about the subject. This survey studies the role of negation in an opinion-oriented information-seeking system. We investigate the problem of determining the polarity of sentiments in movie reviews when negation words, such as *not* and *hardly* occur in the sentences. We examine how different negation scopes (window sizes) affect the classification accuracy. We used term frequencies to evaluate the discrimination capacity of our system with different window sizes. The results show that there is no significant difference in classification accuracy when different window sizes have been applied. However, negation detection helped to identify more opinion or sentiment carrying expressions. We conclude that traditional negation detection methods are inadequate for the task of sentiment analysis in this domain and that progress is to be made by exploiting information about how opinions are expressed implicitly.

Categories and Subject Descriptors

H.3.1 [**Information Systems**]: Content Analysis and Indexing – *Linguistic processing.*

General Terms

Reliability, Experimentation, Languages, Human Factors, Information Systems

Keywords

Scope Modelling, Movie Review Analysis, Opinion Mining

1. INTRODUCTION

With the rapid expansion of e-commerce, more products are being sold online. Industry or manufacturing companies that produce these products want to know how their customers feel about them. This information can be acquired by studying opinions from review portals (for example, Amazon and ConsumerReports). At the same time, users or consumers want to know which product to buy or which movie to watch, so they also read reviews and try to make their decisions accordingly. However, gathering all this online information manually is time consuming. Therefore automatic sentiment analysis is important. Sentiment analysis is defined here as the task of identifying the opinions expressed in text and classifying texts accordingly. To do so, the main task is to extract the opinions, facts and sentiments expressed in these reviews. Example applications are, classifying products or reviews into 'recommended' or 'not recommended' [1, 2], opinion summarization [3] and subjectivity classification [1, 4] which is the task of determining whether a sentence or a paragraph contains the opinion of the writer. There are also other applications for sentiment analysis, for example, comparison of products, or general opinions on public policy. Sentiment analysis aims at classifying the sentiment of the opinions into polarity types (the common types are positive and negative). This text classification task is also referred to as polarity classification.

Negation is one of the most common linguistic means that can change text polarity. Therefore in sentiment analysis negation has to be taken into account [5, 6]. The scope size of a negation expression determines which sequence of words in the sentence is affected by negation words, such as, *no, not, never* [6]. Negation terms affect the contextual polarity of words but the presence of a negation word in a sentence does not mean that all of the words conveying sentiments will be inverted [7]. That is why we also have to determine the scope of negation in each sentence. One of the most noticeable works done on examining the affect of different scope models for negation is [7]. Jia et al. have used some linguistic rules to identify the scope of each negation term. The impact of scope modelling for negation applied for sentiment analysis has not been studied a lot compared to domains such as biomedical studies [8-10].

Linguistic negation is a complex topic and there are several forms to express a negative opinion. Negation can be morphological where it is either denoted by a prefix ("dis-", "non-") or a suffix ("-less") [11]. It can be implicit, as in *with this act, it will be his first and last movie.* Although this sentence carries a negative opinion, no negative words are used. Negation can also be explicit, *this is not good*. This last type of negation will be the focus of our experiments. In this paper we studied the effect of scope modelling for negation by comparing the effect of different scope sizes (or window sizes) in the context of sentiment analysis, particularly with respect to sentiments expressed in movie reviews. Scope in negation detection is defined here as the window in which a negation word may affect the other elements of the sentence. We studied how opinions were expressed in each category of reviews and how adjectives and adverbs were used.

This paper is organized as follows; in section 2 the related work on scope detection for negation is introduced. Sections 3 and 4 explain the method and experimental setup. The results and evaluation of the model is presented in section 5 and we round off the paper with the discussion and conclusion in sections 6 and 7.

2. RELATED WORK

Recently [6] did a review on negation and its scope in sentiment analysis. This work presents various computational approaches to modelling negation in sentiment analysis. The focus of this paper is particularly on the scope of negation. It also discusses limits and challenges of negation modelling. For example, recognition of polar expressions (sentences which carry sentiments) is still a challenging task. The authors also discussed that the effectiveness of negation models can change in different corpora because of the specific construction of language in different contexts.

On the effect of negation on sentiment analysis, [7] introduces the concept of the scope of a negation term. The authors employ a decision tree to determine the polarity of the documents. The proposed scope detection method, considers static delimiters (unambiguous words) such as, *because*, dynamic delimiters (ambiguous words) such as, *like,* and heuristic rules which focus on polar expressions. For negation detection they have tried three window sizes; 3, 4 and 5. Their experimental results show that their method outperforms other methods in accuracy of sentiment analysis and the retrieval effectiveness of polarity classification in opinion retrieval. [12] suggests that the scope of negation should be the adjectives close to the negation word. Authors have suggested that the scope of a negation term to be its next 5 words.

In [1] the scope of a negation term is assumed to be the words between the negation term and the first punctuation mark following it. The accuracy of this work is 0.69 based on the previous version (Ver. 0.9) of movie review data. [13] introduces the concept of contextual valence shifters which consist of negation, intensifier and diminisher. Intensifiers and diminishers are terms that change the degree of the expressed sentiments. The sentence, *this movie is* ***very*** *good*, is more positive than *this movie is good.* In the sentence, *this movie is* ***barely*** *any good*, the term *barely* is a diminisher, which makes this statement less positive. They have used a term-counting method, a machine learning method and a combination of both methods on the same data collection as was used in our experiment. They found that combining the two systems slightly improved the results compared to machine learning or term-counting methods alone.

There are other studies on determining the scope of negation mostly in biomedical texts, using machine learning techniques. In recent work by Morante et al. [15], a metalearning approach to processing the scope of negation signals is studied, involving two classification tasks: identifying negation signals and finding the scope, using supervised machine learning methods. They achieved an error reduction of 32.07 %.

3. METHODS

The experiment to determine the sentiments expressed in movie reviews is based on term frequencies. We count the number of occurrences of positive words and negative words in each document. These numbers are compared with each other and the documents are classified accordingly as positive or negative. If the numbers of positive and negative words are equal the document is neutral.

When an explicit negation word occurs in a sentence, it is important to determine the range of words that are affected by this term. The scope may be only the next word after the negation word, for example, *the movie was not interesting* (window size = 1), or a wider range, for instance, *I do not call this film a comedy movie* (window size = 5). In the second sentence the effect of *not* is until the end of the sentence and not only the word following it. A negation does not negate every subsequent word in the sentence. There is no fixed window size. The window can be affected by different combinations of textual features such as adjectives, adverbs, nouns and verbs. When a positive or negative word falls inside the scope of a negation, its original meaning shifts to the opposite one and it is counted as the opposite polarity.

For extracting the opinion words we use the two wordlists. We do not use part of speech tags in our experiment. Considering the word senses given by WordNet[1], it was verified that almost all of the words in the wordlists are adjectives. Few of them belong to other categories (verbs or adverbs) which again only occur in one form, for example verbs such as "adore" and "detest".

Negation terms are not restricted to *not.* The set of negation terms that we have used in this paper also includes *no, not, rather, hardly* and all the verbs that the word *not* can be concatenated to in the form of *n't*.[2]

4. EXPERIMENTAL SETUP

We used the Movie Review data set prepared by [14]. This data set contains 2000 movie reviews: 1000 positive and 1000 negative. These reviews were originally collected from the Internet Movie Database (IMDb) archive [3]. Their classification as positive or negative was automatically extracted from the ratings and will be used as ground truth.

In order to identify the positive and negative terms in the documents we use two wordlists. The positive wordlist[4] consists of 136 words which are used to express positive opinions. For example, "good" is one of the positive words along with its synonyms such as, "fascinating" and "absorbing" which were also added to the list. The negative wordlist[5] contains 109 negative words which are used to express negative opinions (for example "boring" and its synonyms "awful", "dull" and "tedious"). These lists are derived from online dictionaries such as synonyms.com and the words proposed in [1]. Following [1] we also use "?" and "!" as negative words in the wordlist.

Of the total number of the words in the positive list, 20 never occurred in any of the reviews and the rest of the words occurred on average 44 times in the whole corpus. From the negative list, 18 have never occurred in any of the documents and the rest of the words were used 75 times on average in the corpus. Figures 1 and 2 illustrate the frequency of the 30 most repeated positive and negative words in the corpus.

[1] http://wordnet.princeton.edu/ [Accessed 24 October 2010]

[2] List of the negation words is accessible online: http://wwwhome.ewi.utwente.nl/~dadvarm/dir2011/negation.txt

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/ [Accessed 24 October 2010]

[4] The positive words list is accessible online: http://wwwhome.ewi.utwente.nl/~dadvarm/dir2011/positive.txt

[5] The negative words list is accessible online: http://wwwhome.ewi.utwente.nl/~dadvarm/dir2011/negative.txt

Our aim in this work is to examine whether negation detection affects sentiment analysis and improves the classification. Moreover, we evaluated the effect of different window sizes (scope) in negation detection. We started our experiment by classifying the movie reviews, without considering the negation (step 1). In each document, the numbers of occurrences of the wordlists' words were counted. Accordingly the reviews were classified as positive, negative or neutral.

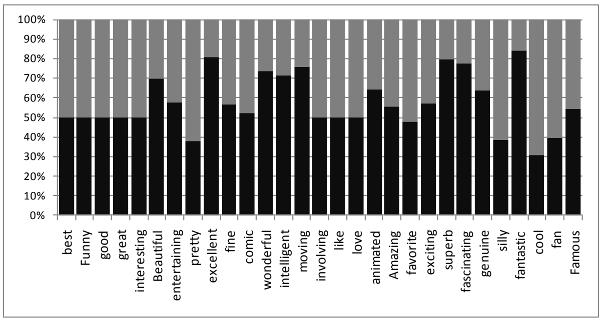


Figure 1. Frequency comparison (%) of the 30 most repeated positive words in positive (black) and negative (grey) documents.

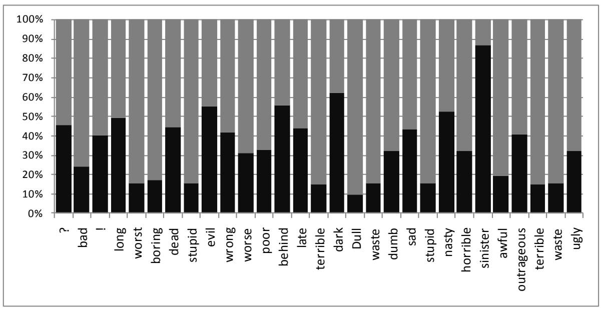


Figure 2. Frequency comparison (%) of the 30 most repeated negative words in positive (black) and negative (grey) documents.

In the second step we employed negation detection, considering only the term *not* as the negation word. We checked how results would change in different window sizes. Then (step 3) we extended our negation words by adding the words *no, rather, hardly*. The verbs which were negated with *n't* were then added to the negation word lists in step 4. We repeated our experiment with different window sizes from 1 up to and including 5.

We used Accuracy as our evaluation metric. We evaluated the classification of our system by comparing it with the Naive baseline which all the documents are classified as positive, i.e., precision 0.5, recall 1, and accuracy 0.5.

5. RESULTS

We repeated the [1] experiment using same word lists and corpus to evaluate our system. Pang et al. have used more limited wordlists (Negatives = 7, Positives = 7). Our results of sentiment analysis (without negation detection) with accuracy of 0.70 comply with the results of [1] with an accuracy of 69%. The overall accuracy of the first step of our experiment (sentiment analysis without negation) was 65%, true positive rate (recall) was 84% and precision was 62%. Table 1 shows the accuracy results after applying negation detection using only *not* as negation word (step 2). Accuracy results of step 3 and step 4 are shown in tables 2. There are no significant differences in the results with different negation words and window sizes. Negation detection in window sizes 4 and 2, and using *no, not, rather, hardly* as the negation words, resulted in more accurate classification. Recall was always higher than precision in all experiments which suggests poor negative review classification.

We also counted the number of adjectives and adverbs in the dataset. There were more adjectives and adverbs in positive documents compared to negative documents. (Table 3)

Table 1. Accuracy results of sentiment analysis (SA) before and after applying negation detection (ND) using only *not* as the negation word in different window sizes (WS)

Experiment	Recall	Precision	Accuracy
SA without ND	0.83	0.62	0.65
WS 5	0.83	0.62	0.65
WS 4	0.83	0.62	0.65
WS 3	0.83	0.62	0.65
WS 2	0.83	0.61	0.65
WS 1	0.83	0.61	0.65

Table 2. Accuracy results after applying negation detection using *no, not, rather, hardly,* and the verbs which were negated with *n't* as the negation words in different window sizes (WS)

Negation words	WS 5	WS 4	WS 3	WS 2	WS 1
not	0.65	0.65	0.65	0.65	0.65
no, not, rather, hardly	0.66	0.70	0.66	0.70	0.65
no, not, rather, hardly and the verbs which were negated with *n't.*	0.67	0.66	0.66	0.66	0.66

Table 3. Mean number of adjectives and adverbs in each review

Dataset	Type	Mean	Std. Dev.
Positives	Adjectives	66.0	31.6
	Adverbs	47.8	26.2
Negatives	Adjectives	57.5	24.6
	Adverbs	44.9	22.6

6. DISCUSSION

We studied the impact of negation detection in sentiment analysis in movie reviews. We tested different negation scopes to investigate how it would affect the polarity identification of the sentences. We hypothesized to observe significant improvements on the classification of the documents after applying negation detection. In our experiment we assumed that opinions are mostly expressed by the use of adjectives and adverbs. Therefore, we classified the reviews as negative or positive according to the number of occurrences of these types of words. After failure analysis, we realized that most of the sentiments and opinions are expressed implicitly, for example, " ... *I have a problem even regarding it as a film, it's more of a show"*.

The negation words that we have used in our experiment, according to grammatical rules should usually be followed by either an adjective or an adverb. Therefore, in our case (adjective and adverbs are not commonly used to express the opinions), negation did not have much influence on the outcome. The majority of reviewers have used sarcasm sentences, comparison and metaphor, for instance the sentences;

> *"now , I saw this scene coming from a mile away , but I said to myself , " that is impossible . there's no way they'll do that . . . oh god ! " they did do it . it's there . "*
> *"Now what didn't work in this movie? would be the rest of it".*

Although we extended the word lists compare to [1], the result of classification did not improve significantly. This can support our claim that since the opinions are mostly expressed indirectly, the number of adjectives does not have much effect on the outcome.

As it is illustrated in the figures 1 and 2, there are also words which are considered to be positive but are equally or even more occurred in the negative documents than the positive ones and vice versa. For example, the word *cool*, which is one of the words from the positive word list, it is more frequently occurred in negative documents than the positive documents. This can also be another reason for misclassifications. A pre-enhancement of the wordlists, considering the language used in the dataset, may also improve the classifications.

Many emotions and opinions are expressed in the form of question or surprise. The results show that "?" and "!" are the most repeated ones in the documents, *!* in negative documents = 527, in positive documents = 352 and *?* in negative documents = 1092, in positive documents = 913. As it was mentioned in [1], negative sentiments are most likely to be expressed by – at least – one of these punctuation marks.

Our results also illustrate higher recall than precision which implies a better discrimination capacity in positive documents (in step 1, TP = 794 vs. TN = 431). A possible reason for higher misclassifications in negative documents can also be the number of adjectives and adverbs. In the positive documents more opinions are conveyed by explicit use of adjectives or adverbs in comparison to the negative documents (Table 3).

More investigation on falsely classified documents revealed that in some cases the negation word appears after the words which convey sentiments. For example, "*sounds great huh? well it's not*", where the adjective *great* is located four words before the negation word *not*.

7. CONCLUSION

We conclude that traditional negation detection methods are inadequate for sentiment analysis in this domain. In addition to the explicit elements, there are other indirect elements that affect the polarity of sentences, either positively or negatively. In some cases the opinion words are used before the negation word, therefore, it might be wise to also take them into account while setting the negation scope. It is important to study which lexical features are mainly used to express the sentiments implicitly. Sarcasm and metaphor detection may also improve the classifications accuracy. We also would like to extend our research by performing more detailed analysis using machine learning approaches.

ACKNOWLEDGMENTS

We thank Thijs Verschoor and Dolf Trieschnigg for their useful advice on programming. Thanks also to Alessandro Valitutti for his helpful comments and technical supports. The language editing was kindly done by Lynn Packwood. We are grateful to four anonymous reviewers for their valuable comments and suggestions. This research is funded by the EU project PuppyIR http://www.puppyir.eu (EU FP7 231507).

REFERENCES

[1] Pang, B., L. Lee, and S. Vaithyanathan. *Thumbs up? Sentiment classification using machinelearning techniques*. in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*. 2002.

[2] Turney, P.D. *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*. in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2002. Philadelphia, USA.

[3] Ku, L., et al. *Major topic detection and its application to opinion summarization*. in *SIGIR '05 Proceedings of the 28th annual international ACM conference on Research and development in information retrieval*. 2005. Salvador, Brazil.

[4] Yu, H. and V. Hatzivassiloglou. *Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences*. in *Conference on Empirical Methods in Natural Language Processing*. 2003: Association for Computational Linguistics.

[5] Jia, L., C. Yu, and W. Meng. *The effect of negation on sentiment analysis and retrieval effectiveness*. 2009: ACM.

[6] Wiegand, M., et al. *A survey on the role of negation in sentiment analysis*. in *'10 Proceedings of the Workshop on Negation and Speculation in Natural Language Processing* 2010: Association for Computational Linguistics.

[7] Jia, L., C. Yu, and W. Meng. *The effect of negation on sentiment analysis and retrieval effectiveness*. in *8th International Conference on Information and Knowledge Management*. 2009. Hong Kong.

[8] Chapman, W., et al., *A simple algorithm for identifying negated findings and diseases in discharge summaries.* Journal of biomedical informatics, 2001. **34**(5): p. 301-310.

[9] Goldin, I. and W. Chapman. *Learning to detect negation with 'not'in medical texts*. in *Workshop at the 26th ACM SIGIR Conference*. 2003.

[10] Morante, R., A. Liekens, and W. Daelemans. *Learning the scope of negation in biomedical texts*. in *EMNLP '08 Proceedings of the Conference on Empirical Methods in Natural Language Processing* 2008.

[11] Councill, I., R. McDonald, and L. Velikovich. *What's great and what's not: learning to classify the scope of negation for improved sentiment analysis*. in *'10 Proceedings of the Workshop on Negation and Speculation in Natural Language Processing* 2010.

[12] Hu, M. and B. Liu. *Mining and summarizing customer reviews*. in *Tenth ACM International Conference on Knowledge Discovery and Data Mining*. 2004. Seattle, WA.

[13] Kennedy, A. and D. Inkpen, *Sentiment classification of movie reviews using contextual valence shifters.* Computational Intelligence, 2006. **22**(2): p. 110-125.

[14] Pang, B. and L. Lee. *A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts*. in *Proceedings of the ACL*. 2004: Association for Computational Linguistics.

[15] Morante, R. and W. Daelemans. A metalearning approach to processing the scope of negation. in Proceedings of the CoNLL. 2009: Association for Computational Linguistics.

A multi-dimensional model for search intent

Max Hinne
Institute for Computing and
Information Sciences (iCIS)
Radboud University Nijmegen
mhinne@sci.ru.nl

Maarten van der Heijden
Institute for Computing and
Information Sciences (iCIS)
Radboud University Nijmegen
m.vanderheijden@cs.ru.nl

Suzan Verberne
Centre for Language and
Speech Technology
Radboud University Nijmegen
s.verberne@cs.ru.nl

Wessel Kraaij
Institute for Computing and
Information Sciences (iCIS)
Radboud University Nijmegen
TNO, Delft
kraaijw@acm.org

ABSTRACT

The interaction of users with search engines is part of goal driven behaviour involving an underlying information need. Information needs range from simple lookups to complex long-term desk studies. This paper proposes a new multi-dimensional model for search intent, which can be used for the description of search sessions. Using examples from a search engine log we show that our model allows a more comprehensive description of information need than existing categorizations.

1. INTRODUCTION AND BACKGROUND

User interaction with search engines is an object of study in different domains of science. This may be the reason that key concepts such as *intent, information need* and *query session* lack a consistent definition in the literature. Many definitions of query sessions have been suggested and explored in the literature [4], It seems well accepted that sessions can consist of multiple queries that are often topically related. Gayo-Avello [2] introduce the term *searching episode* for all queries by a user during a single day, consisting of one or more *search sessions* where the "successive queries are related to a single information need or goal". Session boundaries are usually determined by looking at lexical or temporal cues or a combination of these cues.

Classifications of search patterns that can help to determine session boundaries have been presented in e.g. Lau and Horvitz [5] and He et al. [3]. A key element of the search patterns within a search session is that there is some form of lexical overlap. Queries can be refined by specialization, generalization or reformulation. These refinement classes are examples of what Lau and Horvitz call *user's intents relative to his prior query.* Thus in an IR context, intents could be defined as intermediate goals that are the result of a certain knowledge state, which is the result of the interaction with the search engine so far. Intents represent the (sub)goals motivating the user's search behaviour.

We introduce a multi-dimensional notion of intent, with information need as the driving force behind search behaviour, and search intent as specializations of that force.

Since information need is an abstract concept, it is not necessarily restricted to a specific search session. This aspect is important, since the overall information need is a core part of the context that can help to define the relevance of search results. If a search engine can detect that e.g. a request for booking skiing lessons is connected to a previous search session concerning renting an apartment in a specific ski resort, it would be helpful to rank the pages about ski-schools in the vicinity of this ski-resort higher than pages about other ski-schools.

In this paper, we will show that such a multi-dimensional view on intent can be supported by click data. We propose three facets of search intent, explained in Section 2. We claim that these facets can help to create a more fine grained taxonomy to discuss and analyze search intent. We are also able to relate several existing intent classification schemas (e.g. Broder [1], Lau and Horvitz [5]) to our model (Section 3 and Section 4). Section 5 provides some examples from data followed by some concluding remarks and future work in Section 6.

2. OUR MODEL FOR INFORMATION NEED AND SEARCH INTENT

Following survey studies such as [2] and [7] , we conclude that the concepts *information need* and *search intent* (or *query intent*) are widely used in the literature about user interactions with search engines, but lack a uniform interpretation. Before we discuss extensions to existing classification schemes for search intent, we present our view on the relation between information need and search intent:

At the basis of a user interaction with a search engine lies the *information need.* This can be anything from an abstract, unexpressed need to a clearly formulated request. A complex information need generates one or more *search intents.* A search intent is a clear-cut element of the information need that the user hopes to solve with a well-formulated query. In practice, a search intent leads to the realization of one or more queries; it is possible that a user needs to formulate multiple queries until the local search intent is satisfied. In that case multiple queries are related, motivated by the user's desire to refine a query.

This hierarchical process, starting at an information need

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.
.

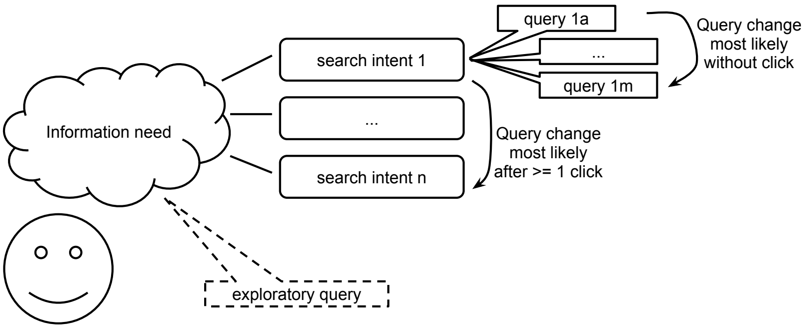


Figure 1: Our model for information need and search intent

and ending with a series of queries, is visualized in Figure 1.[1] The process can be exemplified by the following case: Consider the complex information need "Collecting information about the Dutch prime minister for an essay". It is composed of several search intents: finding out who the Dutch prime minister is, collecting biographical facts about Mark Rutte, finding a good picture, and foraging opinions and media performances related to him. These search intents may require multiple queries to be satisfied, and perhaps the user has to reformulate his queries multiple times in order to obtain a useful result.

3. CLASSIFICATIONS OF SEARCH INTENT

The search intents generated by an information need are traditionally classified according to the *actions* the user wants to execute with the results. These can be *informational, transactional* or *navigational* [1]. We argue that although this classification is sound, it is not complete.[2] It forms one dimension of the three-dimensional classification of search intent that we propose in this section.

Sushmita et al. [8] propose that search intents should be classified by the requested *form* of the results. For example, a search may be aimed at retrieving pictures, maps, videos or Wikipedia entries. They refer to this aspect of the search intent as a combination of *query domain* and *query genre*. We will instead use the term *mode* to refer to this second dimension of the search intent. The user's choice along this dimension is sometimes made explicit in the query, by adding terms such as "pictures" or "movies".

The third dimension that characterizes the search intent is its *topic*. This is most strongly connected to the textual realization of the query: the query "Mark Rutte" is a request for items 'about' Mark Rutte. Within one session of interaction with a search engine, the user may consider multiple topics, that each relate to a series of queries.

In most papers addressing information need, queries are classified according to the search intent that generated them, using the navigational-transactional-informational scheme. We propose to extend this scheme to a three-dimensional classification, of which the axes are *action*, *mode* and *topic*. In the remainder of this paper, we investigate the relevance and applicability of these dimensions by considering series of queries in search engine interactions. We use search engine log data for this purpose, the Microsoft "Accelerating Search in Academic Research Spring 2006 Data Asset", which contains one month of MS search queries from the spring of 2006 together with the URLs clicked, a timestamp and a session identifier. Because of privacy concerns, session lengths have been cut-off at 30 minutes.

4. CLASSIFICATION OF QUERY TRANSITIONS

The usefulness of the additional dimensions for query classification become apparent when we consider the transitions between different queries. In the three-dimensional model, we expect a new query within a user session to change on one or more axes of the model. Therefore, in this section, we study the *transitions* of one query to another within one session and try to classify these transitions according to the multi-dimensional model of search intent proposed above. From our hierarchical model of information need, it follows that there are three levels on which a query transition can take place:

1. Starting to work on a new information need.
2. Introducing a new search intent within the same information need.
3. A query reformulation (correction) for the same search intent.

When a user moves from one search intent to another, then he will reformulate the query along one of the three axes of search intent *action*, *mode* or *topic*. In other words, the change of intent is realized as a query transition. Here, the query transition categorization as proposed by Lau and Horvitz [5] can play a role. Lau and Horvitz classify query transitions according to the change in surface form (textual content) of the query, labelled as generalization, specialization, reformulation etc.

The change in surface form does not have a direct link to the change in search intent, but categorizing the query

[1] If the user has an information need that he is not able to directly express in the form of a clear search intent – what Taylor refers to as the *visceral* information need [9] – the user may generate an exploratory query. The results that are presented to the user help him in formulating his search intent.

[2] In addition, navigational search intent seems are more aimed at bypassing a browser's address bar than to actually find information, but that is not an issue that we address in the current paper.

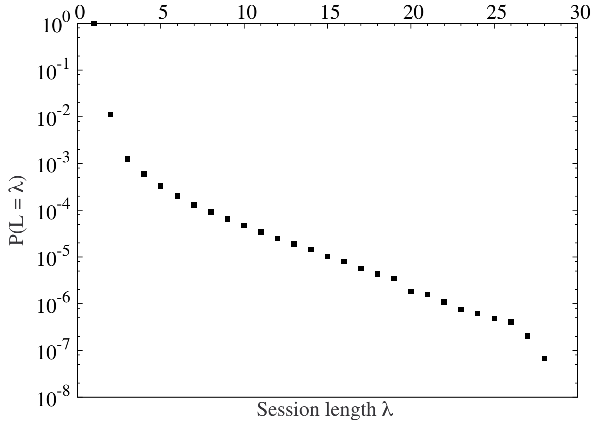


Figure 2: The distribution of session lengths in the MSN query data set as the probability for a session to contain λ clicks.

transitions may be helpful for understanding the changing intent.

We analyzed query transition behaviour with the aid of the Microsoft search log. The distribution of session length (measured in the number of clicks per session, see Figure 2) shows that 1.4% of sessions contain more than one click.[3] We implemented an automatic classification of query transitions for the MSN click data, using the following heuristics for transition types based on [5]:

- Request for additional results: query Q_{i-1} is equal to Q_i (the query is not necessarily reissued, multiple clicks for a single query show up the same way in the query log).
- Generalization: query Q_i is a substring of Q_{i-1}. E.g. "Mark Rutte prime minister", followed by "Mark Rutte".
- Specialization: query Q_{i-1} is a substring of Q_i. I.e. "Mark Rutte", followed by "Mark Rutte prime minister".
- Reformulation: query Q_i has at least one word in common with Q_{i-1} without the transition being generalization or specialization. E.g. "Mark Rutte Netherlands" followed by "Mark Rutte pictures".
- New topic: query Q_i has no words in common with Q_{i-1}.

These heuristics are oversimplified as a model for query transition because they consider queries as sequences of words that are compared literally. As a result, coincidental word overlap between queries Q_{i-1} and Q_i (such as repeating the word 'the') is categorized as a reformulation instead of a new topic. And two queries that are very similar in meaning but use different wordings (e.g. when 'pictures' is changed to 'photos') are categorized as a change to a new topic. A better implementation of the query transition categorization would be to take into account semantic relatedness between two queries. We will implement this in the near future with the use of the WordNet Relatedness tool [6].

Table 1: 2 Million queries from the MSN click data set automatically classified into the query transition classification by [5].

Number of queries	2000000	100%
Number of sessions	1008656	
Number of follow-up queries	991344	
New topic	1339910	67.0%
New topic in same session	331254	16.6%
Request for additional results	270958	13.5%
Reformulation	247033	12.4%
Specialization	98585	4.9%
Generalization	43514	2.2%

We applied the heuristics-based classification of query transitions to the MSN click data set. In this way, all queries in a session are automatically annotated with transition information. The counts over 2 Million queries are shown in Table 1. The transition types do not explicitly inform us on the user's search intent. We argue that our suggested multi-dimensional search intent model can aid in explaining the different query transitions within a session in terms of query intent. In the next section, we use a number of examples from the click data to manually classify query transitions along the axes of our model.

5. EXAMPLES FROM CLICK DATA

We manually analyzed a number of the annotated sessions in order to gain insight in the type of transitions occurring in the data and how they relate to presumed search intents. Table 2 shows two example sessions from the click data, automatically annotated with transition information. Before analyzing this sequence of queries, we should note that since this is a retrospective analysis, the actual intents are unknown, and the analysis just shows how our model can be applied to user behaviour data.

The first three queries (0, 1, 2) in the first session seem to be informational queries about specific event locations, presumably known to the searcher (a manual check shows that query 0 leads to a restaurant chain and 1 and 2 to venues that advertise themselves as wedding reception locations). Then with query 3 the search intent seems to change, asking about wedding reception locations in a particular town in Texas, followed by a generalization in query 4. This query could be interpreted as a request for a different *mode*, i.e. a map of Seguin. Query number 5 seems to be a reformulation continuing the informational intent of query 3. Query 6, although still topically related, deals with a different facet of the information need, specifically the average cost of a wedding. After apparently finding such an estimate, the new search intent in query 7 includes the modifier 'cheap'. The last query is a specialization to the specific location 'Austin'.

Thus, the overall information need of this session seems to be about planning a wedding reception, with search intents changing to reflect different topical aspects (location and cost); different modes (information and maps); and possibly once a satisfactory location is found, the action intent might change from informational to transactional. This example thus shows that our model at least allows for a more fine grained analysis of search intents: subsequent queries can belong to different search intents while having the same underlying information need.

[3] This number is quite low. It may partly be caused by the artificial cut-off of search sessions.

Table 2: Example sessions from click data, automatically annotated with transition information according to the model by Lau and Horvitz [5].

0:The Salt Lick	New topic
1:Texas Old Town Kyle , TX	New topic in same session
2:Old Glory Ranch	Reformulation of query 1 (words overlapping: Old)
3:Seguin wedding receptions	New topic in same session
4:Seguin, TX	Generalization of query 3 (words overlapping: Seguin)
5:Reception Site in Seguin, Texas	Specialization of query 4 (words overlapping: Seguin)
6:Average Cost of a wedding with 150 guests	Specialization of query 3 (words overlapping: wedding)
7:Cheap Texas Weddings	Specialization of query 5 (words overlapping: Texas)
8:Austin, Texas Wedding sites	Specialization of query 7 (words overlapping: Texas Wedding)

0:ceramic paint	New topic
1:color chart	New topic in same session
2:paint color chart	Specialization of query 1
3:paint color chart	Request for additional results (same as query 2)
4:ceiling paint that will not allow water spots	Reformulation of query 3 (words overlapping: paint)
5:ceiling problems	Reformulation of query 4 (words overlapping: ceiling)
6:water repellant ceiling	Reformulation of query 5 (words overlapping: ceiling)
7:no water stane ceiling	Reformulation of query 6 (words overlapping: water ceiling)
8:no water stain ceiling	Reformulation of query 7 (words overlapping: no water ceiling)

Let us consider an additional example, shown in the bottom half of Table 2, to gain some feeling for the classifications that our model allows. Query 0 introduces a topic as start of the session, with a query that appears *informational* and given the usual mode of internet search, *textual*. Then, with query 1 a transition is made not only in the *topic* dimension (from ‘paint’ to ‘color’) but also in the *mode* dimension, as a chart is requested. Queries 2 and 3 combine the first topics. A slight topic shift is introduced in query 4, which gives a specialization of what the paint is needed for, followed by a number of reformulations that appear to be aimed at satisfying the same search intent on water stains (one of which is just correcting a spelling error).

Again we see that a session of queries has a single information need, that is, finding information about paint that can cover water stains. Although all queries can be called informational, the topics do change from looking for ceramic paint, to colours and to paint specifically well suited to cover water stains. Queries 1–3 also clearly request a mode of information that is different from text, which we would be unable to express in Broder’s classification of intents.

6. CONCLUSION AND FUTURE WORK

In this paper we proposed a multi-dimensional model for search intent. It combines three classification schemes that form its axes, viz.: the *topic* of the query; the *action* that the search results should aid in and the *mode* in which the search results are expected. A change in search intent leads to a change in query text. As a result, the changes in query texts can provide information on how the search intent of the user changed. We automatically annotated 2 Million queries from an MSN click data set with query transition classifications. We performed a manual analysis on examples of annotated sessions, showing how our model can be used to describe user search behaviour.

The added complexity of the model makes it better suited to model real data. On the down side, however, the complexity of the model makes validation more difficult. It is hard to recover what a user’s search intent was, based on nothing more than the click data. As a consequence there is currently no hard validation that our model indeed captures the necessary aspects of information need and search intent. However we do believe that our more fine grained approach is valuable in understanding user queries.

In future research we will (1) make the query transition classification more informative by taking into account the semantic relatedness between subsequent queries; (2) investigate human agreement on the classification of query transitions into search intent (human agreement is a good proxy for the complexity of the problem for automatic analysis); (3) conduct a user study in which we will ask search engine user to categorize their queries in retrospect. We expect that this will provide insights in the structure search sessions and the several types of query reformulations in relation to the underlying intents.

7. REFERENCES

[1] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[2] D. Gayo-Avello. A survey on session detection methods in query logs and a proposal for future evaluation. *Information Sciences*, 179:1822–1843, 2009.

[3] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38:727–742, 2002.

[4] B. Jansen, A. Spink, C. Blakely, and S. Koshman. Defining a session on Web search engines. *Journal of the American Society for Information Science and Technology*, 58(6):862–871, 2007.

[5] T. Lau and E. Horvitz. Patterns of search: analyzing and modeling Web query refinement. In *UM ’99: Proceedings of the seventh international conference on User modeling*, pages 119–128, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[6] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet::Similarity — Measuring the Relatedness of Concepts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1024–1025. AAAI Press, 2004.

[7] F. Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, 2010.

[8] S. Sushmita, B. Piwowarski, and M. Lalmas. Dynamics of genre and domain intents. In *Proceedings of The Sixth Asia Information Retrieval Society Conference*, 2010.

[9] R. S. Taylor. The process of asking questions. *Amer. Doc.*, 13(4):391–396, 1962.

Improving Result Diversity using Probabilistic Latent Semantic Analysis

Peter Lubell-Doughtie
University of Amsterdam
Amsterdam, Netherlands
lubell@science.uva.nl

Katja Hofmann
University of Amsterdam
Amsterdam, Netherlands
khofmann@uva.nl

ABSTRACT

IA-SELECT is a recently developed algorithm for increasing the diversity of a search result set by reordering an original document list based on manually generated clusters. In this paper we extend this approach to create a diversification framework in which arbitrary clustering methods can be used, and where the influence of clusters can be balanced against the original rank of documents. We study whether clusters that are automatically generated using probabilistic latent semantic analysis (PLSA) can compete with manually created clusters, and investigate how balancing the influence of clusters and original document rank affects diversity scores. As there are currently few datasets for evaluating diversity, we develop a new dataset, which is released with this paper. Our results show that diversification using PLSA can improve diversity, but that there is a large gap in performance between automatically and manually created clusters.

Keywords

result diversification, latent semantic indexing, clustering

1. INTRODUCTION

As search engines struggle to return well-ranked and relevant information for ambiguous queries from large and growing sets of documents, interest in improving accuracy through alternative methods has increased [7]. One approach is to ensure that documents representing multiple topics, or aspects of a query are highly ranked, by reducing redundancy within the same topics. This can be measured by the *diversity* of a set of search results, which reflects that set's coverage of multiple interpretations of a query.

We present a result diversification system that extends the recently developed IA-SELECT method [1]. IA-SELECT reorders documents based on manually created clusters reflecting different interpretations of a query. We create a diversification framework which can use arbitrary clustering methods, and where the influence of clusters is balanced against the original rank of documents.

We assume that the interpretations of a query contain documents that are conditionally independent given this interpretation, and can therefore be represented by a mixture of conditionally independent clusters. Additionally, in order to cope with ambiguity in query term meaning, we desire a model that can represent polysemy. These conditions justify our use of a conditionally independent latent class model, such as PLSA.

Because there are currently few datasets for evaluating diversification approaches, we contribute a new small scale dataset to complement the TREC ClueWeb09 dataset[1]. It is created from a question answering corpus in which ideal clusters are given by human judges. We are releasing this dataset with our paper.[2]

Our results show that while diversification using PLSA can improve diversity, there is a significant gap between the performance of automatically and manually created clusters. The best diversity scores were achieved with a non-linear function that weights a document's original rank higher for highly ranked documents and places more importance on cluster structure at lower ranks.

2. RELATED WORK

Our result diversification system is a continuation of related research focusing on measuring the diversity of a list of search results and designing algorithms that optimize result order to increase diversity. The earliest diversity metric and algorithm formally explored is maximum marginal relevance (MMR), which maximizes a linear interpolation of the similarity between each document and the query, minus the similarity between that document and previously returned documents [2]. In [8] the authors apply MMR to subtopic retrieval and find that gains obtained by increasing the rank of novel documents are offset by the cost of increasing the rank of non-relevant documents, as is confirmed in our experiments. Both the original [2] and modified [8] MMR do not directly measure subtopic coverage and assume document novelty is independent from document relevance.

Clarke et al. [4] address the shortcomings of MMR by explicitly measuring subtopic retrieval . The summarization and question answering community defines *information nuggets* (or nuggets) as representations of facts, topicality, or any binary property of a document or information need. Clarke et al. assign nuggets to the query and its returned documents and define the probability that a document is relevant based on the intersection of its nuggets and the query's nuggets. Based on nDCG, the authors define α-nDCG, which rewards novelty through a gain vector accounting for the relevant nuggets within a document.

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.
.

[1] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[2] http://code.helioid.com/diversity/ and http://ilps.science.uva.nl/webclef_diversity

α-nDCG unrealistically assumes all nuggets are equally relevant. Agrawal et al. [1] address this by defining *intent aware* (IA) metrics, which sum evaluation scores over categories, weighted by the probability of a category given a query. Categories are defined as locations in a taxonomy of information and user intents, for our purposes they can be seen as equivalent to nuggets. Agrawal et al. also present the IA-SELECT algorithm, which reorders results to maximize the likelihood that the top k results will covers all the query's categories relative to their likelihood.

Dou et al. [5] present a more general algorithm which combines various subtopic indicators and further improves diversity scores. It is possible that greedy strategies exclude a relevant but rare nugget which co-occurs only in documents containing other nuggets already returned. To address this [3] assumes one "correct" interpretation of a query and returns documents covering all its *facets* (which are defined similarly to nuggets or categories).

3. APPROACH

Our diversification system performs three steps: (i) retrieval, (ii) clustering, and (iii) reordering. We use clusters generated from an initial ranked document list to ensure documents from different clusters are highly ranked and a single cluster is not overrepresented. A function of rank and cluster membership likelihood balances the importance of rank and cluster diversity. Clusters represent query interpretations and diversifying over clusters will diversify over interpretations.

We use a modified version of IA-SELECT to reorder documents [1]. Given the query q, a category (nugget) c, and a document d, IA-SELECT builds a set of documents, labeled S, which maximizes utility. Using a measure of document quality, $V(d|q,c)$, and the conditional distribution over categories for the query and current S, $U(c|q,S)$, the algorithm calculates utility as:

$$g(d|q,c,S) = \sum_{c \in C(d)} U(c|q,S)V(d|q,c) \qquad (1)$$

where $C(d)$ is the set of categories for document d. $U(c|q,S)$ is initialized as $P(c|q)$, defined below, and then at each iteration the algorithm adds the d with greatest utility and updates $U(c|q,S)$. This allows us to approximate the S which maximizes utility.

Given a query, in step (i) our implementation uses the successful BM25 formula to create an ordered set of documents. To model each cluster as a possible interpretation of the query, we assume clusters are independent and that documents can belong to an arbitrary number of clusters. This motivates using PLSA to assign cluster membership probabilities to the top k documents in step (ii).

Using the cluster probabilities for the top k documents we calculate our initial conditional category distribution as:

$$P(c|q) = \sum_{d \in D} p(c|d)^{\phi_p(\text{rank}(q,d))}, \qquad (2)$$

where $p(c|d)$ is the probability that document d is a member of cluster c, rank(q,d) returns the rank of d for q, and ϕ_p determines the importance rank plays in calculating cluster to query relevance. Document quality is similarly calculated as:

$$V(d|q,c) = p(c|d)^{\phi_v(\text{rank}(q,d))}, \qquad (3)$$

where ϕ_v is the importance of rank in calculating relevance. When ϕ_p and ϕ_v are constant rank is irrelevant, otherwise the greater their convexity the greater the influence of rank.

Using these definitions of document quality and conditional category distribution we perform step (iii) and reorder the top k documents. As opposed to IA-SELECT, we explicitly define the value and conditional category distributions in terms of rank and cluster membership probability. This allows us to adjust their influence to benefit the system's goals, in our case, increased diversity scores.

4. EXPERIMENTS AND RESULTS

There are few standard information retrieval evaluation sets that can be used to evaluate diversification because most do not define ground truth categories for documents. In addition, many evaluation sets used in diversity research are either proprietary and unreleased, or are incompletely evaluated versions of question answering corpuses, and can be used only after preprocessing and result extrapolation. The recent ClueWeb09 dataset provides a large diversification task. To complement this, we develop a smaller scale dataset based on the WebCLEF 2007 question answering corpus [6].

In this corpus, information nuggets are assigned to each document and defined by a set of passages taken directly from the document text. To convert this dataset into a retrieval task with subtopics we parse the assessments file, letting the topic of each question form the query and the answer nuggets form the query's subtopics. We then search for the nuggets in the corpus' documents to generate a subtopic document list.

We run experiments with the following settings. The baseline is generated by retrieving the top 200 documents using BM25 with $k_1 = 1$ and $b = 0.3$.

To test the influence of induced clusters, we apply PLSA to the top 20 and 200 returned results to create 20 subtopics, and then reorder with $\phi_p(x) = 1 + \log(x)$ and $\phi_v(x) = x^2$. After experimenting with different functions we found that these produce the best results by appropriately weighting the influence of rank and cluster membership. We evaluate our system using α-nDCG and P-IA, following [1, 4].

The results of our experiments are shown in Table 1. We see that reordering based on 20 documents has the best performance for α-nDCG@{5,10,20} with scores of 0.151, 0.157, and 0.180 respectively. It is unable to beat the baseline for P-IA@10 but does so for P-IA@{5,20} with scores of 0.055 and 0.049 respectively, where the P-IA@20 score is significant at the 0.001 level using a paired student's t-test. Reordering based on 200 documents has the worst performance on all metrics.

Increasing the influence of rank by increasing the convexity of ϕ_v increases diversity scores up to a point. In addition to the results displayed in Table 1, we tested $\phi_v(x) = \{1, 1 + \log(x), x, x^2, x^3\}$. Excluding P-IA@10, $\phi_v(x) = x^2$ produces the best scores. Ignoring rank, with a constant $\phi_v(x) = 1$, produces the lowest scores in all runs except P-IA@20. Up to and including $\phi_v(x) = x^2$, α-nDCG scores increase as function convexity increases, but further increasing convexity decreases scores.

To determine how reordering with 20 results is able to improve on the baseline scores, we plot the α-nDCG@5 scores per query in Fig. 1. Considering individual queries, the reordered list produces better results by matching or im-

Table 1: Diversity scores for all diversification systems. Significant differences from the baseline are marked with $^{\triangledown}$(decrease, $p = 0.01$) and $^{\vartriangle}$(improvement).

Experiment	α-nDCG@5	α-nDCG@10	α-nDCG@20	P-IA@5	P-IA@10	P-IA@20
Baseline	0.145	0.155	0.175	0.051	**0.046**	0.031
PLSA 20	**0.151**	**0.157**	**0.180**	**0.055**	0.044	**0.049**$^{\vartriangle}$
PLSA 200	0.136	0.152$^{\triangledown}$	0.173	0.049	0.040	0.032
QRELS 20	0.324	0.305	0.287	0.080	0.050	0.033
QRELS 200	0.611	0.632	0.621	0.134	0.102	0.079

proving over the baseline on most queries and substantially beating the baseline on a few queries (2 and 11). The algorithm takes a conservative approach and maximizes utility by reordering in cases where both document rank and subtopic relevance are high.

In order to measure the effect of the unequal distribution of subtopics per query, Fig. 1 also plots the number of subtopics in each query. The correlation between reordering performance and the number of subtopics is low, with a Pearson correlation coefficient between the number of subtopics and the baseline, rank2, and constant runs of 0.08, 0.03, and -0.17 respectively. This indicates that performance is not directly related to the number of subtopics in a query.

The IA measures have an undefined upper bound which is less than 1 unless there is a single perfect ordering for all subtopics [1]. In addition, if the best ordering relies on a document outside the top k reordered documents it will be impossible to achieve the maximum score. To estimate this upper bound we reorder based on the ground truth subtopics and assignments in the dataset. The results are labeled as QRELS and shown in the lower part of Table 1. Except for P-IA@20 these scores are substantially higher than either the baseline scores or those achieved when reordering using PLSA clusters. In this case, ignoring rank with constant $\phi_v(x) = 1$ gives the best scores and increasing the influence of rank decreases scores, the opposite of what occurs when using induced subtopics. This is expected if subtopics are more relevant to improving diversity than rank, and provides anecdotal evidence that these estimates may form a reasonable upper bound.

5. DISCUSSION

In our diversification system, changes in the importance of diversity are expressed by changing the influence of rank through the ϕ_v function. We would expect the influence of document rank to be inversely correlated with the diversity of reordered results. However, this is not strictly the case as we achieve maximum diversity scores by balancing the influence of rank and relevance.

In an approach based on reordering results according to subtopics, the effectiveness of the system depends on generating subtopics aligned with those used by the scoring function. Putting significant emphasis on a document's rank appears to be successful primarily due to the poor quality of induced clusters. Experiments generating an upper bound demonstrate that increasing cluster quality and decreasing rank's influence correlate with higher diversity scores.

This is exemplified by results for the query "plastic tableware and the environment" (topic 26), in which the PLSA and QRELS orderings disagree for the second document returned. QRELS returns a document about restrictions on plastic products, fitting the nugget "restricted use of disposable plastic," while PLSA returns a document listing various plastic products for sale, including biodegradable products. Although the document returned by PLSA does not fit any given subtopics it could arguably fit an appropriate subtopic, such as "environmentally friendly plastic products." Here the diversity score is decreased by an understanding of the query's subtopics that is discordant with the subtopics used in evaluation, although not necessarily incorrect.

Concerning the diversification algorithm, other variations in the influence of rank, i.e. $\phi_v(x) = x^n$ for $1 < n < 3$, may further improve scores. That increasing n — the influence of rank — eventually leads to decreasing scores shows that clusters provide valuable information about how to best reorder documents. A more effective strategy would bias towards the original ranking when doing so benefits diversity scores and away when it does not. Figure 1 presents the α-nDCG@5 score per query using a method that heavily weights rank, *rank2* with $\phi_v(x) = x^2$, and a method that ignores rank, *constant* with $\phi_v(x) = 1$. Although the overall score of constant is much lower, on certain queries (2 and 16) it significantly outperforms the baseline and rank2. We suspect that constant performs well on these queries because the clusters generated for them closely match the known clusters.

In our PLSA 20 experiments, which reorder 20 documents using IA-SELECT with 20 PLSA topics, our implementation may reduce to a maximum likelihood estimator by assigning one document to each class, leading to equivalent class and document language models. Work remains to be done in testing that one document is indeed assigned to each class. In this case our implementation would be very similar to the original MMR algorithm and future work could investigate this connection and its implications for the usefulness of PLSA in search result diversification.

6. CONCLUSIONS AND FUTURE WORK

The expansion of online documents and users has led to increases in the number of documents a query is applicable to and in the number of users using the same or similar queries to express different information needs. This, in turn, has led to an increase the number of valid yet differing ways in which we can interpret queries. A complementary challenge arises when different queries express similar information needs. This has also been exacerbated by increasing numbers of documents and users. Search result diversification methods address these challenges by satisfying users' multifarious needs. Diversity research has moved beyond independent analysis of document novelty and relevance (as in MMR) to measuring a document's contribution in relation to the additional information it provides.

In this paper we have shown that using PLSA to create

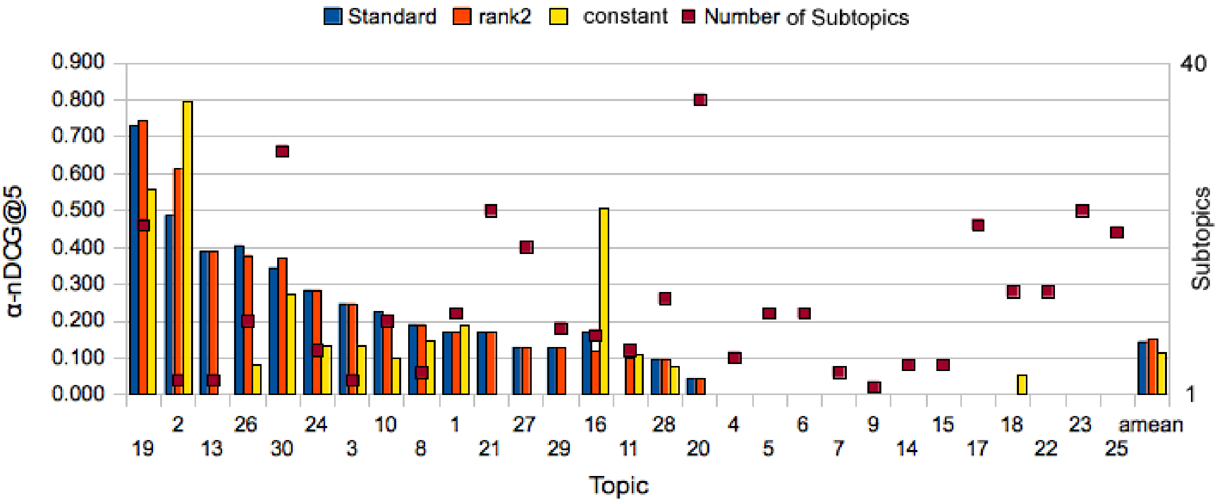


Figure 1: α-nDCG@5 per query for 20 documents, the final column is the arithmetic mean. Topics are ordered by decreasing score for rank2 ($\phi_v(x) = x^2$).

an external partition for reordering search results can improve diversity. The functioning of the reordering algorithm is sensitive to, and can be tuned through, changes in the influence of a document's original rank. Decreasing the influence of rank puts more trust in the accuracy of induced clusters and vice versa.

Future work includes inducing clusters with alternative algorithms and adapting to specific queries. In our experiments we use a fixed number of clusters. This could be improved by changing the number of clusters relative to vocabulary cardinality or other heuristics. In addition, our results show that some queries greatly benefit from diversification while for others the original ranking performs better. Diversification could be applied selectively, for example based on measures developed for query performance prediction or topic models.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM '09*, pages 5–14, 2009.

[2] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, pages 335–336, 1998.

[3] B. Carterette and P. Chandar. Probabilistic Models of Novel Document Rankings for Faceted Topic Retrieval. *CIKM '09*, pages 1287–1296, 2009.

[4] C. Clarke, M. Kolla, G. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR '08*, pages 659–666, 2008.

[5] Z. Dou, K. Chen, R. Song, Y. Ma, S. Shi, and J. Wen. Microsoft Research Asia at the Web Track of TREC 2009. In *TREC '09*. NIST, 2009.

[6] V. Jijkoun and M. de Rijke. Overview of webclef 2007. *Advances in Multilingual and Multimodal Information Retrieval*, pages 725–731, 2008.

[7] M. Sanderson. Ambiguous queries: test collections need more sense. In *SIGIR '08*, pages 499–506, 2008.

[8] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03*, pages 10–17, 2003.

Query Load Balancing by Caching Search Results in Peer-to-Peer Information Retrieval Networks

Almer S. Tigelaar
a.s.tigelaar@cs.utwente.nl
Database Group, University of Twente

Djoerd Hiemstra
hiemstra@cs.utwente.nl
Database Group, University of Twente

ABSTRACT

For peer-to-peer web search engines it is important to keep the delay between receiving a query and providing search results within an acceptable range for the end user. How to achieve this remains an open challenge. One way to reduce delays is by caching search results for queries and allowing peers to access each others cache. In this paper we explore the limitations of search result caching in large-scale peer-to-peer information retrieval networks by simulating such networks with increasing levels of realism. We find that cache hit ratios of at least thirty-three percent are attainable.

Keywords

distributed query processing, peer-to-peer simulation.

1. INTRODUCTION

In peer-to-peer information retrieval a network of peers provide a search service collaboratively. We define a peer as a computer system connected to the Internet. The term peer refers to the fact that in a peer-to-peer system all peers are considered equal and can both supply and consume resources. In a peer-to-peer network each additional peer adds extra processing capacity and bandwidth in contrast with typical client/server search systems where each additional client puts extra strain on the search server. When such a peer-to-peer network has good load balancing properties it can scale up to handle millions of simultaneous peers. However, the performance of such a network is strongly affected by how well it can deal with the constant and rapid joining and departing of peers which is called *churn.*

We study peer-to-peer information retrieval systems where the collection is split over the peers. Each peer contains a subset of all the documents in the collection, and thus also contains a partial index. Since presumably relevant search results can be located at any peer in the network it is often difficult to route a query to the right peer. This problem is commonly approached by using different network topologies and replication of index data. Indeed, query routing is a difficult problem in peer-to-peer information retrieval [4].

In this paper we explore search result caching as a technique that can be used to both perform load balancing and increase the availability of search results. Instead of forward push-based replication of an index, we use a pull-based caching approach [1]. We experiment with fifty times more peers than any existing scientific peer-to-peer experiments we know of.

We define the following research questions:

1. What fraction of queries can be potentially answered from a cache?

2. How can the cache hit distribution in a peer-to-peer network be characterised?

3. How does churn affect caching?

2. RELATED WORK

Markatos [5] analyses the effectiveness of caching search results for a centralised web search engine combined with a caching web accelerator. Their experiments suggest that one out of three queries submitted has already been submitted previously. They conclude that cache hit ratio's between 25 to 75 percent are possible.

Skobeltsyn and Aberer [7] investigate how search result caching can be used in a peer-to-peer information retrieval network. When a peer issues a query it first looks in a distributed meta-index, kept in a distributed hash table, to see if there are peers with cached results for this query. If so, the results are obtained from one of those peers, but if no cached results exist, the query is broadcast through the entire network. The costs of this fallback are $O(n)$ for a network of n peers. The authors further try to increase the performance of their system by using query subsumption: obtaining search results for subsets of the terms of the full query. They show that with subsumption cache hit rates of 98 percent are possible as opposed to 82 percent without. Interestingly, only 18 percent of the queries in the query log they use appear only once. Perhaps this is because their log is a Wikipedia trace as this is inconsistent with our findings.

3. EXPERIMENTS

3.1 Introduction

Our experiments are intended to give insight into the *maximum benefits* that can be gained by caching. Each experiment has been repeated five times, averages are reported,

Copyright is held by the author/owner(s)
DIR'2011, February 4, 2011, Amsterdam.
Copyright © 2011 Almer S. Tigelaar.

Table 1: Query log statistics.

Queries (incl. duplicates)	21,082,980
Users	651,647

no differences between runs were observed that exceeded 0.5 percent. We assume that there are three types of peers: *supplier peers* that have their own locally searchable index, *consumer peers* that issue queries to the network, and *mixed peers* that have both an index and issue queries. We further assume that all peers in our network are willing to cooperate by caching search results. For query routing we introduce a party called the *tracker* which keeps track of which peer can answer what query. The usage of a tracker is inspired by BitTorrent [3]. However, in BitTorrent the tracker is used for locating a specific file: *exact search*, and not for searching to obtain a list of peers which have presumably relevant search results: *approximate search*. In reality the tracker can be implemented in various ways: as a central machine, as a group of high capacity machines in the network, as a distributed hash table or by fully replicating a global data index over all peers. In our experiments we make two important assumptions: firstly, that caches are unbounded in size, and secondly that cached results retain their validity: they need not be invalidated. When dropping either of these two assumptions, caching would become less effective.

3.2 Collection

To simulate a network of peers posing queries we use a large search engine query log [6]. This log consists of over twenty million queries of users recorded over a period of three months. Each unique user in the log is a distinct peer in our experiment. We made several adjustments to it to make our simulations more realistic. Firstly, some queries are censored and appear in the log as a single dash [2]: these were removed. Secondly, we removed results by one user in the log that poses an unusually high number of queries: likely some type of proxy.

Furthermore, we assume that a search session lasts at most one hour. If the exact same query is recorded multiple times in this time window, they are assumed to be requests for subsequent search result pages and thus we use it only once in the simulation. Table 1 shows statistics regarding the log. While the log is sorted by numeric user identifier, for realistic simulation we play back the log in chronological order. We noticed that one day in the log, May 17th 2006, is truncated and does not contain data for the full day, but only for about half an hour after midnight. This has consequences for one of our experiments described later. For clarity: we do not use real search results for the queries in the log. In our experiments we make the assumption that specific subsets of peers have search results and obtain experimental results by counting hits only.

3.3 Centralised

Let us first consider the case where one supplier peer in the system is the only peer that can provide search results. This peer does not pose queries itself. This scenario provides a baseline which resembles a centralised search system. Calculating the query load on the peer-to-peer network is trivial in this case: all 21 million queries *have to be* answered by this single central supplier peer.

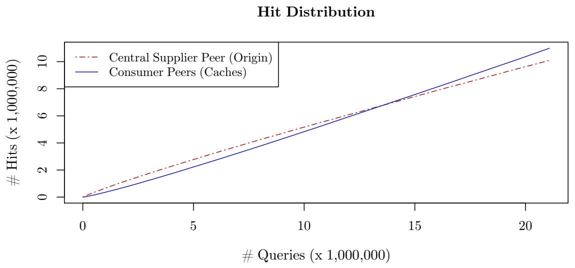


Figure 1: Distribution of hits when peers perform result caching (N=651,647 peers).

However, what if the search results provided by the central supplier peer can be cached by the consuming peers? In this scenario the tracker makes the assumption that all queries can initially be answered by the central supplier peer. However, when a consuming peer asks the tracker for advice for a particular query, this peer is registered at the tracker as caching search results for that query. Subsequent requests for that same query are offloaded to caching peers by the tracker. When there are multiple possible caching peers for a query, one is selected randomly.

Figure 1 shows the number of search results provided by the origin central supplier peer and the summed number of hits on the caches at the consumer peers. It turns out that results for about half of the queries need to be given by the supplier at least once. The other half can be served from the caches of the other peers. Since the maximum achievable cache hit ratio is approximately 0.5, caching can reduce the load on a central peer by about 50 percent. Caching becomes more effective as more queries flow through the system. This is due to the effect that there are increasingly more repeated queries and less unique queries. So, you always see slightly fewer new queries than queries you have already seen as the number of queries increases.

How many results can a peer serve from its local cache and for how many does it have to consult caches at other peers? The local cache hit ratio climbs from around 22 percent for several thousand queries to 39 percent for all 21 million queries. So, the majority of cache hits is on external peers (between 61 and 78 percent).

Let us take a closer look at those external hits. We define a peer's share ratio as follows:

$$shareratio = \#cachehits/\#queries \qquad (1)$$

Where *cachehits* is the number of external hits on a peer's cache, meaning: all cache hits that are not queries posed by the peer itself. *Queries* is the number of queries issued by the peer. A *shareratio* of 0 means that a peer's cache is never used for answering external queries, between 0 and 1 means that a peer is sending more queries than it answers, and a ratio above 1 indicates that a peer is actually serving results for more queries than it sends.

Figure 2 shows that about 20 percent of peers does not share anything at all. It turns out that the majority of peers, 68 percent, at least serve results for some queries, whereas only 12 percent, about 80,000 peers, serve results for more queries than they issue.

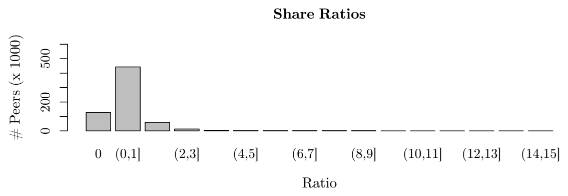


Figure 2: Observed share ratios (N=651,647 peers).

3.4 Decentralised

Now that we have shown the effectiveness of caching for offloading one central peer, we make the scenario more realistic. Instead of a central peer we introduce n peers that are *both* suppliers and consumer at the same time. These mixed peers are chosen at random. They serve search results, pose queries and also participate in caching. The remaining peers are merely consumers that can only cache results.

The central hits in the previous sections become hits per supplier in this scenario. How does the distribution of search results affect the external cache hit ratios of the supplier peers? We examine two distribution cases:

1. For each query there is always only exactly one supplier with unique relevant search results.
2. The number of supplier peers that have relevant search results for a query depends on the query popularity. There is always at least one supplier for a query, but the more popular a query the more suppliers there are (up to all n suppliers for very popular queries).

For simplicity we assume in both cases that there is only one set of search results per query. In the first case this set is present at exactly one supplier peer. However, the second case is more complicated: among the mixed peers we distribute the search results by considering each peer as a bin covering a range in the query frequency histogram. We assume that for each query there is at least one peer with relevant results. However, if a query is more frequent it can be answered by more mixed peers. The most frequent queries can be served by *all* n supplier peers. The distribution of search results is, like the queries themselves, *zipf* over the mixed peers. We believe that this is realistic, since popular queries on the Internet tend to have many search results as well. In this case the random choice is between a variable number m of n peers that supply search results for a given query. Thus, when the tracker receives a query for which there are multiple possible peers with results it chooses one randomly.

We performed two experiments to examine the influence on query load. The first is based on case 1, where there is always one supplier given an input query. The second is based on case 2 where the number of suppliers varies per query. For case 2 we first used the query log to determine the popularity of queries and then used this to generate the initial distribution of search results over the suppliers. This distribution is performed by randomly assigning the search results to a fraction of the suppliers depending on the query popularity. Since normally the query popularity can only be approximated, the results represent an ideal outcome.

Table 2: Original search results and cache hits (21,082,980 queries; 651,647 peers of which 10,000 are suppliers). All suppliers operate in mixed mode.

	Case 1	Case 2
Suppliers (origin)	11,599,060	12,110,592
Consumers internal (caches)	3,682,995	3,930,025
Consumers external (caches)	5,800,925	5,042,363

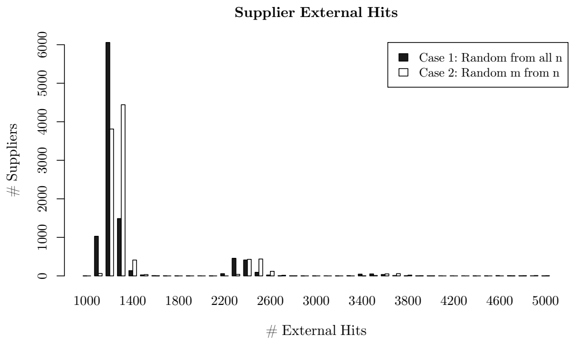


Figure 3: Supplier external hit distributions (N=651,647 peers, n=10,000 suppliers).

We used n = 10,000 supplier peers in a network of 651,647 peers in total (about 1.53 percent). This mimics the Internet which has a small number of websites compared to a very large number of surfing clients.

Figure 3 and Table 2 show the results. The number of original search results provided by the suppliers is about five percent higher than in the central peer scenario. This is the combined effect of no explicit offloading of the supplier peers by the tracker, and participation of the suppliers in caching for other queries. In the second case there is slightly more load on the supplier peers than in the first case: 57 percent versus 55 percent. The hit distribution in Figure 3 is similar even though the underlying assumptions are different. About 87 percent of peers answer between 1000 and 1500 queries. A very small number of peers answers up to about five times that many queries. Differences are found near the low end, which seems somewhat more spread in the first than in the second case. Nevertheless, all these differences are relatively small. The distribution follows a wave-like pattern with increasingly smaller peaks: near 1300, 2500, 3700 and 4900. The cause of this is unknown.

3.5 Churn

The experiments thus far have shown the maximum improvements that are attainable with caching. In this section we add one more level of realism: we no longer assume that peers are on-line infinitely. We base this experiment on case 1 above where the search results are uniformly distributed over the suppliers. The query log contains timestamps and we assume that if a specific user has not issued a query for some period of time, that his session has ended and its cache is temporarily no longer available. If the same user issues a query later on (comes back on-line), its cache becomes available again. This simulates churn in a peer-to-peer network

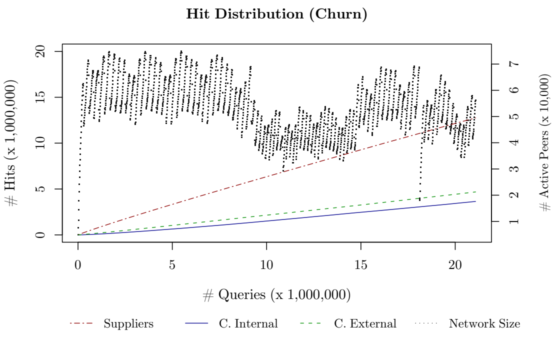


Figure 4: Distribution of hits under churn conditions (N=651,647 peers).

where peers join and depart from the network. All peers, including supplier peers, are subject to churn. For bootstrapping: if there are no suppliers on-line at all, an off-line one is randomly chosen to provide search results.

Assuming that all peers are on-line for a fixed amount of time is unrealistic. Stutzbach and Rejaie [8] show that download session lengths, post-download lingering time and the total up-time of peers in peer-to-peer file sharing networks are best modelled by using *Weibull distributions*. However, our scenario differs from file sharing. An information retrieval session does not end when a search result has been obtained, rather it spans multiple queries over some length of time. Even when a search session ends, the machine itself is usually not immediately turned off or disconnected from the Internet. This leads us to two important factors for estimating how long peers remain joined to the network. Firstly, there should be some reasonable minimum that covers at least a browsing session. Secondly, up-time should be used rather than 'download' session length. As soon as a peer issues its first query we calculate the remaining up-time of that peer in seconds as follows :

$$remaininguptime = 900 + (3600 \cdot 8) \cdot w \qquad (2)$$

where w is a random number drawn from a Weibull distribution with $\lambda = 2$ and $k = 1$. The w parameter is usually near 0 and very rarely near 10. The up-time thus spans from at least 15 minutes to at most about 80 hours. About 20 percent of the peers is on-line for longer than one day. This mimics the distribution of up-times as reported in [8].

Figure 4 shows the results: the number of origin search results served by suppliers as well as the number of internal and external hits on the caches of consumer peers. We see that the number of supplier hits increases to over 12.75 million: over 1.16 million more compared to the situation with no churn. The majority of this increase can be attributed to a decrease in the number of external cache hits. The dotted cloud shows the size of the peer-to-peer network on the right axis: this is the number of peers that is on-line simultaneously. We can see that this varies somewhere between about 30,000 and 80,000 peers. There is a dip in the graph caused by the earlier described log truncation.

4. CONCLUSION

We conducted several experiments that simulate a large-scale peer-to-peer information retrieval network. Our research questions can be answered as follows:

1. At least 50 percent of the queries can be answered from search result caches in a centralised scenario. This drops to 45 percent for the decentralised case.
2. Share ratios are skewed which suggests that additional mechanisms are needed for cache load balancing.
3. Introducing churn into a peer-to-peer network reduces the maximum cache hits by 12 percent to 33 percent.

We have shown the potential of caching under increasingly realistic conditions. Caching search results significantly offloads the origin suppliers that provide search results under all considered scenarios. This could be even further improved by applying query subsumption, term re-ordering and stemming. These techniques may decrease the quality of the search results, but also offer more effective usage of caches. This is needed when extra layers of realism are added to the experiments by working with individual search results instead of result sets, by experimenting with finite size caches, and by invalidating cached results over time. It would be useful to experiment with a combination of caching *and* replication. Finally, much work remains to be done in peer-to-peer information retrieval, especially in investigating the properties that hold in large-scale simulations.

5. ACKNOWLEDGEMENTS

We wish to thank Dolf Trieschnigg. This paper was created using only Free and Open Source Software. We gratefully acknowledge the support of the Netherlands Organisation for Scientific Research (NWO) under project 639.022.809.

References

[1] Baentsch, M., Baum, L., Molter, G., Rothkugel, S., and Sturm, P. 1997. Enhancing the web's infrastructure. *Internet Computing 1,* 2 (Mar.), 18–27.

[2] Brenes, D. J. and Gayo-Avello, D. 2009. Stratified analysis of aol query log. *Information Sciences 179,* 12, 1844 – 1858.

[3] Cohen, B. 2003. Incentives build robustness in bittorrent. In *Proceedings of P2PEcon.*

[4] Lu, J. and Callan, J. 2006. Full-text federated search of text-based digital libraries in peer-to-peer networks. *Information Retrieval 9,* 4, 477–498.

[5] Markatos, E. P. 2001. On caching search engine query results. *Computer Communications 24,* 2 (Feb.), 137–143.

[6] Pass, G., Chowdhury, A., and Torgeson, C. 2006. A picture of search. In *Proceedings of InfoScale.* Hong Kong, 1.

[7] Skobeltsyn, G. and Aberer, K. 2006. Distributed cache table: efficient query-driven processing of multi-term queries in p2p networks. In *Proceedings of P2PIR.* Arlington, Virginia, US, 33–40.

[8] Stutzbach, D. and Rejaie, R. 2006. Understanding churn in peer-to-peer networks. In *Proceedings of IMC.* Rio de Janeiro, BR, 189–202.

Compressed contributions

Ranking Related Entities: Components and Analyses (Abstract) [*]

Marc Bron
m.m.bron@uva.nl

Krisztian Balog
k.balog@uva.nl

Maarten de Rijke
derijke@uva.nl

ISLA, University of Amsterdam
Science Park 904, 1098 XH Amsterdam

ABSTRACT

Related entity finding is the task of returning a ranked list of homepages of relevant entities of a specified type that need to engage in a given relationship with a given source entity. We propose a framework for addressing this task and perform a detailed analysis of four core components; co-occurrence models, type filtering, context modeling, and homepage finding. Results show that pure co-occurrence is useful to select initial candidates, that type filtering is an instrument for tuning towards either recall or precision, and that context models successfully promote entities engaged in the right relation with the source entity. Our method achieves very high recall scores on the end-to-end task and is able to incorporate additional heuristics that lead to state-of-the-art performance.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.3 Information Search and Retrieval

General Terms

Algorithms, Experimentation, Measurement, Performance

Keywords

Entity search, Language modeling, Wikipedia

1. INTRODUCTION

Over the past decade, increasing attention has been devoted to retrieval technology aimed at identifying entities relevant to an information need. The TREC 2009 Entity track introduced the *related entity finding* (REF) task: given a source entity, a relation and a target type, identify homepages of target entities that enjoy the specified relation with the source entity and that satisfy the target type constraint [1]. E.g., for a source entity ("Michael Schumacher"), a relation ("His teammates while he was racing in Formula 1") and a target type ("people") return entities such as "Eddie Irvine" and "Felipe Massa." To address the REF task we consider an entity finding system architecture as shown in Fig. 1. The first component is a co-occurrence-based model that selects candidate entities. While a co-occurrence-based model can be effective in identifying the potential set of related entities, it fails to rank them effectively. Our failure analysis reveals two types of error that affect precision: (1) entities of the wrong type pollute the ranking and (2) entities are retrieved that are associated with the source entity without engaging in the right relation with it. To address (1), we add type filtering based on category information in Wikipedia. To correct for (2), we complement the pipeline with contextual information, represented as statistical language models derived from documents in which the source and target entities co-occur.

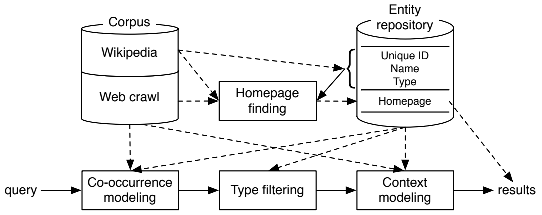


Figure 1: Components of our REF system.

2. APPROACH

The goal of the REF task is to return a ranked list of relevant entities e for a query consisting of a source entity (E), target type (T) and a relation (R) [1]. We formalize REF as the task of estimating the probability $P(e|E,T,R)$. As this probability is difficult to estimate directly we apply Bayes' Theorem and rewrite $P(e|E,T,R)$. After dropping the denominator as it does not influence the ranking of entities, we derive the following ranking formula:

$$\begin{aligned}
&P(E,T,R|e)\cdot P(e)\\
&\quad\propto P(E,R|e)\cdot P(T|e)\cdot P(e) \qquad (1)\\
&\quad= P(E,R,e)\cdot P(T|e) = P(R|E,e)\cdot P(E,e)\cdot P(T|e)\\
&\quad\overset{\text{rank}}{=} P(R|E,e)\cdot P(e|E)\cdot P(T|e)
\end{aligned}$$

In (1) we assume that type T is independent of source entity E and relation R. We rewrite $P(E,R|e)$ to $P(R|E,e)$ so that it expresses the probability that R is generated by two (co-occurring) entities (e and E). Finally, we rewrite $P(E,e)$ to $P(e|E)\cdot P(E)$ and drop $P(E)$ as it is assumed uniform. We are left with the following components: (i) pure co-occurrence model ($P(e|E)$), (ii) type filtering ($P(T|e)$) and (iii) contextual information ($P(R|E,e)$).

Co-Occurrence Modeling. The pure co-occurrence component ($P(e|E)$) expresses the association between entities based on occurrences in documents, independent of context (i.e., document content). To express the strength of co-occurrence between e and

[*] The full version of this paper appeared in *CIKM 2010*.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DIR'11, February 4, 2011, Amsterdam, the Netherlands.
Copyright 2011 ACM X-XXXXX-XX-X/XX/XX ...$10.00.

E we use a function $\text{cooc}(e, E)$ and estimate $P(e|E)$ as follows:

$$P(e|E) = \frac{\text{cooc}(e, E)}{\sum_{e'} \text{cooc}(e', E)}.$$

We consider two settings of $\text{cooc}(e, E)$: (i) as maximum likelihood estimate and (ii) χ^2 hypothesis test [3].

Type Filtering. In order to perform type filtering we exploit category information available in Wikipedia. We map each of the target types ($T \in \{PER, ORG, PROD\}$) to a set of Wikipedia categories ($\text{cat}(T)$) and create a similar mapping from entities to categories ($\text{cat}(e)$). The former is created manually, while the latter is granted to us in the form of page-category assignments in Wikipedia. Note that we only consider entities that have a Wikipedia page. With these two mappings we estimate $P(T|e)$ as follows:

$$P(T|e) = \begin{cases} 1 & \text{if } \text{cat}(e) \cap \text{cat}^{L_n}(T) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

We expand the set of categories assigned to each target entity type T, hence write $\text{cat}^{L_n}(T)$, where L_n is the chosen level of expansion and a parameter to be determined empirically.

Adding Context. The $P(R|E, e)$ component is the probability that a relation is generated from ("observable in") the context of a source and candidate entity pair. We represent the relation between a pair of entities by a co-occurrence language model (θ_{Ee}), a distribution over terms taken from documents in which the source and candidate entities co-occur. By assuming independence between the terms in the relation R we arrive at the following estimation:

$$P(R|E, e) = P(R|\theta_{Ee}) = \prod_{t \in R} P(t|\theta_{Ee})^{n(t,R)}, \qquad (2)$$

where $n(t, R)$ is the number of times t occurs in R. To estimate the co-occurrence language model θ_{Ee} we aggregate term probabilities from documents in which the two entities co-occur:

$$P(t|\theta_{Ee}) = \frac{1}{|D_{Ee}|} \sum_{d \in D_{Ee}} P(t|\theta_d), \qquad (3)$$

where D_{Ee} denotes the set of documents in which E and e co-occur and $|D_{Ee}|$ is the number of these documents. $P(t|\theta_d)$ is the probability of term t within the language model of document d:

$$P(t|\theta_d) = \frac{n(t, d) + \mu \cdot P(t)}{\sum'_t n(t', d) + \mu}, \qquad (4)$$

where $n(t, d)$ is the number of times t appears in document d, $P(t)$ is the collection language model, and μ is the Dirichlet smoothing parameter, set to the average document length in the collection.

Homepage Finding. To gather possible homepage URLs we get the external links on an entity's Wikipedia page and submit the entity name as a query to an index of a large web crawl, collecting URLs of the top relevant documents. We then rank the URLs through a linear combination of their retrieval score and a score proportional to a URL's rank among the external links, with equal weights to both components.

3. EXPERIMENTS AND RESULTS

Our document collection is the ClueWeb09 Category B subset [2]. Named entity recognition is difficult to realize on a data set the size of ClueWeb. Instead we use Wikipedia as a repository of known entities. For our estimations we use the entity names as queries to an index of this collection to obtain co-occurrence counts. We perform two types of evaluation: first, on the intermediate components by comparing the entity strings to a ground truth established by extracting all primary Wikipedia pages from the TREC 2009 Entity qrels. Our second type of evaluation, is the end-to-end evaluation on the original TREC REF task. Specifically, we use R-Precision (R-prec), where R is the number of relevant entities for a topic, and recall at rank 100 (R@100). We also report on the metrics used at the TREC 2009 Entity track: P@10, nDCG@R, and the number of primary homepages retrieved (#pri). We forego significance testing as the minimal number of topics (25) recommended is not available. Table 1 shows the results when ranking without (left half) and with (right half) context; type filtering is always applied, optimized either for precision (top half) or recall (bottom half). The left half of the table shows R-precision and R@100 of the pure co-occurrence model including type filtering. We find that of the two estimators for the pure co-occurrence component χ^2 performs best in terms of precision and that MLE performs best in terms of recall. Comparing the top half with the bottom half of the table we find that the type filtering component can be used to increase either precision or recall. The highest recall is obtained by using MLE and level 6 category expansion. The right half of Table 1 shows results for the context dependent model. In both cases (optimized for precision/recall), R-precision and R@100 are improved further.

Co-occ.	Pure Co-Occurrence		Context Dependent	
	R-Prec	R@100	R-Prec	R@100
Optimized for Precision				
MLE	.1512	**.5423**	.1898	**.5423**
χ^2	**.2382**	.4891	**.2623**	.4747
Optimized for Recall				
MLE	.0799	**.5821**	.0966	**.6982**
χ^2	**.2281**	.5474	**.2399**	.5418

Table 1: Results for the pure co-occurrence and context dependent model with filtering for either precision or recall.

On the end to end evaluation when optimized for precision (P@10=.2100, nDCG@R=.1198) we improve substantially over the median results achieved at TREC 2009 (P@10=.1030, nDCG@R= .0650). When optimized for recall our model surpasses the top performing team in terms of the number of primary homepages retrieved (#pri: 171 vs. 137, out of 396). We use this as a starting point for improving precision of our model by adding additional heuristics: (i) improved type filtering by utilizing high quality type definitions in the DBpedia ontology and (ii) co-occurrence based on anchor text. Anchor based co-occurrence emphasizes candidate entities that occur on the Wikipedia page of the source entity as anchor text or vice-versa. We find that with these additional heuristics our model (P@10=.3000, nDCG@R=.1562) achieves performance comparable to the median of the top 3 at TREC (P@10=.3100, nDCG@R=.1689) in terms of precision, while maintaining exceptionally high recall scores (#pri=174/396).

Acknowledgements. Supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.512, 612.061.814, 612.061.815, 640.004.802, and partially by the Center for Creation, Content and Technology (CCCT).

4. REFERENCES

[1] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *TREC '09*.

[2] ClueWeb09. The ClueWeb09 dataset, 2009. URL: `http://boston.lti.cs.cmu.edu/Data/clueweb09/`.

[3] C. D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

Result Diversification Based on Query-Specific Cluster Ranking (Abstract) [*]

Jiyin He, Edgar Meij and Maarten de Rijke
ISLA, University of Amsterdam
The Netherlands
{j.he, edgar.meij, derijke}@uva.nl

ABSTRACT

Result diversification is a retrieval strategy for dealing with ambiguous or multi-faceted queries by providing documents that cover as many potential facets of the query as possible. We propose a result diversification framework based on query-specific clustering and cluster ranking, in which diversification is restricted to documents belonging to a set of clusters that potentially contain a high percentage of relevant documents. Empirical results on the TREC 2009 Web track test collection show that the proposed framework improves the performance of several existing diversification methods, including MMR, IA-select, and FM-LDA. The framework also gives rise to a simple yet effective cluster-based approach to result diversification that selects documents from different clusters to be included in a ranked list in a round robin fashion.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software

General Terms: Algorithms, Experimentation

Keywords: Result diversification, Query-specific clustering

1. INTRODUCTION

Queries submitted to web search engines are often ambiguous or multi-faceted in the sense that they have multiple interpretations or sub-topics. One retrieval strategy that attempts to cater for multiple interpretations of such a query is to *diversify* the search results. Without explicit or implicit user feedback or history, the retrieval system makes an educated guess as to the possible facets of the query and presents as diverse a result list as possible by including documents pertaining to different facets of the query within the top ranked documents.

Following the Cluster Hypothesis [6], query-specific cluster-based retrieval is the idea of clustering retrieval results for a given query, which was shown to improve retrieval effectiveness if one can place documents from high quality clusters (in which a relatively large fraction of documents is relevant) at the top of the ranked list [5]. In this paper, we consider a ranking approach based on query-specific cluster-based retrieval in the context of result diversification. Specifically, we propose to rank and select a set of high quality clusters and then apply diversification only to the documents within these clusters. We posit that such a strategy should lead to improved results as measured in terms of both relevance and diversity since it only diversifies documents that are likely to be relevant.

2. METHOD

The overall goal of our approach is to rank query-specific clusters with respect to their relevance to the query and to limit the diversification process to documents contained in the top ranked clusters only, in order to improve the effectiveness of diversification as measured in terms of both relevance and diversity.

For a query q and a ranked list of top n documents D_q^n retrieved in response to q, we cluster D_q^n into K clusters. Assume that we have a ranking method $cRanker(\cdot)$ that ranks clusters with respect to their relevance to a query and a diversification method $Div(\cdot)$ that diversifies a given ranked list of documents. We propose the following procedure for diversification. The input of the procedure is the output of $cRanker$, that is, a ranked set of clusters $RC = c_1, \ldots, c_K$, and the documents contained in each cluster, D_q^c. A free parameter T is used to indicate the number of top ranked clusters to be selected for diversification. Furthermore, $dRanker(\cdot)$ is assumed to be a document ranker that ranks documents according to certain criteria, for example, ranking documents in descending order of their retrieval scores. The diversification procedure first applies $Div(\cdot)$ to the documents assigned to the top T ranked clusters; documents assigned to clusters ranked below the top T are ranked by $dRanker(\cdot)$ and appended to the ranked list of documents obtained from the top T clusters.

Clustering. We use latent Dirichlet allocation (LDA) [2] to cluster the initial retrieved ranked list. First, we train the topic models over D_q^n with a pre-fixed number of K clusters (or latent topics). A document d is then assigned to a cluster c^* such that

$$c^* = \arg\max_c p(c|d), \tag{1}$$

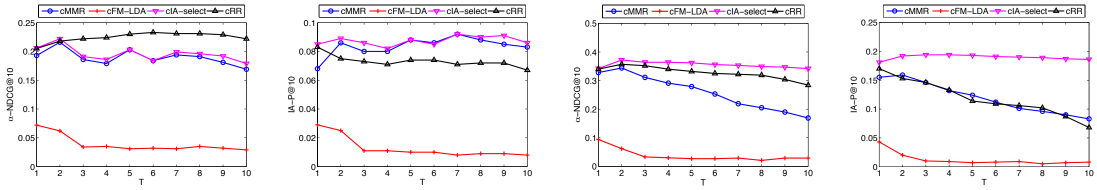
where $p(c|d)$ is estimated using the LDA model.

Diversification. For $Div(\cdot)$ we consider the following three diversification methods: Maximal Marginal Relevance (MMR) [3], Facet Model with LDA (FM-LDA) [4], and Intent Aware select (IA-select) [1]. In addition, we propose a cluster-based approach referred to as Round-Robin (RR). For this, we first rank the clusters according to their relevance of the clusters to the query. Then, documents within each cluster are ranked in the order of their original retrieved scores and, finally, we select documents belonging to different clusters in a round robin fashion.

Cluster ranking. For simplicity, we only discuss two ways to rank clusters that are necessary for investigating the effectiveness of our proposed framework for result diversification: *query likelihood* and *oracle*. For an input query, the query likelihood ranker ranks the clusters in descending order of the probability $p(c|q)$, which is inferred from the LDA model as described above. In other words, the clusters are ranked according to their likelihood given the query. Presumably, if a cluster has a high probability to generate a query, the documents contained in this cluster are more likely to be relevant to the query. Hence, the cluster is more likely to contain

[*] The full version of this paper is accepted for publication by *Journal of American Society for Information Science and Technology*.

Copyright is held by the author/owner(s).
DIR'11 February 4, 2011, Amsterdam, the Netherlands
ACM X-XXXXX-XX-X/XX/XX.



(a) (b) (c) (d)

Figure 1: Diversification with cluster ranking using query likelihood ranker (1(a), 1(b)) and oracle ranker (1(c), 1(d)) over different numbers of selected top ranked clusters (T). K is set to 10 (30 and 50 show similar trends).

relevant documents. The oracle ranker, on the other hand, ranks the clusters using information from explicit relevance judgments. Here, the probabilities $p(c|q)$ are estimated using the judgments of retrieved documents in D_q^n, computed as

$$p(c|ora_q) = \frac{|D_q^c \cap D_q^R|}{|D_q^c|}. \quad (2)$$

where D_q^R are the documents judged to be relevant. That is, we rank clusters according to the number of relevant documents contained in them, normalized by the size of the cluster.

Determining the cut-off T. Automatically determining the optimal cut-off T is non-trivial. We typically do not have sufficiently many test queries to learn the optimal value of T, hence we apply leave-one-out cross-validation to find the optimal value of T for each query. Specifically, we optimize T over a set of training queries for a given K and a given diversification method for a given evaluation metric by exhaustive search, i.e., over all possible values of $T = 1, ..., K$. Then we apply the learned T on the test query.

3. RESULTS AND DISCUSSION

We apply our proposed diversification framework on the TREC 2009 Web track catB test collection. We use the Markov Random Field model (MRF) [7] to generate the initial ranked list and set $n = 1000$. We then conduct the LDA clustering on the initial ranked list, setting $K = 10, 30,$ and 50.

Figure 1 shows the trends of the performance of each diversification method with cluster ranking (cMMR, cFM-LDA, cIA-select and cRR) across values of T, the number of top-ranked clusters whose documents are used for diversification. For each method, when $T = K$, diversification with cluster ranking is equivalent to diversifying the complete list of initially retrieved documents. Here, we only show the results measured using α-NDCG@10 and IA-P@10; a similar trend can be observed for α-NDCG@X and IA-P@X, for X =5 and 20. We observe that with both the query likelihood and the oracle cluster ranker, diversification performance is hardly influenced by selecting all clusters, i.e., by diversifying the complete ranked list of documents. Also, for each method there is an optimal value of T that maximizes the performance of the method, the value of which is smaller than the total number of clusters, i.e., for which the optimal value of T satisfies $T < K$.

If we compare the query likelihood ranker to the oracle cluster ranker, we see that the retrieval performance fluctuates a lot as T increases in Fig. 1(a) and 1(b), that is, with many local maximums, while in Fig. 1(c) and 1(d), the performance curves are relatively smooth: they remain the same or decrease once an initial maximum has been reached. This implies that, with a near perfect ranking of clusters, we can find the global optimal T by simply adding documents belonging to a cluster ranked next, until the performance starts to decrease. On top of that, we clearly see that optimal results are achieved by selecting a small number of top ranked clusters.

Table 1 compares diversification with cluster ranking against diversifying the complete list of retrieved documents. cX indicates the runs with cluster ranking and selection, where X is the name of a diversification method. K is the total number of clusters. Here we only list the results from K that result in best performance for the original diversification method (i.e., without cluster ranking). We also list the average predicted value of T. On top of that, we include the performance achieved by each method when T is optimal, indicated by T^*. These values correspond to the peaks in Figure 1.

K	**Method**	**α-NDCG@5**		**α-NDCG@10**		**IA-P@5**		**IA-P@10**	
		score	avg. T	score	avg. T	score	avg. T	score	avg. T
	MMR	0.122	–	0.169	–	0.066	–	0.083	–
10	cMMR	**0.191**$^{\triangle}$	1.98	**0.216**	2.00	0.070	2.44	0.069	6.82
	cMMR$^{T^*}$	**0.191**$^{\triangle}$	2	**0.216**	2	**0.090**	2	**0.092**	7
10	FM-LDA	0.027	–	0.029	–	0.011	–	0.008	–
	cFM-LDA	**0.058**	1.00	**0.072**$^{\triangle}$	1.00	**0.031**$^{\triangle}$	1.00	**0.029**$^{\triangle}$	1.00
	cFM-LDA$^{T^*}$	**0.058**	1	**0.072**$^{\triangle}$	1	**0.031**$^{\triangle}$	1	**0.029**$^{\triangle}$	1
50	IA-select	0.146	–	0.193	–	0.078	–	0.092	–
	cIA-select	0.181$^{\triangle}$	15.06	0.208	27.14	0.100	31.36	0.092	23.54
	cIA-select$^{T^*}$	**0.199**$^{\triangle}$	9	**0.226**$^{\triangle}$	27	**0.105**$^{\triangle}$	32	**0.096**	23
10	RR	0.198	–	0.222	–	0.079	–	0.067	–
	cRR	0.199	2.68	**0.233**$^{\triangle}$	6.00	0.085	2.00	**0.083**	1.00
	cRR$^{T^*}$	**0.204**	2	**0.233**$^{\triangle}$	6	**0.091**	2	**0.083**	1

Table 1: Results of proposed diversification framework. $^{\triangle}$indicates a significant difference given by a paired t-test with p-value<0.05.

We observe that diversification with cluster ranking outperforms the original algorithms in nearly all cases, even though query likelihood is not a perfect ranker for ranking clusters and T has not been fully optimized. If we take the optimal T with respect to the average performance over all queries, i.e., T^*, we see further improvements, and more improvements are statistically significant compared to that of the predicted T.

Acknowledgements This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, by the Center for Creation, Content and Technology (CCCT), and by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.512, 612.061.814, 612.061.815, 640.004.802.

4. REFERENCES

[1] R. Agrawal and S. Gollapudi and A. Halverson and S. Ieong. Diversifying search results. In *WSDM'09*, 2009.

[2] D. M. Blei and A. Y. Ng and M. I. Jordan. Latent Dirichlet Allocation. In *Journal of Machine Learning Research*, 2003.

[3] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries In *SIGIR'98*, 1998.

[4] B. Carterette and P. Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *CIKM'09*, 2009.

[5] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR'96*, 1996.

[6] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. In *Information Storage and Retrieval*, 1971.

[7] D. Metzler and W. B. Croft. A Markov Random Field Model for Term dependencies. In *SIGIR'05*, 2005.

Accounting for Negation in Sentiment Analysis

Bas Heerschop
basheerschop@gmail.com

Paul van Iterson
paulvaniterson@gmail.com

Alexander Hogenboom
hogenboom@ese.eur.nl

Flavius Frasincar
frasincar@ese.eur.nl

Uzay Kaymak
kaymak@ese.eur.nl

Econometric Institute
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

ABSTRACT

Automated ways of analyzing sentiment in Web data are becoming more and more urgent as virtual utterances of opinions or sentiment are becoming increasingly abundant on the Web. The role of negation in sentiment analysis has been explored only to a limited extent until now. In this paper, we investigate the impact of accounting for negation in sentiment analysis. To this end, we utilize a basic sentiment analysis framework – consisting of a wordbank creation part and a document scoring part – taking into account negation. Our experimental results show that by accounting for negation, precision relative to human ratings increases with 1.17%. On a subset of selected documents containing negated words, precision increases with 2.23%.

1. INTRODUCTION

In recent years, utterances of opinions or sentiment have become increasingly abundant on the Web through messages on Twitter, online customer reviews, etcetera. The information contained in this ever-growing data source is invaluable to key decision makers, e.g., those making decisions related to politics, reputation management, or marketing. An understanding of what is going on in their particular markets is crucial for decision makers, yet the analysis of sentiment in an overwhelming amount of data is far from trivial.

Sentiment analysis aims to determine the attitude, evaluation, or emotions of the author with respect to the subject of a text. This may involve word sentiment scoring (i.e., learning the sentiment scores of single words), subject/aspect relevance filtering (i.e., determining the subject and/or aspect a sentiment carrying word is relevant to), subjectivity analysis (i.e., determining whether a sentence is subjective or objective), or sentiment amplification and negation (i.e., modifying sentiment strength on amplifying words and reversing sentiment scores on negated words). The impact of taking into account negation in sentiment analysis has not been demonstrated yet. Therefore, we present our first steps towards insight in the impact of negation on sentiment analysis. A more elaborate analysis may be found in an extended version of this work [2].

DIR 2011 February 4, 2011, Amsterdam, The Netherlands.
Copyright 2011 by the author(s).

2. SENTIMENT ANALYSIS

Most approaches to sentiment analysis (i.e., classification) of documents essentially adhere to more or less similar frameworks consisting of creating a list of words and their associated sentiment from a training corpus and a subsequent method for scoring documents. An example of such a framework is the basic framework proposed by Ceserano et al. [1], who provide two word scoring algorithms based on supervised learning and three sentence-level document scoring algorithms with topic relevance filtering. Despite adhering to similar frameworks, document sentiment analysis approaches have several characteristic features distinguishing them from one another.

Sentiment may be scored on document level, sentence level, or window level. In this process, most approaches rely on a wordbank, typically containing per-word sentiment scores. Creation methods include supervised learning on a set of manually rated documents, learning through related word expansion, completely manual creation, or a combination of these methods. In matching words in a text with words in a wordbank, some approaches as lemmatization are designed to cope with syntactical variations. Part-of-speech tagging is also considered to be helpful in sentiment analysis, as it may help algorithms to, for example, distinguish sentiment-carrying words like adjectives or adverbs. Additionally, some algorithms attempt to identify subjective phrases or phrases relevant to the topic considered in order to boost sentiment analysis performance. Other helpful techniques include taking into account amplification or negation of sentiment carrying words. The role of negations has however been explored only to a limited extent until now. Therefore, we propose to shed some light onto the impact of accounting for negation in sentiment analysis.

3. SENTIMENT NEGATION

In order to assess the impact of sentiment negation, we propose a very simple sentiment analysis framework, consisting of wordbank creation and subsequent lexicon-based document scoring. Both parts have optional support for sentiment negation. We classify a document as either positive (1), neutral (0), or negative (-1). The score range of individual words is [-1, 1]. We focus on adjectives.

The first part of our framework facilitates wordbank creation, involving scoring sentiment of individual words (adjectives) w in a training corpus D_{train}. Our word scoring function is based on a pseudo-expected value function [1]. The sentiment score of any adjective w, $\text{score}(w)$, is based on its total relative influence on the sentiment over all documents $d \in D_w$, where $D_w \subseteq D$, with each document containing w:

$$\text{score}(w) = \frac{\sum_{d \in D_w} \text{score}(d) \times \inf(w, d, neg)}{|D_w|}, \quad (1)$$

where score (d) is a document d's manually assigned score, $|D_w|$ is the number of documents in D_w, and $\inf(w,d,neg)$ is the relative influence of an adjective w in document d, with *neg* indicating whether to account for negation or not. This influence is calculated as the count freq (w,d,neg) of w in d in terms of the total frequency $\sum_{w' \in d} \text{freq}(w', d, neg)$ of all sentiment carrying words w' in d:

$$\inf(w,d,neg) = \frac{\text{freq}(w,d,neg)}{\sum_{w' \in d} \text{freq}(w',d,neg)}. \quad (2)$$

In order to support negation in our framework, we use a variation of Hu and Liu's method [3] of negation. We first focus on a one-word scope for negation words in an attempt to tease out the effects of accounting for even the simplest forms of negation, as opposed to not accounting for negation at all. We only handle negation words that precede a sentiment word, as larger distances might cause noise in our results due to erroneously negated words. Support for negation is considered in the frequency computations by subtracting the number of negated occurrences of word w or w' in d from the number of non-negated occurrences of w or w' in d.

In the second part of our framework, the score eval (d) of a document d containing n adjectives $\{w_1, w_2, \ldots, w_n\}$ is simply computed as the sum of the scores of the individual adjectives (the same adjective can appear multiple times), as determined using (1) and (2). In case negation is accounted for, we propose to use the following document scoring function:

$$\text{eval}(d) = \sum_{w_i \in d} (-1)^{\text{negated}(w_i,d)} \times \text{score}(w_i), \quad (3)$$

where negated (w_i,d) is a Boolean indicating whether the ith adjective in w is negated in d (1) or not (0). Using (3), the classification class (d) of a document d can finally be determined as follows:

$$\text{class}(d) = \begin{cases} 1 & \text{if eval}(d) > 0.002, \\ 0 & \text{if } -0.021 \leq \text{ eval}(d) \leq 0.002, \\ -1 & \text{if eval}(d) < -0.021, \end{cases} \quad (4)$$

where the thresholds have been optimized through hill-climbing.

4. EVALUATION

We have implemented our framework in C#, combined with a Microsoft SQL Server database. We have used a corpus of 13,628 human-rated Dutch documents on 40 different topics. Sentiment in these documents is classified as positive, negative, or neutral. In order to be able to asses the impact of negation, we have implemented two versions of our framework. The first version has no support for negation, whereas the second version supports negation both in the wordbank creation and in the document scoring part. Our framework only handles adjectives for sentiment analysis and uses the Teezir part-of-speech tagger (based on OpenNLP and trained on Dutch corpora) to identify adjectives in the corpus.

We have used 60% of our documents for training and 40% for testing. The training set was used to create wordbanks and to determine the best threshold level for document classification. Our software first retrieves all adjectives from the training corpus, where multiple occurrences of an adjective are not allowed. The list of adjectives thus extracted is subsequently used for creating a wordbank, by scoring all adjectives in the training set with word scoring function (1). Our software then scores documents in accordance with document scoring functions (3) and (4).

In order to evaluate the human judgements, we manually rated documents in our corpus for sentiment. As manually rating documents for sentiment is a laborious activity, we have decided to use a random sample of 224 documents in this process. We observed 56% strong agreement and 99% weak agreement between our judgement and the human annotations, where strong agreement means an exact match and weak agreement means that one rating is positive or negative, whereas the other is neutral. Most discrepancies between ratings can be explained by interpretation differences. It is for instance difficult for humans to pick up on subtle cases of sentiment, which can be expressed in irony and tone. The interpretation of such subtle uses of sentiment can differ from person to person. The two observed cases of strong disagreement are due to misinterpretation of the text.

We have evaluated the performance of our framework against human ratings in two set-ups: one with support for negation and one without support for negation. Precision improves with 1.17% from 70.41% without taking into account negation to 71.23% when accounting for negation. This improvement is even more evident when our framework is applied to a subset of the corpus, where each document contains negated words (not necessarily adjectives). On this subset of the corpus, precision increases with 2.23% from 69.44% without accounting for negation to 70.98% when taking into account negation. These results are notable given that only 0.85% of the sentences in the original corpus contain negations.

5. CONCLUSIONS AND FUTURE WORK

The main contribution of this paper lies in our reported endeavors of shedding some light onto the impact of accounting for negation in sentiment analysis. Our experiments with a basic sentiment analysis framework show that a relatively straightforward approach to accounting for negation already helps to increase precision. On a subset of selected documents containing negated words, precision increases somewhat more; a notable result if we consider the fact that negation is sparsely used in our data set.

Nevertheless, it appears to be worthwhile to investigate the effects of optimizing the scope of influence of negation words in order to obtain more detailed insights in the impact of negation in sentiment analysis. We also want to experiment with other types of words in our wordbank (e.g., adverbs, possibly combined with adjectives). Finally, we plan to consider various degrees of negation.

6. ACKNOWLEDGEMENTS

We would like to thank Teezir (`http://www.teezir.com`) for their technical support, fruitful discussions, and for supplying us with data for this research.

7. REFERENCES

[1] Cesarano, C., Dorr, B., Picariello, A., Reforgiato, D., Sagoff, A., Subrahmanian, V.: OASYS: An Opinion Analysis System. In: AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs (CAAW 2006). pp. 21–26, AAAI Press (2006)

[2] Heerschop, B., van Iterson, P., Hogenboom, A., Frasincar, F., Kaymak, K.: Analyzing Sentiment in a Large Set of Web Data while Accounting for Negation. In: 7th Atlantic Web Intelligence Conference (AWIC 2011), Springer (Forthcoming 2011)

[3] Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004), pp. 168–177. ACM (2004)

Contextual Factors for Finding Similar Experts (Extended Abstract)[*]

Katja Hofmann[†], Krisztian Balog[†], Toine Bogers[‡] and Maarten de Rijke[†]
[†]ISLA, University of Amsterdam and [‡]ILK, Tilburg University
K.Hofmann@uva.nl, K.Balog@uva.nl, A.M.Bogers@uvt.nl, deRijke@uva.nl

ABSTRACT

Expertise seeking research studies how people search for expertise and choose whom to contact in the context of a specific task. An important outcome are models that identify factors that influence expert finding. Expertise retrieval addresses the same problem, expert finding, but from a system-centered perspective. The main focus has been on developing content-based algorithms similar to document search. These algorithms identify matching experts primarily on the basis of the textual content of documents experts are associated with. Other factors, such as the ones identified by expertise seeking models, are rarely taken into account.

In this paper we extend content-based expert finding approaches with contextual factors that have been found to influence human expert finding. We focus on a task of science communicators in a knowledge-intensive environment, the task of *finding similar experts*, given an example expert. Our approach combines expertise seeking and retrieval research. First, we conduct a user study to identify contextual factors that may play a role in the studied task and environment. Then we design expert retrieval models to capture these factors. We combine these with content-based retrieval models and evaluate them in a retrieval experiment.

Our main finding is that, while content-based features are the most important, human subjects also take contextual factors into account, for example media experience and organizational structure. We develop two principled ways of modeling the identified factors and integrate them with content-based retrieval models. Our experiments show that models combining content-based and contextual factors can significantly outperform existing content-based models.

1. INTRODUCTION

The increasing amount of information available is making the need to critically asses information more important. The burden of credibility assessment and quality control is partly shifting onto individual information seekers, but the need for information intermediaries—e.g., experts—has not disappeared and is actually increasing in cases where the credibility of information has to meet high standards [8]. Against this background, *expert finding* is a particularly relevant task: identifying and selecting individuals with specific expertise, for example to help with a task or solve a problem.

The goal of *expertise retrieval* is to support search for experts using information retrieval technology. Following the experimental paradigm and evaluation framework established in the information retrieval community, expertise retrieval has been addressed in world-wide evaluation efforts [9]. Promising results have been achieved, particularly in the form of algorithms and test collections [1, 2]. State-of-the-art retrieval algorithms model experts on the basis of the documents they are associated with, and retrieve experts on a given topic using methods based on document retrieval, such as language modeling [3, 4]. In evaluations of these algorithms user aspects have been abstracted away.

While research into expertise retrieval has primarily focused on identifying good topical matches between needs for expertise and the content of documents associated with candidate experts, behavioral studies of human *expertise seeking* have found that there may be important additional factors that influence how people locate and select experts [11]—such factors include accessibility, reliability, physical proximity, and up-to-dateness. We term these *contextual factors* to distinguish them from content-based factors that have been explored in previous work.

2. RESEARCH QUESTIONS AND METHOD

Our aim in this paper is to explore the integration of contextual factors into content-based retrieval algorithms for finding similar experts. We look at this problem in the setting of the public relations department of a university, where communication advisors employed by the university get requests for topical experts from the media. The specific problem we are addressing is this: the top expert identified by a communication advisor in response to a request is not available because of meetings, vacations, sabbaticals, or other reasons. In this case, communication advisors have to recommend similar experts and this is the setting for our expert finding task. Based on this task we address three main research questions:

1. Which contextual factors influence (human) decisions when finding similar experts in the university setting we study?
2. How can such factors be integrated into content-based algorithms for finding similar experts?
3. Can integrating contextual factors with existing, content-based approaches improve retrieval performance?

To answer our research questions, we proceed as follows. Through a set of questionnaires completed by a university's communication advisors, we identify contextual factors that play a role in how similar experts are identified in this situation, and we construct a test data set to evaluate retrieval performance. From the questionnaire, we identify contextual factors that play a role in the studied setting.

Based on the questionnaire results, we develop models of contextual factors, and integrate these with existing, content-based retrieval methods. We explored modeling factors as input-dependent, similar to content-based similarity methods, and as input-independent,

[*]The full version of this paper has appeared in [6].

Copyright is held by the author/owner(s).
DIR2011, February 4, 2011, Amsterdam.
.

similar to a prior probability. The intuition between the first model is that candidates with similar characteristics to the given target expert would be likely to be recommended. The intuition behind the second model is that there may be certain characteristics that make a candidate to be more likely to be recommended, independent of the target expert. In our experiments, we evaluate our contextual retrieval models against a baseline consisting of the optimal combination of content-based retrieval models.

3. FINDINGS

Our first goal was to identify contextual factors that play a role in the task of finding similar experts in response to media requests for expertise. We find that, while *topic of knowledge* appears to be the most important factor in the studied setting, contextual factors play a role as well, such as *position*and *contacts*. In addition to these factors that had been identified in previous studies, we were able to identify two new factors that played a role, namely *organizational structure* and *media experience*.

The individual contextual factors that appear to have the most impact are *media experience*, *organizational structure*, and *position*. This finding suggests that there may be a strong task-specific component to the contextual factors that play a role in finding similar experts, and possibly in other retrieval tasks as well. In future work, it would be interesting to perform similar studies of contextual factors in information seeking tasks in other settings. Based on findings from several such studies it may be possible to develop more general models of how tasks relate to other factors, and how these relations influence people's relevance decisions.

Our second research question was how to model contextual factors and integrate them with existing retrieval models. To this end, we explored modeling factors as input-dependent, similar to content-based similarity methods, and as input-independent, similar to a prior probability. We found that both types of models improved upon the baseline using content-based factors only. Overall, input-independent models led to better performance, except for the input-dependent model of *organizational structure*. Thus, the studied setting, it is important that a candidate expert is part of the same department as the topic expert, but in addition to that there are attributes that are common to frequently recommended experts, such as having prior media experience, or being a professor. Best performance was achieved with a run that combined both types of models. These results show that both types of models are useful and that it is not enough to identify a factor, but that it also needs to be modeled appropriately.

The third question was whether integrating contextual factors with content-based retrieval methods would improve retrieval performance. Our results show that our models that include contextual factors indeed achieve significant improvements over the content-based baseline methods.

Overall, our results indicate that identifying contextual factors and integrating them with content-based expertise retrieval models is indeed a promising research direction. The method used for collecting data on contextual factors is an extension of normal relevance assessment and could be applied in other settings where the original topic creators are available for relevance assessment.

4. CONCLUSION

In this paper we started from the observation that contextual factors appear to play a role in expertise seeking. We explored the role of contextual factors in the task of finding similar experts. First, we identified contextual factors that play a role in the task of finding similar experts in the public relations department of a university. The identified factors were modeled in two principled ways and implemented using available data. We integrated the resulting models with existing, content-based models and evaluated them to assess retrieval performance. Our results demonstrate that it is possible to identify and model contextual factors in the studied task of finding similar experts, and we think that this may be the case for other retrieval tasks as well.

In information seeking research, models of how contextual factors play a role have been developed and it has been shown that information seeking behavior changes with, for example, specifics of the task [5, 7] and the user's problem stage [10]. From an information retrieval perspective, these contextual factors are difficult to model and researchers typically design experiments where they abstract from context to make results generalizable. In this paper we have argued that, in order to arrive at generalizable results, we need to model context and develop models of how contextual factors influence expertise seeking. We have shown that the factors can be modeled, that it is possible to integrate them with retrieval models, and that the resulting models can improve retrieval performance.

Acknowledgements We are extremely grateful to the communication advisors at Tilburg University for their participation in our study: Clemens van Diek, Linda Jansen, Hanneke Sas, Pieter Siebers, Michelle te Veldhuis, and Ilja Verouden. We wish to thank Lilian Woudstra for her extensive feedback and Joost Kircz for discussions and feedback.

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 017.001.190, 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.-815, 640.004.802, by the Dutch and Flemish Governments (http://www.stevin-tst.org) under projects DAESO (STE-05-24) and DuOMAn (STE-09-12) and by the IOP-MMI program of SenterNovem / The Dutch Ministry of Economic Affairs, as part of the À Propos project.

5. REFERENCES

[1] P. Bailey, N. Craswell, I. Soboroff, and A. P. de Vries. The CSIRO enterprise search test collection. *SIGIR Forum*, 41(2): 42–45, 2007.

[2] K. Balog. *People Search in the Enterprise*. PhD thesis, University of Amsterdam, June 2008.

[3] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR '06*, pages 43–50. ACM Press, 2006.

[4] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Information Processing and Management*, 45(1):1–19, 2009.

[5] K. Byström and K. Järvelin. Task complexity affects information seeking and use. *Information Processing and Management*, 31(2):191–213, 1995.

[6] K. Hofmann, K. Balog, T. Bogers, and M. de Rijke. Contextual factors for finding similar experts. *JASIS&T*, 61(5):994–1014, 2010.

[7] J. Kim. Describing and predicting information-seeking behavior on the web. *JASIS&T*, 60(4):679–693, 2009.

[8] M. Metzger. Making sense of credibility on the web: Models for evaluating online information and recommendations for future research. *JASIS&T*, 58(13):2078–2091, 2007.

[9] TREC. Enterprise track, 2005. URL: http://www.ins.cwi.nl/projects/trec-ent/wiki/.

[10] P. Vakkari. Changes in search tactics and relevance judgements when preparing a research proposal a summary of the findings of a longitudinal study. *Information Retrieval*, 4(3-4):295–310, 2001.

[11] L. S. E. Woudstra and B. J. Van den Hooff. Inside the source selection process: Selection criteria for human information sources. *Information Processing and Management*, 44:1267–1278, 2008.

Automatically Annotating Web Pages Using Google Rich Snippets

Frederik Hogenboom
fhogenboom@ese.eur.nl

Flavius Frasincar
frasincar@ese.eur.nl

Damir Vandic
vandic@ese.eur.nl

Jeroen van der Meer
jeroenvdmeer@gmail.com

Ferry Boon
ferry.boon@gmail.com

Uzay Kaymak
kaymak@ese.eur.nl

Econometric Institute
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

ABSTRACT

We propose the Automatic Review Recognition and annOtation of Web pages (ARROW) framework, a framework for Web page review identification and annotation using RDFa Google Rich Snippets. The ARROW framework consists of four steps: hotspot identification, subjectivity analysis, information extraction, and page annotation. We evaluate an implementation of the framework by using various Web sites. Based on the evaluation we conclude that our framework is able to properly identify the majority of reviews, reviewed items, and review dates.

1. INTRODUCTION

Despite the technological advances of the last decades, it remains difficult for machines to understand information contained in Web pages on the World Wide Web. One of the pillars of the Semantic Web is to define the content of the Web pages semantically (i.e., as concepts with meaning) in order to make data machine understandable. The ability of computers to automatically process and interpret data will support new functionality on the Web.

Google's Rich Snippets is a service for Web page owners to add semantics to their (existing) Web page using the semantic vocabulary provided by Google. Up until now the vocabulary is rather limited in its number of concepts (Person, Review, Review Aggregate, Product, and Organization, Recipe, and Video). Future applications are promising, e.g., when searching for "Christian Dior" products, with Rich Snippets one is able to state that all results with "Christian Dior" as a person should be ignored.

For annotating Web sites built from structured data from a database, it would be sufficient to identify concepts in the generated pages and add the corresponding attributes to the Web page while generating the HTML output. Not all Web pages are built from databases and thus pre-generation of annotations is not always possible. The latter type of Web pages require manual annotation, which can be a tedious task. Hence, we present a method to automatically read and annotate Web pages, using the RDFa attributes as defined in Google Rich Snippet's vocabulary. The Automatic Review Recognition and annOtation of Web pages (ARROW) framework reads Web pages, identifies reviews, and annotates the pages with the RDFa attributes defined by Google Rich Snippets. An extended version of this paper containing more details on the framework is to be presented at the 26th ACM Symposium on Applied Computing [4].

DIR 2011 February 4, 2011, Amsterdam, The Netherlands.
Copyright 2011 by the author(s).

2. RELATED WORK

In this paper, we focus on unsupervised Web information extraction systems, as they can be fully automated and do not require pre-annotated documents for training. Based on Web page contents, unsupervised methods try to find a pattern on the Web page, e.g., a set of recurring HTML tags or specific text strings. Examples of unsupervised Web information extraction systems are RoadRunner [2] and DeLa [5]. To identify the attributes of the reviews, e.g., author, date, etc., these systems employ unsupervised information extraction methods for Web pages. These methods can be divided into tag-based approaches, text-based approaches, and hybrid approaches. The tag-based approaches derive a wrapper for the Web site based on the structural characteristics of a Web page. Text-based approaches focus on the textual content of a Web page. Last, the hybrid approaches are a combination of the tag-based and text-based approaches and hence contain elements of both methods.

There are three different approaches to review annotation. First of all, Microformats [3] is a collection of formats that makes the representation of semi-structured information such as reviews possible. In the case of reviews, the hReview microformat can be encountered on various Web sites. Second, the W3C is working on extending the HTML language, as part of the HTML5 specification, to allow native support for annotations as described by the Microdata format. The third and final option is RDFa. RDFa extends XHTML with a set of attributes that allow the XHTML code to be enriched with metadata. Although RDFa is aimed towards extending XHTML, its attributes can also be used in HTML as most RDFa parsers will recognize these attributes.

3. ARROW FRAMEWORK

Google Rich Snippets supports a limited vocabulary of RDFa entities and their attributes. Our main focus is on recognizing and annotating the review entities and their attributes in Web pages. The proposed ARROW framework for automatically annotating review pages by adding RDFa annotations to a Web page is composed of four stages: hotspot identification, subjectivity analysis, information extraction, and page annotation.

After normalizing the data, i.e., converting the HTML documents to DOM trees, we continue with identifying the potential reviews or *hotspots* of the page. Usually, reviews are characterized by blocks of text. These blocks are less often found in page headers, navigation elements, footers, etc. Text blocks are usually structured by small amounts of HTML elements, such as `h1` and `div`. Hence, for identifying reviews, we aim to find the elements that contain a lot of textual content. For this, we calculate a text-to-content ratio, the $TTCR$, which can be denoted as

$$TTCR = \frac{L_{text}}{L_{DOM}} \,, \qquad (1)$$

where the number of characters in text is denoted by L_{text} and the total number of characters within the DOM tree is represented by L_{DOM}. HTML elements with a high text-to-content-ratio are labeled as hotspots.

After hotspot detection, we need to verify the hotspots, as they might contain reviews. A review can be defined as a subjective view on a certain topic, as opposed to an objective view which describes only facts about a topic. In order to be able to analyze the hotspots, we use an improved version of the LightWeight subjectivity Detection mechanism (LWD) as proposed by [1], which now also takes into account the length of the review. More precisely, hotspots where a certain number of sentences contain a minimum number of subjectivity words per sentence are considered to represent reviews.

For review attribute extraction we employ several methods. Authors are identified by means of a Named Entity Recognizer (NER), whereas dates and ratings are recognized by means of regular expression patterns. Products are filtered from titles, as it is often hard to identify the product in the review content due to the frequent mentioning of related products.

Finally, after reviews and attributes have been identified in the Web pages, the framework annotates pages using Google's RDFa vocabulary designed by Google for its Rich Snippets. Annotating involves tagging the identified key elements of the review.

4. ARROW EVALUATION

We have implemented the ARROW framework as a Web application[1]. The approach is evaluated on data from various review Web sites[2]. We evaluate the framework on review identification and attribute identification. On average, review annotation is a subsecond process for each Web page.

To assess the review recognition performance, we test the tool on a selection of 100 English review Web pages and 100 non-review English Web pages. When comparing manually annotated reviews with ARROW's annotations, we obtain good results on precision and specificity, yet varying results on accuracy and recall. The results also show us that the framework works better on some Web sites than on others, caused by type of content, specific Web site structures, etc. When performing a similar experiment in order to assess the performance of review attribute identification, we can conclude that our framework does a good job on finding the item reviewed, date, and rating, but performs poorly on detecting the authors. This can be explained by the ambiguity of the names used on the Internet, as many people use nicknames on the Internet rather than their real (full) names. This makes the automatic identification of people difficult.

5. CONCLUSIONS

Using Google Rich Snippets for semantic annotation allows for a more appealing presentation by emphasizing some specific concept properties. Unfortunately, there are not yet many Web sites that support this vocabulary. In order to allow existing Web sites to make use of Google Rich Snippets, we have proposed the ARROW framework in this paper, which aims to automatically identify and annotate reviews on Web pages using Google's vocabulary. We have evaluated an implementation of the framework, which yields good results on precision and specificity, yet varying results on accuracy and recall.

As future work, we suggest to extend our framework to cover other elements from the Google Rich Snippets vocabulary, e.g., recipes, videos, and organizations. Also, one could take into consideration that many reviews lack an explicit rating, e.g., a grade or a number of stars. As Google Rich Snippets accepts a rating based on a scale of 1 to 5, it would be useful to investigate ways of calculating ratings based on review texts using, for example, sentiment analysis methods.

6. REFERENCES

[1] L. Barbosa, R. Kumar, B. Pang, and A. Tomkins. For a Few Dollars Less: Identifying Review Pages Sans Human Labels. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2009)*, pages 494–502. ACL, 2009.

[2] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In *27th International Conference on Very Large Data Bases (VLDB 2001)*, pages 109–118. Morgan Kaufmann Publishers Inc., 2001.

[3] R. Khare and T. Çelik. Microformats: A Pragmatic Path to the Semantic Web. In *15th International World Wide Web Conference (WWW 2006)*, pages 865–866. ACM, 2006.

[4] J. van der Meer, F. Boon, F. Hogenboom, F. Frasincar, and U. Kaymak. A Framework for Automatic Annotation of Web Pages Using the Google Rich Snippets Vocabulary. In *26th ACM Symposium on Applied Computing (SAC 2011)*, pages 763–770. ACM, 2011.

[5] J. Wang and F. H. Lochovsky. Data Extraction and Label Assignment for Web Databases. In *12th International World Wide Web Conference (WWW 2003)*, pages 187–196. ACM, 2003.

[1] Available at `http://www.arrow-project.com/`.

[2] Data extracted from `http://www.tripadvisor.com`, `http://www.epinions.com`, `http://www.imdb.com`, `http://www.yelp.com`, and `http://www.cnn.com`.

The Search Behavior of Media Professionals at an Audiovisual Archive: A Transaction Log Analysis (Abstract)[*]

Bouke Huurnink
ISLA, University of Amsterdam
bhuurnink@uva.nl

Laura Hollink
Delft University of Technology
l.hollink@tudelft.nl

Wietske van den Heuvel
Netherlands Institute for Sound and Vision
wvdheuvel@beeldengeluid.nl

Maarten de Rijke
ISLA, University of Amsterdam
derijke@uva.nl

ABSTRACT

Finding audiovisual material for reuse in new programs is an important activity for news producers, documentary makers, and other media professionals. Such professionals are typically served by an audiovisual broadcast archive. We report on a study of the transaction logs of one such archive. The analysis includes an investigation of commercial orders made by the media professions, as well as a characterization of sessions, queries, and the content of terms recorded in the logs. We identify a strong demand for short pieces of audiovisual material in the archive. Also, searchers are generally able to quickly navigate to a usable audiovisual broadcast, but it takes them longer to place an order for a subsection of a broadcast than it does for them to order an entire broadcast. Queries are found to predominantly consist of (parts of) broadcast titles and of proper names. Our observations imply that it may be beneficial to increase support for fine-grained access to audiovisual material, for example, through manual segmentation or content-based analysis.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation, Measurement

Keywords

Transaction log analysis, audiovisual archive

1. INTRODUCTION

Documentary makers, journalists, news editors, and other media professionals routinely require previously recorded audiovisual material for reuse in new productions. For example, a news editor might wish to reuse footage shot by overseas services for the evening news. To complete production, the media professional must locate audiovisual material that has been previously broadcast in another context. One of the sources for reusable broadcasts is the audiovisual archive, which specializes in the preservation and management of audiovisual material [1]. Where audiovisual material was once primarily stored on analog carriers, in recent years audiovisual archives have started making their content available in digital format and enabling online access [2, 3]. In such an environment, the media professional can search for and purchase multimedia material. In addition, with audiovisual acquisition being done through a digital interface, the archive can record information about the media professional's information seeking process. Despite the fact that an increasing amount of audiovisual programming is produced digitally, little is known about the search behavior of media professionals locating material for production purposes.

We aim to characterize the behavior of users of an audiovisual archive, and in addition to give insight into the content of their searches. We work in the context of a large national audiovisual broadcast archive, actively used by media professionals from a range of production studios. The archive presents a rich source of information because of its specialist nature. Our study is performed through an analysis of *transaction logs* — the electronic traces left behind by users interacting with the archive's online retrieval and ordering system. The transaction log analysis is enhanced by leveraging additional resources from the audiovisual broadcasting field, which can be exploited due to the specialist nature of the archive. In particular we analyze purchase orders of audiovisual material, and we use catalog metadata and the a structured audiovisual thesaurus to investigate the content of query terms.

2. METHOD

Our study takes place within the context of the *Nederlands Instituut voor Beeld en Geluid* — the Netherlands Institute for Sound and Vision, a large audiovisual archive, which we will refer to below as "the archive." The archive functions as the main provider of archive material for broadcasting companies in the Netherlands. The collection contains more than 700,000 hours of radio, television, documentaries, films and music. Nowadays, all digitally broadcast television and radio programs made by the Dutch public broadcasting companies are automatically ingested in the archive's digital asset management system. The digital multimedia items available in the archive can be divided into two types: video and audio. The video items consist largely of television broadcasts, but also include movies, amateur footage, and internet broadcasts. The audio portion of the collection consists primarily of radio broadcasts and music recordings. Each catalog entry contains multiple fields which contain either freely entered text or structured terms.

[*] The full version of this paper appeared in *Journal of the American Society for Information Science and Technology*, 61(5):994-1014, May 2010.

Copyright is held by the author/owner(s).
DIR'11, February 4, 2011, Amsterdam.
ACM xxx-x-xxxxx-xxx-x/xx/xx.

The archive is primarily used by media professionals who work for a variety of broadcasting companies, public and commercial, and are involved in the production of a range of programs. Once purchased, ordered audiovisual material may be re-used in many types of programs, especially news and current affairs programs. In addition, it is sometimes used for other purposes, for example to populate online platforms and exhibitions.

Search through audiovisual material in the archive is based on manually created catalog entries. These entries are created by the archival staff, and include both free text as well as structured terms contained in a specialized audiovisual thesaurus. The thesaurus is called the *GTAA* — the Dutch acronym for "Common Thesaurus for Audiovisual Archives."

3. EXPERIMENTAL DESIGN

Transaction logs from *Beeld en Geluid*'s online search and purchasing system were collected between November 18, 2008 and May 15, 2009. The logs were recorded using an in-house system tailored to the archive's online interface.

In total the logs contained $290,429$ queries after cleaning. The transaction logs often reference documents contained in the archive's collection. In order to further leverage the log information, we obtained a dump of the catalog descriptions maintained by the archive on June 19, 2009. The size of the catalog at that time was approximately $700,000$ unique indexed program entries.

We define five key units that will play a role in our analysis below: the three units common in transaction log analysis of session, query, term; and two units specific to this study, facet and order. The specialized knowledge sources available within the archive allowed (and motivated) us to develop the last two units of analysis.

4. MAIN FINDINGS

Our analysis was structured around four main research questions, the answers to which we summarize here. With respect to our first question, *what characterizes a typical session at the archive?*, we found the typical session to be short, with over half of the sessions under a minute in duration. In general, there were also few queries and result views in a session, with a median value of one query issued and one result viewed. Sessions resulting in orders had a considerably longer duration, with over half of the sessions having a median duration of over seven minutes, but no increase in terms of the number of queries issued and results viewed.

In answer to our second question, *what kinds of queries are users issuing to the audiovisual archive?*, we found nearly all of the queries contained a keyword search in the form of free text, while almost a quarter specified a date filter. The advanced search option, for searching on specific catalog fields, was used in 9% of the queries. The most frequently occurring keyword searches consisted primarily of program titles. Advanced search on specific catalog fields, when utilized, frequently specified the media format or copyright owner of the results to be returned, for example that only results available in high-quality digital format should be returned.

In addressing our next research question, *what kinds of terms are contained in the queries issued to the archive?*, we performed a content analysis of the query terms. This was accomplished by using catalog information as well as session data; terms in a query were matched to the titles and thesaurus entries of the documents that were clicked during a user session. This allowed us to leverage the thesaurus structure for identifying different kinds of query terms. The approach does have limitations, as terms can only be identified in sessions where users click at least one result, and even then, a term can only be identified if it is present as a title or thesaurus entry. Of all the queries where users clicked a result during the session, 41% contained a title term. Thesaurus terms were identified in 44% of the queries. Approximately one quarter of thesaurus terms consisted of general subjects such as *soccer*, *election*, and *child*. Another quarter consisted of the names of people, especially of politicians and royalty. The remaining terms were classified as locations, program makers, other proper names, or genres.

To answer our final research question, *what are the characteristics of the audiovisual material that is ordered by the professional users?*, we isolated the orders placed to the archive. Orders were for recent and historical material, with 46% of orders for items that were broadcast over one year before the order date. We identified three units of ordering: *programs*, *stories*, and *fragments*. We saw that less than a third of orders placed to the archive were for entire broadcasts, while 17% of the orders were for subsections of broadcasts that had been previously defined by archivists. Nearly half of the orders were for audiovisual fragments with a start and end time specified by the users. The fragments were typically on the order of a few minutes in duration, with 28% of fragments being one minute or less. In these cases, where users specified the fragment boundaries manually, sessions typically took more than two and half times as long as when ordering an entire broadcast.

5. CONCLUSION

Our main contributions in this paper include: a description of the search behavior of professionals in an audiovisual archive in terms of sessions and queries, and orders; a categorization of their query terms by linking query words to titles and thesaurus terms from clicked results; and an analysis of the orders made from the archive in terms of their size relative to the broadcast length and the time taken to get from query to purchase. Our study is significant in that there is a relatively large time span covered (almost half a year), and in that the users are specialists in audiovisual search, looking for broadcasts and fragments of broadcasts for reuse in new productions. In addition, we utilize catalog annotations to provide additional detail about the data recorded in the transaction logs. The results of the study can serve to give researchers and archives insight into aspects of multimedia search related to the specific use case of media professionals. They may also be used by audiovisual broadcast archives to better adjust their services to the user.

Acknowledgments. This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430 (GALATEAS), by the 7th Framework Program of the European Commission, grant agreement no. 258191 (PROMISE), by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.-066.512, 612.061.814, 612.061.815, 640.004.802, 380-70-011 and by the Center for Creation, Content and Technology (CCCT).

References

[1] R. Edmondson. *Audiovisual Archiving: Philosophy and Principles*. UNESCO, Paris, France, 2004.

[2] J. Oomen, H. Verwayen, N. Timmermans, and L. Heijmans. Images for the future: Unlocking value of audiovisual heritage. In *Museums and the Web 2009: Proceedings*, Toronto, Ontario, Canada, 2009. Archives & Museum Informatics.

[3] R. Wright. Annual report on preservation issues for European audiovisual collections. Deliverable PS_WP22_BBC_D22.4_Preservation Status_2007, BBC, 2007.

Generating Focused Topic-specific Sentiment Lexicons

Valentin Jijkoun Maarten de Rijke Wouter Weerkamp
ISLA, University of Amsterdam, The Netherlands
jijkoun,derijke,w.weerkamp@uva.nl

This paper is a compressed version of Jijkoun et al. (2010).

1. INTRODUCTION

In the area of *media analysis*, one of the key tasks is collecting detailed information about opinions and attitudes toward specific topics from various sources, both offline (traditional newspapers, archives) and online (news sites, blogs, forums). Specifically, media analysis concerns the following system task: given a topic and list of documents (discussing the topic), find all instances of attitudes toward the topic (e.g., positive/negative sentiments, or, if the topic is an organization or person, support/criticism of this entity). For every such instance, one should identify the source of the sentiment, the polarity and, possibly, subtopics that this attitude relates to (e.g., specific targets of criticism or support). Subsequently, a (human) media analyst must be able to aggregate the extracted information by source, polarity or subtopics, allowing him to build support/criticism networks etc. Recent advances in language technology, especially in *sentiment analysis*, promise to (partially) automate this task.

Sentiment analysis is often considered in the context of the following two tasks: *sentiment extraction* (identify subjective phrases/ sentence in a document) and *sentiment retrieval* (identify/rank documents with subjective attitude on a topic).

How can technology developed for sentiment analysis be applied to media analysis? In order to use a *sentiment extraction* system for a media analysis problem, a system would have to be able to determine which of the extracted sentiments are actually relevant, i.e., it would not only have to identify specific targets of all extracted sentiments, but also decide which of the targets are relevant for the topic at hand. This is a difficult task, as the relation between a *topic* (e.g., a movie) and specific targets of sentiments (e.g., acting or special effects in the movie) is not always straightforward, in the face of ubiquitous complex linguistic phenomena such as referential expressions ("… this beautifully shot *documentary*") or bridging anaphora ("the *director* did an excellent jobs").

In *sentiment retrieval*, on the other hand, the topic is initially present in the task definition, but it is left to the user to identify sources and targets of sentiments, as systems typically return a list of documents ranked by relevance and opinionatedness. To use a traditional sentiment retrieval system in media analysis, one would still have to manually go through ranked lists of documents returned by the system.

To be able to support media analysis, we need to combine the specificity of (phrase- or word-level) sentiment analysis with the topicality provided by sentiment retrieval. Moreover, we should be able to identify sources and specific targets of opinions.

In order to move towards the requirements of media analysis, in this paper we focus on two of the problems identified above: (1) pinpointing evidence for a system's decisions about the presence of sentiment in text, and (2) identifying specific targets of sentiment.

We address these problems by introducing a special type of lexical resource: a topic-specific subjectivity lexicon that indicates specific relevant targets for which sentiments may be expressed; for a given topic, such a lexicon consists of pairs (*syntactic clue*, *target*). We present a method for automatically generating a topic-specific lexicon for a given topic and query-biased set of documents. We evaluate the quality of the lexicon both manually and in the setting of an opinionated blog post retrieval task. We demonstrate that such a lexicon is highly *focused*, allowing one to effectively pinpoint evidence for sentiment, while being competetive with traditional subjectivity lexicons consisting of (a large number of) clue words.

Unlike other methods for topic-specific sentiment analysis, we do not expand a seed lexicon. Instead, we make an existing lexicon more focused, so that it can be used to actually pin-point subjectivity in documents relevant to a given topic.

2. GENERATING TOPIC-SPECIFIC LEXICONS

In this section we describe how we generate a lexicon of subjectivity clues and targets for a given *topic* and a list of *relevant documents* (e.g., retrieved by a search engine for the topic). As an additional resource, we use a large background corpus of text documents of a similar style but with diverse subjects; we assume that the relevant documents are part of this corpus as well. As the background corpus, we used the set of documents from the assessment pools of TREC 2006–2008 opinion retrieval tasks (described in detail in section 3). We use the Stanford lexicalized parser to extract labeled dependency triples (*head, label, modifier*). In the extracted triples, all words indicate their category (*noun, adjective, verb, adverb,* etc.) and are normalized to lemmas.

Figure 1 provides an overview of our method.

3. DATA AND EXPERIMENTAL SETUP

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DIR '11 Amsterdam, The Netherlands
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

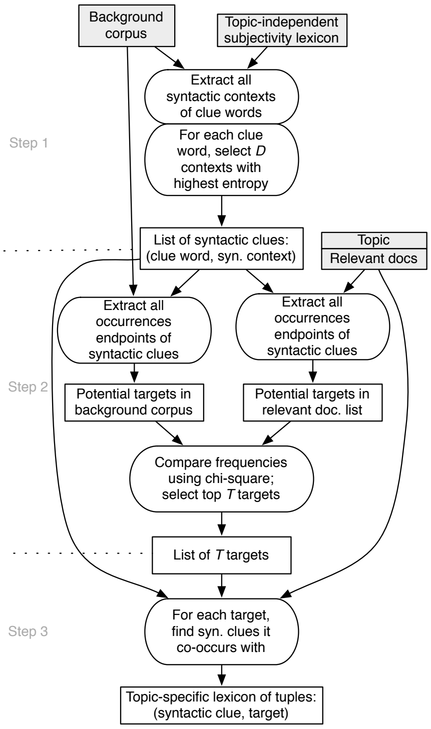

Background corpus
Topic-independent subjectivity lexicon
Extract all syntactic contexts of clue words
Step 1
For each clue word, select *D* contexts with highest entropy
List of syntactic clues: (clue word, syn. context)
Topic
Relevant docs
Extract all occurrences endpoints of syntactic clues
Extract all occurrences endpoints of syntactic clues
Step 2
Potential targets in background corpus
Potential targets in relevant doc. list
Compare frequencies using chi-square; select top *T* targets
List of *T* targets
Step 3
For each target, find syn. clues it co-occurs with
Topic-specific lexicon of tuples: (syntactic clue, target)

Figure 1: Our method for learning a topic-dependent subjectivity lexicon.

For extrinsic evaluation we apply our lexicon generation method to a collection of documents containing opinionated utterances: the TREC Blog06 collection.

TREC 2006–2008 came with the task of *opinionated blog post retrieval*. For each year a set of 50 topics was created, giving us 150 topics in total. Every topic comes with a set of relevance judgments: Given a topic, a blog post can be either (i) nonrelevant, (ii) relevant, but not opinionated, or (iii) relevant and opinionated. TREC topics consist of three fields (*title*, *description*, and *narrative*), of which we only use the *title* field: a query of 1–3 keywords.

4. QUANTITATIVE EVALUATION OF LEXICONS

In this section we assess the quality of the generated topic-specific lexicons numerically and extrinsically. To this end we deploy our lexicons to the task of opinionated blog post retrieval. A commonly used approach to this task works in two stages: (1) identify topically relevant blog posts, and (2) classify these posts as being opinionated or not. In stage 2 the standard approach is to rerank the results from stage 1, instead of doing actual binary classification. We take this approach, as it has shown good performance in the past TREC editions and is fairly straightforward to implement. For all experiments we use the collection described in Section 3.

Our experiments have two goals: to compare the use of topic-independent and topic-specific lexicons for the opinionated post retrieval task, and to examine how different settings for the parameters of the lexicon generation affect the empirical quality.

4.1 Reranking using a lexicon

To rerank a list of posts retrieved for a given topic, we opt to use the method that showed best performance at TREC 2008. The approach taken by Lee et al. (2008) linearly combines a (topical) relevance score with an opinion score for each post. In addition to using Okapi BM25 for opinion scoring, we also consider a simpler method:a simple count of lexicon matches in a document.

4.1.1 Results and observations

There are several parameters that we can vary when generating a topic-specific lexicon and when using it for reranking: the number of syntactic contexts per clue, the number of extracted targets, the opinion scoring function, the weight of the opinion score in the linear combination with the relevance score.

First, we note that reranking using all lexicons significantly improves over the relevance-only baseline for all evaluation measures. When comparing topic-specific lexicons to the topic-independent one, most of the differences are not statistically significant, which is surprising given the fact that most topic-specific lexicons we evaluated are substantially smaller.

The only evaluation measure where the topic-independent lexicon consistently outperforms topic-specific ones, is Mean Reciprocal Rank that depends on a single relevant opinionated document high in a ranking. A possible explanation is that the large general lexicon easily finds a few “obviously subjective” posts (those with heavily used subjective words), but is not better at detecting less obvious ones, as indicated by the recall-oriented MAP and R-precision.

Interestingly, increasing the number of syntactic contexts considered for a clue word (parameter D) and the number of selected targets (parameter T) leads to substantially larger lexicons, but only gives marginal improvements when lexicons are used for opinion retrieval. This shows that our bootstrapping method is effective at filtering out non-relevant sentiment targets and syntactic clues.

The evaluation results also show that the choice of opinion scoring function (Okapi or raw counts) depends on the lexicon size: for smaller, more focused lexicons unnormalized counts are more effective. This also confirms our intuition that for small, focused lexicons simple presence of a sentiment clue in text is a good indication of subjectivity, while for larger lexicons an overall subjectivity scoring of texts has to be used, which can be hard to interpret for (media analysis) users.

Acknowledgements

This research was supported by the European Union’s ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, and by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.-512, 612.061.814, 612.061.815, 640.004.802.

References

Jijkoun, V., de Rijke, M., and Weerkamp, W. (2010). Generating focused topic-specific sentiment lexicons. In *48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, Uppsala, Sweden. ACL, ACL.

Lee, Y., Na, S.-H., Kim, J., Nam, S.-H., Jung, H.-Y., and Lee, J.-H. (2008). KLE at TREC 2008 Blog Track: Blog Post and Feed Retrieval. In *Proceedings of TREC 2008*.

Query graphs analyzing for query similarity evaluation

Rushed Kanawati
LIPN - CNRS UMR 7030
University of Paris Nord
99 Av. J.B. Cément 93430
Villetaneuse, FRANCE
rushed.kanawati@lipn.univ-paris13.fr

ABSTRACT

Query similarity is a core function in many information retrieval applications. Different query similarity functions can be defined. However, no clear evaluation measurement of different query similarity functions is yet provided. In this paper we propose to evaluate the quality of a query similarity function by the quality of the induced graph defined as follows: Let $sim()$ be a query similarity function. We define a query graph over a query set induced by function $sim()$ as : $G^{\sigma}_{sim} =< \mathcal{Q}, E \subseteq \mathcal{Q} \times \mathcal{Q} >$ where $\mathcal{Q}$ is a set of queries and two queries $Q_i, Q_j \in \mathcal{Q}$ are linked if $sim(Q_i, Q_j) \geq \sigma$. σ is a given threshold. The intuition we are searching to confirm is that effective similarity functions induce scale-free similarity graphs.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

1. INTRODUCTION

Web search engines keep track of all received queries in *log* files. For each query Q submitted by a user u, we usually find the following information in the log:

1. Q^t the query processing time.
2. Q^u the user identifier. This is usually represented by the IP address of the machine that have sent the query. This limits seriously the usefulness of such an identifier for distinguishing real users[1]
3. Q^T is the set of the query terms. In this work we only consider simple queries composed of a set of words. No boolean operators (i.e. and, or, not) or filtering operators (i.e. near, language filtering, etc.) are considered.
4. Q^R is a ranked list of results returned by the search engine in answer to Q.
5. $Q^S \subseteq Q_R$ is a ranked list of results selected by the user among the list Q^R. For each selected document we may save also in the log file the selection time as well as the visualization time of the document.

Mining query log files has a quite a number of useful applications for enhancing web information retrieval. Main application fields are: search result personalization [13, 14, 22, 11], result re-ranking [21], query expansion and reformulation [18, 2]. , query recommendation [16, 23] and queries clustering and classification [16].

A core question in all the above mentioned application fields is how to define an effective query similarity function. Different similarities has been proposed in the scientific literature [6, 3]. These cover term-based similarities, result-based similarities, selection-based similarities and graph-bases similarities [3]. In each of the above mentioned types of similarities a wide variety of concretes similarities can be conceived. The problem we tackle in this work is the how to compare and evaluate different similarity functions in a task-independent way ? Actually, as far as we are aware, no clear methodology is given in the scientific literature for evaluating query similarity measures. Some partial work is given in [19, 6]. The idea we explore in this work is based on defining the concept of *similarity induced graph*. This is simply defined as follows. let $\mathcal{Q}$ be a set queries. Let $sim()$ be a query similarity function : $sim : \mathcal{Q} \times \mathcal{Q} \rightarrow [0, 1]$ et let $\sigma \in [0, 1]$ be a given threshold. The similarity induced query graph is then defined by: $G_{(sim,\sigma)} =< \mathcal{Q}, E \subseteq \mathcal{Q} \times \mathcal{Q} >$, where E denotes the set of links in the graph. Two queries Q_i, Q_j are linked if their similarity, as computed by $sim()$, is greater than σ. The intuition we want to confirm is that effective similarity functions induce a free-scale similarity graphs [20]. One of the major characteristics of free-scale graphs is that they exhibit a high clustering coefficient, as compared to random graphs of the same size [20]. The clustering coefficient defines the probability of having two neighbors of a random selected node linked in the graph. In social graphs, which are one of the most studied scale-free graphs, this can be expressed by the high probability of having " friends of friends are friends". This property is not natural in similarity induced graphs since similarity function in general are not transitive. We claim, that a similarity function inducing a scale-free graph over the set of queries would be an efficient similarity function.As a first step towards assessing this claim, we propose here to compute different similarities

[1] Mainly because of the wide use of Dynamic Host Configuration servers (DHCP) and Network Address Translation services (NAT).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 2011 ACM 978-1-60558-906-0/10/09 ...$10.00.

over a real dataset of query log and to examine if the re is any correlation between the scale-free nature of obtained similarity induced graphs and performances obtained by applying the correspondent similarity function in the context of a result re-ranking application [17]. **This constitutes no formal proof in any way**. However results we obtain allow us to be more confident in believing this intuition.

Next in section 2. we give a short review of the most used query similarity functions. Our approach for evaluating similarities is then described in detail in section 4. Experimental results and learned lessons are given and commented in section 5.

2. QUERY SIMILARITY FUNCTIONS

Query similarity metrics already proposed in the scientific literature fall into one of the four following categories [3, 7]. next we give some examples of similarity metrics used later in this work. The given list is not an exhaustive one.

Term-based similarities.

Query similarity is computed by evaluating the differences between the terms used in two queries. One first example is the classical Jaccard metric:

$$Jaccard-T(Q_i, Q_j) = \frac{|Q_i^T \cap Q_j^T|}{|Q_i^T \cup Q_j^T|} \tag{1}$$

Another metric, taking term's order into consideration is the edit distance metric:

$$Edit-T(Q_i, Q_j) = 1 - \frac{editDistance(C_i, C_j)}{max(len(C_i), len(C_j))} \tag{2}$$

where $editDistance$ is a function computing the minimal cost of transforming C_i into C_j applying atomic edition operations: adding and suppressing characters. $len(c)$ returns the length of the string c.

Result-based similarities.

These constitute a less direct way to compare two queries by examining result' sets returned by the same search engine in response to them. Queries may have no terms in common but have an important overlap in their result sets.

$$Jaccard-R(Q_i, Q_j) = \frac{|Q_i^R \cap Q_j^R|}{|Q_i^R \cup Q_j^R|} \tag{3}$$

A more sophisticated result-based similarity metrics can be conceived using URL similarity metric. A general formula would be the following:

$$Content{-}R(Q_i, Q_j) = \frac{\sum_{URL_i \in Q_i^R} \sum_{URL_j \in Q_j^R} simURL(URL^i, URL^j)}{|Q_i^R| * |Q_j^R|} \tag{4}$$

Where $simURL()$ is a basic URL similarity metric. A basic metric from computing similarities of URL contents is the classical $cosin()$ metric given by:

$$simURL(URL^i, URL^j) = \frac{\sum_{k=1}^{n}(w_k^i * w_k^j)}{\sqrt{\sum_{l=1}^{n}(w_l^i)^2} * \sqrt{\sum_{f=1}^{n}(w_f^j)^2}} \tag{5}$$

Where w_k^i is the wight of term w_k in the document indexed by URL_i. The term-vector representation of documents is generated using classical information retrieval techniques as described in [5].

Selection-based similarities.

These are basically the same as the result-based functions but applied only on documents selected by users from the whole set of results returned by the search engine. Next in this paper, this type of similarity function will not be considered since the target application we use is a result re-ranking approach in which result selection information is not available at time of similarity computing.

Graph-based similarity.

Different types of relations can be defined between two queries as described in [4]. These relations can be coded in form of a graph defined over the query set. Notice that these are different form similarity-induced graphs are introduced earlier in this paper. Relational graphs can then be used to detect similarities among queries in [8, 1, 9].

3. SCALE-FREE GRAPHS

Different graphs modeling real complex systems have been showed recently to exhibit a common set of features that distinguish them from pure random graphs [20]. Let $G =< V, E \subseteq V \times V >$ be a graph. scale-free graphs have the following main characteristics:

- **Small diameter**. The diameter of a graph is given by highest shortest distance between any couple of nodes. In scale-free graphs, this distance is very short compared to the number of nodes in the graph . In many real graphs the diameter is less than 6 stating that we can reach any node from any other nodes by making 6 hops at most. This is the main reason why lot os scale free graphs are also called small-world graphs [20].

- **Low density**. The density of non-oriented graph G is given by $d_G = \frac{|E|}{|V| \times (|V|-1)}$ the number of effective links over the number of possible links. In scale-free graphs little links do exist, compared to $|V|$ the number of nodesin the graph.

- **Power-law degree distribution** : the number of nodes that have K direct neighbors in the graph is proportional to $K^{-\alpha}$. For many real graphs we have $\alpha \in [2, 3]$ [15].

- **Hight clustering coefficient** The clustering coefficient is given by

$$cc(G) = \sum_{v \in V} \frac{2|E \cap (\Gamma(v) \times \Gamma(v))|}{d(v) \times (d(v) - 1)}$$

where $\Gamma(v)$ denotes the set of neighbors of node v in the graph. $d(v)$ denotes the degree of node v. This measure estimates the probability that two neighbors of a randomly selected node are linked directly.

4. QUERY SIMILARITY EVALUATION APPROACH

The approach we propose, for evaluating and comparing different query similarity metrics is structured into two main steps:

- Construct the similarity induced graphs. For each such graph, we compute the above mentioned measures characterizing scale-free graphs.

- Apply query similarity metric in the context of a web result re-ranking approach [17]. We search if there is any correlation between the performances obtained from applying a similarity metric and the characteristics of the similarity graph induced by the same similarity metric. Evaluating the obtained performances when applying a given query similarity metric.

The results re-ranking approach is based on mining the log of past processed queries. For each past query Q_i we compute a *voting function* $Q_i^V()$ that compute a permutation of Q_i^R such that Q_i^S is a prefix of $Q_i^V(Q_i^R)$. In other terms, the voting function can give the ranked result list selected by the user from the list of results returned by the search engine in answer to Q. Now having a target query Q_T, the system searches for past *similar* queries. Let k be the number of retrieved past similar queries. For each retrieved similar query we apply the voting function on Q_T^R. We obtain k potentially different permutations of Q_T^R. These different permutations are then merged to obtain the final re-raking of Q^R [12]. Hence, the re-ranking framework we propose is structured in three main hotsopts[2]: 1) The query similarity metric to use, 2) The voting function to apply and 3) the Permutation merging procedure to apply. Each of these steps can be implemented by a variety of technical approaches. In the current prototype, we have implemented four different query similarity metrics summarized in next table.

In the current implementation of the system, we apply a voting function inspired from the classical voting algorithm [10]. The voting function is implemented as follows: let Q_{target}^R be the set of results returned in answer to target query Q_{target}. Let Q_s be a past query similar to Q_{target}. Let $pos(r, Q_j^R)$ a function returning the rank of document r in the list of results Q_j^R. For each result $r \in Q_{target}^R$ we compute the following weight

$$w_r = \sum_{rs \in Q_s^R} simURL(rs, r) \times pos(rs, Q_s^R)$$

Where $simURL(r_i, r_j)$ is a given document similarity metric. Currently this is takes to be the classical cosine document similarity metric. The result of the voting function of past query Q_s is the list Q_{target}^R sorted in ascending order with respect to computed weights w_r. We apply, the original Borda voting algorithm [10] for merging voting results obtained from k similar past queries. We propose to evaluate the correctness of the re-ranking approach by the value of the edit similarity between the rank proposed by the system and the selection order performed by users (as registered in a log file).

5. EXPERIMENTATION

Experiments are conducted on a real query log file provided by Microsoft. Data follow the description of a classical query log file as described in section 1. In this experiment we use a set of 200000 queries. These contains 80800 distinct query terms and results are composed of 754000 distinct URLs. We have applied the above describes results reranking approach using foud different query similarity metrics: $Jaccard-T$, $Edit-T$, $Jaccard-R$, and $Content-R$. For each metric we vary the similarity threshold σ from 0.6 to 0.9. Characteristics of induced similarity graphs are given in table 1. In this table, diameter, density and power are those of the biggest connected component.

For all experiments a classical 3-cross validation approach is applied: the query log is divided into three folds; 2 are used as a learning set and the third as a validation set. Each experiment is repeated three times by changing each round the selected learning/validation folds. Average results of three rounds are given in table1 (last colmoun).

We clearly found that result-based similarities outperform term-based ones. And that result-based similarity induced graphs exhibit more scale-free features. Result-content based similarities give a slightly more enhanced results that results overlap similarity. Again results-content graphs is more similar to scale-free graphs (especially in terms of clustering coefficient which a major metric for characterizing scale-free graphs [20]). These results enforce our intuition that effective query similarity metric induce scale-free similarity graphs.

6. CONCLUSION

In this work we've propose a new approach for evaluating query similarity metrics that can be applied independently for the type of the target application. First experiments, reported here, show that effective similarity metrics define also a scale-free like graphs. Obviously, current experimentation does not allow to generalize these findings. More experimentations are needed in order to take into account other types of similarity metrics as well as other types of information retrieval related tasks (other than results re-ranking).

7. REFERENCES

[1] Baeza-Yates, R., Tiberi, A.: The anatomy of a large query graph. JOURNAL OF PHYSICS A: MATHEMATICAL AND THEORETICAL 41, 1–13 (2008)

[2] Baeza-Yates, R.A.: Applications of web query mining. In: ECIR. pp. 7–22 (2005)

[3] Baeza-Yates, R.A.: Graphs from search engine queries. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plasil, F. (eds.) SOFSEM (1). Lecture Notes in Computer Science, vol. 4362, pp. 1–8. Springer (2007)

[4] Baeza-Yates, R.A.: Mining queries. In: Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A. (eds.) ECML/PKDD. Lecture Notes in Computer Science, vol. 4702, p. 4. Springer (2007)

[5] Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press / Addison-Wesley (1999)

[6] Balfe, E., Smyth, B.: An analysis of query similarity in collaborative web search. In: ECIR. pp. 330–344 (2005)

[7] Balfe, E., Smyth, B.: A comparative analysis of query similarity metrics for community-based web search. In: Muñoz-Avila, H., Ricci, F. (eds.) ICCBR. Lecture

[2] In a component framework a hot spot is a place where adaptations can occur.

Table 1: Characteristics of induced similarity graphs with obtained re-ranking correctness

Similarity	Threshold	Density	Clustering Coeff.	Diameter	# connected components	Power	Re-ranking
Jaccard-T	0.6	1.29E-04	0.506	22	16 383	1.581	0.459
	0.7	1.93E-04	0.541	1	13 621	1.102	0.501
	0.8	2.29E-04	0.561	1	12 259	0.997	0.511
	0.9	2.37E-04	0.574	1	11 975	0.992	0.512
Edit-T	0.6	1.43E-04	0.455	25	7 646	2.003	0.406
	0.7	1.03E-04	0.459	51	15 286	2.0659	0.430
	0.8	1.50E-04	0.530	11	15 652	1.547	0.472
	0.9	2.14E-à4	0.577	1	12 936	1.259	0.499
Jaccard-R	0.6	3.40E-04	0.784	6	17 250	1.498	0.420
	0.7	2.14E-04	0.668	6	15 348	1.95	0.461
	0.8	1.65E-04	0.548	5	11 844	2.027	0.514
	0.9	1.76E-04	0.399	7	6 646	1.918	0.609
Content-R	0.6	3.23E-03	č0.835	5	987	1.28	0.421
	0.7	5.29E-03	0.809	3	507	1.287	0.439
	0.8	1.15E-02	0.731	2	237	1.032	0.501
	0.9	3.15E-02	0.684	2	110	0.849	0.754

Notes in Computer Science, vol. 3620, pp. 63–77. Springer (2005)

[8] Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: Shanahan, J.G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D.A., Kolcz, A., Choi, K.S., Chowdhury, A. (eds.) CIKM. pp. 609–618. ACM (2008)

[9] Boldi, P., Bonchi, F., Castillo, C., Donato, D., Vigna, S.: Query suggestions using query-flow graphs. In: Workshop on Web Search Click Data. pp. 56–63. ACM Press, Barcelona, Spain (2009)

[10] Borda, J.: Mémoire sur les élections au scrutin. Comptes rendus de l'Académie des sciences, traduit par Alfred de Grazia comme Mathematical Derivation of a election system , Isis, vol 44, pp 42-51 (1781)

[11] Boydell, O., Smyth, B.: Enhancing case-based, collaborative web search. In: Weber, R., Richter, M.M. (eds.) ICCBR. Lecture Notes in Computer Science, vol. 4626, pp. 329–343. Springer (2007)

[12] Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: WWW. pp. 613–622 (2001)

[13] Fitzpatrick, L., Dent, M.: Automatic feedback using past queries: Social searching? In: SIGIR. pp. 306–313. ACM (1997)

[14] Freyne, J., Smyth, B., Coyle, M., Balfe, E., Briggs, P.: Further experiments on collaborative ranking in community-based web search. Artif. Intell. Rev. 21(3-4), 229–252 (2004)

[15] Guillaume, J.L., Latapy, M.: Bipartite graphs as models of complex networks. 371, pages, 2006. Physica A 37(1), 795–813 (2006)

[16] Hosseini, M., Abolhassani, H., Harikandeh, M.S.: Clustering search engine log for query recommendation. In: Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization. pp. 201–206 (2007)

[17] Kanawati, R.: A cbr framework for implementing community-aware web search engine. In: Proceedings of Second Internbational Workshop on Adfpative Information Retrieval (AIR-08). pp. 32–38 (2008)

[18] Kanawati, R., Jaczynski, M., Trousse, B., Anderloi, J.M.: Applying the Broadway recommendation computation approach for implementing a query refinement service in the CBKB meta-search engine. In: Trousse, B., Mille, A. (eds.) Conférence français sur le raisonnement à partir de cas (RàPC'99). pp. 17–26. AFIA, Palaiseau, France (june 1999)

[19] Krömer, P., Snásel, V., Platos, J.: Comparing query similarity measures for collaborative web search. In: Pan, J.S., Abraham, A., Chang, C.C. (eds.) ISDA (2). pp. 383–388. IEEE Computer Society (2008)

[20] Li, L., Alderson, D., Doyle, J.C., Willinger, W.: Towards a theory of scale-free graphs: Definition, properties, and implications. Internet Mathematics 2(4), 431–523 (Mar 2006), `http://netlab.caltech.edu/publications/IM06.pdf`

[21] Rohini, U., Varma, V.: A novel approach for re-ranking of search results using collaborative filtering. In: ICCTA. pp. 491–496. IEEE Computer Society (2007)

[22] Tan, B., Shen, X., Zhai, C.: Mining long-term search history to improve search accuracy. In: Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopulos, D. (eds.) KDD. pp. 718–723. ACM (2006)

[23] Yang, X., Procopiuc, C.M., Srivastava, D.: Recommending join queries via query log analysis. In: ICDE. pp. 964–975. IEEE (2009)

How Different are Language Models and Word Clouds?

Rianne Kaptein[1] Djoerd Hiemstra[2] Jaap Kamps[1,3]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] Database Group, University of Twente
[3] ISLA, Informatics Institute, University of Amsterdam

ABSTRACT

Word clouds are a summarised representation of a document's text, similar to tag clouds which summarise the tags assigned to documents. Word clouds are similar to language models in the sense that they represent a document by its word distribution. In this paper[1] we investigate the differences between word cloud and language modelling approaches, and specifically whether effective language modelling techniques also improve word clouds. We evaluate the quality of the language model and the resulting word clouds using a system evaluation test bed, and a user study. Our experiments show that different language modelling techniques can be applied to improve a standard word cloud that uses a TF weighting scheme in combination with stopword removal. Including bigrams in the word clouds and a parsimonious term weighting scheme are the most effective in both the system evaluation and the user study.

1. INTRODUCTION

This paper investigates the connections between tag or word clouds popularised by Flickr and other social web sites, and the language models as used in IR. The new generation of the Internet, the social Web, allows users to do more than just retrieve information and engages users to be active. Users can now add tags to categorise web resources and retrieve their own previously categorised information. By sharing these tags among all users large amounts of resources can be tagged and categorised. These generated user tags can be visualised in a tag cloud where the importance of a term is represented by font size or colour. Of course, the majority of documents on the web are not tagged by users. An alternative to clouds based on user-assigned tags, is to generate clouds automatically by using statistical techniques on the document contents, so-called 'word clouds'. Figure 1 shows a word cloud summarising 10 documents. Our main research question is: do words extracted by language modelling techniques correspond to the words that users like to see in word clouds?

2. EXPERIMENTS

Since there is no standard evaluation method for word clouds, we created our own experimental test bed. Our experiments comprise of two parts, a system evaluation and a user study. For both experiments we use query topics from the 2008 TREC Relevance Feedback track. The system evaluation consists of two parts, first we test if adding the word cloud as a whole to the original query leads to improvements in retrieval performance. Secondly, for each topic we generate 25 queries where in each query one word from the word cloud is added to the original query. For each query we measure the difference in performance caused by adding the expansion term to the original query, words are considered relevant if adding the word leads to an improvement in retrieval performance, words are considered acceptable if there is no large decrease (more than 25%) in retrieval results. In the user study test persons rank different groups of word clouds. The 13 test persons consisted of 4 females and 9 males with ages ranging from 26 to 44 and were recruited at the university.

Table 1: Effectiveness of unigrams and bigrams

Approach	MAP	P10	% Rel. words	% Acc. words
Unigrams	0.2575	0.5097	**35**	**73**
Mixed	**0.2706**$^{-}$	**0.5226**$^{-}$	31	71
Bigrams	0.2016°	0.4387^{-}	25	71

Table 2: Effectiveness of term weighting approaches

Approach	MAP	P10	% Rel. words	% Acc. words
TF	0.2575	0.5097	**35**	**73**
TFIDF	0.1265$^{\bullet}$	0.3839°	22	67
Pars.	**0.2759**$^{\circ}$	**0.5323**$^{-}$	31	68

Clouds from Pseudo Relevant and Relevant Results

First, we compare a TF cloud made from 10 pseudo-relevant documents to a cloud of 100 relevant documents. We make this comparison to get some insights on the question whether there is a mismatch between words that improve retrieval performance, and words that users like to see in a word cloud. Our standard word cloud (shown in Figure 1) uses pseudo-relevant results. The cloud in Fig. 2 is based on 100 pages judged as relevant.

When we look at the system evaluation the relevant documents lead to better performance than the pseudo-relevant documents. The test persons in our user study however clearly prefer the clouds based on 10 pseudo-relevant documents: 66 times the pseudo-relevant cloud is preferred, 36 times the relevant cloud, and in 27 cases there is no preference (significant at 95% using a two-tailed sign-test). There seem to be three groups of words that often contribute positively to retrieval results, but are probably not appreciated by test persons: numbers, general and frequently occurring words which do not seem specific to the query topic e.g. 'year' or 'up', words that test persons don't know like abbreviations or technical terms .

Non-Stemmed and Conflated Stemmed Clouds

We look at the impact of stemming by generating conflated stemmed

[1] This paper is a compressed version of Kaptein, R., Hiemstra, D., and Kamps, J. (2010). How different are language models and word clouds? In Advances in Information Retrieval: 32nd European Conference on IR Research (ECIR 2010), volume 5993 of LNCS, pages 556-568. Springer.

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.
.

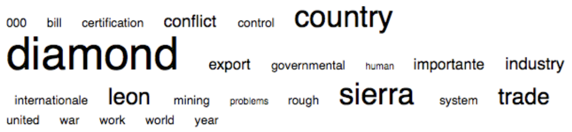


Figure 1: Word cloud from 10 results for the topic "diamond smuggling"

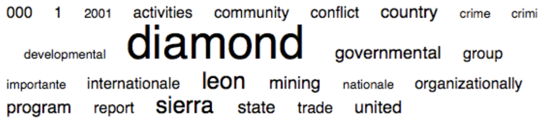


Figure 2: Word cloud from 100 relevant results

clouds. To stem, we use the most common English stemming algorithm, the Porter stemmer [2]. To visualize terms in a word cloud, Porter word stems are not a good option. A requirement for the word clouds is to visualize correct English words, and not stems of words which are not clear to the user, therefore in our conflated word clouds, word stems are replaced by the most frequently occurring word in the collection that can be reduced to that word stem. The effect of stemming is only evaluated in the user study. Looking at pairwise preferences, we see that there is no significant preference for the conflated cloud or the non-stemmed cloud. Often the difference between the clouds is so small that it is not noticed by test persons.

Bigrams

For users, bigrams are often easier to interpret than single words, because a little more context is provided. We have created two models that incorporate bigrams, a mixed model that contains a mix of unigrams and bigrams, and a bigram model that consists solely of bigrams. For the user study we placed bigrams between quotes to make them more visible as can be seen in Figure 3. In Table 1 the system evaluation results are shown. For query expansion, the model that uses a mix of unigrams and bigrams performs best. Using only bigrams leads to a significant decrease in retrieval results compared to using only unigrams. Looking at the percentages of relevant and acceptable words, the unigram model produces the most relevant words. The mixed model performs almost as good as the unigram model.

In the user study, the clouds with mixed unigrams and bigrams and the clouds with only bigrams are selected most often as the best cloud. There is no significant difference in preference between mixed unigrams and bigrams, and only bigrams. Users do indeed like to see bigrams, but for some queries the cloud with only bigrams contains too many meaningless bigrams such as 'http www'. An advantage of the mixed cloud is that the number of bigrams in the cloud is flexible. When bigrams occur often in a document, also many will be included in the word cloud.

Term Weighting

Besides the standard TF weighting we investigate two other variants of language models to weigh terms, the TFIDF model and the parsimonious model. Before weighting terms we always remove an extensive stopword list consisting of 571 common English words. In the TFIDF algorithm, the text frequency (TF) is now multiplied by the inverse document frequency (IDF).

The third variant of our term weighting scheme is a parsimonious model [1]. The parsimonious language model concentrates the probability mass on fewer words than a standard language model.

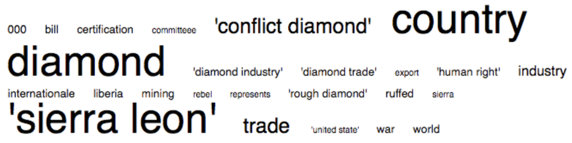


Figure 3: Word cloud of 10 results with mixed unigrams and bigrams

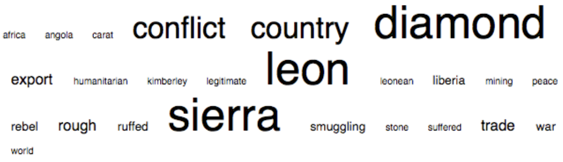


Figure 4: Word cloud of 10 results with parsimonious term weighting.

In Figure 4 the parsimonious word cloud of our example topic is shown. Table 2 shows the system evaluation results for the different term weighting schemes.

The parsimonious model performs best on both early and average precision. The TFIDF model performs significantly worse than the TF and the parsimonious model. Our simplest model, the TF model, actually produces the highest number of relevant and acceptable words. The weighting scheme of the parsimonious model is clearly more effective than the TF model though, since for query expansion where weights were considered the parsimonious model performed better than the TF model.

In the user study the parsimonious model is preferred more often than the TF model, and both the parsimonious and the TF model are significantly more often preferred over the TFIDF model. The parsimonious model contains more specific and less frequently occurring words than the TF model.

3. CONCLUSION

This paper investigated the connections between word clouds and the language models as used in IR. We have investigated how we can create word clouds from documents and use language modelling techniques which are more advanced than only frequency counting and stopword removal. We find that different language modelling techniques can indeed be applied to create better word clouds. Including bigrams in the word clouds and a parsimonious term weighting scheme are the most effective improvements. We find there is some discrepancy between good words for query expansion selected by language modelling techniques, and words liked by users. This will be a problem when a word cloud is used for suggestion of query expansion terms. The problem can be partly solved by using a parsimonious weighting scheme which selects more specific and informative words than a TF model, but also achieves good results from a system point of view.

REFERENCES

[1] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings SIGIR'04*, pages 178–185. ACM Press, New York NY, 2004.

[2] M. Porter. An algorithm for suffix stripping. *Program*, 14(3): 130–137, 1980.

Entity Ranking using Wikipedia as a Pivot

Rianne Kaptein[1] Pavel Serdyukov[2] Arjen de Vries[2,3] Jaap Kamps[1,4]
kaptein@uva.nl p.serdyukov@tudelft.nl arjen@acm.org kamps@uva.nl

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam, The Netherlands
[2] Delft University of Technology, The Netherlands
[3] Centrum Wiskunde & Informatica, The Netherlands
[4] ISLA, Informatics Institute, University of Amsterdam, The Netherlands

ABSTRACT

In this paper we investigate the task of Entity Ranking on the Web[1] Searchers looking for entities are arguably better served by presenting a ranked list of entities directly, rather than a list of web pages with relevant but also potentially redundant information about these entities. Since entities are represented by their web homepages, a naive approach to entity ranking is to use standard text retrieval. Our experimental results clearly demonstrate that text retrieval is effective at finding relevant pages, but performs poorly at finding entities. Our proposal is to use Wikipedia as a pivot for finding entities on the Web, allowing us to reduce the hard web entity ranking problem to easier problem of Wikipedia entity ranking. Wikipedia allows us to properly identify entities and some of their characteristics, and Wikipedia's elaborate category structure allows us to get a handle on the entity's type.

1. INTRODUCTION

Just like in document retrieval, in entity ranking the document should contain topically relevant information. However, it differs from document retrieval on at least three points: i) returned documents have to represent an entity, ii) this entity should belong to a specified entity type, and iii) to create a diverse result list an entity should only be returned once. The main goal of this paper is to demonstrate how the difficult problem of web entity ranking can often be reduced to the easier task of entity ranking in Wikipedia.

Our proposal is to exploit Wikipedia as a pivot for entity ranking. For entity types with a clear representation on the web, like living persons, organisations, products, movies, we will show that Wikipedia pages contain enough evidence to reliably find the corresponding web page of the entity. For entity types that do not have a clear representation on the web, returning Wikipedia pages is in itself a good alternative. So, to rank (web) entities given a query we take the following steps:

1. Associate target entity types with the query
2. Rank Wikipedia pages according to their similarity with the query and target entity types
3. Find web entities corresponding to the Wikipedia entities

We evaluate our approach using the entity ranking test collection created in the TREC 2009 Entity Ranking track [1].

2. ENTITY RANKING ON THE WEB

To investigate whether the hard problem of web entity ranking can be in principle reduced to the easier problem of Wikipedia entity ranking we look at the coverage of relevant TREC entities in Wikipedia. We find that the overwhelming majority of relevant entities (160 out of 198) of the TREC 2009 Entity ranking track are represented in Wikipedia, and that 85% of the topics have at least one relevant Wikipedia page. We also find that with high precision and coverage relevant web entities corresponding to the Wikipedia entities can be found using Wikipedia's "external links", and that especially the first external link is a strong indicator for primary homepages.

Furthermore we examine the value of entity type information for entity retrieval in Wikipedia. We find that entity types are valuable retrieval cues. Automatically assigned entity types are effective, but less so than manually assigned types. We can exploit the structure of Wikipedia to significantly improve entity ranking effectiveness.

In the remainder of this section we examine our research question: Can we improve web entity retrieval by using Wikipedia as a pivot? We compare our entity ranking approach of using Wikipedia as a pivot to the baseline of full-text retrieval.

We experiment with three approaches for finding webpages associated with Wikipedia pages:
1. External links: Follow the links in the External links section of the Wikipedia page.
2. Anchor text: Take the Wikipedia page title as query, and retrieve pages from the anchor text index. A length prior is used here.
3. Combined: Since not all Wikipedia pages have external links, and not all external links of Wikipedia pages are part of the Clueweb category B collection, we can not retrieve webpages for all Wikipedia pages. In case less than 3 webpages are found, we fill up the results to 3 pages using the top pages retrieved using anchor text.

2.1 Experimental Setup

In this experimental section we discuss experiments with the TREC Entity Ranking topics. We use the Indri search engine. We have created separate indexes for the Wikipedia part and the Web part of the Clueweb Category B. Besides a full text index we have also created an anchor text index. On all indexes we applied the Krovetz stemmer, and we generated a length prior. All runs are created with a language model using Jelinek-Mercer smoothing with a collection λ of 0.15.

[1] This paper is a compressed version of Kaptein, R., Serdyukov, P., Kamps, J., and de Vries, A. P. (2010). Entity ranking using Wikipedia as a pivot. In Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010), pages 69-78. ACM Press, New York USA.

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.
.

Table 1: TREC Web Entity Ranking Results

	Full Text	Wikipedia	
Run		Link	Cat+Link
Rel. WP	**73**	**73**$^{-}$	57°
Rel. HP	**244**	69$^{\bullet}$	70$^{\bullet}$
Rel. All	**316**	134$^{\bullet}$	121$^{\bullet}$
NDCG Rel. WP	**0.2119**	**0.2119**$^{-}$	0.1959^{-}
NDCG Rel. HP	**0.1919**	0.0820$^{\bullet}$	0.0830$^{\bullet}$
NDCG Rel. All	**0.2394**	0.1429$^{\bullet}$	0.1542$^{\bullet}$
Primary WP	78	78^{-}	**96**$^{\bullet}$
Primary HP	6	29°	**34**$^{\circ}$
Primary All	86	107°	**130**$^{\bullet}$
P10 pr. WP	0.1200	0.1200^{-}	**0.1700**$^{\circ}$
P10 pr. HP	0.0050	0.0300°	**0.0400**$^{\bullet}$
P10 pr. All	0.1200	0.1300^{-}	**0.1850**$^{\bullet}$
NDCG pr. WP	0.1184	0.1184^{-}	**0.1604**$^{\bullet}$
NDCG pr. HP	0.0080	0.0292^{-}	**0.0445**$^{\circ}$
NDCG pr. All	0.1041	0.1292^{-}	**0.1610**$^{\bullet}$

Significance of increase or decrease over full text according to t-test, one-tailed, at significance levels 0.05($^{\circ}$), and 0.01($^{\bullet}$).

Our baseline run uses standard document retrieval on a full text index. The result format of the TREC entity ranking runs differs from the general TREC style runs. One result consists of one Wikipedia page, and can contain up to three webpages from the non-Wikipedia part of the collection. The pages in one result are supposed to be pages representing the same entity.

For our baseline runs we do not know which pages are representing the same entity. In these runs we put one homepage and one Wikipedia page in each result according to their ranks, they do not necessarily represent the same entity. The Wikipedia based runs contain up to three homepages, all on the same entity. When a result contains more than one primary page, it is counted as only one primary page, or rather entity found.

2.2 Experimental Results

Recall from the above that the ultimate goal of web entity ranking is to find the homepages of the entities (called primary homepages). There are 167 primary homepages in total (an average of 8.35 per topic) with 14 out of the 20 topics having less than 10 primary homepages. In addition, the goal is to find an entity's Wikipedia page (called a primary Wikipedia page). There are in total 172 primary Wikipedia pages (an average of 8.6 per topic) with 13 out of the 20 topics having less than 10 primary Wikipedia entities.

The results for the TREC Entity Ranking track are given in Table 1. Our baseline is full text retrieval, which works well (NDCG 0.2394) for finding relevant pages. It does however not work well for finding primary Wikipedia pages (NDCG 0.1184). More importantly, it fails miserably for finding the primary homepages: only 6 out of 167 are found, resulting in a NDCG of 0.0080 and a P10 of 0.0050. Full text retrieval is excellent at finding relevant information, but it is a poor strategy for finding web entities.

We now look at the effectiveness of our Wikipedia-as-a-pivot runs. The Wikipedia runs in this table use the external links to find homepages. The second column is based on the baseline Wikipedia run, the third column is based on the run that uses the manual categories that proved effective for entity ranking on Wikipedia. Considering primary pages, we find more primary Wikipedia pages, translating into a significant improvement of retrieval effectiveness (up to a P10 of 0.1700, and a NDCG of 0.1604). Will this also translate into finding more primary homepages? The first run is a straightforward run on the Wikipedia part of ClueWeb, using the external links to the Web (if present). Recall that we established that primary pages linked from relevant Wikipedia pages have a high precision. This strategy finds 29 primary homepages (so 11 more than the baseline) and improves retrieval effectiveness to an NDCG of 0.0292, and a P10 of 0.0300. The second run using the Wikipedia category information improves significantly to find 34 primary homepages with a NDCG of 0.0445 and a P10 of 0.0400.

Table 2: TREC Homepage Finding Results

Run	Cat+Link	Anchor	Comb.
Rel. HP	70	127	**137**
Rel. All	121	178	**188**
NDCG Rel. HP	0.0830	0.0890	**0.1142**
NDCG Rel. All	0.1542	0.1469	**0.1605**
Primary HP	34	29	**56**
Primary All	130	125	**152**
P10 pr. HP	0.0400	0.0450	**0.0550**
P10 pr. All	**0.1850**	0.1750	**0.1850**
NDCG pr. HP	0.0445	0.0293	**0.0477**
NDCG pr. All	0.1041	0.1472	**0.1610**

Recall again that the external links have high precision but low recall. We try to find additional links between retrieved Wikipedia pages and the homepages by querying the anchor text index with the name of the found Wikipedia entity. This has no effect on the found Wikipedia entities, so we only discuss the primary homepages as presented in Table 2. Ignoring the existing external links, searching for the Wikipedia entities in the anchor text leads to 29 primary homepages. The combined run, supplementing the existing external links in Wikipedia with the automatically generated links, finds a total of 56 primary homepages. For homepages this improves the P10 over the baseline to 0.0550, and NDCG to 0.0447.

3. CONCLUSION

This paper investigates the problem of entity retrieval on the Web. Our main findings are the following. Our first finding is that, in principle, the problem of web entity ranking can be reduced to Wikipedia entity ranking. We found that the majority of entity ranking topics in our test collections can be answered using Wikipedia, and that with high precision relevant web entities corresponding to the Wikipedia entities can be found using Wikipedia's 'external links'. Our second finding is that we can exploit the structure of Wikipedia to improve entity ranking effectiveness. Entity types are valuable retrieval cues in Wikipedia. Automatically assigned entity types are effective, and almost as good as manually assigned types. Our third finding is that web entity retrieval can be significantly improved by using Wikipedia as a pivot. Both Wikipedia's external links and the enriched Wikipedia entities with additional links to homepages are significantly better at finding primary web homepages than standard text retrieval.

REFERENCES

[1] K. Balog, A. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 entity track. In *The Eighteenth Text REtrieval Conference (TREC 2009) Notebook*. National Institute for Standards and Technology, 2009.

What is the Importance of Anchor Text for Ad Hoc Search?

Marijn Koolen[1] Jaap Kamps[1,2]

[1] Archives and Information Studies, University of Amsterdam, The Netherlands
[2] ISLA, Informatics Institute, University of Amsterdam, The Netherlands
{m.h.a.koolen,kamps}@uva.nl

ABSTRACT

It is generally believed that propagated anchor text is very important for effective Web search, but many years of TREC Web retrieval research failed to establish the effectiveness of link evidence for ad hoc retrieval on Web collections. In this paper we use the new TREC 2009 Web Track collection to study the impact of collection size and link density on the effectiveness of anchor-text for Web ad hoc retrieval. Our main findings are that anchor-text outperforms full-text retrieval in terms of early precision and an improvement in overall precision when combined with it. Other findings are that, contrary to expectations, link density has little impact on effectiveness, while the size of the collection has a substantial impact on the quantity, quality and effectiveness of anchor text. This paper is based on [6].

1. INTRODUCTION

The use of anchor text for Web retrieval is well studied, with the broad conclusion that it is very effective for finding entry pages of sites–often outperforming approaches based on document text alone–but not for ad hoc search. Some speculated that the number of (inter-server) links in the TREC collections was too low and that the collections might be too small for anchors to be effective [3]. Others pointed at the difference between traditional ad hoc retrieval studied at TREC and actual Web search. Web searchers tend to "prefer the entry page of a well-known topical site to an isolated piece of text, no matter how relevant" [4]. Although the switch to more Web-centric search tasks like home page and named page finding showed link information to be very effective [2, 7], there is no clear explanation of why anchor text is not effective for ad hoc retrieval. To study the value of link information, Gurrin and Smeaton [3] suggested a representative test-collection needs to be sufficiently large and have sufficiently high inter- and intra-server link densities. At the TREC 2009 Web Track [1] a new, large Web collection—ClueWeb09—was introduced, which is much larger than previous collections and was crawled to reflect Tier 1 of a commercial search engine, so has a relatively dense link structure, urging us to revisit the question:

- What is the importance of anchor text for ad hoc search?

2. INITIAL EXPERIMENTS

We indexed the ClueWeb09 category B, which is a 50 million pages subset of the full ClueWeb09, using Indri with Krovetz stemming and stopword removal. We created two indexes, a *full-text* index and an *anchor text* index containing only the propagated anchor text of ClueWeb09 B. The full-text and anchor text runs use the Indri language model approach and linear smoothing with $\lambda_{collection} = 0.15$. Documents are scored using the document length as a prior probability $p(d) = \frac{|d|}{|D|}$, where d is a document in collection D. We also made a mixture run, combining the full-text and anchor runs using the weighting $S_{mix}(d) = 0.7 \cdot S_{full}(d) + 0.3 \cdot S_{anchor}(d)$.

Table 1: Results for the 2009 Adhoc Task. Significant differences ($p > 0.95$, denoted °) are with respect to the full text run

	Full collection		No Wikipedia	
Run	**statMAP**	**MPC(30)**	**statMAP**	**MPC(30)**
Text	0.1442	0.3079	0.1038	0.2557
Anchor	0.0567	**0.5558**	0.0617	0.4289
Mix	**0.1643**°	0.4812°	**0.1213**	**0.4773**
Text · In-degree	0.1098	0.2694	0.0746	0.2059
UDWAxQEWeb	0.1999	0.5010	–	–
uogTrdphCEwP	0.2072	0.4966	–	–
ICTNETADRun4	0.1746	0.4368	–	–

2.1 Results

The results are shown in Table 1. We test for significant changes with respect to the full-text baseline using a one-tailed bootstrap test with 100,000 resamples. The *Anchor* run has a low statMAP compared to the *Text* run. A possible explanation is that many pages in the collection have no or few incoming links, including many relevant pages. In contrast, anchor text is effective for early precision. The *Anchor* run scores better on MPC(30) than the *Text* run and supports the above explanation for its low statMAP score. More importantly, the *Mix* run leads to significant improvements in statMAP showing that the two indexes are complementary and that Web structure can be used to improve ad hoc search. To put this into perspective, we compared them against the top 3 groups of the TREC 2009 Web Ad hoc task (according to MPC(30), bottom 3 rows). The runs of the top 3 groups score substantially better on statMAP, but lower on MPC(30). This shows that anchor text alone can meet or exceed the precision of the top-performing systems.

Perhaps anchor text is more effective than in previous TREC experiments because this collection contains the full Wikipedia, which has a dense link structure and many anchors matching the titles of the target pages. Columns 4 and 5 in Table 1 show the results of these runs. The *Anchor* run still has higher early precision and the *Mix* run still has higher statMAP than the *Text* run. Wikipedia is not the reason for the effectiveness of anchor text. In sum, this new Web collection finally shows the long expected value of Web link structure for ad hoc search.

Copyright is held by the author/owner(s).
DIR'10, 4 Feruary, 2011, Amsterdam, Netherlands.
.

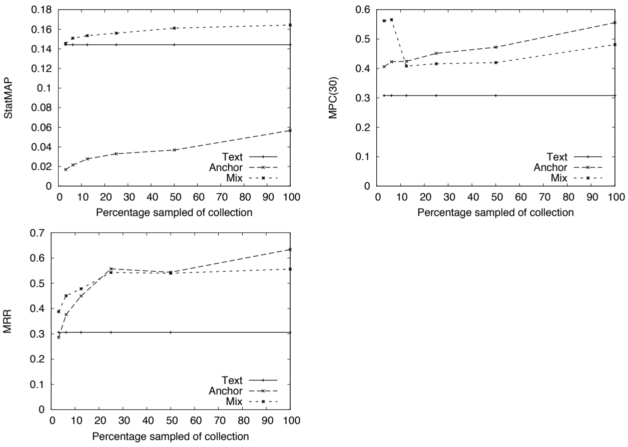


Figure 1: Impact of link sampling on effectiveness of full-text, anchor text and mixture runs.

3. WHY ANCHOR TEXT WORKS

In this section we seek to understand what makes the anchor text representation effective. We look at the impact of link density and collection size, which we do by down-sampling either links or pages.

3.1 The Impact of Link Density

We filter links by randomly selecting n% of all documents and removing their outgoing links. If we randomly sample 50% of the pages and remove the outgoing links of those pages, we would expect to end up with roughly 50% of all the links. The impact of sampling links on the effectiveness of full-text and anchor text is shown in Figure 1. The full-text index is not affected by link sampling, hence the straight line in the figures. The statMAP (top left) of the *Anchor* run slowly decreases as we remove more links because the index covers fewer pages. The *Mix* run scores better at statMAP with even the smallest samples of links, indicating that even very few links can improve the *Text* run. The MPC(30) scores (top right) of the *Anchor* run stay well above the *Text* score. We note that below 12.5% of the links (less than 3 incoming links per page), the density is well below the link densities of earlier TREC Web collections. The impact of link density seems small. To rule out that the MPC(30) score is over-estimated we transformed the relevance judgements to traditional binary judgements and looked at the Mean Reciprocal Rank (MRR, bottom left of Figure 1), which cannot over-estimate. It supports that anchor text gives better early precision than full-text. Link density plays a role at low densities, but its impact stabilises quickly.

3.2 The Impact of Collection Size

Next, we look at the impact of the collection size. We randomly remove n% pages from the collection, and thereby lose both the outgoing and incoming links of those pages. Thus, if we sample 50% of the pages, we remove more than 50% of the links. One of the favourable aspects of randomly sampling pages is that the probability of relevance is unaffected [5]. The impact of sampling pages on the effectiveness of full-text and anchor text is shown in Figure 2. The statMAP (left figure) of the *Text* run goes up slowly—possibly due to losing topics with little relevance—while for the *Anchor* run it goes down slowly. The *Text* run gains precision at rank 30 (MPC(30), right figure) as the collections grows, as predicted [5]. The anchor text precision is more affected by collection size. With half the collection, anchor text is nowhere near as effective as full-text. With fewer relevant documents left, and an increasingly smaller coverage of the collection, it becomes harder to find relevant pages through anchor text. For precision at a fixed cut-off, the impact of the collection size is much larger for anchor text than for full-text.

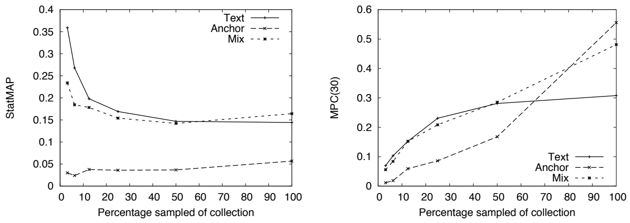


Figure 2: Impact of page sampling on effectiveness of full-text, anchor text and mixture runs.

4. CONCLUSIONS

Our main finding is that in contrast with earlier results, the anchor text leads to significant improvements in retrieval effectiveness for ad hoc informational search. Link density has little impact on anchor text effectiveness, while collection size has a big impact on the anchor text representations, affecting quantity, quality and effectiveness. Full-text search is less affected by collection size.

Perhaps the main contribution of this paper is that it solves the apparent contradiction between the experiences of Internet search engines, and the results of experiments at TREC. This turns the earlier negative results into something positive in a sense: they aid to our understanding of when and why link evidence works, and when not.

Acknowledgments

This work was generously supported by the Netherlands Organisation for Scientific Research (NWO, grants # 612.066.513, 639.-072.601, and 640.001.501).

REFERENCES

[1] C. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2009 Web Track. In *TREC*, 2009.

[2] N. Craswell, D. Hawking, and S. E. Robertson. Effective site finding using link anchor information. In *SIGIR*, pages 250–257, 2001.

[3] C. Gurrin and A. F. Smeaton. Replicating web structure in small-scale test collections. *Inf. Retr.*, 7:239–263, 2004.

[4] D. Hawking and N. Craswell. Very large scale retrieval and web search. In *TREC: Experiment and Evaluation in Information Retrieval*, chapter 9. MIT Press, 2005.

[5] D. Hawking and S. Robertson. On collection size and retrieval effectiveness. *Inf. Retr.*, 6(1):99–105, 2003.

[6] M. Koolen and J. Kamps. The importance of anchor-text for ad hoc search revisited. In H.-H. Chen, E. N. Efthimiadis, J. Savoy, F. Crestani, and S. Marchand-Maillet, editors, *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 122–129. ACM Press, New York NY, USA, 2010.

[7] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *SIGIR*, pages 27–34. ACM, 2002.

News Comments: Exploring, Modeling, and Online Prediction (Abstract)[*]

Manos Tsagkias
e.tsagkias@uva.nl

Wouter Weerkamp
w.weerkamp@uva.nl

Maarten de Rijke
mdr@science.uva.nl

ISLA, University of Amsterdam
Science Park 409, 1098 XH Amsterdam

ABSTRACT

Online news agents provide commenting facilities for their readers to express their opinions or sentiments with regards to news stories. The number of user supplied comments on a news article may be indicative of its importance, interestingness, or impact. We explore the news comments space, and compare the log-normal and the negative binomial distributions for modeling comments from various news agents. These estimated models can be used to normalize raw comment counts and enable comparison across different news sites. We also examine the feasibility of online prediction of the number of comments, based on the volume observed shortly after publication. We report on solid performance for predicting news comment volume in the long run, after short observation. This prediction can be useful for identifying potentially "hot" news stories, and can be used to support front page optimization for news sites.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics

General Terms

Algorithms, Theory, Experimentation, Measurement

Keywords

Comment volume, prediction, user generated content, online news

1. INTRODUCTION

As we increasingly live our lives online, huge amounts of content are being generated, and stored in new data types like blogs, discussion forums, mailing lists, commenting facilities, and wikis. In this environment of new data types, online news is an especially interesting type for mining and analysis purposes. Much of what goes on in social media is a response to, or comment on, news events, reflected by the large amount of news-related queries users ask to blog search engines [3]. Tracking news events and their impact as reflected in social media has become an important activity of media analysts [1]. We focus on online news articles plus the comments they trigger, and attempt to uncover the factors underlying the commenting behavior on these news articles. We explore the dynamics of user generated comments on news articles, and undertake the challenge to model and predict news article comment volume shortly after publication.

Let us take a step back and ask why we should be interested in commenting behavior and the factors contributing to it in the first place? We briefly mention two types of application for predicting the number of comments shortly after publication. First, in *reputation analysis* one should be able to quickly respond to "hot" stories and real-time observation and prediction of the impact of news articles is required. Second, the *lay-out decisions* of online news agents often depend on the expected impact of articles, giving more emphasis to articles that are likely to generate more comments, both in their online news papers (e.g., larger headline, picture included) and in their RSS feeds (e.g., placed on top, capitalized).

Our aim is to gain insight on the commenting behavior on online news articles, and use these insights to predict comment volume of news articles shortly after publication. To this end, we seek to answer the following questions: (i) What are the dynamics of user generated comments on news articles? Do they follow a temporal cycle? The answers provide useful features for modeling and predicting news comments. (ii) Can we fit a distribution model on the volume of news comments? Modeling the distribution allows for normalizing comment counts across diverse news sources. (iii) Does the correlation between number of responses at early time and at later time found in social media such as Digg and Youtube hold for news comments? I.e., are patterns for online responses potentially "universal"? And can we use this to predict the number of comments an article will receive, having seen an initial number?

This paper makes several contributions. First, it explores the dynamics and the temporal cycles of user generated comments in online Dutch media. Second, it provides a model for news comment distribution based on data analysis from eight news sources. And third, it tries to predict comment volume once an initial number of comments is known, using a linear model.

We explore the dataset in §2, model news comments in §3 and report on prediction results of comment volume in §4.

2. EXPLORING NEWS COMMENTS

The dataset consists of aggregated content from seven online news agents: *Algemeen Dagblad* (*AD*), *De Pers*, *Financieel Dagblad* (*FD*), *Spits*, *Telegraaf*, *Trouw*, and *WaarMaarRaar* (*WMR*), and one collaborative news platform, *NUjij*. We have chosen to include sources that provide commenting facilities for news stories, but differ in coverage, political views, subject, and type.

We turn to our first research question: What are the dynamics of user generated comments on news articles? News comments are found to follow trends similar to blog post comments as reported in [4]. The news agent commenting facilities (is it easy to comment or not) and content nature (accessible, require less understanding) show to influence the number of comments a news source receives. The time required for readers to leave a comment is on average slower for news than for blogs, although this

[*] The full version of this paper appeared in *ECIR 2010*.

Copyright is held by the author/owner(s).
DIR'11, February 4, 2011, Amsterdam.
ACM xxx-x-xxxxx-xxx-x/xx/xx.

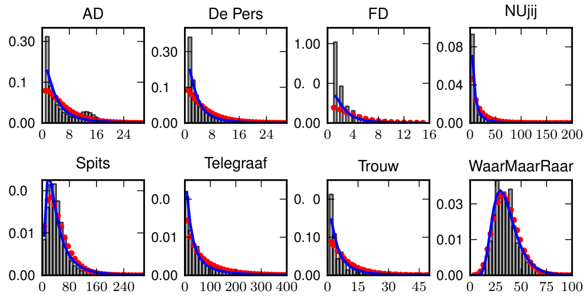


Figure 1: Modeling comment volume distribution per source using the continuous log-normal (blue line), and the discrete negative binomial distribution (red dots). Grey bars represent observed data. Probability density is on y-axis, and number of comments (binned) is on x-axis.

differs significantly per news source possibly due to differences in news readers demographics. With regards to temporal cycles, we look at monthly, weekly and daily cycles. March shows the highest comment volume across the board, and November shows the least for most sources. Weekdays receive more comments compared to weekends, with Wednesday being, on average, the most active day and Sunday the least active day across the board. The daily cycle reveals a correlation between comment volume, sleep and awake time, as well as working, lunch and dinner time: The comment volume peaks around noon, starts decreasing in the afternoon, and becomes minimal late at night. These aspects of online news seem to be inherent characteristics of each source possibly reflecting the credibility of the news organisation, the interactive features they provide on their web sites, and their readers' demographics [2].

3. MODELING NEWS COMMENTS

With regards to our second research question we seek to identify models (i.e., distributions) that underly the volume of comments per news source. We do so (1) to understand our data, and (2) to define "volume" across sources. Our approach is to express a news article's comment volume as the probability for an article from a news source to receive x many comments. We consider two types of distribution to model comment volume: log-normal and negative binomial. For evaluating the models' goodness of fit we choose the χ^2 test due to its applicability to both continuous and discrete distributions [5]. Both distributions fit our dataset well with low χ^2 scores (see also Fig. 1) leaving the final decision on which distribution to favor on the data to be modeled and the task at hand.

4. COMMENT PREDICTION

We now turn to our third research question. First, we are interested in finding out whether the correlation between early and late popularity found by [6] also holds for the news comments space. Then, assuming such a relation has been confirmed, it can be employed for predicting the comment volume of a news story. The existence of a circadian pattern implies that a story's comment volume depends on the publication time. We account for this by introducing a temporal transformation from real-time to *source-time*, a function of the comment volume entering a news site within a certain time unit.

We graph the Pearson's correlation coefficient ρ to visualize the correlation strength between comment volume at times close (early) and farther away (late) from publication. *Spits* displays a very steep comment volume curve meaning that most stories stop receiving comments short after publication. In contrast to our expectations that *NUjij* follows a fast correlation pattern similar to Digg, our findings suggest that a strong correlation is achieved much later possibly due to the different levels of effort required for digg-ing and commenting.

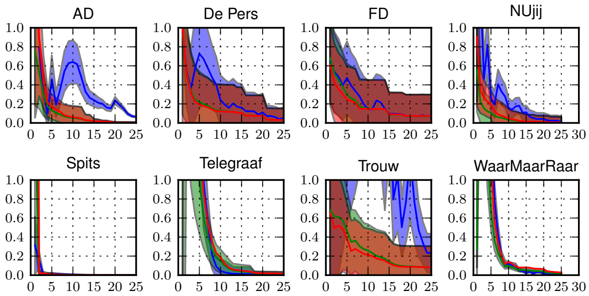


Figure 2: Relative square error using Model 1 (blue line), Model 2 (green line), and Model 3 (red line). Standard deviation is shown in the shaded areas around the lines. QRE on y-axis, observation time (hrs) on x-axis.

We follow [6] and estimate a linear model on a logarithmic scale for each source in our dataset. For evaluating our model we choose the relative squared error (QRE) metric averaged over all stories from a certain source. Fig. 2 shows that from the three models we study, the one using all stories and having the slope fixed at 1 (M2) performs the best. M2 demonstrates strong predictive performance indicated by low QRE < 0.2 for all sources, in less than 10 hours of observation. The QREs converge to 0 faster for some sources and slower for others, exposing the underlying commenting dynamics of each source as discussed earlier.

In this section we looked at natural patterns emerging from news comments, such as the possible correlation of comment counts on news stories between early and later publication time. A relation similar to the one observed for Digg and Youtube has been confirmed, allowing us to predict long term comment volume with very small error. We observed that different news sources ask for different observation times before a robust prediction can be made. QRE curves can indicate the optimum observation time per source, that balances between short observation period and low error.

Acknowledgments. This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430 (GALATEAS), by the 7th Framework Program of the European Commission, grant agreement no. 258191 (PROMISE), by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.-066.512, 612.061.814, 612.061.815, 640.004.802, 380-70-011 and by the Center for Creation, Content and Technology (CCCT).

References

[1] D. L. Altheide. *Qualitative Media Analysis (Qualitative Research Methods).* Sage Pubn Inc, 1996.

[2] D. S. Chung. Interactive features of online newspapers: Identifying patterns and predicting use of engaged readers. *Journal of Computer-Mediated Communication*, 13(3):658–679, 2008.

[3] G. Mishne and M. de Rijke. A study of blog search. In *ECIR'06*, pages 289–301. Springer, April 2006.

[4] G. Mishne and N. Glance. Leave a reply. In *Third annual workshop on the Weblogging ecosystem*, 2006.

[5] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, USA, 2000.

[6] G. Szabó and B. A. Huberman. Predicting the popularity of online content. *CoRR*, abs/0811.0405, 2008.

Cluster-Based Information Retrieval in Tag Spaces

Damir Vandic
vandic@ese.eur.nl

Jan-Willem van Dam
jwvdam@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

Econometric Institute
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

ABSTRACT

Many of the existing tagging systems fail to cope with syntactic and semantic tag variations during user search and browse activities. As a solution to this problem, we propose the Semantic Tag Clustering Search. The framework consists of three parts: removing syntactic variations, creating semantic clusters, and utilizing the obtained clusters to improve search and exploration of tag spaces. Using our framework, we are able to find relevant clusters and achieve a higher search precision by utilizing these clusters. The advantages of a cluster-based approach for searching and browsing through tag spaces have been exploited in XploreFlickr.com, the implementation of our framework.

1. INTRODUCTION

Today's Web offers many services that enable users to label content on the Web by means of tags. Flickr and Delicious (also known as del.icio.us) are two well-known applications utilizing tags. Registered Flickr users are allowed to upload and tag photographs. As with most tagging systems the user has no restrictions on the tags that can be used, i.e., the user can use any tag to his or her likings. Even though tags are a flexible way of categorizing data, they have their limitations. Tags are prone to typographical errors or syntactic variations due to the amount of freedom users have. This results in different tags with similar meanings, e.g., 'waterfal' and 'waterfall'. A query for 'waterfall' on Flickr returns $1,158,957$ results, whereas 'waterfal' returns $1,388$ results. This implies that potentially $1,157,569$ results are lost due to a typographical mistake. Users also describe pictures in different ways. For a picture which shows the interior of a house, most users would use the tag 'interior', where others would use a tag like 'inside' or 'furniture'. This is a problem for search engines which only implement keyword-based searching, as 'interior', 'inside', and 'furniture' are all semantically related.

As a solution to the previous problem, we define the Semantic Tag Clustering Search (STCS) framework, which consists of three parts. The first part deals with syntactic variations, whereas the second part is concerned with deriving semantic clusters. The last part of the framework consists of a part where search methods utilize these clusters to improve search for pictures. In the STCS framework, we consider non-hierarchical clusters, where we select the method proposed by [3]. Different from other methods, this algorithm allows tags to appear in multiple clusters, which enables easy detection of different contexts for tags. Each cluster is considered to be a context for a tag. Also, we propose an adaptation of this method that improves the clustering results. Finally, we devise a search method, of which the results are compared with a case without knowledge about the semantic clusters or syntactic variation clusters. We have made available an implementation of the STCS framework in the form of a Web application called XploreFlickr.com [4].

DIR 2011 February 4, 2011, Amsterdam, The Netherlands.
Copyright 2011 by the author(s).

2. RELATED WORK

Syntactic variations between tags form a widely studied research subject, as they represent a well-known symptom in tagging systems. In [1], the authors analyze the performance of the Levenshtein distance [2] and the Hamming distance. The authors state that Levenshtein and Hamming distances provide similar results for some syntactic variation types, e.g., for typographic errors. In contrast, for variation identification based on the insertion or deletion of characters, the Levenshtein distance performs significantly better than the Hamming distance. This does not imply that the Levenshtein distance performs well enough, as it has problems with for instance identifying variations based on the transposition of adjacent characters, although results can be improved by ignoring candidate tags with less than four characters.

In previous approaches, the semantic symptoms are dealt with by either using a clustering technique which results in non-hierarchical clusters of tags, or a hierarchical graph of either tags or clusters of tags. There is an extensive body of literature available on tag clustering. Several measures which create clusters of related tags are based on cooccurrence data, a commonly used similarity being the cosine similarity.

In this paper we focus on non-hierarchical clustering, as hierarchical clustering is more complex and thus more time consuming, because it first needs to build the tag hierarchy from which subsequently the clusters are deduced. The amount of data that we are dealing with asks for fast clustering procedures. Further, we observe that current non-hierarchical clustering approaches, e.g., the algorithm proposed by Specia and Motta [3], suffer from merging issues, i.e., larger clusters merge too easily and smaller clusters merge too difficultly. In this paper, we provide a solution to this problem.

3. STCS FRAMEWORK

Due to space limitations, we only discuss the first and second part of the STCS framework in this version of the paper. An extended version of this paper also discusses the third part, i.e., how we use the clusters to improve the performance of tag search engines. This extended version of the paper is to be presented at the 26th ACM Symposium on Applied Computing (SAC 2011) [5].

3.1 Syntactic clustering

In the first part of the framework, the syntactic clustering algorithm uses an undirected graph $G = (T, E)$ as input. The set T contains tags, and E is the set of weighted edges (triples (t_i, t_j, w_{ij})) representing the similarities between tags. Weight w_{ij} is calculated as a weighted average based on the normalized Levenshtein distance and the cosine similarity between tags i and j using the co-occurrence vectors. Normalized Levenshtein values are not representative for short tags, which is why we increase the weight for the cosine value as the length of the two tags decreases. The algorithm then proceeds by cutting edges that have a weight lower than a threshold β. The syntactic clusters are computed by determining the connected components in the resulting graph.

3.2 Semantic clustering

For semantic clustering, we propose a modified version of the algorithm that is proposed in [3]. The algorithms loops over all tags that are present in the data set and creates a new cluster which only includes the current tag. The algorithm then loops over all tags again and adds a tag to the cluster if it is sufficiently similar to the cluster. The tag is sufficiently similar when the average cosine of the tag with respect to all tags currently present in the cluster is larger than a threshold χ. Because many tags are similar to each other, this procedure produces many duplicate or near duplicate clusters. Hence, there is a need for cluster merging.

The authors of [3] propose two heuristics for the semantic clusters merging process. The first heuristic merges two clusters if the one contains the other and the second heuristic merges clusters if the number of different elements between two clusters is below a certain threshold. We propose a merging heuristic with a dynamic threshold, depending on the cluster sizes. With a constant threshold the larger clusters often merge too easily and the smaller clusters merge too difficultly. The STCS heuristic fits the clustering process better, as it is less sensitive to the size of smaller clusters than the method proposed in [3].

4. STCS EVALUATION

In order to analyze the performance of the syntactic variations detection algorithm, we use a test set which contains 200 randomly chosen tag combinations. These tags are subject to the weighted average of the normalized Levenshtein value and the cosine similarity. In our experiments, the weighted average for all tag combinations is calculated with a threshold value β of 0.62 for cutting edges, which is determined by result evaluation using a hill climbing procedure. After manually checking these tags on correctness, we identify 10 mistakes that are produced by the framework, resulting in a syntactic error rate of 5%.

For the analysis of the semantic clustering process, we follow a similar procedure. For 100 random clusters, which contained 458 tags, the number of misplaced tags is counted, i.e., the tags that should have been placed in another cluster. We encounter 44 misplaced tags and thus the error rate is 9.6%. We report an error of 13.1% for the method of [3], which shows that the STCS method outperforms the original method on this data set. We observe that the STCS algorithm finds many relevant clusters, such as {rainy, Rain, wet, raining} and {iPod, iphone, mac}.

5. CONCLUSIONS

The Semantic Tag Clustering Search (STCS) framework is used for building and utilizing semantic clusters based on information retrieved from a social tagging system. The framework has three core tasks: removing syntactic variations, creating semantic clusters, and utilizing obtained clusters to improve search and exploration of tag spaces. For the syntactic clustering process we have proposed a measure based on the normalized Levenshtein value combined with the cosine value based on co-occurrence vectors. Results show that the framework obtains an error rate for syntactic clustering of 5% and 9.6% for semantic clustering. We compared the non-hierarchical clustering method proposed by Specia and Motta [3] to our adapted version and have found that the adapted version has a lower error rate than the original method.

As future work, we would like to improve the process of removing syntactic variations by using two ideas. First, we want to take into account abbreviations, as the Levenshtein distance does not address this issue. Second, we would like to experiment with variable cost Levenshtein distances, which associate different weights to edit operations depending on update characters and their location.

6. REFERENCES

[1] F. Echarte, J. J. Astrain, A. Córdoba, and J. Villadangos. Pattern Matching Techniques to Identify Syntactic Variations of Tags in Folksonomies. In *1st World Summit on The Knowledge Society (WSKS 2008)*, volume 5288 of *LNCS*, pages 557–564. Springer, 2008.

[2] V. I. Levenshtein. Binary Codes Capable of Correction Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[3] L. Specia and E. Motta. Integrating Folksonomies with the Semantic Web. In *4th European Semantic Web Conference (ESWC 2007)*, volume 4519 of *LNCS*, pages 503–517. Springer, 2007.

[4] J. W. van Dam, D. Vandic, F. Hogenboom, and F. Frasincar. XploreFlickr.com, 2010. From: `http://www.xploreflickr.com/`.

[5] D. Vandic, J. W. van Dam, F. Hogenboom, and F. Frasincar. A Semantic Clustering-Based Approach for Searching and Browsing Tag Spaces. In *26th Symposium on Applied Computing (SAC 2011)*, pages 1698–1704. ACM, 2011.

Constructing Collections for Learning to Rank

Emine Yilmaz[‡], Evangelos Kanoulas[†], Stephen E. Robertson[‡], Javed A. Aslam[⋆]
eminey@microsoft.com, e.kanoulas@shef.ac.uk, ser@microsoft.com, jaa@ccs.neu.edu

[‡] Microsoft Research Cambridge, UK [†] Information School, University of Sheffield, UK
[⋆] College of Computer & Information Science, Northeastern University, USA

ABSTRACT

In this paper we summarize our previous research on the construction of training sets and development of metrics for learning to rank. In particular, we consider the case of a fixed budget of total judgments to be spent and we discuss the effect of (a) the allocation of the budget between documents and queries, (b) the documents to be selected per query, and (c) the choice of the metric to be optimized on the effectiveness of learning to rank algorithms.

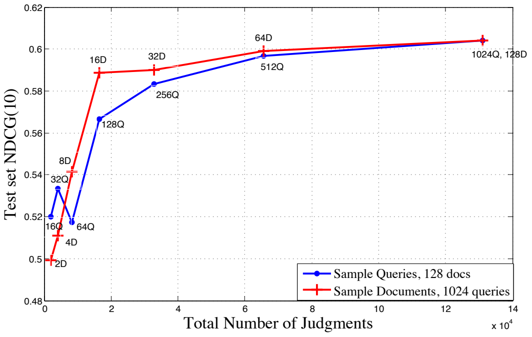
1. INTRODUCTION

Most algorithms for building modern search engines are based on learning to rank. Given a training set of (feature vector, relevance) pairs, a machine learning procedure learns how to combine the query and document features to rank the underlying collection upon a user's request by optimizing an effectiveness metric correlated with user satisfaction. Much thought and research has been placed on feature extraction and the development of sophisticated learning to rank algorithms. However, relatively little research has been conducted on the choice of queries, documents and effectiveness metrics to be used for learning to rank nor on the effect of these choices on the effectiveness and efficiency of the learning algorithm.

The main bottleneck in constructing learning to rank collections is obtaining labels by annotating documents with relevance grades since this task requires extensive human effort. Thus, given a fixed judgment budget researchers and practitioners often need to make a number design decision, "Is it better to judge many queries with few judgments per query (shallow judgments) or to judge few queries but more documents per query (deep judgments)?", "Which documents per query should be selected to be annotated?", "Which metric should be optimized to obtain the best performing ranking function with respect to an end user?". In this work we summarize recent results in an attempt to answer all of the above questions [4, 1, 5, 2].

2. DEEP VS. SHALLOW JUDGMENTS

In order to test the effect of deep versus shallow judgments we use data obtained from a commercial search engine. The dataset contains 382 features extracted from a set of 5K queries with an average of 350 judged documents per query and it is split into train, validation and test sets with 2K, 1K and 2K queries, respectively. Due to the high variability of the number of documents judged per query, we select an 1K subset of from the training queries with at least 128 judgments each so that we can better control the experiment. Using this data we form different data sets by halving the number of queries in the training set, resulting in training sets with 1024, 512, 256, 64, 32 and 16 queries, each containing 128 documents. Similarly, we also form different sets by halving the number of judgments per query (128, 64, 32, 16, 8, 4 and 2 documents), keeping the number of queries fixed (1K). The LambdaRank algorithm is then used to train the ranking function over the different training sets.



The test set NDCG(10) value using these different training sets is reported in the figure. The x-axis shows the total number of judged documents in the training set. The line with the plus marks corresponds to halving the queries (having 128 judgments per query) and the line with the dotted marks corresponds to halving the number of judged documents per query (keeping all the 1K queries). Next to each plus (or dot), we report the number of documents in the training set (or the number of queries in the training set). The results suggest that given a fixed judgment budget, it is better to judge more queries with fewer documents per query. It can be seen that with as few as 16 documents per query, test set NDCG(10) values are comparable to using the entire 128 documents. However, decreasing the number of judged documents further results in a sharp decrease in performance [4].

3. DOCUMENT SELECTION

In the experiments above having no information about the process used to construct the original data set of 350

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.
.

judged documents per query on average we uniformly sampled p% of them to construct the training subsets. Nevertheless, some documents may be more useful than others (i.e. hold more information) for learning to rank. Here we examine a number of alternative document selection methods that has been previously used for low-cost evaluation of retrieval systems to choose documents to annotate. In order to examine the effect of different mechanisms to select documents we used TREC data since these collections provide some further information about the documents to be picked (e.g. their ranks by the submitted to TREC retrieval systems). Our *complete* document collection consists 150 queries and depth-100 document pools from TREC 6, 7 and 8 adhoc tracks, along with the corresponding relevance judgments. A small set of 22 content features (a subset of LETOR3.0 features [3]) is extracted from all query-document pairs. Using different document selection methodologies, for each query, documents from the complete collection are selected with different percentages from 0.6% to 60%, forming different sized subsets of the complete collection for each methodology. The document selection methodologies used vary from sampling (uniform and stratified) to depth-k pooling and greedy on-line algorithms along with the current approach of selecting documents used in the construction of the LETOR 2.0 and 3.0 datasets.[1] We employ five different learning-to-rank algorithms to test the document selection methodologies, RankBoost, Regression, RankingSVM, RankNet, and LambdaRank.

Based on the results obtained by training the six learning to rank algorithms over the different training data set a number of observations were made. First, some learning to rank algorithms are more robust to document selection methodologies than other (e.g. LambdaRank). Second, some document selection methods are more effective than others, with depth-pooling and stratified sampling being the best performing ones. The fact that different document selection methods produce training sets of different characteristics allows us to examine what makes one training set better than another. Using model selection we determined that the precision of the dataset (proportion of relevant documents) and the similarity between relevant and non-relevant documents the most influential characteristics. Surprisingly our results suggest that it is harmful to select too many relevant documents and relevant and non-relevant documents that are too similar.

4. EFFECTIVENESS METRICS

Most current learning to rank algorithms are based on the assumption that if a metric X evaluates the utility of the search engine to an end user, then a search engine should be trained to optimize for that particular metric. For instance, in section 2 the LambdaRank algorithm was optimized for NDCG(10) given that our test metric was NDCG(10). Nevertheless, evaluation metrics used in optimization act as bottlenecks that summarize the training data. Given that some metrics are more informative than others we hypothesize that even if user satisfaction can be measured by a metric X, optimizing the ranking function for a more informative metric Y may result in better test performance according to X. To test our hypothesis we extended the LambdaRank algorithm to optimize for a number of evaluation metrics (Precision(10), Average Precision (AP), nDCG and nDCG(10)), we used the original data set described in section 2 to train the ranking function and measured the performance of the obtained retrieval systems by all aforementioned measures. When binary judgments were used our results suggested that even if one is interested in user oriented metrics such as PC(10) or nDCG(10) it is better to optimize for more informative metrics such as AP and nDCG. Further, optimizing for AP appeared to lead to better results than optimizing for nDCG [5]. Given that AP appeared the best metric to optimize in the case of binary judgments and given that a multi-graded measure can certainly hold more information than a binary one, we extended the definition of AP to accommodate multi-graded judgments [2]. Then we tested our hypothesis by optimizing for nDCG, AP and Graded Average Precision (GAP). The results of our experiments can be view in the table above and they suggest that GAP is indeed the best measure to optimize for even when you care about different measures.

LambdaRank	Test Metric		
	nDCG	AP	PC(10)
Opt nDCG	0.6301	0.6158	0.5355
Opt GAP	**0.6363**	**0.6287**	**0.5388**
Opt AP	0.6296	0.6217	0.5360

5. CONCLUSIONS

When constructing training collections for learning to rank with a limited budget researchers and practitioners face a number of design question regarding how to distribute judgments across query-document pairs and what metric to optimize for to obtain the best performing ranking function. In our work we've shown than distributing budget across a large number of queries with few judgments per query is better than deeply judging a few queries. Further, we've shown that the manner of selecting these few documents to be judged per query makes a difference for most learning algorithms. Finally, we've observed that learning algorithms can make best use of these limited judgments when optimized for an informative metric, with AP appearing to be one of the most informative binary metrics. Motivated by that, we've extended AP to graded judgments so we can further increase the informativeness of the metric and improve the effectiveness of the learning to rank algorithm.

6. REFERENCES

[1] J. A. Aslam, E. Kanoulas, V. Pavlu, S. Savev, and E. Yilmaz. Document selection methodologies for efficient and effective learning-to-rank. In *SIGIR2009*, 2009.

[2] S. E. Robertson, E. Kanoulas, and E. Yilmaz. Extending average precision to graded relevance judgments. In *SIGIR2010*, pages 603–610, New York, NY, USA, 2010. ACM.

[3] J. X. Tao Qin, Tie-Yan Liu and H. Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval Journal*, 2010.

[4] E. Yilmaz and S. Robertson. Deep versus shallow judgments in learning to rank. In *SIGIR2009*, pages 662–663, New York, NY, USA, 2009. ACM.

[5] E. Yilmaz and S. E. Robertson. On the choice of effectiveness measures for learning to rank. *Information Retrieval Journal*, 13(3):271–290, 2010.

[1] More details can be found in Aslam et al. [1]

Demonstrations

Fuzzy faceted search

Expand your search strategy

Wouter Alink
Spinque
wouter@spinque.com

Roberto Cornacchia
Spinque
roberto@spinque.com

Arjen P. de Vries
Centrum Wiskunde &
informatica
arjen@acm.org

ABSTRACT

Our demo at DIR 2011 shows the benefits of fuzzy faceted search. Fuzzy facets can be used to re-rank result sets with vague predicates - not only to filter them with hard selection criteria. This makes facets a better match with IR search applications that are based on the ranking approach. First class citizens in the score-based world, fuzzy facets can be more powerful and useful tools to interactively improve results for an improved and more natural user experience.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval

General Terms

Design, Experimentation

Keywords

DB and IR, probabilistic databases, fuzzy facets, probabilistic relational algebra, search by strategy

1. INTRODUCTION

In recent years faceted search has become commodity in search interfaces, and provides users with the ability to quickly zoom in on result sets using various views on the data. Faceted search is often used in IR-oriented search applications, but does not blend naturally rank-based approach. Faceted search provides DB-style methods for filtering results, often ignoring the computed scores for the items in the result set.

If facets were regarded as vague predicates, search interfaces could provide interactive refinement of query results, without quickly running into database search issues such as empty results or non-specific filter criteria. The proposed demo shows a working implementation of fuzzy facets applied to the intellectual property domain with a real-life sized data set.

The technology showed in this demo is particularly interesting about two aspects: firstly, our approach towards flexible and efficient query processing and facet computation; secondly, a novel novel user interaction is enabled, in which an end-user can quickly refine her initial query using fuzzy faceted search.

Flexible query processing is achieved by introducing a clear separation of concerns in various layers of query formulation (conceptual search strategy, probabilistic relational algebra, SQL). Efficient execution of the automatically derived query plans is achieved by using a next-generation column-oriented database back-end.

When having such an efficient query processing back-end, facet options can be computed on the fly and are not restricted to pre-calculated bins, which in turn makes it possible to provide novel user interface elements.

2. REFERENCES

[1] W. Alink, R. Cornacchia, and A. de Vries. Searching clef-ip by strategy. In *CLEF 2009, Revised Selected Papers, Part I.* Springer, 2011. To appear.

[2] S. Chaudhuri, R. Ramakrishnan, and G. Weikum. Integrating DB and IR Technologies: What is the Sound of One Hand Clapping? In *Proc. CIDR*, pages 1–12, Asilomar, CA, USA, 2005.

[3] R. Cornacchia and A. P. de Vries. A parameterised search system. In *ECIR 2007*, pages 4–15, Apr. 2007.

[4] R. Cornacchia, S. Héman, M. Zukowski, A. de Vries, and P. Boncz. Flexible and efficient IR using Array Databases. *VLDB Journal*, 17(1):151–168, Jan. 2008.

mediaWalker: Tracking and browsing news video along the topic thread structure [*]

Ichiro Ide[†]
Nagoya Univ.
Furo-cho, Chikusa-ku
Nagoya, Japan
ide@is.nagoya-u.ac.jp

Tomokazu Takahashi
Gifu Shotoku Gakuen Univ.
1-38 Naka-Uzura
Gifu, Japan
ttakahashi@gifu.shotoku.ac.jp

Tomoyoshi Kinoshita
NetCOMPASS Ltd.
6F, 2-14-4 Shinkawa
Chuo-ku, Tokyo, Japan
kino@netcompass.co.jp

Shin'ichi Satoh
Nat'l Inst. of Informatics
2-1-2 Hitotsubashi
Chiyoda-ku, Tokyo, Japan
satoh@nii.ac.jp

Hiroshi Murase
Nagoya Univ.
Furo-cho, Chikusa-ku
Nagoya, Japan
murase@is.nagoya-u.ac.jp

Frank Nack
Univ. of Amsterdam
Science Park 907
Amsterdam, The Netherlands
nack@uva.nl

ABSTRACT

We introduce a news video tracking and browsing interface "mediaWalker" that allows users to explore throughout a news video archive by tracking news topics along a chronological semantic structure of news stories.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces

General Terms

Design

Keywords

News video, video archive, topic tracking, interface

1. INTRODUCTION

In the last ten years, we have been creating a news video archive composed of more than 1,800 hours of video recorded from a daily news show. In order to efficiently retrieve information from such a large news video archive, analysis of the chronological semantic structure of its contents is necessary.

We have previously proposed a method that retrieves the chronological semantic structure; "topic thread structure", that originates from a specified news story and chains stories on subsequent events on related news topics in the form of a directed graph. This was done by measuring the relation of news stories based on both text similarity and chronological order. Details of the method could be found in [1].

In this paper, we will introduce an interface "mediaWalker" that allows users to browse throughout the archive by tracking news topics along the topic thread structure. We believe that such an interface facilitates the users to understand news topics along the timeline, while it saves time than browsing through a linear list of video clips on related topics as in a traditional video retrieval interface.

2. THE MEDIAWALKER INTERFACE

We will briefly introduce the functions of the interface, according to the search flow shown in Fig. 1.

2.1 Initial story listing

First, a user searches the initial story-in-focus, either from a list of manually arranged set of stories, or by issuing a query by combining keywords and dates (Fig. 1(a)).

2.2 Initial story selection

Next, a list of stories that match the criteria in the previous screen is listed. The list can be rotated, while video clips corresponding to stories could be played. The user then chooses one story, and selects to browse a topic thread structure either towards the past (left) or the future (right), originating from the story (Fig. 1(b)).

2.3 Topic tracking and browsing

Finally, video clips (stories) are placed as nodes on the topic thread structure that originates from the story specified in the previous screen (Fig. 1(c)). Figure 2 shows an example of topic tracking in the interface; The development of news topics could be tracked by playing clip by clip along the structure. The interface also shows the difference of keywords between stories, to provide the users with information for selecting the desired topic thread during the tracking.

In addition to the tracking function, the interface also provides the following functions:

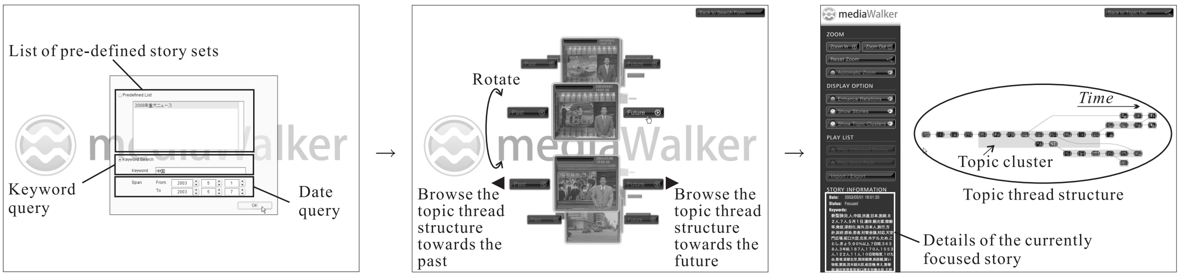
- Automatic playing and exporting
  As shown in Fig. 3(a), when two stories are selected by

[*]This paper is a summarized version of papers [1, 2, 3].

[†]Currently staying at Univ. of Amsterdam.

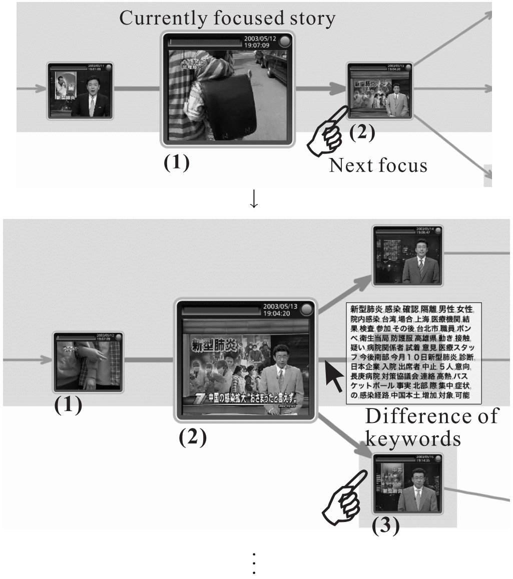
Copyright is held by the author/owner(s).

DIR'2011, February 4, 2011, Amsterdam.
.


List of pre-defined story sets
Keyword query
Date query
Rotate
Browse the topic thread structure towards the past
Browse the topic thread structure towards the future
Time
Topic cluster
Topic thread structure
Details of the currently focused story
mediaWalker
→
→

(a) Initial story listing (b) Initial story selection (c) Topic tracking and browsing

Figure 1: Search flow in the mediaWalker interface.



新型肺炎 感染 確認 隔離 男性 女性 院内感染 台湾 場合 上海 医療機関 結果 検査 参加 その後 台北市 職員 ボンベ 衛生当局 防護服 高雄県 動き 接触 疑い 病院関係者 試着 意見 医療スタッフ 今後南部 今月１０日新型肺炎 診断 日本企業 入院 出席者 中止 ５人 意向 長庚病院 対策協議会 連絡 高熱 バスケットボール 事実 北部 際 集中 症状 の 感染経路 中国本土 増加 対象 可能

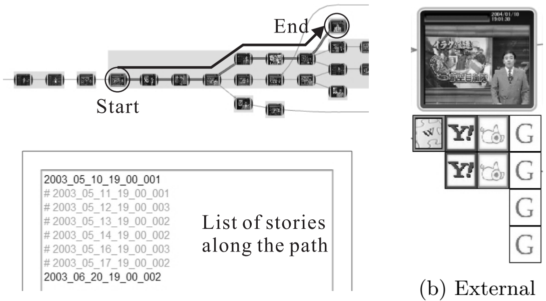
Figure 2: An example of topic tracking in the topic tracking and browsing screen; Tracking stories (1), (2), (3), ...

checking a button at the corner of the video players, the interface finds a path that connects them. The user can then, let the interface automatically play along the path, or export the list of stories along the path for external use or post-processing.

- External link
  As shown in Fig. 3(b), each story is linked to external Web pages related to its contents. It is linked first to Wikipedia articles, and then to other contents by issuing queries based on the title of the articles. Details on the linking method could be found in [3, 4].

3. CONCLUSIONS

We briefly introduced an interface that allows users to track the development of news topics along the topic thread structure. Future work includes story telling along the structure.



```
2003_05_10_19_00_001
# 2003_05_11_19_00_001
# 2003_05_12_19_00_003
# 2003_05_13_19_00_002
# 2003_05_14_19_00_002
# 2003_05_16_19_00_003
# 2003_05_17_19_00_002
2003_06_20_19_00_002
```

(a) Automatic playing and exporting (b) External links

Figure 3: Other functions in the topic tracking and browsing interface.

4. ACKNOWLEDGMENTS

Parts of the work were supported by Hoso Bunka Foundation, Grants-in-aid for Scientific Research from the Japanese Ministry of Education, Culture, Sports, Science and Technology, and the Excellent Young Researcher Overseas Visit Program from the Japan Society for the Promotion of Science.

5. REFERENCES

[1] I. Ide, T. Kinoshita, T. Takahashi, H. Mo, N. Katayama, S. Satoh, and H. Murase. Exploiting the chronological semantic structure in a large-scale broadcast news video archive for its efficient exploration. In Proc. 2010 APSIPA Annual Summit and Conf., pages 996–1005, Dec. 2010.

[2] I. Ide, T. Kinoshita, T. Takahashi, S. Satoh, and H. Murase. mediaWalker: A video archive explorer based on time-series semantic structure. In Proc. 15th ACM Int. Multimedia Conf., pages 162–163, Sept. 2007.

[3] I. Ide, T. Takahashi, T. Kinoshita, T. Okuoka, D. Deguchi, S. Satoh, and H. Murase. mediaWalker II: A news video archive browsing interface associated with Web contents (in Japanese). In Proc. Meeting on Image Recognition and Understanding (MIRU) 2010, pages 1320–1321, July 2010.

[4] T. Okuoka, T. Takahashi, D. Deguchi, I. Ide, and H. Murase. Labeling news topic threads with Wikipedia entries. In Proc. 11th IEEE Int. Symposium on Multimedia, pages 501–504, Dec. 2009.

PuppyIR Unleashed

A Framework for Building Child-Oriented Information Services

Richard Glassey
School of Computing Science
University of Glasgow
Scotland, UK
rjg@dcs.gla.ac.uk

Tamara Polajnar
School of Computing Science
University of Glasgow
Scotland, UK
tamara@dcs.gla.ac.uk

Leif Azzopardi
School of Computing Science
University of Glasgow
Scotland, UK
leif@dcs.gla.ac.uk

ABSTRACT

Children now encounter information technology in most environments (home, school, leisure) from the earliest of ages. Whilst children naturally adapt to technology, it is less clear whether technology is meeting their particular needs, specifically in their quest for information. The PuppyIR Project is working towards a better understanding of children as information seekers, and incorporating this knowledge into a common framework for building information services for children. This paper reports the development of three prototypes based upon the framework and its children-oriented information processing components.

1. INTRODUCTION

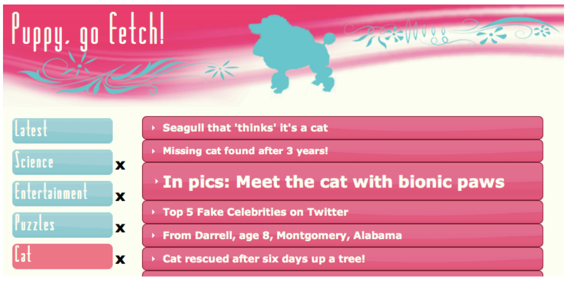
Children naturally adapt to new technology in a way that is often quite surprising to adults, but the question remains whether services they use to find and access information are appropriate for *their* needs. This question has prompted research that investigates how children and adults are different (and similar) in their information seeking behaviour [3], usability concerns across younger user groups [10], and the implication of the different developmental stages that children experience as they grow up [1]. It has been suggested that a complete rethink is required in how services incorporate support for the information needs of children [6], and that some progress may be made by actually involving children themselves within the design process [8].

In light of this, the main goal of the PuppyIR project[1] is to design, develop and deliver an open source framework for building information services specifically intended for children. The project considers a wide range of service aspects, including: the user interface and experience; information processing of queries, query suggestions and results; and providing access from different types of devices.

The framework has been designed using a prototype-driven methodology. Three prototypes have been created that have helped both guide framework development, and identify gaps in current service provision. FiFi (Find and Filter), a topic-based aggregator service for news and other interests, revealed the limited amount of good quality information feeds targeted at a younger audience; SeSu (Search and Suggest), a visual search suggestion service to help build better queries; and JuSe (Junior Search), a completely visual search service.

[1] www.puppyir.eu

Copyright is held by the author/owner(s).
*DIR'*2011, February 4, 2011, Amsterdam.
.



Seagull that 'thinks' it's a cat
Missing cat found after 3 years!
In pics: Meet the cat with bionic paws
Top 5 Fake Celebrities on Twitter
From Darrell, age 8, Montgomery, Alabama
Cat rescued after six days up a tree!

Figure 1: Screenshot of FiFi – Find and Filter

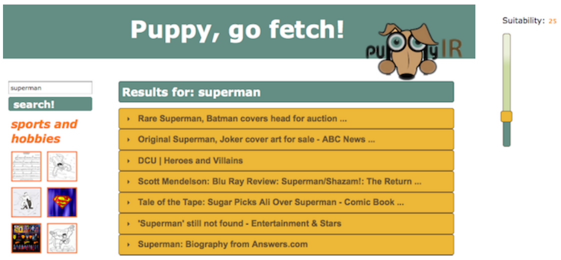
Whilst the framework provides the means to create complete services with relative ease, its component-based architecture allows new components to be developed independently by third parties and integrated into a common platform. This provides a wider benefit to the Interactive Information Retrieval (IIR) community by allowing developments, such as a better form of query suggestion service, to be rapidly integrated and evaluated within existing services.

2. PROTOTYPES

To assist with the design and development of the framework, three prototype information services were created.

FiFi – Find and Filter: Information filtering for children, based on their interests, has not been specifically studied within the literature. However, it provides an opportunity to investigate the information interests of children, whilst facilitating their *information encountering* [5]. For instance, a child may have an ongoing interest in gossip surrounding High School Musical. A filtering service would be ideal for meeting this information need, delivering fresh content regularly, instead of favouring relevance.

FiFi was developed to support this use case. RSS feeds for children on diverse topics (e.g. entertainment, news, science, etc) were retrieved (from an manually curated list) and indexed. The collection of articles can be searched by adding *topics* to the user interface, which act as queries. Instead of listing results (entries extracted from all feeds) based on relevance, they are presented in reverse-chronological order, however, relevance cues for the topic are preserved and indicated by varying the text size of an entry's title based on its relevance score for that topic (see Fig. 1).



Results for: superman

- Rare Superman, Batman covers head for auction ...
- Original Superman, Joker cover art for sale - ABC News ...
- DCU | Heroes and Villains
- Scott Mendelson: Blu Ray Review: Superman/Shazam!: The Return ...
- Tale of the Tape: Sugar Picks Ali Over Superman - Comic Book ...
- 'Superman' still not found - Entertainment & Stars
- Superman: Biography from Answers.com

Figure 2: Screenshot of SeSu – Search and Suggest

The development of FiFi required the integration of a local search engine into the filtering service, but improving the result presentation to make attractive and user friendly for children. To separate the aspects of presentation and search, a middleware component was developed to ensure that the framework was not tied to a particular presentation method or search engine technology, avoiding limitations in the flexibility of the framework.

SeSu – Search and Suggest: Whilst FiFi provided a novel means for children to encounter recently published content aligned with their interests, it did not address a key challenge faced by young information seekers: query formulation [4]. It is well established that adults struggle to adequately convey their information needs as a query, but this struggle is worse for children who possess a smaller vocabulary and limited cognitive development [9].

SeSu (Search and Suggest) was developed to provide query assistance to children using query term and visual suggestions. SeSu differs from FiFi by integrating an online search service (provided by Yahoo), instead of a local search engine. Figure 2 shows the user interface of SeSu. A panel on the left-hand side contains the textual and visual suggestions for the current query, whilst the search results fill the central panel. The right-hand panel contains an experimental feature: a slider bar to control a suitability filter, which moderates the displayed search results.

Building on the experience of developing FiFi, more attention was focused on improving the user interface in line with the research [2, 7]. Beyond user interface improvements, a suitability filter was developed to mitigate the risk of using an external search service (despite using its *safe mode*, inappropriate content can still be found), whilst making it easier to identify results based upon more positive features (i.e. a page that was specifically designed for children).

JuSe – Junior Search: Search suggestion is a useful technique to assist children as they search. However, using text based interfaces requires a certain level of dexterity and cognitive ability, or at least the assistance of an adult. For the final prototype, attention was focused on building an information service for children that required no query terms whatsoever, relying instead upon a completely visual interface. In effect, a service that very young children could make use of to independently to encounter information without assistance.

JuSe (Junior Search) presents the user with a central panel of clip art images organised by category (see Fig. 3). Categories can be created on demand by supplying a category name, along with a list of associated keywords (e.g. *Animals: Cat, Dog, Mouse*). Images for the keywords in a category are sourced from Google Image Search service, selecting the top n clip art images per keyword. To further assist the user, audio snippets of each image/keyword pair are automatically generated, introducing an educational aspect to the service. One or more images can be dragged from the central panel to the left panel, generating a query from the associated keywords.

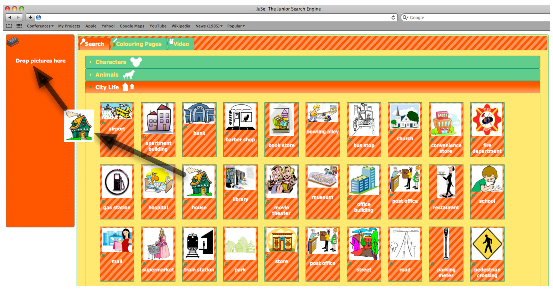


Figure 3: Screenshot of JuSe – Junior Search

3. REFERENCES

[1] M. Baumgarten. Kids and the internet: A developmental summary. *Computers in Entertainment*, 1(1), 2003.

[2] D. Bilal. Draw and Tell: Children As Designers of Web Interfaces. *American Society for Information Science and Technology*, 40(1):135–141, 2003.

[3] D. Bilal and J. Kirby. Differences and Similarities in Information Seeking: Children and Adults As Web Users. *Information Processing & Management*, 38(5):649–670, 2002.

[4] A. Druin, E. Foss, H. Hutchinson, E. Golub, and L. Hatley. Children's roles using keyword search interfaces at home. In *In Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI)*, pages 413–422, 2010.

[5] S. Erdelez. Information Encountering, Theories of Information Behaviour. *American Society. for Info. Science and Tech.*, pages 179–184, 2005.

[6] H. E. Jochmann-Mannak, T. W. C. Huibers, and T. J. M. Sanders. Children's information retrieval. In *Proceedings. of Future Directions in Information Access*, pages 64–72, 2008.

[7] A. Large and J. Beheshti. Int. design, web portals, and children. *Lib. Trends*, 54(2):318–342, 2005.

[8] A. Large, J. Beheshti, and T. Rahman. Design Criteria for Children's Web Portals: the Users Speak Out. *Journal of the American Society for Information Science and Technology*, 2(53):79–94, 2002.

[9] P. Moore and A. S. George. Children as information seekers: The cognitive demands of books and library systems. *School Library Media Quarterly*, 19(3):161–168, 1991.

[10] J. Nielsen. Usability of websites for children: Design guidelines for targeting users aged 3–12 years, http://www.useit.com/alertbox/children.html, 2010.

Peilend.nl: Exploring Dutch-language Online News

Valentin Jijkoun Fons Laan Maarten de Rijke
ISLA, University of Amsterdam, The Netherlands
jijkoun,a.c.laan,derijke@uva.nl

We present Peilend.nl, a system for analysing Dutch-language online news.

Essentially, Peilend.nl consists of three components:

- Data collection: the online news articles and users' comments on the articles (when available) are continuously tracked using Ssscrape, an open source system for collecting dynamic online data[1].
- Data processing: the collected data is indexed using Lucene[2] and is sent to Fietstas[3], a text analysis web service, that performs, in particular, extraction and resolution of named entities; document processing results are available through a REST web service.
- User interface provides functionality such as keyword search and visualization of search results in terms of word and entity clouds.

Peilend.nl provides feedback functionality: logged-in users can correct system's decisions, such as types of entities (person, organization, location), canonical names or URIs for entities. The system uses such feedback to correct the display of the information, and moreover, collects it for future use in retraining entity extractor and resolver.

Peilend.nl is a demonstrator for the technology developed for the online media analysis, where opinions towards entities and topics are studied. Within this user scenario, we will demonstrate the use of simple sentiment analysis techniques, based on hand-crafted and automatically-derived polarity lexicons.

Acknowledgements

Peilend.nl is supported by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12. The research is also supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, by and by the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.066.512, 612.061.814, 612.061.815, 640.004.802.

[1] http://ilps.science.uva.nl/resources/ssscrape

[2] http://lucene.apache.org

[3] http://fietstas.science.uva.nl

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
DIR '11 Amsterdam, The Netherlands
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

Annotation Strategies for Audiovisual Heritage

Roeland Ordelman
Netherlands Institute for Sound and Vision
Hilversum, The Netherlands
rordelman@beeldengeluid.nl

Johan Oomen
Netherlands Institute for Sound and Vision
Hilversum, The Netherlands
joomen@beeldengeluid.nl

ABSTRACT

In order to safeguard audiovisual heritage for future generations, large amounts of audiovisual content are being digitized. Unlocking the social and economic value of the collections requires the availability of metadata, preferably containing rich, fragment level annotations. Ideally, the content is linked to relevant contextual information sources as well. As available resources for manual annotation and contextualization are typically not in line with the quantities of digitized content, support from (semi) automatic annotation strategies and/or strategies that deploy crowdsourcing mechanisms, are widely investigated. This paper describes practical solutions that are developed at the Netherlands Institute for Sound and Vision in collaboration with academic partners in The Netherlands.

Keywords

Audiovisual archives; access; metadata; automatic annotation; crowdsourcing.

1. INTRODUCTION

Audiovisual archives are transforming from archives of analogue materials to very large stores of digital data. Audiovisual recordings preserve the history of the 20th Century: events and personalities can be seen and heard, having unique impact and meaning to all people. The exploitation model for this content -unlocking the social and economic value of the collections-implies investments concentrated on both preservation and access. Hence, in the context of a large digitization program, such as the Images for the Future[1] program at The Netherlands Institute for Sound and Vision (NISV), the study of access requirements of potential user groups plays an important role in the exploitation model.

Although different types of end users have different backgrounds, different needs, different expectations and different goals, studies such as [1,2] focusing on transaction log analysis of *broadcast professionals* and multidisciplinary collaborations investigating requirements for *oral historians* [3], endorse the general impression that rich, fragment level annotations are becoming a prerequisite for successful exploitation of audiovisual collections. Moreover, in order to be able to link community knowledge (e.g., wiki) or multimedia context sources (e.g., broadcast websites) to archival content, anchor points -high level entities- need to be localized such as a particular person or place, a topic or event.

As audiovisual archives are simply not capable to allocate resources for manual annotation and contextualization of today's quantities of digitized content in such levels of detail, support from (semi) automatic annotation strategies and/or strategies that deploy crowdsourcing mechanisms, are widely investigated. An impressive set of methods and tools already exist and have proven their value in laboratory settings. However, in a more diversified, real world setting, scaling up and putting the complex pieces together remains a challenge.

In this demonstration session, we showcase three practical annotation strategies: speech recognition, video concept labeling with users in the loop and a video labeling game.

2. SPEECH RECOGNITION

There is common agreement that deploying speech recognition technology for generating time-labeled annotations for audiovisual content based on the spoken words therein can be a useful strategy. However, success stories of the application of speech-based annotation for real world archives lag behind. After less successful attempts to use speech recognition technology for annotating highly heterogeneous historical data with low audio quality, NISV now focuses on automatic speech-based annotation strategies for Radio and content digitized in Images for the Future. Radio content is only sparsely annotated and as a consequence practically inaccessible in the archive. The speech application in the demonstration session shows the interface that is used by professional archivists within the archive to monitor transcription quality. This stems from the fact that task domains for speech recognition need to be selected carefully (and monitored).

3. VIDEO CONCEPT LABELING

This demonstration showcases video concept labeling and crowdsourcing using video footage of the Pinkpop rock festival that was digitized in the Images for the Future project [4]. The application is a real-world video search engine based on advanced multimedia retrieval technology, which allows for user-provided feedback to improve and extend automated content analysis results, and share video fragments. The main mode of user interaction with the video search engine is by means of a timeline-based video player. The player enables users to watch and navigate through a single video concert. Little colored dots on the timeline mark the location of an interesting fragment corresponding to an automatically derived label. To inspect the label and the duration of the fragment, users simply move their mouse cursor over the colored dot. By clicking the dot, the player instantly starts the specific moment in the video. If needed, the user can manually select more concept labels in the panel on the left of the video player. If the timeline becomes too crowded as a result of multiple labels, the user may decide to zoom in on the timeline. Besides providing feedback on the automatically detected labels, we also allow our users to comment on the individual fragments, share the fragment through e-mail or Twitter, and embed the integrated video player, including the crowdsourcing mechanism, on different websites.

[1] http://beeldenvoordetoekomst.nl/en

4. VIDEO LABELING GAME

The third demonstration shows how engaging users in tagging videos through so called "games with a purpose" could work. "Waisda" is a multi-player video labeling game [5,6], launched in 2009, where players describe video by entering tags. Players score points based on various temporal tag agreements. The underlying assumption is that tags are probably valid if they are entered independently by at least two players within a given time-frame.

5. REFERENCES

[1] B. Huurnink, L. Hollink, W. van den Heuvel, and M. de Rijke. 2010. Search Behavior of Media Professionals at an Audiovisual Archive: A Transaction Log Analysis. *Journal of the American Society for Information Science and Technology*, 61(6):1180-1197.

[2] W. van den Heuvel. 2010 Expert search for radio and television: a case study amongst Dutch broadcast professionals *Proceedings of the 8th International Interactive Conference on Interactive TV&Video*. Tampere, July 2010. Pages: 47-50.

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

[4] Cees G. M. Snoek, Bauke Freiburg, Johan Oomen, and Roeland Ordelman, "Crowdsourcing Rock N' Roll Multimedia Retrieval," in *Proceedings of the ACM International Conference on Multimedia*, Firenze, Italy, 2010.

[5] L. B. Baltussen. Waisda? video labeling game: Evaluation report, 2009. http://research.imagesforthefuture.org/index.php/waisda-video-labeling-game-evaluation-report/

[6] J. Oomen, and L. Belice Baltussen, and S. Limonard, and A. van Ees, and M. Brinkerink, and L. Aroyo, and J. Vervaart, and K. Asaf, Kamil and R. Gligorov 2010 Emerging Practices in the Cultural Heritage Domain - Social Tagging of Audiovisual Heritage. In: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, April 26-27th, 2010, Raleigh, NC: US.

Q-go Natural Language Search

Fabrice Nauze
Q-go
Eekholt 40
1112 XH Diemen
+31 20 53 13 800
fabrice.nauze@q-go.com

ABSTRACT

In this demo, we showcase Q-go Natural Language Search technology.

Keywords

Q-go, Natural Language Search, online customer service.

1. INTRODUCTION

Q-go's online customer service is based on a sophisticated semantic and natural language search software. Q-go helps consumers find what they seek and achieve their goals on your website. We support organizations and help them cope with the following key issues. Q-go enables people to ask questions using their own words and language. The questions are analyzed both grammatically and semantically and a small set of relevant answers is returned within two clicks along with contextual services or offerings that can be easily read even on a mobile device.

2. WEBSITE SEARCH SPECIFICITY

2.1 Customer language

Q-go enables consumers to ask questions on corporate webpages using their own words and language. Most search queries can be classified into three types: full sentences, telegram style and keywords. It is furthermore in the nature of webpage queries to contain typos as well as linguistic errors. A key requirement of Q-go's system is thus to be robust to customer query formulation.

2.2 Customer facing

One specific point concerning website search as well as customer self-care and support is its intertwinement with marketing and branding material. Quality and relevance of the answers are therefore central to the customer experience.

In order to give the most relevant results in a controlled fashion Q-go has opted for an indirect construction where the central concept is that of a model question. In a nutshell, a model question represents an information need that is most often, though not necessarily, related to the webpage content.

Q-go's core natural language technology is thus not based on the indexing of the website's content.

3. Q-GO NATURAL LANGUAGE SEARCH

3.1 Basic idea

The system is thus based on the idea to find out which model questions match best a given customer query. In order to make optimal use of any lexical, syntactic, and semantic information available in a query, Q-go applies its proprietary natural language technology.

3.2 Lexical analysis

The first process of the linguistic analysis is lexical analysis. The customer query is tokenized and the resulting strings are looked up in Q-go's dictionaries. Spelling correction is an integral part of this process. Because Q-go has historically been active on the Dutch, German as well as the Spanish market, compound analysis and clitic splits are also applied whenever possible. Multiword units and regular expressions are also analyzed. This is particularly relevant in the context of corporate webpage with product names or numbers.

The end output of this process is a list of lemmatized lexical entries with their parts of speech. Synonyms like 'ATM' and 'Cash machine' are retrieved by the system. The lexical entries returned also contain spelling corrected words.

3.3 Syntactic analysis

Based on the output of the lexical analysis and on Q-go's grammar a syntactic analysis of the input query is built. This process is also tied up to the building of a semantic representation for the query. Q-go's syntactic analysis is an adaptation of an Earley parser. The main deviation from a standard Earley parser comes from the fact that Q-go's context-free grammar also codes semantic information together with its syntactic rules. The parser performs thus two tasks 1) building a parse tree and 2) building its associated semantic representation.

As the input lexical entries for the parsing algorithm may contain spelling correction Q-go's system can also correct real-words errors. The output of syntactic analysis is a structured representation containing merely lemmas.

The basic idea of the system being of finding the best model question to answer a customer query it is only natural to analyze our model questions in the manner just described. We therefore end up with having to compare the meaning representation of a user query with the meaning representations of the model questions.

3.4 Matching

Because the customers must be able to formulate their queries in their own words and language we must provide some flexibility to the matching. We cannot expect to match exactly lemmas from the customer query's representation with lemmas of the model question's representation. To smooth this process Q-go uses hierarchical information. Connected lemmas, i.e. conceptually connected concepts, are matched at the cost of a penalty.

To further improve the flexibility and range of the matching Q-go also implements rules to automatically translate representations.

The matching process outputs a ranked list of best matches of which the top best 5-7 matches are shown to the customer.

3.5 Fallback mechanism

Finally a keyword matching mechanism based on an index of the model questions is consistently used in parallel of the natural language technology.

4. CONCLUSION

Q-go provides a flexible and robust natural language search technology that enable companies to provide excellent search and support capabilities as well as the related sales opportunities.

Finally Q-go supports research on ways to improve its core technology and functionalities with machine learning methods and continuously tries to improve customer usability with advanced interaction schemes.

Textkernel Semantic Search for Recruiters

Henning Rode
Textkernel BV, Amsterdam
rode@textkernel.nl

Jakub Zavrel
Textkernel BV, Amsterdam
zavrel@textkernel.nl

ABSTRACT

We present the new Textkernel CV search application that uses automatic information extraction of domain-specific semantics for searching in a collection of unstructured CV documents. The state-of-the-art user interface enables non-expert users to effectively search and explore a set of CV's in order to quickly zoom in on the most relevant candidates. The user interface provides faceted search, tag clouds, and a simple means of constructing and manipulating complex structured match profiles. The ranking makes use of CV-specific knowledge such as career weighted relevance, and synonyms for job titles, degrees and skills. Queries are automatically interpreted with respect to the domain specific semantics and the application aims to provide federated search across multiple CV repositories.

1. INTRODUCTION

Recruiters need to search in large collections of CV's (Curriculum Vitae) for suitable candidates to find those with the best possible match on a multitude of criteria, including work experience, education, availability, location, seniority, specific languages and computer skills. However, the perfect candidate usually does not exist. CV search tasks therefore often require to construct complex structured queries and to find a set of best possible candidates ranked in order of relevance. Some query terms are hard criteria, others are important but not required, yet others are nice to have. The candidate profiles are usually are present in recruitment CRM systems or applicant tracking systems in the form of unstructured CV documents. Since manual data entry is often too time consuming, and structured search would otherwise not be optimal, Textkernel has over the past 10 years developed CV parsing software for many languages. CV parsing software automatically recognizes the document structure and extracts information such as the personal data of a candidate, language or computer skills, and even derived information such as the total number of work experience years. The demo shows that the automatic extraction, while not 100% perfect, provides a level of precision that offers a breakthrough in effective CV search without any manual correction of the extracted data.

Our approach in building a usable semantic search application for recruiters started out with the following design goals:

(1) Recruiters are most often not information retrieval experts. We need to provide a user-friendly interface that allows non-expert users to issue complex queries intuitively;

(2) The relevance of the top candidates must be beyond doubt. We must provide advanced relevance ranking using state-of-the-art language models, document structure and external domain-specific knowledge;

(3) Fast retrieval performance on large CV databases;

(4) Candidates must be searchable from the moment they apply. This makes live indexing of new CV's required;

(5) The right candidate is just as likely to be on LinkedIn or Facebook as in the recruiters set of applicants. The application should provide integrated searching in internal as well as external (online) CV databases.

We focus in the reminder of the article on the first two design goals, the search interface and strategies to improve the ranking based on domain-specific knowledge.

2. USER INTERFACE

The search interface of our CV search application provides several means to construct queries and to tune and manipulate them to further refine the results:

- facet browsing
- field-based tag-clouds
- "bread crumb"-like query overview
- controls to weight query parts
- robust powerful query language

The faceted search interface enables to explore the collection and to drill down search results. We show aggregated counts on facet items to guide the user when trying to fine-tune the query results. The interface is highly configurable and can show arbitrary numeric range and/or category facets.

For fields containing short strings, not limited to a small set of unique items, the interface offers to show tag-clouds instead of static facets. Tag-clouds can visualize a larger number of items than a common facet menu. They summarize the characteristics of the current result set and provide useful suggestions for query expansion. Instead of clicking a tag-cloud term, we also allow the user to enter free terms that are not shown in the cloud. This way users can easily construct structured queries without an additional advanced

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

search interface.

Moreover, we provide control both on facet and tag-cloud selections to mark query parts as mandatory or desirable. The latter will relax the specific query condition, allowing to re-rank the result set by the facet selection instead of filtering them.

Since queries often become complex during refinement, it is helpful to provide an overview on all selected constraints coming from the different interface elements. To this end, we decided to show all currently selected query parts in the style of bread crumbs. In contrast to the original bread crumb idea, query selections are grouped according to fields instead of presenting them as a search history in chronological order. However, our bread crumbs still provide means to control the entire query and to easily deselect specific query parts. Bread crumbs also play a role in giving the user feedback about the system's interpretation of natural language queries, synonym expansions, etc.

Besides the graphical elements of the user interface, the CV search also provides a robust and powerful query language for expert users. The query language offers all controls that can be selected by other interface elements and even extends their power by offering proximity or weighting features.

3. SEMANTIC RANKING STRATEGIES

The CV search application currently applies three strategies to improve the ranking beyond standard language model based full-text retrieval by using domain knowledge:

- synonym expansion
- automatic query field interpretation
- parametrized indexing

Synonym expansion exploits specially created thesauri on e.g. job titles, or skills to expand the query. The employed thesauri contain not only synonyms but also weights representing the relation strength between two expressions. The weights are first used to determine which synonyms should be used for query expansion, and secondly for weighting the terms inside the expanded query.

If a user states a query such as "*web developer london*", we would like to automatically recognize that *web developer* is a job description, whereas *london* refers to a city and should be matched against the address of the candidate. Using the same thesauri as employed for synonym expansion we can automatically recognize job titles, skills or cities in queries and assign those query parts to the corresponding fields of the CV. The CV search clearly shows the advantage of structured queries over simple full text queries. The above full-text query would e.g. also match a candidate working years ago in *London* for an internship. Since recognition in queries and documents does not succeed in all cases we use the full-text ranking as a fall-back strategy.

Special attention is given in our CV search to the ranking of previous work experiences. The recognition of job titles in the CV is not enough for effective ranking. A recruiter would expect candidates with a longer or more recent work experience to rank higher than other candidates that have worked only briefly or years ago in the specified area. However, the statistical language model of a document does not take into account the recency or duration of experiences. Since our CV parsing engine can recognize work experiences in the text as well as their corresponding begin and end dates, we have all necessary input to modify the representations of term statistics used for relevance ranking before indexing. Such parametrizable indexing shows to be quite effective for experience ranking. It can also be useful for ranking skills or education items of the CV, but following different configurations for boosting term statistics in such cases.

4. CONCLUSION

Our CV search application combines a number of state-of-the-art search interface elements that enable users to effectively search large databases of CVs. We also demonstrate how to improve standard text ranking models by making use of domain knowledge in form of specialized thesauri and parametrized indexing strategies that capture our external insight in the relative importance of different CV sections. Current development focuses on more advanced query interpretation and on integrating federated search in external CV databases.

AquaBrowser – associative catalog search

ir. R.C.P. van der Veer
Serials Solutions Medialab
Modemstraat 2
1033 RW Amsterdam
+31 (0)20 635 3190

rob.vanderveer@serialssolutions.com

ABSTRACT

This demo paper describes the information retrieval system AquaBrowser, as developed by Serials Solutions Medialab.

Categories and Subject Descriptors

H3.3.3 [Information storage and retrieval]: Information Search and Retrieval – *Clustering, information filtering*

General Terms

Algorithms, Design

Keywords

Information retrieval, search, text mining, word association, fuzzy search, clustering, relevance ranking, library

1. INTRODUCTION

The main challenge with searching library catalogs is a universal information retrieval challenge: how to understand what the user means. AquaBrowser combines several methods for query understanding by providing ways for the user to make clearer what is meant. According to user tests, this substantially improves search results.

AquaBrowser is used by visitors of several hundreds of libraries worldwide to search through catalogs and external sources for books, articles and music.

2. QUERY UNDERSTANDING

AquaBrowser starts by providing a search box in which the user enters a query. Next, the system provides a search result, based on relevance ranking, accompanied by visualizations of terms that provide the user with several ways to better indicate what to look for: facets, associations, spelling variations and synonyms.

By clicking on these terms, new search results appear and the relevance ranking will take the indications into account. For example: search on 'Jaguar' and then click the right associated term to indicate the animal was meant, not the car brand.

Copyright is held by the author/owner(s).
DIR'2011, February 4, 2011, Amsterdam.

2.1 Conceptual grouping

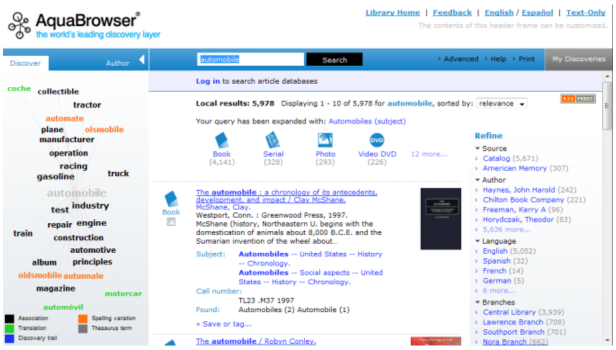
Words that can be associated with the search terms are shown in an interactive visual presentation of a graph – the so-called *word cloud*. These associations are based on a conceptual grouping algorithm[1] that has scanned reference documents to look at word co-occurrence, applying the necessary language handling such as stemming.

2.2 Other search extensions

The so-called *facets* that are presented with a search result refer to properties of the items in the catalog: year of publication, author, type of material, language, etc. By clicking these facets, the search is refined.

The entire clicking behavior during the search session is used in the ranking to try to understand the direction the user wants to go with the search. The rich visual presentation of the suggestions encourages users to explore and discover more information.

By offering social networking options, users can create personal profiles they can share with others, providing even more information that allows the system to better associate what people are interested in.



Log in to search article databases

Local results: 5,978 Displaying 1 - 10 of 5,978 for automobile, sorted by: relevance

Your query has been expanded with: Automobiles (subject)

The automobile : a chronology of its antecedents, development, and impact / Clay McShane.
McShane, Clay.
Westport, Conn. : Greenwood Press, 1997.
McShane (history, Northeastern U. begins with the domestication of animals about 8,000 B.C.E. and the Sumarian invention of the wheel about...
Subject: Automobiles -- United States -- History -- Chronology.
Automobiles -- Social aspects -- United States -- History -- Chronology.
Call number: TL23 .M37 1997
Found: Automobiles (2) Automobile (1)

The automobile / Robyn Conley.

Figure 1: AquaBrowser screenshot

3. REFERENCES

[1] Veling, A. and Van der Weerd, P. 1999. *Conceptual grouping in word co-occurrence networks*. In *Proceedings of the International Joint Conference on Artificial Intelligence* (Stockholm, Sweden, July 31-August 6,1999*)*

Termtreffer: an Automatic Term Extractor for Dutch

Dennis de Vries
GridLine
Keizersgracht 520
1017EK Amsterdam
+31 20 6162050
dennis@gridline.nl

ABSTRACT

TermTreffer is an application for automatically extracting domain terms from documents. For extraction it uses a large variety of linguistic and statistical modules. On the one hand users can use standard configurations of these modules, but on the other hand, expert users can combine modules to create their own custom extraction pipelines.

Apart from automatic term extraction, TermTreffer can also be used to browse, compare and edit (extracted) termbanks. It also contains functionality for exporting and importing termbanks to and from most common formats used for terminology.

Keywords

Term extraction, Dutch, terminology, computational linguistics, statistics, unithood, termhood

1. DESCRIPTION

Terminology is an important ingredient for language related software applications at governmental institutions and companies. For example, knowledge management, searching in text and speech, opinion mining, classification and improvement of communication. GridLine supports this kind of applications by using language technology for the Dutch language. It forms the basis for terminology extraction, term enrichment, automatic merging of termbanks, terminology management and standardization for writing assistance.

Earlier this year, GridLine built an application called TermTreffer for the Nederlandse Taalunie. It enables organisations to keep full control of creating and maintaining lists of their specific terminology. With TermTreffer, users can easily create term lists based on their own documents, and their own organisation language. This is done by performing automatic term extraction using various linguistic and statistical algorithms. Apart from this, TermTreffer offers functionality for (semi)automatic merging of termbanks and term enrichment (= adding term properties).

The extraction process starts with a number of linguistic processing steps which enrich the plain text and create an initial set of term candidates. Some examples of these linguistic modules are: Tokenizer, Lemmatizer, POS-tagger, Compound Splitter, Spellchecker, Chunker and Multi Word Unit Detector. These last two modules are able to make a first selection of term candidates, based on their syntactic profile.

The terms extracted by the linguistic modules are then passed on to the statistical modules. These modules can be split up into two categories: Unithood and Termhood.

Unithood modules are used to determine the strength of multi word terms. For example, the term "baseball bat" will have a high Unithood value because these words, in this order, have a strong connection. The term "big bat" on the other hand will have a low Unithood value because this combination of words is less common. TermTreffer offers a number of statistical evaluation methods for calculating Unithood values of term candidates and filtering out bad candidates.

Termhood modules determine the measure in which a term is representative for a document collection, and thus for a domain. A commonly used method that TermTreffer offers is corpus comparison, in which term frequencies in the user's documents are compared to their frequencies in a general corpus. Terms that have a significantly higher relative frequency in the user's own documents are considered important term within the domain represented by these documents.

Another Termhood method available in TermTreffer is called Distance Statistics. It calculates a Termhood value based on the assumption that important domain terms don't occur evenly spread throughout document collections, but are concentrated in certain documents or paragraphs. Term candidates that occur regularly in all sections of the user's texts get a low score, whereas term candidates that have a high frequency in a few passages get a high score.

For unexperienced users, there is a one-click extraction option which uses the standard configuration of extraction modules. Expert users can fully configure their own extraction pipelines, including or excluding modules and setting module parameters. This distinction makes TermTreffer very usable for a wide variety of users.

After extraction, users can manage and modify the resulting termbank as they wish. This view also shows linguistic properties of terms which were determined during extraction, like their lemma, their head (for compounds and multi word terms), their gender or their Part of Speech tag. Also, occurrences of terms in the text can be viewed as well as the different left and right contexts a term appears in and termbanks can be compared to other termbanks.

Termbanks can be exported and imported to and from a variety of commonly used formats, enabling compatibility with other terminology software applications.

2. DEMO

The demo will consist of a live demonstration of the TermTreffer application, showing automatic extraction of terms from a text, ways in which extractions can be customized and functionality for analyzing and editing resulting termbanks.

eCare: Online Sentiment Monitoring

Thijs Westerveld, Stefan de Bruijn, Arthur van Bunningen, Rolf Schellenberger, and Sylvain Perenes

Teezir
Kanaalweg 17 L-E
3526 KL Utrecht
+31 30 267 9648
info@teezir.com

ABSTRACT

This paper presents eCare, a web-based dashboard for online reputation monitoring. eCare allows companies to monitor what is said online about their brands and products, to gauge the online sentiment, to identify the opinion leaders in the sector, and to find the sources where their brands and products are actively discussed.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms

Algorithms, Experimentation, Verification.

Keywords

Online reputation management, Sentiment analysis, Web crawling

1. INTRODUCTION

People are influenced by their peers. Recommendations from friends, family and colleagues are an important factor in deciding where to eat, what places to visit, which movies to see and what to buy. We get spontaneous, unsolicited advice from people around us all the time; often we trust the information and act on it. In fact, word of mouth is one of the most important factors in consumers' behavior. When consumers are asked which forms of advertisement they trust most, *recommendations from other users* and *opinions posted online* are amongst the top answers [1].

Traditionally, word of mouth has been limited to face-to-face, spoken communication, but today, online forms of communication have become equally important. Blogs, forums and social media (Twitter, Hyves, Facebook, etc.) support word of mouth product recommendation and become more and more important to determine consumer's behaviour. This demo shows eCare, Teezir's online sentiment monitoring dashboard [2]. eCare is a flexible tool to monitor the online communication around brands, products and topics. This paper describes the underlying technology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DIR'11, February 2, 2011, Amsterdam, The Netherlands.

Copyright 2010 ACM 1-58113-000-0/00/0010…$10.00.

2. eCare Technology

Two important aspects of eCare as an online reputation monitoring tool are the flexibility of the tool and the quality of the results. Since eCare's data collection is not centred on pre-defined search terms, users are free to query for their own topics. This way, even new users have instant access to a wealth of historic information that has been collected over the years. Moreover, users can construct their own dashboards choosing from a variety of result presentations.

2.1 Crawling

The content in eCare is collected by continuously monitoring a fixed set of the most important blogs, forums, news sites and social media platforms in the Dutch language domain. Specific crawlers have been trained to recognize the patterns of the various source types (news, blogs, and forums). Based on a small set of training examples, the crawlers have learned the characteristics of the links to follow and the content elements to store. This way content can be extracted cleanly from both known and new sources: individual posts and their metadata are collected while menu structures, advertisements and other distractors that may be present on a webpage are ignored. Additional content is collected from RSS feeds and social media APIs.

While the set of sources monitored for eCare is not the complete Dutch internet, the carefully selected set provides a good picture of the topics that are discussed online and of the corresponding sentiments. Still, in some cases it is useful to know whether more discussion takes place outside these sources or whether important sources are missing. To identify these *blind spots*, eCare makes use of external search engine APIs. A change in the volume of relevant content for a specific site can lead to further study of the site by the eCare user, or eventually to adding this site to the set of monitored sources from which clean content is collected.

2.2 Sentiment Analysis

All collected content is automatically analysed to determine the predominant sentiment in the document. The main sentiment expressed in a document is formed by the words and sentences of the document. Based on lists of terms that are known to have a clear positive or negative connotation, the overall sentiment of phrases, sentences and documents is determined. We use part of speech tagging to take the context of a term into account. For example, the term *sound* can be opinionating or neutral depending on its context. Compare for example the following phrases: *"their judgment is always sound"*, *"…with sound foundations"*, *"I have a 16bit sound card…"* The first two are clear positive statements, the last is neutral.

Context is also analysed for modifiers that strengthen, weaken or reverse the sentiment of a phrase. This way we can deal with negations and subtleties like *"…the mouse itself is not exactly ergonomically shaped…"*, *"…simple, but somewhat awkward…"* and *"…extremely solid and easy to use…"*. The final sentiment expressed in a document is a function of the weights of the document's opinionating expressions.

The base list of positive and negative sentiment terms originates from the Instituut voor de Nederlandse Lexicografie. This list is constantly adapted to the language encountered in our data sources. In regular tuning sessions we manually adapt and extend the lists to keep up with the evolving language use in social media and other online sources.

2.3 Index

To efficiently combine relevance and sentiment scores, we developed our own inverted-file based indexing structures. These indexes contain the collected content, their metadata and the computed sentiment scores.

Adding newly collected content and searching previously indexed content happens on the same files. This way there is no need to swap indexes to be able to access new data. Newly added data is initially kept in memory and only committed to the on-disk indexing structures later. A carefully designed recovery process is in-place to be able to reconstruct the non-committed indexing data when something goes wrong during the indexing process.

The eCare indexing structures are designed for efficient computation of both relevance and sentiment scores. We need to be able to efficiently rank documents on either of these scores. At the same time computing aggregate volume, sentiment and relevance scores for sets of documents should be efficient.

2.4 Dashboard

eCare allows its users to slice and dice the collected content, and learn what people say, either at the very aggregated level: "What is the share of positive versus negative views about our new product?", or at the very detailed level: "Which sources reflect this negative sentiment, and what exactly are people saying?". Choosing from a range of available widgets, users can compose their own dashboards and share them with team members.

A query can be formulated using terms, phrases and Boolean operators. On top of that, filters can be applied to zoom in on specific timeframes, sources or authors. The available views of the data include the following:

- Document results: a relevance ranking of the matching documents
- Sentiment overview: an indication of the overall sentiment in the matching documents
- Timeline: showing the development of volume and sentiment over time (Figure 1)
- Related terms: showing the most distinguishing terms in the matching documents as well as whether they appear mostly in positive or negative documents (Figure 2)
- A breakdown of volume and sentiment by source type (news/blogs/forums/etc.), by individual source or by author

eCare sends email alerts when the sentiment drops below a user-defined threshold. Users can export results to further analyse or combine with external data. Finally, users can directly engage with the authors of blog or forum posts and administer these actions in the eCare dashboards.

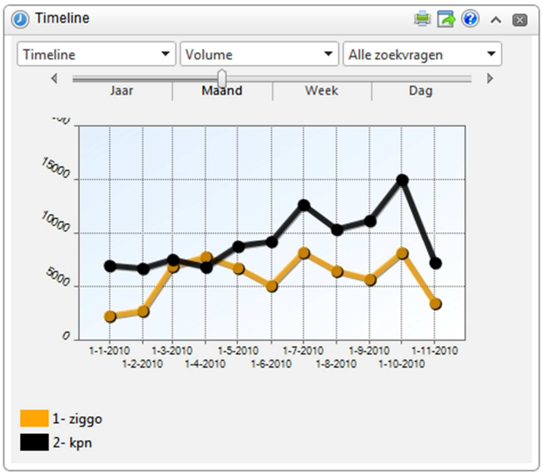


Figure 1 - Timeline comparing the volume of results for Ziggo and KPN between January and November 2010

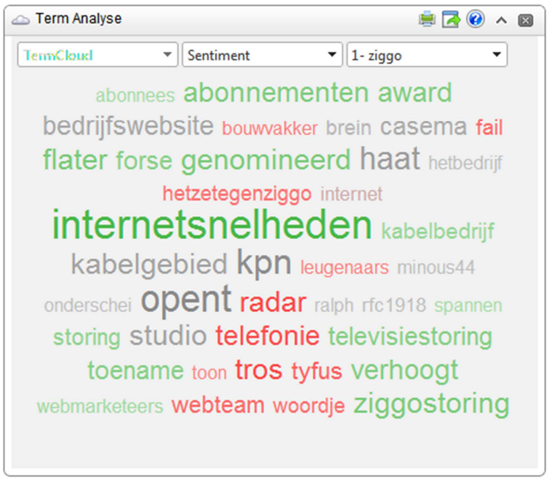

Term Analyse
TermCloud
Sentiment
1- ziggo
abonnees abonnementen award
bedrijfswebsite bouwvakker brein casema fail
flater forse genomineerd haat hetbedrijf
hetzetegenziggo internet
internetsnelheden kabelbedrijf
kabelgebied kpn leugenaars minous44
onderschei opent radar ralph rfc1918 spannen
storing studio telefonie televisiestoring
toename toon tros tyfus verhoogt
webmarketeers webteam woordje ziggostoring

Figure 2- Term cloud showing the most distinguishing terms in the context of Ziggo, including the context in which they appear (green for positive, red for negative)

3. REFERENCES

[1] The Nielsen Company. 2009. Nielsen Online Consumer Survey. http://blog.nielsen.com/nielsenwire/consumer/global-advertising-consumers-trust-real-friends-and-virtual-strangers-the-most

[2] http://www.ecarewebcare.nl/

Notes

Notes

DIR 2011 is organized under the auspices of: