PROCEEDINGS

26/04/2013

The 13th Dutch-Belgian
Information Retrieval
Workshop
Delft, The Netherlands

DIR '13
DUTCH-BELGIAN
INFORMATION RETRIEVAL WORKSHOP
13TH edition

PROCEEDINGS DIR 2013

We would like to welcome you to the 13th edition of the Dutch-Belgian Information Retrieval Workshop. The DIR series serves as a platform for exchange and discussion on information retrieval research and application in the Netherlands and Belgium, as well as internationally. Again, this year, we are boasting a high-quality program of research contributions, discussing relevant challenges in the domains of information retrieval, natural language processing and data mining.

The DIR team,

Carsten Eickhoff
General Chair

Ester Smith
Local Organization & Graphic Design

Saskia Peters
Local Organization

Jeroen Vuurens
Local Organization

Thomas Demeester
Local Organization

Claudia Hauff
Local Organization

Stefan Ferdinandus
Web Hosting

TABLE OF CONTENTS

KEYNOTE

FULL RESEARCH CONTRIBUTIONS

COMPRESSED CONTRIBUTIONS

POSTERS & DEMONSTRATORS

The Return of the Probability of Relevance

Norbert Fuhr
The University of Duisburg-Essen
Duisburg, Germany
norbert.fuhr@uni-due.de

Abstract

The probability ranking principle (PRP) proves that ranking documents by decreasing probability of relevance yields optimum retrieval quality. Most research on probabilistic models has focused only on producing a probabilistic ranking, without estimating the actual probabilities. In this talk, we discuss models for three types of modern IR applications which rely on calibrated values of the probability of relevance.

1. Vertical search deals with the aggregation of documents with different types or media (such as, e.g., Web pages, news, tweets, videos, images) in response to a query. Based on the probabilistic estimation of the number of relevant documents per resource, the decision-theoretic selection model describes the optimum solution for this problem.
2. The optimum clustering framework provides not only the first theoretic foundation for document clustering, it also proves the clustering hypothesis. Its key idea is to base cluster analysis and evaluation on a set of queries, by defining documents as being similar if they are relevant to the same queries.
3. The interactive PRP generalizes the classical PRP for interactive retrieval. It characterizes each situation in interactive retrieval as a list of choices, where each choice is described as the effort for evaluating it, the probability that the user will accept it, and the benefit resulting from acceptance. By developing appropriate parameter estimation methods, we can describe interactive retrieval by Markov models, which allow for a number of predictions.

With these models, it becomes possible to implement approaches based on solid theoretic foundations, which are more transparent than heuristic approaches, thus allowing for theory-guided adaptation and tuning.

About the Speaker

Dr. Norbert Fuhr is a full professor in the Department of Computer Science at the University of Duisburg-Essen. He obtained his Ph.D in Computer Science from the Technical University of Darmstadt in 1986 where he served as an assistant professor. He became Associate Professor in the computer science department of the University of Dortmund in 1991, before taking up his current position in 2002.

He has published more than 300 papers in the fields of IR, databases and digital libraries. His current research interests are retrieval models, networked digital library architectures, user-oriented retrieval methods and the evaluation of digital libraries.

He has served as regular PC member of many major international conferences related to information retrieval and digital libraries, such as ACM-SIGIR, CIKM, ECIR, SPIRE, ICDL, ECDL, ICADL, FQAS. He was PC chair of ECIR 2002, IR track chair of CIKM 2005 and Co-Chair of SIGIR 2007. For the German IR-group GI-FGIR, he served as Chair from 1992-2008. He also is a member of the editorial boards of the journals Information Retrieval, ACM Transactions on Information Systems, International Journal of Digital Libraries, and Foundations and Trends in Information Retrieval.

In 2012, he received the prestigious Gerald Salton Award in recognition of his significant, sustained and continuing contributions to research in information retrieval.

The committee particularly emphasised his *"pioneering contributions to the theoretical foundations of information retrieval and database systems. His work describing how learning methods can be used with retrieval models and indexing anticipated the current interest in learning ranking functions, his development of probabilistic retrieval models for database systems and XML was ground-breaking, and his recent work on retrieval models for interactive retrieval has inspired new research. His rigorous approach to research and research methods is an outstanding example for our field."*

An Adaptive Window-Size Approach for Expert-Finding

Fawaz Alarfaj, Udo Kruschwitz, and Chris Fox
School of Computer Science and Electronic Engineering
University of Essex
Colchester, CO4 3SQ, UK
{falarf, udo, foxcj }@essex.ac.uk

ABSTRACT

The goal of expert-finding is to retrieve a ranked list of people as a response to a user query. Some models that proved to be very successful used the idea of association discovery in a window of text rather than the whole document. So far, all these studies only considered fixed window sizes. We propose an adaptive window-size approach for expert-finding.

For this work we use some of the document attributes, such as document length, average sentence length, and number of candidates, to adjust the window size for the document. The experimental results indicate that taking document features into consideration when determining the window size, does have an effect on the retrieval outcome. The results shows an improvement over a range of baseline approaches.

Categories and Subject Descriptors

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

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Expert-Finding, Entity Search, Adaptive Window, Proximity Search

1. INTRODUCTION

With the massive and ever-growing amount of electronic data, search engines have become crucial for any organisation that wants to help its employees with their day-to-day information needs. Traditionally, search engines, or information retrieval systems in general, function by returning a list of documents for the user's query, although the user's information need may not necessarily be in the form of documents. In fact, users more often search for specific things (people, organisations, or products) [9]. Many user information needs therefore, would be better answered by specific entities. Studies on user search behaviour show that entity search is the most prominent type of search on the web [5]. This led to the introduction of some entity search engines, such as product search (Google Product Search and Yahoo Shopping).

One special type of entity search is expert-finding. In expert-finding we are concerned with identifying experts who possess the relevant skills and knowledge on a given topic [1]. Today, expert-finding is considered an important task in the area of information retrieval, and it has attracted a great deal of attention and interest within the information retrieval community over the past few years [3]. People have different motives for seeking experts. Yimam-Seid and Kobsa [12] categorise these motives into two main groups, (i.e. expert finding and expert profiling). Firstly, in expert finding, users seeks expert as a source of information, where users are mostly interested in the question, ‘Who knows about topic X?’. Secondly, in expert profiling, the motive is to find someone who can perform a given organisational or social function, where in this case users are equally interested in other questions; for example, ‘How much does Y know about topic X?’, ‘What else does Y know?’ or ‘How is Y compared with others in his/her knowledge of X?’

Given a search topic, state-of-the-art expert-finding systems typically measure the knowledge of candidates from the textual content of top ranking documents, which are used to derive associations between candidates and search topics based on co-occurrences [7, 3]. The co-occurrence of candidate identifiers with query terms is considered to provide evidence of expertise. In addition, the nature and frequency of co-occurrences is used in estimating the probability of a person being an expert. The general assumption is that the more often a candidate is found in a document containing many terms describing the topic, the more likely he or she will be an expert on this topic. The second assumption is that the closer the candidate identifiers are to the query terms, the stronger the association between them. Using these assumptions, some studies consider the proximity of query terms and candidate identifiers using fixed-size windows. Zhu *et al.* tested 31 window sizes on the W3C collection[1] ranging from 5 to 1100. They found the best window size to be around 200 words. According to Zhu *et al.*, small window sizes could lead to high precision, but low recall. On the other hand, large window sizes lead to high recall, but low precision [14]. Some studies therefore, consider multiple

DIR 2013, April 26, 2013, Delft, The Netherlands.

[1] The same collection is used in this paper.

levels of associations in documents by combining multiple fixed window sizes [14, 2].

In this paper, we consider the idea of an *adaptive* window size, where the size of the window is a function of various document features. We argue that each document has distinct features that differ from other documents in the collection. Using these features to set the window size could improve the overall ranking function. There are many document features that could be examined. We focus on three of them: document length, average sentence length, and candidate frequency (i.e. the number of candidates that appear in a document). To the best of our knowledge, no existing work has dealt with using the document features to determine the optimal window size for the proximity function.

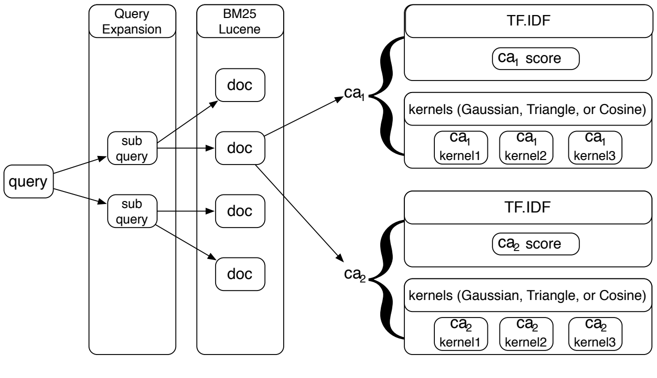
It is important to note that the adaptive window size approach could be applied to any proximity search, in particular for an entity-oriented search, a generalisation of expert search. We carried out the study in the expert search domain due to the availability of an expert-search benchmark.

The main research question considered here is whether an adaptive window size leads to improvements over fixed window size methods.

2. EXPERT-FINDING FRAMEWORK

As described above, the input for any expert-finding system is the user's submitted query. This query then could be normalised and different query expansion techniques could be applied to it. Next, the query is passed to an underlying search engine; in this work, we used Lucene[2] as our search engine, with a BM25 ranking function. For each query, only the top 100 documents returned by the search engine were considered. We used these documents to rank the candidates based on two measures. First, based on their frequency occurrence, and second, based on the proximity between the candidate's evidence and the query occurrences in the document (Figure 1).

Figure 1: Expert-Finding Framework



3. ADAPTIVE WINDOW SIZE FOR PROXIMITY RANKING

Proximity approaches have been successfully used in different applications, which enhance the quality of the retrieval systems. In particular, the work of Petkova & Croft, [11] directly addresses the use of a kernel for proximity of name / query term occurrences in expert search. The other works, among others, which examine proximity in expert search include Macdonald *et al.* [6] and Petkova & Croft [10].

In this work, the window size for the proximity function will be determined for each document based on the following features.

(1) Document Length: According to Miao *et. al.* [8], in large documents, it is more likely to find more occurrences of a query topic. It is also more likely to have irrelevant words (noise) in such documents. Thus, in order to minimise the negative influence of noise, the window size should be relatively smaller as the document gets bigger. **(2) Candidate Frequency**: This term is used to refer to the number of candidates found in a document. When a document has more occurrences of candidates' evidence, the window size should be relatively larger to accommodate more occurrences. **(3) Average Sentence Length**: The window size is adjusted in proportion to the average sentence length (in tokens) in the document. We combine these features in the following equation:

$$\begin{aligned} &Window\ Size = \\ &\frac{\sigma}{3} * (\log(\frac{1}{DocLength}) * \beta_1 \\ &\quad + CanFreq * \beta_2 \\ &\quad + AvgSentSize * \beta_3) \end{aligned} \tag{1}$$

σ is a variable that allows to scale the window size. We explore a wide range of values for σ, (see below). The β weighting factors, which determine each feature's contribution in the equation, have been set empirically, where $\sum_{i=1} \beta_i = 1$. The TREC2005 data includes ten training topics[3]. We used these topics to train our β variables, thus having a clear distinction between test and training data.

Although the proposed model used the three features, we will also report experiments for each feature individually.

After establishing the size of the window, it is applied to every full match for the query found in the document. Then, the candidate evidence neighbouring this term is extracted; each one within the window will be given a weight depending on its distance from the query.

The advantage of this window is that it provides a graded proximity boost. Candidates with an index close to the query terms will receive the highest boost. As the candidate indexes drift further and further away, the boost will gradually decrease until it reaches the end of the window. A document can contain multiple query terms. In this case, we place a window at each occurrence. If, for example, a document has two query terms, two windows are placed, but centred at different locations. If the two windows are close to each other, both windows could boost candidates that appear between them.

Three different kernel functions were used to calculate the weight: Gaussian, Triangle, and Cosine [13].

4. EXPERIMENTS

To evaluate our approach, we used the document collection of the W3C corpus and the test sets of the 2005 TREC Enterprise track. The W3C corpus includes a predefined

[2] `http://lucene.apache.org/core/`

[3] `http://trec.nist.gov/data/enterprise/05/ent05.expert.trainingtopics`

Run		σ	MAP	r-prec	bpref	P@5	P@10	P@20
Baseline		N/A	0.1532	0.2531	0.2749	0.3210	0.2519	0.1908
Gaussian	baseline	N/A	0.3001	0.3554	0.4297	0.5092	0.3595	0.3089
		350	0.3363	0.3808	0.4787	0.5200	0.3900	0.3350
		400	0.3342	**0.3975**	0.4737	0.5200	0.4000	0.3300
		450	0.3454	0.3955	**0.4954**	0.5200	0.4099	**0.3450**
		500	**0.3454**	0.3905	0.4861	0.5200	0.4199	0.3350
		550	0.3443	0.3905	0.4890	0.5200	**0.4299**	0.3400
		600	0.3402	0.3905	0.4851	0.5200	0.4199	0.3350
		650	0.3357	0.3821	0.4792	0.5200	0.4099	0.3350
Triangle	baseline	N/A	0.2358	0.3331	0.3602	0.4023	0.3329	0.2750
		350	0.3126	0.3642	0.4494	0.4800	0.4099	0.3199
		400	0.2974	0.3509	0.4427	0.4800	0.4099	0.3199
		450	**0.3261**	0.3793	**0.4623**	0.5199	**0.4299**	**0.3300**
		500	0.3169	**0.3804**	0.4330	0.5600	0.4200	0.3050
		550	0.3144	0.3776	0.4209	0.5600	0.4099	0.3050
		600	0.3036	0.3767	0.4093	**0.5800**	0.3800	0.2950
		650	0.2836	0.3490	0.3869	0.5400	0.3900	0.2800
Cosine	baseline	N/A	0.2700	0.3605	0.4078	0.4102	0.3495	0.3095
		350	0.2735	0.3557	**0.4494**	0.4219	0.3999	0.3499
		400	0.2757	0.3414	0.3149	0.4191	0.4199	0.3599
		450	0.2761	0.3498	0.3149	0.4191	0.4199	0.3599
		500	**0.2811**	0.3639	0.3199	**0.4241**	**0.4399**	**0.3599**
		550	0.2800	0.3639	0.3199	0.4232	0.4399	0.3599
		600	0.2756	0.3639	0.3149	0.4155	0.4399	0.3599
		650	0.2744	0.3639	0.3149	0.4155	0.4199	0.3599

Table 1: The performance of the Adaptive Window-size Approach for different proximity functions. Highest scores for each category are typeset in boldface. The best run overall are typeset in boldface and underlined.

list of 1092 candidates, 331,037 documents, and 50 topics, each of which is provided with a relevance judgement. We selected this collection in order to test our method on a simple and most basic form of expert-finding[4].

We removed stopwords and HTML markup, and treated all documents as plain text. For evaluation, we applied a range of standard IR measures, but in our discussion we focus on Mean Average Precision (MAP).

In this work, we use the two-stage model for the initial candidate ranking by calculating the probability of the candidate given the query, $P(ca|q)$, as follows:

$$P(ca|q) = \sum_{d} P(d|q) \cdot P(ca|d) \qquad (2)$$

where $P(d|q)$ is the document relevance to the query, which is calculated by the underlying search engine, and $P(ca|d)$ is the candidate's probability given the document. In our baseline, $P(ca|d)$ is calculated using the full document without a proximity function. Whereas in all other experiments, we apply Equation 1 to find the optimal window size for the current document. The proximity functions will only consider the occurrences within this window of text.

Feature	CanFreq	AvgSentSize	DocLength
Best σ value	250	600	450
MAP	0.2806	0.2798	0.2777
bpref	0.3452	0.3269	0.3452
r-prec	0.4147	0.4199	0.4112
P@5	0.4199	0.4189	0.4199
P@10	0.3599	0.3499	0.3499
P@20	0.3100	0.3100	0.3050

Table 2: The performance of the Adaptive Window-Size Approach using a single feature. Only the best result for each feature is reported.

Our first baseline is a frequency-based approach. In this baseline, a $TF-IDF$ weighting scheme is used in order to obtain the candidate's importance in a particular document, while at the same time integrating it with the candidate's general importance [2]:

$$P(ca|d) = \frac{n(ca,d)}{\sum_{ca'} n(ca'd)} \cdot \log \frac{|D|}{|\{d' : n(ca,d') > 0\}|} \qquad (3)$$

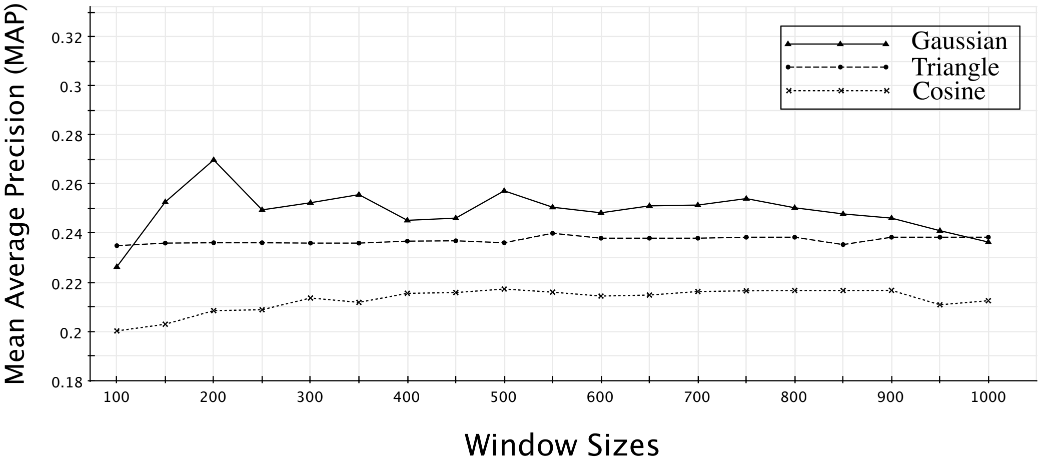
where $n(ca,d)$ is the number of times the candidate ca appears in the document d and $|D|$ is the total number of documents in the collection.

Starting from the baseline, we used the proximity functions with adaptive window size to boost the relevance score.

To test the effect of each document feature separately, we first generated an adaptive window size with only one feature and used it with a Gaussian proximity function. In Table 2, we report the best runs for each feature separately (i.e. CanFreq with $\sigma = 250$, AvgSentSize with $\sigma = 600$, and DocLength with $\sigma = 450$).

[4] Other forms of expert finding include finding similar experts and finding all expertise for a given candidate.

Figure 2: MAP for fixed window sizes



We used our adaptive window-size method (Equation 1), with the three proximity functions at different σ values ranging from 0 to 1000 with an increment of 50. We only report the results for σ values between 350 and 650. The results below 350 and above 650 drop gradually, so they were not reported. Furthermore, we calculate a baseline for each proximity function. In this baseline, we set the window size to be equal to the document length. Our results are summarised in Table 1.

The top MAP of 0.3454 is achieved using a Gaussian proximity function with an adaptive window size where $\sigma = 500$.[5] We found that the difference between our best run and the baseline is statistically significant (using paired t-tests on average precision values at $p < 0.05$). Moreover, we found that the differences between the best run for each proximity function and its baseline were also statistically significant.

For comparison, we used a range of fixed window sizes. We calculated MAP for fixed windows in a range from 100 to 1000 in increments of 50. We repeated the experiments using the three proximity functions (Gaussian is shown to be significantly better than the other two functions, with a top result of MAP=0.27 at a window size of 200); see Figure 2.

5. CONCLUSIONS

We introduced the idea of an adaptive window size for expert-finding. Thus, for the proximity function, the size of the window will be set based on current document features rather than a fixed window for all documents in the collection. Adopting this method results in significant improvements over standard metrics. This is true for all proximity functions used in this study (i.e. Gaussian, Triangle, and Cosine). We found that the best results were achieved using a Gaussian function. As for future work, we plan to investigate the effectiveness of using other document features such as the readability index for determining the optimal window size. We also plan to test the adaptive window size method on other expert-finding collections and also on other TREC benchmarks.

6. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval: the concepts and technology behind search.* Addison-Wesley, Pearson, 2ed edition, 2011.

[2] K. Balog, L. Azzopardi, and M. de Rijke. A language modeling framework for expert finding. *Inf. Process. Manage.*, 45(1):1–19, Jan. 2009.

[3] K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. Expertise retrieval. *Foundations and Trends in Information Retrieval*, 6(2-3):127–256, 2012.

[4] N. Craswell, A. de Vries, and I. Soboroff. Overview of the TREC-2005 enterprise track. In *TREC 2005 Conference Notebook*, pages 199–205, 2005.

[5] B. Jansen and A. Spink. How are we searching the world wide web? a comparison of nine search engine transaction logs. *Information Processing & Management*, 42(1):248–263, 2006.

[6] C. Macdonald, D. Hannah, and I. Ounis. High quality expertise evidence for expert search. *Advances in Information Retrieval*, pages 283–295, 2008.

[7] C. Macdonald and I. Ounis. Searching for expertise: Experiments with the voting model. *The Computer Journal*, 52(7):729–748, 2009.

[8] J. Miao, J. X. Huang, and Z. Ye. Proximity-based rocchio's model for pseudo relevance. SIGIR '12, pages 535–544, Portland, Oregon, 2012.

[9] G. Mishne and M. de Rijke. A study of blog search. *Advances in information retrieval*, pages 289–301, 2006.

[10] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. pages 599–608, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[11] D. Petkova and W. B. Croft. Proximity-based document representation for named entity retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 731–740, New York, NY, USA, 2007.

[12] D. Yimam-Seid and A. Kobsa. Expert-finding systems for organizations: Problem and domain analysis and the demoir approach. *Journal of Organizational Computing & Electronic Commerce*, 13(1), 2003.

[13] J. Zhao, J. X. Huang, and B. He. CRTER: using cross terms to enhance probabilistic information retrieval. SIGIR '11, pages 155–164, Beijing, China, 2011.

[14] J. Zhu, D. Song, and S. Rüger. Integrating multiple windows and document features for expert finding. *J. Am. Soc. Inf. Sci. Technol.*, 60(4):694–715, Apr. 2009.

[5] For comparison, the best run at TREC 2005 reported a MAP value of 0.2749 [4], but do note that this was in 2005.

Distributional Similarity of Words with Different Frequencies

Christian Wartena
Hochschule Hannover, University of Applied Sciences and Arts
Expo Plaza 12
30359 Hannover, Germany
christian.wartena@hs-hannover.de

ABSTRACT

Distributional semantics tries to characterize the meaning of words by the contexts in which they occur. Similarity of words hence can be derived from the similarity of contexts. Contexts of a word are usually vectors of words appearing near to that word in a corpus. It was observed in previous research that similarity measures for the context vectors of two words depend on the frequency of these words. In the present paper we investigate this dependency in more detail for one similarity measure, the Jensen-Shannon divergence. We give an empirical model of this dependency and propose the deviation of the observed Jensen-Shannon divergence from the divergence expected on the basis of the frequencies of the words as an alternative similarity measure. We show that this new similarity measure is superior to both the Jensen-Shannon divergence and the cosine similarity in a task, in which pairs of words, taken from Wordnet, have to be classified as being synonyms or not.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing Methods, Linguistic Processing*; G.3 [**Probability and Statistics**]: [Correlation and regression analysis]; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language Models,Text Analysis*

General Terms

Experimentation

Keywords

Distributional Similarity, Synonymy

1. INTRODUCTION

For many applications dealing with texts it is useful or necessary to know what words in a language are similar. Similarity between words can be found in hand crafted resources, like WordNet [8], but methods to derive word similarities from large text corpora are at least an interesting alternative. Intuitively, words that occur in the same texts or, more generally, the same contexts are similar. Thus we could base a similarity measure on the number of times two words occur in the same context, e.g. by representing words in a document space. Especially, if we consider small contexts, like a window of a few words around a word, this approach gives pairs of words that are in some dependence relation to each other. De Saussure [3] calls such such relations, defined by co-presence in a linguistic structure (e.g. a text, sentence, phrase, fixed window, words in a certain grammatical relation to the studied word and so on), *syntagmatic* relations. The type of similarity that is much closer to synonymy and much more determined by the meaning of a word, is obtained by comparing the contexts in which a word occurs. This type of similarity is usually called *paradigmatic* similarity or distributional similarity.

Though distributional similarity has widely been studied and has established as a method to find similar words, there is no consensus on the way the context of a word has to be defined and on the best way to compute the similarity between contexts. In the most general definitions the context of a word consists of words and their relation to the given word (see e.g. [6, 2]). In the following we will only consider the simplest case in which there is only one relation: the relation of being in the same sentence. Now each word can be represented by a *context vector* in a high dimensional word space. Since these context vectors are very sparse, often dimensionality reduction techniques are applied. In the present paper we use random indexing, introduced by Karlgren and Sahlgren [7] and Sahlgren [9] to reduce the size of the context vectors. For random indexing each word is represented by a random index vector. The context vector of a word is constructed by addition of the index vectors of all words in the context. Thus the dimensionality of the context vector is the same as the dimensionality chosen for the index vectors. It was shown by Karlgren and Sahlgren [7] that this technique gives results that are comparable to those obtained by dimensionality reduction techniques like singular value decomposition, but requires less computational resources. The similarity of the context vectors, finally, can be used as a proxy for the similarity of words.

In order to evaluate the various methods to define context vectors and the various similarity measures that can be used subsequently, usually the computed similarity of words is tested in a task in which words have to be classified as being synonym or not to a given word. Often the data are taken from the synonym detection task from TOEFL (Test of English as a Foreign Language) in which the closest related word from a set of four words has to be chosen. Görnerup and Karlgren [5] found that best results are obtained using L1-norm or Jensen-Shannon divergence (JSD).

DIR 2013, April 26, 2013, Delft, The Netherlands.

Curran and Moens [2] obtain best results using a combination of the Jaccard coefficient and the T-test while Van der Plas and Bouma [10] report best results using a combination of the Dice coefficient and pointwise mutual information. Both Curran and Moens and Van der Plas and Bouma use a number of different relations and need a similarity measure that is able to assign different weights to the relations. This makes their results less relevant for the present paper. The differences between the latter two studies show how strongly the results depend on the exact settings of the experiment. Many authors, however, use cosine similarity as a generally well established similarity measure for vectors in high dimensional word spaces.

Weeds et al. [13] do not compare similarity measures to hand crafted data sets but studied characteristic properties of various measures. They find that, in a task where words related to a given word have to be found, some similarity measures tend to find words with a similar frequency as the target word, while others favor highly frequent words. The Jensen-Shannon divergence (JSD) is one of the measures that tends to favor more general terms. In the following we will investigate this in more detail. We show that a better similarity measure can be defined on the base of the JSD, when we use our knowledge about the dependency of the JSD on the frequency of the words. Finally, we show that this new similarity measure outperforms the original JSD and the cosine similarity in a task in which a large number of word pairs have to be classified as synonyms or non-synonyms.

2. INFLUENCE OF WORD FREQUENCY

As already mentioned above, Weeds et al. [13] observed that, in tasks in which related words have to be found, some measures prefer words with a frequency similar to that of the target word while others prefer highly frequent words, regardless of the frequency of the target word. The JSD belongs to the latter category. In Wartena et al. [12] we also made this observation. There we compared context vectors of words with the word distribution of a document with the goal of finding keywords for the document. In order to compensate for the strong bias to highly frequent words, we introduced specificity as an explicit second condition for finding keywords. As long as we try to find synonyms for a given word, i.e. if we compare pairs of words in which one component is fixed, like in the TOEFL tests, the problem usually is tolerable. Moreover, the problem is not that apparent if the range of the lowest and highest frequencies is not too large, e.g. when only words with certain minimal frequency are considered and the size of the corpus gives a low upper bound on the frequency. Length effects are completely avoided if for every word the same amount of contexts is sampled, as e.g. is done by Giesbrecht [4]. As we will see below, JSD becomes completely useless if we compare arbitrary word pairs and do not pose any lower or upper bound on the frequency of the words.

The JSD between two probability distributions is defined as the average of the relative entropy of each of the distributions to their average distribution. It is interesting to note, that the JSD can be written as

$$\begin{aligned}\mathrm{JSD}(p,q) &= \tfrac{1}{2}D(p||\tfrac{1}{2}p+\tfrac{1}{2}q)+\tfrac{1}{2}D(q||\tfrac{1}{2}p+\tfrac{1}{2}q)\\ &= \log 2+\frac{1}{2}\sum_{t:p(t)\neq 0\,\wedge\,q(t)\neq 0}\Big(p(t)\log(\frac{p(t)}{p(t)+q(t)})\\ &\qquad +q(t)\log(\frac{q(t)}{p(t)+q(t)})\Big). \qquad (1)\end{aligned}$$

This formulation of the JSD explicitly shows that the value only depends on the words that have a non-zero value in both context vectors. If there is no common word the JSD is maximal. Now suppose that all words are independent. If the context vectors are based on a few instances of a word, the probability that a context word co-occurs with both words is rather low. To be a bit more precise, if we have context vectors v_1 and v_2 that are distributions over d elements, with n_1 and n_2 non zero elements, than the probability that a word is not zero in both distributions is, as a first approximation, $\frac{n_1}{d}\cdot\frac{n_2}{d}$. Even if the words are not independent, we might expect a similar behavior: the probability that a word has a non zero value in two context vectors increases with the number contexts on which the vectors are based.

If we try to predict the JSD of the context vectors of two words, we could base this prediction on the frequency of the words. However, it turns out that this is a very complicated dependency. Alternatively, we could base the prediction on the entropy of the context vector (if we interpret the vector as a probability distribution, as we have to do to compute the JSD): if the entropy of both vectors is maximal, they have to be identical and the JSD will be 0. If the entropy of both vectors is minimal, the JSD of the two vector is most likely to be maximal. Since, in case of independence of all words, the context vectors will not converge to the equal distribution but to the background distribution, i.e. the word distribution of the whole corpus, it is more natural to use the relative entropy to the background distribution. Preliminary experiments have shown that this works, but that JSD of two context vectors can be better predicted by the number of non-zero values in the vectors.

Figure 1 shows the relation between the JSD of two context vectors and the product of the number of non zero values in both distributions. The divergences in this figure are computed for distributions over 20 000 random indices computed on the 2,2 billion words ukWaC Corpus for 9916 word pairs. We found the same dependency for the L1 norm. In contrast, for the cosine similarity we could not find any dependency between the number of instances of the words or the number of non zero values in the context distributions.

3. EXPERIMENTAL RESULTS

To test our hypothesis that the divergence of two context vectors depends on the number of instances on which these vectors are based, we computed divergences for almost 10 000 word pairs on a very large corpus. Furthermore, we show how the knowledge about this dependency can be used to find a better measure to capture the semantic similarity between two words.

3.1 Data

As a corpus to compute the context distribution we use the POS tagged and lemmatized version of the ukWaC Corpus of approximately 2,2 billion words [1]. As the context of a

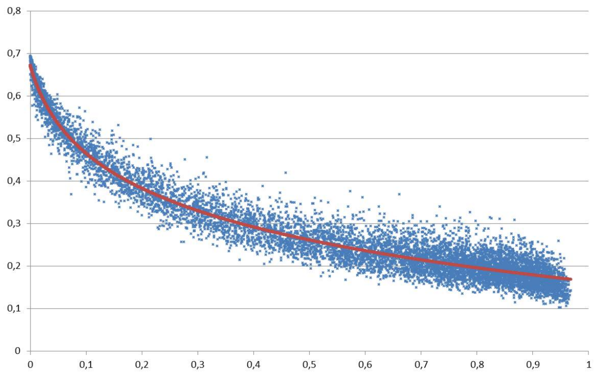


Figure 1: Divergence vs. product of relative number of non-zero values for pairs of context vectors and a function modeling the dependency.

word we consider all lemmata of open class words (i.e. nouns, adjectives, verbs, etc.) in the same sentence. We define a sentence simply as a set of words. A corpus then is a set of sentences. Let C be a corpus and w a word, then we define $C_w = \{S \in C \mid w \in S\}$. Given a corpus C, the context vector p_w of a word w can be defined as

$$p_w = \frac{1}{|C_w|} \sum_{S \in C_w} \frac{1}{|S|} \sum_{v \in S} r_v \tag{2}$$

where r_v is the random index vector of the word v. The random index vector is defined as a probability distribution over d elements, such that for some small set of random numbers $R = \{r \in \mathbb{N} \mid r < d\}$ there are n elements $r_v(i) = \frac{1}{|R|}$ if $i \in R$ and $r_v(i) = 0$ otherwise. In the following we will use distributions with $d = 20\,000$ and $|R| = 8$ unless stated else. Note, that we will always use probability distributions, but stick to the usual terminology of (context) vectors.

For the evaluation of the similarity measures we selected pairs of words from Wordnet [8]. We started with a list of pairs (w_1, w_2) such that (1) w_1 and w_2 are single words, (2) w_1 occurs at least two times in the British National Corpus and (3) w_1 and w_2 share at least one sense. This resulted in a list of 24 576 word pairs. From this list we selected all pairs for which the Jaccard coefficient of the sets of senses of the words is at least 0.7. After filtering out all pairs containing a word that was not found in the ukWaC corpus a list of 849 pairs remained. These word pairs are considered as synonyms in the following. Next from the list of 24 576 word pairs the second components were reordered randomly. The resulting list of new word pairs was filtered such that the two words of each pair both occur in the ukWaC corpus and have no common sense. This resulted in a list of 8967 word pairs.[1]

As a consequence of the requirement of the overlap of Wordnet senses, most words in the synonym list have very few senses and are very infrequent words. Thus the average frequency in ukWaC of the synonyms is much lower than that of the words of the non-synonym list. The most frequent word (*use*) was found 4.57 million times in the ukWaC corpus; 117 words were only found once (e.g. *somersaulting*, *sakartvelo*).

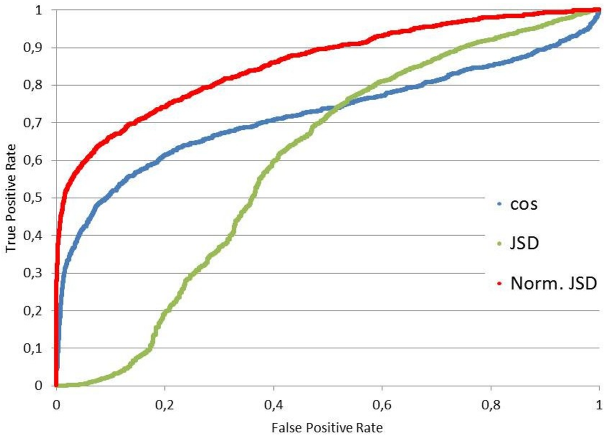


Figure 2: ROC Curves for ranking of word pairs (849 synonym pairs, 8967 non synonym pairs) using different similarity measures.

3.2 Predicting JSD of context vectors

Figure 1 shows that there is a clear dependency between the JSD of a word pair and the product of the relative numbers of non zero values in the context distributions. This dependency can be captured by the following equation:

$$\mathrm{JSD}^{\exp}(p_1, p_2) = a \log\left(1 + \frac{b}{n}\right) + c \tag{3}$$

with $n = \frac{n_1}{d} \cdot \frac{n_2}{d}$ where n_1 and n_2 are the number of non zero values of n_1 and n_2, respectively. Optimal values for a, b and c were found by maximizing the coefficient of determination, R^2, on all non-synonym word pairs. We left out the synonyms, since we try to model the similarity that is caused just by the probability of random words to occur in these context with an increasing number of observations. With $a = -0.34$, $b = 0.032$ and $c = 0.67$ a R^2 score of 0.95 is reached (0.93 for the same constants when synonyms are included). The curve corresponding to these values is displayed in red in Figure 1. Since usually context vectors with much less dimensions are used, we repeated the experiment with context distributions over 1 000 random indices and obtained a R^2 value of $0,92$ ($a = -1.65$, $b = 0.99$ and $c = 0.61$).

3.3 Ranking word pairs

Most of the variance in the JSD of two context distributions can be explained by (3). Now we expect that the remaining variance reflects the degree to which the words have a similar function or even meaning. To test this we define the *(frequency) normalized JSD* as

$$\mathrm{JSD}^{\mathrm{norm}}(p_1, p_2) = \mathrm{JSD}(p_1, p_2) - \mathrm{JSD}^{\exp}(p_1, p_2) \tag{4}$$

Ideally, all word pairs of synonyms will be ranked higher than the non-synonym pairs. We use the area under the ROC curve (AUC) to evaluate the ranking. We compare the ranking according to the normalized JSD with the rankings from the JSD, the cosine similarity and the L1 norm that is used sometimes in combination with random indexing. The L1 norm between two vectors v_1 and v_2 of dimensionality d is defined as $\sum_{0 \le i < d} |v_1(i) - v_2(i)|$. The ROC curves are given in Figure 2 when using context vectors with 20 000 dimensions. The AUC-values are summarized in Table 1, both for

[1] The lists of word pairs are available at `http://nbn-resolving.de/urn:nbn:de:bsz:960-opus-4077`.

Table 1: AUC of classifying wordpairs as synonyms using different numbers of dimensions and different similarity measures

Number of dimensions	Similarity Measure	AUC
1000	Cosine	0,53
1000	JSD	0,41
1000	JSD^{norm}	0,52
20000	Cosine	0,72
20000	JSD	0,41
20000	L1	0,42
20000	JSD^{norm}	**0,86**

the experiment using context distributions over 20 000 and 1 000 random indices.

We see that the JSD gives a ranking worse than a random ranking. The remarkable observation is the large difference between the AUC values, since we are comparing exactly the same context distributions, and thus use exactly the same information. A further observation is the strange behavior of the cosine similarity. For pairs of words for which less than a dozen instances were found, the cosine similarity seems to give almost random results. Thus some positive pairs are ranked very low, explaining the rise of the ROC curve at the right end. The results of the L1 norm are almost the same as those of the JSD, which is not surprising as we also found a linear correspondence between JSD and the L1 norm.

Finally, it should be noted that we did not try to find the best possible ranking. If we would include frequency information (two very frequent words are unlikely to be synonyms) or Levenshtein distance (there are many spelling variants included in the list of synonyms) we could easily obtain a better ranking. The goal of the experiment, however, was evaluation of distance measures for random indexing. The classification is only a means to assess the quality of the distance measure. In [11] we also investigate the possibility to combine various distance measures and other features to get an optimal ranking.

4. DISCUSSION AND CONCLUSIONS

We have clearly found a very strong dependency between the number of non-zero values in random context vectors and the JSD between the vectors. When we use data with an extremely large range in frequencies this leads to JSD values that are useless for ranking word pairs according to their similarity. Note that we included words with frequencies ranging from 1 to 4,57 Million. We used the known dependency between the number of non zero values in the distributions and the JSD to define a new similarity measure, the frequency normalized JSD. This measure clearly outperforms the cosine similarity in the ranking experiment.

Though this result is convincing, we are lacking a theoretical base from which a formula like (3) can be derived. Also, it would be preferable if the constants could be estimated directly from the size of the corpus, the number of dimensions, etc. Now, only one from three constants can easily be explained, namely as the maximum JSD. Alternatively, also smoothing of the context distributions might be a solution to make JSD more useful. The smoothing should then account for the similarities that stem from random words appearing in both contexts. In general, the results show that the choice for the right similarity measure to be used for distributional similarity is not a solved question and more research in this area is needed.

5. REFERENCES

[1] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The wacky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation 43 (3): 209-226*, 43(3):209–226, 2009.

[2] J. R. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *Unsupervised Lexical Acquisition: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLAX).*, pages 59–66. Association of Computational Linguistics, 2002.

[3] F. de Saussure. *Cours de linguistique générale.* V.C. Bally and A. Sechehaye (eds.), Paris/Lausanne, 1916. English translation: Course in General Linguistics. London: Peter Owen, 1960.

[4] E. Giesbrecht. Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 23–28, Los Angeles, California, 2010. ACL.

[5] O. Görnerup and J. Karlgren. Cross-lingual comparison between distributionally determined word similarity networks. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, pages 48–54. ACL, 2010.

[6] G. Grefenstette. Use of syntactic context to produce term association lists for text retrieval. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 89–97. ACM, 1992.

[7] J. Karlgren and M. Sahlgren. From words to understanding. In *Foundations of Real-World Intelligence*, pages 294–308. CSLI Publications, Stanford, Californa, 2001.

[8] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[9] M. Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5, 2005.

[10] L. Van Der Plas and G. Bouma. Syntactic contexts for finding semantically related words. In *Proceedings of Computational Linguistics in the Netherlands*, 2004.

[11] C. Wartena. HsH: Estimating semantic similarity of words and short phrases with frequency normalized distance measures. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, 2013. to appear.

[12] C. Wartena, R. Brussee, and W. Slakhorst. Keyword extraction using word co-occurrence. In *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, pages 54–58. IEEE, 2010.

[13] J. Weeds, D. J. Weir, and D. McCarthy. Characterising measures of lexical distributional similarity. In *COLING 2004, Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

Exploring Real-Time Temporal Query Auto-Completion

Stewart Whiting, James McMinn and Joemon M. Jose
School of Computing Science
University of Glasgow
Scotland, UK.
{stewh,mcminn,jj}@dcs.gla.ac.uk

ABSTRACT

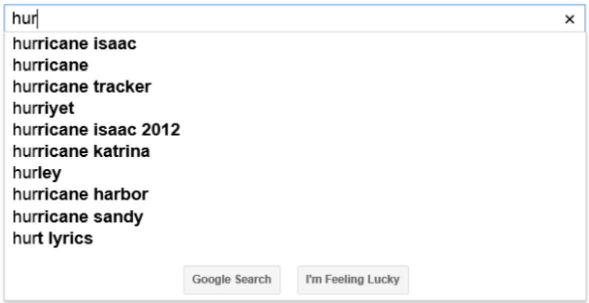
Query auto-completion (QAC) is a common interactive feature for assisting users during query formulation. Following each query input keystroke, QAC suggests queries prefixed by the input characters; allowing the user to avoid further cognitive and physical effort if any are acceptable. To rank suggestions, QAC approaches typically aggregate past query popularity to determine the likelihood of a query being used again. Hence, QAC is usually very effective for consistently popular queries. However, as the web becomes increasingly real-time, more people are turning to search engines to find out about unpredictable emerging and ongoing events and phenomena. QAC approaches reliant on aggregating long-term historic query-logs are not sensitive to very recent real-time events, because newly popular queries will be outweighed by long-term popular queries, especially for less-specific prefix lengths (e.g. 2 or 3 characters). We explore limiting the aggregation period of past query-log evidence to increase the temporal sensitivity of QAC. We vary the query-log aggregation period between 2 and 14 days, for prefix lengths of 2 to 5 characters. Experimentation simulates a real-time environment using openly available MSN and AOL query-log datasets. Analysis indicates a linear relationship between prefix length and QAC performance when using different query-log aggregation periods. In particular, we find QAC for shorter prefix lengths is optimal when a shorter query-log aggregation period is used, and vice-versa, longer prefix lengths benefit from a longer query-log aggregation period.

1. INTRODUCTION

For users, cognitively formulating and physically typing queries is a time-consuming and error prone process. As such, query auto-completion (QAC) [3, 10] has been widely adopted by major web search engines to reduce the effort necessary to submit a query.

As a user types their query into the search box, QAC attempts to predict the completed query the user may have in mind. Following each query input keystroke, QAC suggests possible queries (which we refer to as *completion suggestions*) beginning with the already input character sequence (i.e. *prefix*). The goal for effective QAC is to present the user's intended query after the least possible keystrokes, and at the highest rank in the list of completion suggestions.

Conventional QAC approaches rank completion suggestions by aggregating their popularity in past query-logs. Further work has incorporated personal contextual features for short prefixes [3] and time-series modelling of temporal trends [10]. However, with enough past evidence, completion suggestions ranked solely by their popularity in past query-logs provides reasonably effective QAC [3, 10].



hur

hurricane isaac
hurricane
hurricane tracker
hurriyet
hurricane isaac 2012
hurricane katrina
hurley
hurricane harbor
hurricane sandy
hurt lyrics

Figure 1: Google auto-completion suggestions for the query prefix '***hur***'. Screenshot taken November 8th 2012, 10 days after Hurricane Sandy made landfall on the East Coast of the USA. Browser cookies were cleared to avoid individual personalization effects.

Figure 1 illustrates the ten completion suggestions offered by Google for the three character query prefix '***hur***' on November 8th, 2012. The list of query suggestions indicates the historically most likely queries to be submitted with the given prefix, possibly in the context of some undisclosed ranking features such as geo-location or the user's past queries. Despite the recency and prominence of Hurricane Sandy, the query ranks very low in the completion suggestions, while '*hurricane isaac*' ranks first, regardless that it occurred many months previously. Aside from this issue, QAC for short and unspecific prefixes (i.e. 1 or 2 common characters) is often unsuccessful as there are usually a huge number of possible completion suggestions [3]. Consequently, it is typically long-term '*head*' queries that are suggested as completions for such short prefixes.

With the web increasingly becoming a platform for real-time news and media, time plays a central role in information interaction. A substantial volume of daily top queries are the result of users turning to search engines for up-to-date information about very recent or ongoing events [2, 6]. 15% of the daily queries to an industrial web search engine have never been seen before[1]; a substantial proportion of these queries may be attributed to real-time events and phenomena, rather than the long-tail of very uncommon queries. Similarly, previously unpopular queries may suddenly become extremely popular because of recent developments. It is therefore important that QAC supports queries which become highly popular only during brief periods of time, which we refer to as *real-time temporal queries*.

DIR 2013, April 26, 2013, Delft, The Netherlands.

[1] http://www.google.com/competition/howgooglesearchworks.html

Although the common approach to QAC is to rank completion suggestions by their popularity in the historic query-log (i.e. *past query-log evidence*), there has been very little study on the aggregation period necessary to achieve optimal QAC effectiveness, and whether this varies for each prefix length [10]. Thus, the objective of this paper is to investigate this uncertainty by conducting experiments based upon the AOL and MSN query-log datasets. For each prefix length we use an *N* day sliding window of past query-log evidence to rank completion suggestions, hence making QAC more sensitive to real-time querying distribution changes. We present results and observe overall QAC effectiveness for different periods of *N* days, at prefix lengths of 2 to 5 characters.

2. MOTIVATION

As time undoubtedly plays a central role in user search behaviour [2], it is important for QAC to suggest completions that become highly popular over very short periods (i.e. real-time temporal queries), while also supporting always popular '*head*' queries.

Relying on a long period of past query-log evidence will ensure QAC is robust for continually popular queries, however, it will also have the effect of smoothing over short-term popular queries. For example, imagine a scenario where query q_1 is consistently popular, appearing 1000 times each day in the query-log. Aggregating query popularity over a past 30 day period would mean that query q_2 which is popular only today would need to be appear 30,000 times before it outweighed the long-term popular query in a probabilistic QAC approach. At the same time, reducing the aggregation period may mean the long-term popular query q_1 is not adequately represented, allowing short-term noise to reduce its ranking.

Ultimately, developing an effective QAC system that can respond to real-time temporal trends is a trade-off between robustness and sensitivity. In this paper we aim to study this trade-off in terms of how much past query-log evidence is optimal for aggregating query popularity, and how this changes for each prefix length. Moreover, as there has been little experimentation on open datasets, this work establishes baseline QAC performance for further studies.

The effectiveness of using a shorter query-log evidence aggregation period has been noted previously, particularly for real-time temporal queries [10]. While time-series modelling for query trends is able to improve QAC for recurring predictable temporal trends, for short-term real-time temporal queries it often proved problematic due to lag and over-fitting [10]. Time-series models were not able to model the increasing trend quickly enough, and likewise, continued to predict increased popularity for some time after the brief period of actual popularity.

2.1 Temporal Query-log Analysis

We quantify the extent to which the query-logs are composed of real-time temporal querying, in order to determine the degree to which QAC must support this behaviour. We define real-time temporal queries as those which appear as a 'spike' - with the vast majority of their occurrence within a short period, e.g. hours to days. Similarly, the queries are unlikely to have been recently popular, or even seen previously.

We analyse the temporal trends contained in two publicly available[2] datasets: the AOL [7] and MSN [1] query-log datasets. Extensive temporal analysis of longer-term and larger proprietary query-log data has been performed previously by others [6, 4, 2].

The AOL query-log contains 36.3M user interactions over a 3 month period from the 1st March 2006 to the 31st May 2006. The MSN query-log contains almost 14.9M user interactions over a 1 month period from the 1st May 2006 to the 31st May 2006.

Query-log entries necessary for identifying result clicks were removed. By extracting all the unique query and timestamp combinations, we obtained only queries directly typed by users. Navigational queries containing the URL substrings: .com, .net, .org, http, .edu or www were removed. We were left with 21.8M and 12.2M queries for AOL and MSN, respectively. Preliminary analysis discovered a sizeable number of short bursts of what we suspect is bot spamming activity in the AOL query-logs. Generic queries such as '*personalfinance*', '*aolcelebrity*', '*computercheckup*', appear in high volume with very uniform spacing (e.g. every 30 or 60 seconds). We manually observed and removed around 10,000 instances of these queries from our analysis.

	Volume of Queries	
Window Size (Days)	**AOL**	**MSN**
1	9.2%	3.5%
3	10.1%	4.5%
5	10.4%	5.1%

Table 1: The volume of queries in each query-log which were used ≥ 4 times, and for which 80% of their overall occurrence is within a window of *N* days.

In Table 1 we present the volume of queries (i.e. % of the total queries submitted in the query-log) which occur four or more times, and have at least 80% of their use concentrated within a period of *N* days. In AOL, the most popular 1 to 5 day highly temporal queries include: '*amelia earhart pictures*', '*karl der grosse*', '*the simpsons live action*' and '*leisure suit larry*'. Likewise, in MSN among the most popular are: '*stephen colbert*', '*poison milk*', '*ohio bear attack*' and '*kimberley dozier*'. Investigation shows that many of these queries describe, or are strongly related to significant events.

These results suggest a reasonable volume of real-time temporal queries in both query-logs, at least in the relatively short periods we are able to study. We suspect that the percentage of real-time temporal queries will have substantially increased in more recent query-logs, given the increase in real-time media available on the internet.

3. RELATED WORK

The majority of research has concentrated on the inherent engineering complexity of providing efficient and scalable QAC, which is resilient to typing errors. There have been relatively few studies on improving QAC effectiveness in search engines; likely due to the fact that there are few suitable query-logs available outside industrial search engine companies for experimentation.

Exploiting the user's personal context, and past query sessions has led to considerable QAC improvement, especially for shorter prefixes [3, 8]. Shokouhi and Radinsky [9, 10] used time-series modelling of past temporal query patterns to improve QAC effectiveness. Popular queries recurring during specific temporal intervals, such as day/night, day of week, month, etc. were modelled so that current query popularity could be predicted based on prior evidence only. Shokouhi and Radinsky [10] propose the short time window technique we experiment with in this paper as a baseline (which they refer to as p_1, etc.). They note its relative effectiveness, particularly for correctly predicting short-term highly temporal and unpredictable queries for which time-series modelling is problematic. However, no detailed analysis on the performance impact of aggregation period for each prefix length is performed.

[2] MSN available on request. We justify our use of AOL as we study the data without identifying individuals.

4. AUTO-COMPLETION APPROACH

The common "standard" approach to QAC is Maximum Likelihood Estimation (MLE), based on past query popularity (i.e. '*most popular completion*') [3]. MLE for a prefix ρ_n (of n characters), with each query q in all past queries Q prefixed by ρ_n, is formalised as follows:

$$MLE(\rho_n) = \underset{q \in Q}{\arg\max} P(q) \qquad (1)$$

$P(q)$ is the probability of the query appearing in the past query-log. We refer to this method, aggregating all query-log evidence prior to the current time q_t as our baseline **MLE-ALL**.

4.1 Limiting Past Query-log Evidence

We propose using only the last N days of query-log evidence (e.g., $N = 2$, 4,7 or 14 days) for computing $P(q)$ at q_t (i.e., a *sliding window* of past evidence). We refer to this approach as **MLE-W*N***.

The intuition underlying this approach is that a more recent and limited period of queries may more accurately reflect the current query distribution. Similarly, although consistently popular queries will still be adequately reflected in the distribution, their total frequency will no longer be great enough to outweigh the frequency of popular queries that only spike in shorter periods.

5. METHODOLOGY

The objective of our experiment is to study the trade-off between sensitivity and robustness of QAC, for different prefix lengths. As such, we explore various query-log aggregation periods for each prefix length, and measure the effect on overall QAC performance.

Our experimental methodology simulates a real-time user search scenario; such that the user types a prefix, and receives completion suggestions based only on evidence prior to the time of their query. QAC effectiveness is measured by the presence, and rank of a ground-truth match for each set of suggestions.

A time-ordered query-log provides a stream of ground-truth user queries. We assume that each query present in the query-log is the result of a user having manually typed it into the search box. As such, for each prefix of length n of the query, QAC provides completion suggestions. Each suggestion is matched with the ground-truth of the user's actual query (we discuss matching in the following section).

Evaluation Metric. Similar to past QAC work [3, 10], we rely on Mean Reciprocal Rank (MRR) to observe the effectiveness of each QAC approach. Reciprocal Rank (RR) has typically been used for evaluation in IR situations where there is a single relevant document. For a set of completion suggestions S, RR is computed as:

$$RR(S, q_{intended}) = \frac{1}{S, Rank(q_{intended})} \qquad (2)$$

If no match for $q_{intended}$ is present, then a RR of 0 is assigned (avoiding divide by zero errors). MRR is then computed as the arithmetic average of RR for all queries.

MRR reflects the user interaction model of QAC; a higher-ranked completion suggestion is more beneficial, but the difference in ordering of lower-ranked completion candidates is less significant. That is to say, there is less noticeable difference between a correct completion suggestion ranked at either the 3rd or 4th position, compared to the 1st or 2nd position. We consider a literal lower-case string match between completion suggestion and ground-truth as a successful match.

6. EXPERIMENT

We conduct experimentation using the AOL and MSN query-log datasets. By experimenting with millions of queries contained in each query-log, we achieve a representative indication of how each approach would perform in a real-world setting.

Using two different query-logs validates the approach across two query samples of varying characteristics, and different user populations of the two industrial search engines. The exact sampling and construction of each query-log is unknown. MSN has been filtered for privacy (e.g. clearing known number patterns, such as phone numbers), appears to contain fewer adult queries, and is more in-depth as it is only for a one month period. In contrast, the AOL query-log contains more queries, but has greater breadth as it covers a three month period. However, as noted in [2], the sampling of AOL may not be truly representative of normal querying distributions because of re-finding behaviour.

6.1 Experiment Settings

We report results in Section 7 for two QAC settings: MLE-ALL using all query-log evidence prior to q_t (we treat this as the baseline), and MLE-W*N*, with 2,4,7 and 14 days of past query-log (characterising short and medium-term event/evidence periods). With only sampled query-log datasets, reducing the evidence period further leads to relatively sparse querying data. To emulate a real user interface scenario, we assume the user would see 4 highest-ranking completion suggestions for each prefix they input.

We run each approach for all query prefixes of 2 to 5 characters. Experiments for each prefix length were run independently, hence a successful completion suggestion at a prefix of 2 characters had no effect on the later evaluation for 3 characters.

Learning Period. We report the MRR of MLE QAC computed over the period of the query-log, minus the first N days which we treat as the learning period. Doing this makes the MRR obtained from MLE-ALL and MLE-W*N* directly comparable as both are computed over exactly the same set of queries. In any case, QAC performance during this early period will be extremely low as there is very little query popularity evidence (i.e. the 'cold-start problem'), and wouldn't reflect a real-world scenario where a QAC system would almost always be trained on past evidence.

7. RESULTS

Table 2 presents the overall MRR observed for MLE QAC experiments on the AOL and MSN and AOL query-logs; using the past 2, 4, 7, 14 days as well all past query-log evidence, for prefix lengths of 2 to 5 characters. The MLE-ALL MRR reported beside each MLE-W*N* corresponds to the baseline using all queries prior to q_t, but with the first N days of queries excluded for comparison.

The aggregated statistical power of 21.8M and 12.2M RR measures (i.e. each query) provided by the AOL and MSN experiments, respectively, means that the results we report are statistically significant according to standard *t*-tests [5]. Therefore, our analysis concentrates on the effect size of each window period over the baseline - that is, change in MRR over the corresponding MLE-ALL.

Firstly, for all runs and both query-logs it is clear that QAC is considerably more effective with a longer (i.e. more specific) prefix. This is expected, given that each extra character in the prefix reduces the space of possible completion suggestions, thus increasing the chance of a completion suggestion match [3].

QAC is almost always more effective for MSN than for AOL, especially for prefix lengths of 4 or less characters. Using a sliding window of evidence has a significant effect on overall QAC performance in almost all cases.

For AOL there is a sliding window of evidence which can im-

ρ	MLE-ALL	MLE-W2	MLE-ALL	MLE-W4	MLE-ALL	MLE-W7	MLE-ALL	MLE-W14
AOL								
2	0.090	**0.091 (1.11%)**	0.090	0.091 (1.11%)	0.090	0.091 (1.11%)	0.090	0.091 (1.11%)
3	0.143	**0.147 (2.80%)**	0.143	0.146 (2.10%)	0.143	0.145 (1.40%)	0.143	0.145 (1.40%)
4	0.185	0.189 (2.16%)	0.184	**0.189 (2.72%)**	0.184	0.188 (2.17%)	0.184	0.187 (1.63%)
5	0.217	0.215 (-0.92%)	0.216	0.217 (0.46%)	0.217	0.218 (0.46%)	0.217	**0.219 (0.92%)**
MSN								
2	0.112	**0.117 (4.46%)**	0.111	0.115 (3.60%)	0.111	0.113 (1.80%)	0.110	0.111 (0.91%)
3	0.164	0.163 (-0.61%)	0.164	**0.165 (0.61%)**	0.164	**0.165 (0.61%)**	0.164	0.164 (0.00%)
4	0.197	0.188 (-4.57%)	0.197	0.193 (-2.03%)	0.197	0.196 (-0.51%)	0.197	**0.197 (0.00%)**
5	0.215	0.197 (-8.37%)	0.216	0.205 (-5.09%)	0.216	0.211 (-2.31%)	0.218	**0.216 (-0.92%)**

Table 2: MRR observed for QAC when using all prior query-log evidence, and the past 2, 4, 7 or 14 days of query-log evidence. Prefix (ρ) lengths of 2 to 5 characters are reported for the AOL and MSN query-logs. The best performing sliding window setting is highlighted for each prefix length (although in some cases this is still outperformed by the baseline, at least for the reported window periods).

prove QAC performance over MLE-ALL for all prefix lengths, albeit relatively marginally for 5 characters. Using a shorter 2 day window of evidence improves QAC performance by up to nearly 3% for shorter prefixes of 2 or 3 characters. For a prefix of 4 characters, using a little more evidence, e.g. 4 days is optimal. Similarly, the best performance for a 5 character prefix is obtained when using 14 days of evidence.

For MSN, shorter prefixes (e.g. 2 or 3 characters) can outperform the baseline when using a window of evidence. Specifically, we see the best performance improvement of nearly 4.5% when using 2 days of evidence for a 2 character prefix. However, using between 2 and 14 days of query-log evidence always impairs QAC performance compared to the baseline for 4 or 5 character prefixes. Notably, the detrimental effect on performance is reduced as the sliding window of evidence is increased.

8. DISCUSSION AND CONCLUSION

The baseline QAC performance, and the following sliding window QAC improvement characteristics for each query-log are considerably different between the query-logs. AOL QAC is marginally but consistently improved in almost all cases, whereas MSN QAC is only improved for shorter prefixes, albeit much more so than for AOL. This suggests that the two query-logs have different temporal characteristics. In part, this may be caused by a couple of factors. Firstly, although AOL has more queries, it is spread more sparsely over a three month period, in contrast, MSN queries are concentrated in a 1 month period. Additionally, AOL has a day of missing data [2] which will harm QAC effectiveness following the affected period. Secondly, there may be underlying demographic differences between users of the two search engines that lead to changes in query distributions.

Although the performance improvement characteristics for each prefix and sliding window are different for each query-log, there is a clear overall linear relationship emerging in the results between prefix length and optimal sliding window period. As such, QAC for shorter prefixes performs optimally with a shorter sliding window of evidence, and conversely, QAC for longer prefixes performs best with a longer sliding window of evidence.

This relationship probably arises from the uncertainty posed by short and non-specific prefix lengths [3], where the space of possible completion suggestions is large. In these cases, using a shorter-period of evidence will still reflect long-term popular queries, but also be sensitive to temporal variation. Longer prefixes are more specific and thus narrow possible completion suggestions considerably. In these cases, real-time temporal factors are probably less likely to play a significant role in the already reduced set of possible completion suggestions. Moreover, for less common prefixes (and therefore rarer queries), relying on a longer query-log period is more likely to include the evidence necessary to rank them effectively as they were less likely to be used recently.

Conclusion. In this paper we examined the trade-off between QAC robustness and real-time temporal sensitivity. We found that QAC effectiveness can be improved by up to nearly 5%, simply by selecting the optimal time period of query popularity aggregation for each prefix length. The period necessary to achieve optimal QAC effectiveness varies by prefix length; shorter prefixes (e.g. 2-3 characters) perform best with only short-term evidence (e.g. 2-7 days), whereas longer prefixes (e.g. 4-5 characters) require more long-term evidence (e.g. 7-14 days, or more). Results also indicate the need to train per query-log, in order to capture intrinsic temporal and demographic characteristics. Care must also be taken with the sampling of queries used for training.

Further work will experiment with larger, more recent query-logs and perform cross-validation to verify the preliminary findings we present in this paper. Moreover, we will investigate alternative modelling techniques to improve QAC effectiveness for real-time temporal queries, which are problematic for time-series modelling as they spike so briefly in time.

Acknowledgments. This work was partially supported by the EU LiMoSINe project (288024).

9. REFERENCES

[1] *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, New York, NY, USA, 2009. ACM.

[2] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble. Why we search: visualizing and predicting user behavior. WWW '07, pages 161–170, New York, NY, USA, 2007. ACM.

[3] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *WWW '11*, pages 107–116, 2011.

[4] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. Temporal analysis of a very large topically categorized web query log. *J. Am. Soc. Inf. Sci. Technol.*, 58(2):166–178, Jan. 2007.

[5] P. Ellis. *The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results*. Cambridge University Press, 2010.

[6] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding temporal query dynamics. In *ACM WSDM '11*, pages 167–176, New York, NY, USA, 2011. ACM.

[7] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. InfoScale '06, New York, NY, USA, 2006. ACM.

[8] C. Sengstock and M. Gertz. Conquer: a system for efficient context-aware query suggestions. WWW '11, pages 265–268, New York, NY, USA, 2011. ACM.

[9] M. Shokouhi. Detecting seasonal queries by time-series analysis. In *SIGIR '11*, pages 1171–1172, 2011.

[10] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *SIGIR '12*, pages 601–610, 2012.

Exploiting Semantic Relatedness Measures for Multi-label Classifier Evaluation

Christophe Deloo*
Delft University of Technology
Delft, The Netherlands
c.p.p.deloo@gmail.com

Claudia Hauff
Delft University of Technology
Delft, The Netherlands
c.hauff@tudelft.nl

ABSTRACT

In the multi-label classification setting, documents can be labelled with a number of concepts (instead of just one). Evaluating the performance of classifiers in this scenario is often as simple as measuring the percentage of correctly assigned concepts. Classifiers that do not retrieve a single concept existing in the ground truth annotation are all considered equally poor. However, some classifiers might perform better than others, in particular those, that assign concepts which are semantically similar to the ground truth annotation. Thus, exploiting the semantic relatedness between the classifier-assigned and the ground truth concepts leads to a more refined evaluation. A number of well-known algorithms compute the semantic relatedness between concepts with the aid of general-world knowledge bases such as WordNet[1]. When the concepts are domain specific, however, such approaches cannot be employed out-of-the-box. Here, we present a study, inspired by a real-world problem, where we first investigate the performance of well-known semantic relatedness measures on a domain-dependent thesaurus. We then employ the best performing measure to evaluate multi-label classifiers. We show that (i) measures which perform well on WordNet do not reach a comparable performance on our thesaurus and that (ii) an evaluation based on semantic relatedness yields results which are more in line with human ratings than the traditional F-measure.

Categories and Subject Descriptors: H.3.3 Information Storage and Retrieval: Information Search and Retrieval
Keywords: semantic relatedness, classifier evaluation

1. INTRODUCTION

In this paper, we present a two-part study, that is inspired by the following real-world problem: Dutch Parliamentary papers[2] are to be annotated with concepts from an existing thesaurus[3] (the *Parliament thesaurus*). A multi-label classifier framework exists and each document can be automatically annotated with a number of concepts. Currently, the evaluation of the classifier is conducted as follows: the automatically produced annotations are compared to the ground-truth (i.e. the concepts assigned by domain experts) and the binary measures of precision and recall are computed. This means, that a document labelled with concepts which do not occur in the ground truth receives a precision/recall of zero, even though the assigned concepts may be semantically very similar to the ground truth concepts. As an example, consider Figure 1: the ground truth of the document consists of three concepts {*biofuel, environment, renewable energy*} and the classifier annotates the document with the concepts {*energy source, solar energy*}. Binary precision/recall measures evaluate the classifier's performance as zero, though it is evident, that the classifier does indeed capture the content of the document - at least partially.

Thus, we are faced with the following research question: *Can the evaluation of a multi-label classifier be improved when taking the semantic relatedness of concepts into account?*

To this end, we present two studies (Figure 1):

1. We investigate established semantic relatedness measures on the Parliament thesaurus. Are measures that perform well on WordNet or Wikipedia also suitable for this domain-specific thesaurus?
2. Given the best performing relatedness measure, we include the semantic relatedness in the evaluation of the multi-label classifier framework and investigate if such a semantically enhanced evaluation improves over the binary precision/recall based evaluation.

We find that the best performing measures on WordNet do not necessarily perform as well on a different thesaurus, and thus, they should be (re-)evaluated when a novel thesaurus is employed. Our user study also shows that a classifier evaluation, which takes the semantic relatedness of the ground truth and the classifier assigned concepts into account yields results which are closer to those of human experts than traditional binary evaluation measures.

DIR 2013, April 26, 2013, Delft, The Netherlands.

*This research was performed while the author was an intern at GridLine.

[1] http://wordnet.princeton.edu/

[2] The documents come from the Dutch House of Representatives (de Tweede Kamer), which is the lower house of the bicameral parliament of the Netherlands.

[3] For more details see Section 3.

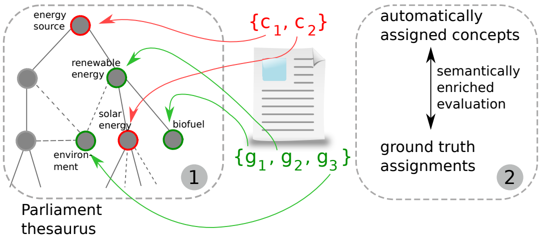


Figure 1: Overview of the two-step process: (1) we first investigate semantic relatedness measures on the Parliament thesaurus. Then, (2) given a document and its assigned ground truth concepts $\{g_1, g_2, g_3\}$ (by human annotators), we evaluate the quality of the classifier-assigned concepts $\{c_1, c_2\}$. The classifier evaluation takes the semantic relatedness between the concepts into account.

2. RELATED WORK

In this section, we first discuss semantic relatedness measures and then briefly describe previous work in multi-label classifier evaluation.

Several measures of semantic relatedness using a variety of lexical resources have been proposed in the literature. In most cases semantic relations between concepts are either inferred from large corpora of text or lexical structures such as taxonomies and thesauri. The state-of-the-art relatedness measures can be roughly organised into graph-based measures [11, 6, 19, 4, 16], corpus-based measures [17, 10] and hybrid measures [12, 5, 7, 1]. The latter combine information gathered from the corpus and the graph structure.

The majority of relatedness measures are graph-based and were originally developed for WordNet. WordNet is a large lexical database for the English language in which concepts (called synsets) are manually organised in a graph-like structure. While WordNet represents a well structured thesaurus, its coverage is limited. Thus, more recently, researchers have turned their attention to Wikipedia, a much larger knowledge base. Semantic relatedness measures originally developed for WordNet have been validated on Wikipedia. Approaches that exploit structural components that are specific to Wikipedia have been developed as well [14, 18, 3].

With respect to multi-label classifier evaluation, our work builds in particular on Nowak et al. [9]. The authors study the behavior of different semantic relatedness measures for the evaluation of an image annotation task and quantify the correctness of the classification by using a matching optimisation procedure that determines the lowest cost between the concept sets of the ground truth and of the classifier.

We note, that besides semantic relatedness measures one can also apply hierarchical evaluation measures to determine the performance of multi-label classifiers, as for instance proposed in [15]. We leave the comparison of these two different approaches for future work.

3. METHODOLOGY

Semantic Relatedness in the Parliament Thesaurus.

We first investigate the performance of known semantic relatedness measures on our domain-specific thesaurus (Figure 1 step (1)). The goal of this experiment is to identify the most promising semantic relatedness measure, i.e. the measure that correlates most closely with human judgements of relatedness. In order to evaluate the different measures, we employ an established methodology: we select a number of concept pairs from our thesaurus and ask human annotators to judge the relatedness of the concepts on a 5-point scale (where 1 means *unrelated* and 5 means *strongly related*). We consider these judgements as our ground truth and rank the concept pairs according to their semantic relatedness. Then, we also rank the concept pairs according to the scores they achieve by the different semantic relatedness measures. The agreement between the two rankings is evaluated with the rank correlation measure Kendall's Tau (τ) and the linear correlation coefficient (r).

The Parliament thesaurus contains nearly 8,000 Dutch terms oriented towards political themes such as defense, welfare, healthcare, culture and environment. As is typical for a thesaurus, the concepts are hierarchically structured and the following three types of relations exist: hierarchical (narrower/broader), synonymy and relatedness. Fifty concept pairs were manually selected by the authors, with the goal to include as many different characteristics as possible, that is, concept pairs of varying path lengths, types of relations, etc. The human ratings were obtained in an electronic survey where Dutch speaking people were asked to rate the fifty concept pairs on their relatedness. As stated earlier, in the 5-point scale, the higher the assigned rating, the stronger the perceived relatedness.

The following relatedness measures were selected for our experiments: Rada [11], Leacock & Chodorow [6], Resnik [12], Wu & Palmer [19], Jiang & Conrath [5] and Lin [7]. The measures of Rada, Leacock & Chodorow and Wu & Palmer are all graph-based measures based on path lengths. The path length is calculated by summing up the weights of the edges in the path. The weights typically depend on the type of relation. The stronger the semantic relation, the lower the weight. Two versions of both Rada's and Leacock & Chodorow's approach were implemented: one including only hierarchical and synonymous relations, and one including all three types of thesaurus relations. The weights of the relations were chosen according to their semantic strength. A weight of 1 was assigned to both hierarchical and related concept relations and a weight of 0 to synonymous concept relations. The remaining three approaches, which are based on the concept of information content, were implemented using the approach of Seco et al. [13].

Multi-label Classifier Evaluation.

Having identified the best performing measure of semantic relatedness on the Parliament thesaurus, we then turn to the evaluation of the existing multi-label classifier framework (Figure 1 step (2)). Matching the concepts from the classifier with the ground truth concepts is performed according to a simplified version (which excludes the ontology and annotator agreement) of the procedure presented in [9]. Nowak et al. define a classification evaluation measure that incorporates the notion of semantic relatedness. The algorithm calculates the degree of relatedness between the set C of classifier concepts and the set E of ground truth concepts with an optimisation procedure. This procedure pairs every label of both sets with a label of the other set in a way that maximises relatedness: each label $l_c \in C$ is matched with a label $l'_e \in E$ and each label $l_e \in E$ is matched with a label $l'_c \in C$. The relatedness values of each of those pairs are summed up and divided by the number of labels occurring

Concept pairs		Av. rating	Std. Dev.
Vaticaanstad *Vatican City*	paus *pope*	4.86	0.25
energiebedrijven *power companies*	elektriciteitsbedrijven *electricity companies*	4.72	0.43
rijbewijzen *driver licenses*	rijbevoegdheid *qualification to drive*	4.64	0.55
	…		
boedelscheiding *derision of property*	gentechnologie *gene technology*	1.2	0.34
roken *smoke*	dieren *animals*	1.17	0.29
makelaars *broker*	republiek *republic*	1.16	0.28

Table 1: Shown are the three concept pairs from our annotation study achieving the highest and the lowest average rating respectively (in Dutch and English).

in both sets. This yields a value in the interval [0, 1]. The higher the value, the more related the sets. Formally:

$$\frac{\sum_{l_c \in C} \max_{l'_e \in E} rel(l_c, l'_e) + \sum_{l_e \in E} \max_{l'_c \in C} rel(l_e, l'_c)}{|C| + |E|} \tag{1}$$

To validate this measure we conduct a study with human experts: three expert users, who are familiar with the thesaurus and the documents, were asked to judge for twenty-five documents the relatedness between the ground truth concepts and the classifier assigned concepts (taking the content of the document into account) on a 5-point scale: *very poor*, *poor*, *average*, *good* and *very good*. It should be emphasised, that our expert users have not created the ground truth concepts (those were created by library experts employed by the Dutch government). The average rating taken over all three individual expert ratings are considered as the ground-truth. The expert evaluations are used to compare the performance of the relatedness evaluation measure and the performance of a frequently used binary evaluation measure (F-measure). We hypothesise, that the classifier evaluation, which takes the semantic relatedness of the concepts into account will correlate to a larger degree with the expert judgements than the traditional binary evaluation measure.

4. EXPERIMENTS & RESULTS

Semantic Relatedness in the Parliament Thesaurus.

Examples of concept pairs that were selected for the annotation study are shown in Table 1; in particular the three concept pairs yielding the highest human annotator relatedness scores and the lowest scores respectively are listed.

The performance of the relatedness measures on the Parliament thesaurus are listed in Table 2. From these results two aspects stand out: (i) the relatively high correlation obtained for Rada's and Leacock & Chodorow's relatedness measure, and, (ii) the relatively poor performance of the remaining measures.

Traditionally, semantic relatedness measures have been evaluated on WordNet, the most well-known manually created lexical database. Seco et al. [13] evaluated all measures from our selection (except Rada) in a similar way on the WordNet graph against a test-bed of human judgements provided by Miller & Charles [8]. They reported significant higher correlations for the selected relatedness measures. Their correlation results range from 0.74 (Wu & Palmer) to 0.84 (Jiang & Conrath) and are in line with similar studies on WordNet such as Budanitsky et al. [2]. We conclude that measures which perform best on WordNet are not performing as well on our domain-dependent Parliament thesaurus.

Measures	r	τ
Rada (similarity)	0.43	0.35
Rada (relatedness)	0.73	0.55
Leacock & Chodorow (similarity)	0.49	0.36
Leacock & Chodorow (relatedness)	0.73	0.55
Wu & Palmer	0.39	0.33
Resnik	0.45	0.37
Jiang & Conrath	0.48	0.41
Lin	0.45	0.39

Table 2: Overview of the correlations of relatedness measures with human judgements of relatedness.

Classifier	Ground truth	Av. rating
toelating vreemdelingen	vreemdelingenrecht vreemdelingen procedures werknemers vluchtelingen	4.67
kinderbescherming kindermishandeling	jeugdigen gezondheidszorg	3.67

Table 3: Two examples of assigned classifier concepts vs. ground truth concepts and the average of the ratings obtained from the three experts users.

Multi-label Classifier Evaluation.

In Table 3 two examples of assigned classifier concepts vs. ground truth concepts are shown. Reported are also the average ratings obtained from the three expert users. Across all 25 evaluated documents, the mean rating was 3.28, indicating that the classifier framework performs reasonably well at assigning concepts related to the ground truth concepts.

Correlation	Semantically-enhanced	$\mathbf{F_1}$
r	0.67	0.48
τ	0.53	0.37

Table 4: Correlations between the expert ratings and the semantically-enhanced and the binary (F_1) classifier evaluation respectively.

The results of the second experiment are summarised in Table 4. Here, we employed Leacock & Chodorow's relatedness as it was our best performing approach (Table 2). The results indicate that for the annotated set of twenty-five documents, the relatedness evaluations correlate more with the expert evaluations than the evaluation based on F_1. The coefficients report an increase in correlation of at least 0.16 in favour of the relatedness evaluations. To emphasise the difference, we also present the scatter plots of the semantically-enhanced (Figure 2) and the binary, F_1 based, evaluation (Figure 3). In both plots, the corresponding trend line is drawn in red. It is evident, that in the binary case, the number of $F_1 = 0$ entries has a significant

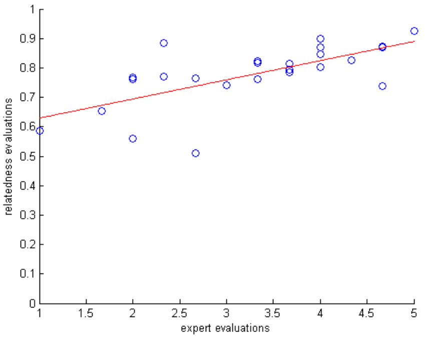


Figure 2: Expert versus relatedness evaluations.

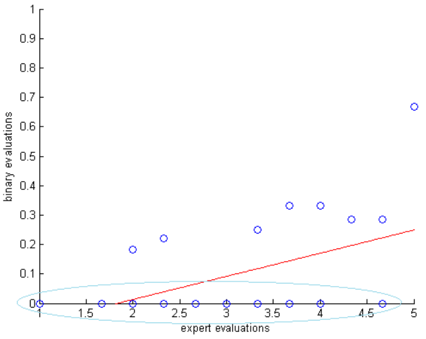


Figure 3: Expert versus binary evaluations.

impact on the obtained correlation. Note that the dispersion of relatedness evaluations in Figure 2 is higher at lower expert evaluations compared to higher expert evaluations. Whether this observation is to be attributed to noise is impossible to say due to the small size of the evaluation. We will investigate this issue further in future work.

5. CONCLUSIONS

In this paper, we have presented a two-step procedure to tackle a real-world problem: namely, the semantically-enhanced evaluation of multi-label classifiers that assign concepts to documents. We first investigated to what extent semantic relatedness measures that perform well on the most commonly used lexical database (WordNet) also perform well on another thesaurus (our domain-specific Parliament thesaurus). To this end, we conducted a user study where we let approximately 100 users annotate fifty concept pairs drawn from our thesaurus. We found that the results achieved on WordNet need to be considered with care, and it is indeed necessary to re-evaluate them when using a different source.

In a second step, we then exploited the semantic relatedness measure we found to perform best in the multi-label classifier evaluation. Again, we investigated the ability of such an evaluation measure to outperform a standard binary measure (F_1) by asking expert users to rate for a small set of documents the quality of the classifier concepts when compared to the ground truth concepts. Our results showed that an evaluation which includes the semantic relatedness of concepts yields results which are more in line with human raters than an evaluation based on binary decision.

Besides the issues already raised, in future work we plan to investigate in which graph/content characteristics WordNet differs from our thesaurus and to what extent these different characteristics can be employed to explain the difference in performance of the various semantic relatedness measures.

6. REFERENCES

[1] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 805–810, 2003.

[2] A. Budanitsky and G. Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[3] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 1606–1611, 2007.

[4] G. Hirst and D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 13:305–332, 1998.

[5] J. J. Jiang and D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. 1997.

[6] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[7] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th international conference on Machine Learning*, volume 1, pages 296–304, 1998.

[8] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

[9] S. Nowak, A. Llorente, E. Motta, and S. Rüger. The effect of semantic relatedness measures on multi-label classification evaluation. In *CIVR '10*, pages 303–310, 2010.

[10] S. Patwardhan. Incorporating dictionary and corpus information into a context vector measure of semantic relatedness, 2003.

[11] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.

[12] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. pages 448–453, 1995.

[13] N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *ECAI*, volume 16, page 1089, 2004.

[14] M. Strube and S. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1419, 2006.

[15] A. Sun and E.-P. Lim. Hierarchical text classification and evaluation. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 521–528. IEEE, 2001.

[16] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *CIKM '93*, pages 67–74. ACM, 1993.

[17] A. Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

[18] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30, 2008.

[19] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL '94*, pages 133–138, 1994.

Estimating the Time between Twitter Messages and Future Events

Ali Hürriyetoğlu, Florian Kunneman, and Antal van den Bosch
Centre for Language Studies, Radboud University Nijmegen
P.O. Box 9103
NL-6500 HD Nijmegen
ali.hurriyetoglu@gmail.com, {f.kunneman,a.vandenbosch}@let.ru.nl

ABSTRACT

We describe and test three methods to estimate the remaining time between a series of microtexts (tweets) and the future event they refer to via a hashtag. Our system generates hourly forecasts. A linear and a local regression-based approach are applied to map hourly clusters of tweets directly onto time-to-event. To take changes over time into account, we develop a novel time series analysis approach that first derives word frequency time series from sets of tweets and then performs local regression to predict time-to-event from nearest-neighbor time series. We train and test on a single type of event, Dutch premier league football matches. Our results indicate that in an 'early' stage, four days or more before the event, the time series analysis produces time-to-event predictions that are about one day off; closer to the event, local regression attains a similar accuracy. Local regression also outperforms both mean and median-based baselines, but on average none of the tested system has a consistently strong performance through time.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Spatial-Temporal Systems

General Terms

Algorithms, Performance

Keywords

Time series analysis, Event prediction, Twitter

1. INTRODUCTION

With the advent of social media, data streams of unprecedented volume have become available. These streams do not only contain text, but also identity markers of the persons who generated the text, and the time at which the messages were published. The availability of massive amounts of time-stamped texts is an invitation to incorporate time series analysis methods into the natural language processing toolbox. For instance, predictive models can be built through time series analysis that can estimate the likelihood and time of future events.

Our study focuses on textual data published by humans via social media about particular events. If the starting point of the event in time is taken as the anchor $t = 0$ point in time, texts can be viewed in relation to this point, and generalizations can be made over texts at different distances in time to $t = 0$. The goal of this paper is to present new methods that are able to automatically estimate the time-to-event from a stream of microtext messages. These methods could serve as modules in news media mining systems[1] to fill upcoming event calendars. The methods should be able to work robustly in a stream of messages, and the dual goal would be to make (i) reliable predictions of times-to-event (ii) as early as possible. Predicting that an event is starting imminently is arguably less useful than being able to predict its start in a number of days. This implies that if a method requires a sample of tweets (e.g. with the same hashtag) to be gathered during some time frame, the frame should not be too long, otherwise predictions could come in too late to be relevant.

In this paper we test the predictive capabilities of three different approaches. The first system is based on linear regression and maps sets of tweets with the same hashtag during an hour to a time-to-event estimate. The second system attempts to do the same based on local regression. The third system uses time series analysis. It takes into account more than a single set of tweets: during a certain time period it samples several sets of tweets in fixed time frames, and derives time series information from individual word frequencies in these samples. It compares these word frequency time series profiles against a labeled training set of profiles in order to find similar patterns of change in word frequencies. The method then adopts local regression: finding a nearest-neighbor word frequency time series, the time-to-event stored with that neighbor is copied to the tested time series. With this third system, and with the comparison against the second system, we can test the hypothesis that it is useful to gather time series information (more specifically, patterns in word frequency changes) over an amount of time.

This paper is structured as follows. We describe the relation of our work to earlier research in Section 2. The three systems are described in Section 3. Section 4 describes the overall experimental setup, including a description of the data, the baseline, and the evaluation method used. The results are presented and analyzed in Section 5. We conclude with a discussion of the results and future studies in Section 6.

[1] For instance, `http://www.zapaday.com/`

2. RELATED RESEARCH

The growing availability of digital texts with time stamps, such as e-mails, weblogs, and online news, has spawned various types of studies on the analysis of patterns in texts over time. An early publication on the general applicability of time series analysis on time-stamped text is [2]. A more recent overview of future predictions using social media is [5]. A popular goal of time series analysis of texts is *event prediction*, where a correlation is sought between a point in the future and preliminary texts.

Ritter *et al.* train on annotated open-domain event mentions in tweets in order to create a calendar of events based on explicit date mentions and words typical of the event [3]. While we also aim to estimate the point in time at which an event will take place, our focus lies on the pattern of anticipation seen in tweets linked to the time until the event occurs rather than specific time references to a future event. [4] do look at anticipation seen in tweets, but focus on personal activities in the very near future, while we aim to predict the time-to-event of potentially large-scale news events as early as possible.

3. METHODS

In this section we introduce the methods adopted in our study. They operate on streams of tweets, and generate hourly forecasts for the events that tweets with the same hashtag refer to. The single tweet is the smallest unit available for this task; we may also consider more than one tweet and aggregate tweets over a certain time frame. If these single tweets or sets of tweets are represented as bag-of-words vectors, the task can be cast as a regression problem: mapping a feature vector onto a continuous numeric output representing the time-to-event. In this study the smallest time unit is one hour, and all three methods work with this time frame.

3.1 Linear and local regression

In linear regression, each feature in the bag-of-words feature vector (representing the presence or frequency of occurrence of a specific word) can be regarded as a predictive variable to which a weight can be assigned that, in a simple linear function, multiplies the value of the predictive variable to generate a value for the response variable, the time-to-event. A multiple linear regression function can be approximated by finding the weights for a set of features that generates the response variable with the smallest error.

Local regression, or local learning [1], is the numeric variant of the k-nearest neighbor classifier. Given a test instance, it finds the closest k training instances based on a similarity metric, and bases a local estimation of the numeric output by taking some average of the outcomes of the closest k training instances.

Linear regression and local regression can be considered baseline approaches, but are complementary. While in linear regression an overall pattern is generated to fit the whole training set, local regression only looks at local information for classification (the characteristics of single instances). Linear regression is unfit for approximating gaussian or other non-linear distributions; as we will see, there are reasons to believe that there are substantial differences in tweets posted in different periods of time before an event. In contrast, local regression is unbiased and will adapt to any local distribution.

3.2 Time series analysis

Time series are data structures that contain multiple measurements of data features over time. If values of a feature change meaningfully over time, then time series analysis can be used to capture this pattern of change. Comparing new time series with memorized time series can reveal similarities that may lead to a prediction of a subsequent value or, in our case, the time-to-event. Our time series approach extends the local regression approach by not only considering single sets of aggregrated tweets in a fixed time frame (e.g. one hour in our study), but creating sequences of these sets representing several consecutive hours of gathered tweets. Using the same bag-of-words representation as the local regression approach, we find nearest neighbors of sequences of bag-of-word vectors rather than single hour frames. The similarity between a test time series and a training time series of the same length is calculated by computing their Euclidean distance. In this study we did not further optimize any hyperparameters; we set $k = 1$.

The time series approach generates predictions by following the same strategy as the simple local regression approach: upon finding the nearest-neighbor training time series, the time-to-event of this training time series is taken as the time-to-event estimate of the test time series. In case of equidistant nearest neighbors, the average of their associated time-to-events is given as the prediction.

4. EXPERIMENTAL SET-UP

4.1 Data collection

For this study we chose football matches as a specific type of event. They occur frequently, have a distinctive hashtag by convention ('#ajafey' for a match between Ajax and Feyenoord) and often generate a useful amount of tweets: up to tens of thousands of tweets per match. For the collection of training and test data we focused on Dutch football matches played in the *Eredivisie*. We harvested tweets by means of `twiqs.nl`, a database of Dutch tweets from December 2010 onwards. We selected the (arbitrary) top 6 teams of the league[2], and queried all matches played between them in 2011 and 2012. For each query, the conventional hashtag for a match was used with a restricted search space of three weeks before the time of the match until the start time of the match (to ensure that the collected tweets were referring to that specific match, and not to an earlier match consisting of the same home and away team and therefore the same hashtag).

The queries resulted in tweets referring to 60 matches between the selected six teams in the period from January 2011 until December 2012. From these, we selected the matches with the most frequent similar starting time, Sundays at 2:30 PM, for our experiment. As we focused on the amount of hours before an event, the actual time when a tweet is posted (for example during the night or in the afternoon) can bias the type of tweet; with the fixed starting time this

[2] Ajax, Feyenoord, PSV, FC Twente, AZ Alkmaar and FC Utrecht

effect is neutralized. To generate training and test events that simulate a system trained on passed events and tested on upcoming events, we selected tweets referring to matches played in 2011 (a calendar year comprising two halfs of a football season) as training data and tweets referring to 2012 matches as test data. This resulted in 12 matches as training events (totaling 54,081 tweets) and 14 matches as test events (40,204 tweets).

The time-to-event in hours was calculated for every tweet, based on their time of posting and the known start time of the event they referred to. For this task we did not take tweets into account that were posted during and after matches. We also constrained the number of days before the event: for both training and test sets, tweets were kept within eight days before the event. Although this is an artificial constraint, the eight days window captures the vast majority, about 98%, of forward-looking tweets.

4.2 Generation of training and test data

The goal of the experiments was to compare systems that generate hourly forecasts of the event start time for each test event. This was done based on the information in aggregated sets of tweets within the time span of an hour. Aggregation is done by treating all training events as one collection during the extraction of features. The linear and local regression methods only operate on vectors representing hour blocks. The time series analysis approach makes use of longer sequences of six hour blocks - this number was empirically set in preliminary experiments.

The aggregated tweets were used as training instances for the linear and local regression methods. To maximize the number of training instances, we generated a sequence of overlapping instances using the minute as a finer-grained shift unit. At every minute, all tweets posted within the hour before the tweets in that minute were added to the instance.

In order to reduce the feature space for the linear and local regression instances, we pruned every bag-of-word feature that occured less than 500 times in the training set. Linear regression was applied by means of R[3]. Absolute occurrence counts of features were taken into account. For local regression we made use of the k-NN implementation as part of TiMBL[4], setting $k = 5$, using Information Gain feature weighting, and an overlap-based metric as similary metric that does not count matches on zero values (features marking words that are absent in both test and training vectors). For k-NN, the binary value of features were used.

The time series analysis vectors are not filled with absolute occurrence counts, but with relative and smoothed frequencies. After having counted all words in each time frame, two frequencies are computed for each word. The first, the overall frequency of a word, is calculated as the sum of its counts in all time frames, divided by the total number of tweets in all time frames in our 8-day window. This frequency ranges between 0 (the word does not occur) and 1 (the word occurs in every tweet). The second frequency is computed per time frame for each word, where the word count in that frame is divided by the number of tweets in the frame. The latter frequency is the basic element in our time series calculations.

As many time frames contain only a small number of tweets, especially the frames more than a few days before the event, word counts are sparse as well. Besides taking longer time frames of more than a single sample size, frequencies can also be smoothed through typical time series analysis smoothing techniques such as moving average smoothing. We apply a pseudo-exponential moving average filter by replacing each word count by a weighted average of the word count at time frames t, $t-1$, and $t-2$, where $w_t = 4$ (the weight at t is set to 4), $w_{t-1} = 2$, and $w_{t-2} = 1$.

4.3 Evaluation and baselines

A common metric for evaluating numeric predictions is the Root Mean Squared Error (RMSE), cf. Equation 1. For all hourly forecasts made in N hour frames, a sum is made of the squared differences between the actual value v_i and the estimated value e_i; the (square) root is then taken to produce the RMSE of the prediction series.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(v_i - e_i)^2} \qquad (1)$$

We computed two straightforward baselines derived from the training set: the median and the mean of time-to-event over all training tweets. For the median baseline, all tweets in the training set were ordered in time and the median time was identified. As we use one-hour time frames throughout our study, we round the median by the one-hour time frame it is in, which turns out to be -3 hours. The mean is computed by averaging the time-to-event of all tweets, and again rounded at the hour. The mean is -26 hours.

5. RESULTS

Table 1 displays the averaged RMSE results on the 14 test events. On average the performance of the linear regression method is worse than both baselines, while the time series analysis outperforms the median baseline. Given that the best performing method is still an unsatisfactory 43 hours off, there is still a lot of improvement needed. The best method per event varies. Even linear regression, which has a below baseline performance on average, leads to the best RMSE for two events. It appears that some negative deviations (110 for 'twefey', 410 for 'tweaja') lead to the poor average RMSE.

The average performance of the different methods in terms of their RMSE according to hourly forecasts is plotted in Figure 1. In the left half of the graph the three systems outperform the baselines, except for an error peak of the linear regression method at around $t = -150$. Before $t = -100$ the time series prediction is performing rather well, with RMSE values averaging 23 hours. The linear regression and local regression methods produce larger errors at first, decreasing as time progresses. In the second half of the graph, however, only the local regression method retains fairly low RMSE

[3] http://www.r-project.org/

[4] http://ilk.uvt.nl/timbl

	Spring 2012								Fall 2012						
	azaja	feyaz	feyutr	psvfey	tweaja	twefey	tweutr	utraz	azfey	psvaz	twefey	utraz	utrpsv	utrtwe	Av (sd)
Baseline Median	63	49	54	62	38	64	96	71	62	67	62	66	61	62	63 (12)
Baseline Mean	51	**40**	44	51	**31**	52	77	58	**50**	55	51	53	49	**51**	51 (10)
Linear regression	52	42	59	54	410	**41**	41	33	111	**31**	110	54	37	68	82 (94)
Local regression	**48**	44	**35**	**41**	43	43	**31**	**20**	57	40	52	**48**	**34**	52	**43** (9)
Time Series	**48**	50	42	43	45	**41**	63	70	48	58	**46**	71	59	63	54 (10)

Table 1: Overall Root Mean Squared Error scores for each method: difference in hours between the estimated time-to-event and the actual time-to-event

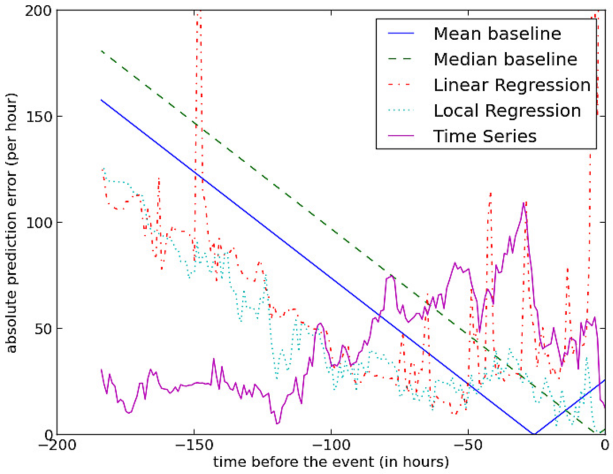


Figure 1: RMSE curves for the two baselines and the three methods for the last 192 hours before $t = 0$.

values at an average of 21 hours, while the linear regression method becomes increasingly erratic in its predictions. The time series analysis method also produces considerably higher RMSE values in the last days before the events.

6. CONCLUSION

In this study we explored and compared three approaches to time-to-event prediction on the basis of streams of tweets. We tested on the prediction of the time-to-event of football matches by generating hourly forecasts. When the three approaches are compared to two simplistic baselines based on the mean and median of the time-to-event of tweets sent before an event, only local regression displays better overall RMSE values on the tested prediction range of $192 \ldots 0$ hours before the event. Linear regression generates some highly erratic predictions and scores below both baselines. A novel time series approach that implements local regression based on sequences of samples of tweets performs better than the mean baseline, but under the median baseline.

Yet, the time series method generates fairly accurate forecasts during the first half of the test period. Before $t < -100$ hours, i.e. earlier than four days before the event, predictions by the time series method are only about a day off (23 hours on average in this time range). When $t \leq -100$, the local regression approach based on sets of tweets in hourly time frames is the better predictor, with RMSE values that are sometimes close to $t = 0$ (21 hours on average in this time range).

On the one hand, our results are not very strong: predictions that are more than two days off and that are at the same time only mildly better than simple baselines cannot be considered precise. However, the results indicate that if we divide the problem into an ‘early’ prediction system based on time series analysis and a ‘late’ prediction system based on local regression, we could limit the prediction error to within a day. If we can detect the point at which the time series analysis starts increasing its predicted time-to-event (which is the wrong trend as the event can only come closer in time), it is time to switch to the local regression system. In our data, this point is around $t = -100$.

In future work we plan to extend the current study in several directions. Most importantly, we plan to extend the study to other events, moving from football to other scheduled events, and from scheduled events to unscheduled events, the ultimate goal of a forecasting system like this. A second extension is to improve on the time series analysis method, particularly to investigate why it is performing well only up to several days before the future event (and what kind of patterns it matches successfully). We also plan to optimize the local regression approach, as we now utilize a fairly standard k-NN approach without optimized hyperparameters, and we have not optimized the selection of features either.

Acknowledgement

This study is financed by the COMMIT program as part of the Infiniti project.

7. REFERENCES

[1] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1–5):11–73, 1997.

[2] J. Kleinberg. Temporal dynamics of On-Line information streams. In *Data stream management: Processing high-speed data streams*. Springer, 2006.

[3] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’12, pages 1104–1112. ACM, 2012.

[4] W. Weerkamp and M. De Rijke. Activity prediction: A Twitter-based exploration. In *Proceedings of TAIA’12*, Aug. 2012.

[5] S. Yu and S. Kak. A survey of prediction using social media. In *ArXiv e-prints*, Mar. 2012.

On the Assessment of Expertise Profiles (Abstract)

Richard Berendsen
University of Amsterdam, The Netherlands
r.w.berendsen@uva.nl

Krisztian Balog
University of Stavanger, Norway
krisztian.balog@uis.no

Toine Bogers
Royal School of Library Information Science, Denmark
tb@iva.dk

Antal van den Bosch
Radboud University Nijmegen, The Netherlands
a.vandenbosch@let.ru.nl

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

1. INTRODUCTION

We summarize findings from [3]. At the TREC Enterprise Track [2], the need to study and understand *expertise retrieval* has been recognized through the introduction of the expert finding task. The goal of *expert finding* is to identify a list of people who are knowledgeable about a given topic. An alternative task, building on the same underlying principle of computing people-topic associations, is *expert profiling*, where systems have to return a list of topics that a person is knowledgeable about [1].

We focus on benchmarking systems performing the topical expert profiling task. We define this task as a ranking task, where knowledge areas from a thesaurus have to be ranked for an expert. We release an updated version of the UvT (Universiteit van Tilburg) expert collection [1]: the *TU* (Tilburg University) *expert collection*.[1] The TU expert collection is based on the *Webwijs* ("Webwise") system[2]: a publicly accessible database of TU employees who are involved in research or teaching. In a back-end for this database, experts can indicate their skills by selecting knowledge areas from an alphabetical list. Prior work has used these *self-selected knowledge areas* as ground truth for both expert finding and expert profiling tasks [1].

One problem with self-selected knowledge areas is that they may be sparse, since experts have to select them from an alphabetically ordered list of well over 2,000 knowledge areas. Using these self-selected knowledge areas as ground truth for assessing automatic profiling systems may therefore not reflect the true predictive power of these systems. To find out more about how well these systems perform in real-world circumstances, we have asked TU employees to judge and comment on profiles that have been automatically generated for them. We refer to this process as the *assessment experiment*. In § 2 we answer the broad research question "How well are we doing at the expert profiling task?" We do this through an error analysis and through a content analysis of free text comments that experts could give. During the assessment experiment, experts judge areas in the system-generated profiles on a five point scale. This yields a new set of graded relevance assessments, which we call the *judged system-generated knowledge areas*. In § 3 our research question is: "Does benchmarking a set of expertise retrieval systems with the judged system-generated profiles lead to different conclusions, compared to benchmarking with the self-selected profiles?" We benchmark eight state-of-the-art expertise retrieval systems with both sets of ground truth and investigate differences in completeness, system ranking, and the number of significant differences detected between systems.

2. THE ASSESSMENT EXPERIMENT

Generating profiles. We use eight expert profiling models. Each of them uses either Model 1 or Model 2 [1], either uses Dutch or English representations of knowledge areas, and either uses relations between knowledge areas extracted from the thesaurus or not. Because experts have limited time and participate in the experiment on a voluntary basis, we rank areas by their estimated probability of being part of the expert's profiles. The more traditional pooling approach would require experts to exhaustively judge the pool. We linearly combine output scores of the eight systems, giving each system equal weight. We boost the top three of each system by adding a sufficiently large constant to the top three scores, to make sure they are judged. System-generated knowledge areas that were in the original self-selected profile of the expert are ticked by default in the interface, but the expert may deselect them, thereby judging them non-relevant.

The assessment interface. Using the assessment interface, each expert can judge retrieved knowledge areas relevant by ticking them. Immediately below the top twenty knowledge areas listed by default, the expert has the option to view and assess additional knowledge areas. For the ticked knowledge areas, experts have the option to indicate a level of expertise. If they do not do this, we still include these knowledge areas in the judged system-generated profiles, with a level of expertise of three ("somewhere in the middle"). At the bottom of the interface, experts can leave any comments they might have on the generated profile.

Error analysis of system-generated profiles. Here, we aim to find properties of experts that can explain some of the variance in nDCG@100 performance. We use the self-selected profiles of all 761 experts we generated a profile for, allowing us to incorporate self-selected knowledge areas that were missing from the system-generated profiles in our analysis. Based on visual inspection, we find no correlation between the number of relevant knowledge areas selected and nDCG@100, and no correlation between the number of documents associated with an expert and nDCG@100 either. Intuitively, the relationship between the ratio of relevant knowledge areas and number of documents associated with the expert is also interesting. However this ratio does not correlate with nDCG@100 either. Looking a bit deeper into the different kinds

[1] http://ilps.science.uva.nl/tu-expert-collection
[2] http://www.tilburguniversity.edu/webwijs/

DIR 2013, April 26, 2013, Delft, The Netherlands.

of document that can be associated with an expert, we find that it matters whether or not an expert has a research description. For the 282 experts without a research description we achieve significantly lower average nDCG@100 performance than for the remaining 479 experts (Welch Two Sample t-test, $p < 0.001$). The difference is also substantial: 0.39 vs. 0.30 for experts with and without a research description, respectively. It is not surprising that these research descriptions are important; they constitute a concise summary of a person's qualifications and expertise, written by the expert himself/herself.

Content analysis of expert feedback. 239 Experts participated in the self-assessment experiment, providing graded relevance judgments. 91 Of them also left free text comments. We study what are important aspects in expert feedback by means of a content analysis. In our analysis, expert comments were coded by two of the authors, based on a coding scheme developed in a first pass over the data. A statement could be assigned multiple aspects. After all aspect types were identified, the participants' comments were coded in a second pass over the data. Upon completion, the two coders resolved differences through discussion. Micro-averaged inter-annotator agreement (the number of times a comment was coded with the same aspect divided by the total number of codings) was 0.97. The main aspects in the feedback of experts are (i) missing a key knowledge area in the generated profile (36%); (ii) only irrelevant knowledge areas in the profile (16.9%); (iii) redundancy in the generated profiles (11.2%); (iv) knowledge areas being too general (11.2%). Based on these results, it seems there is still room for improvement in the performance of expert profiling systems. Also, interesting directions for future work are to address the redundancy in generated profiles, and to take into account the specificity of knowledge areas.

3. BENCHMARKING DIFFERENCES

Completeness. To assess completeness, we estimate the set of all relevant knowledge areas for an expert with the union of the self-selected profile and the judged system-generated profile. Doing this, we find that the judged system-generated profiles are more complete. On average, a judged system-generated profile contains 81% of all relevant knowledge areas, while a self-selected profile contains only 65%.

Changes in system ranking. To better understand the differences in evaluation outcomes between using the self-selected profiles (we call this ground truth set: **GT1**) and the judged system-generated profiles (we call this set **GT5**), we construct three intermediate sets of ground truth (**GT2-4**). Each intermediate set differs from the previous set in only one aspect; in this way we can isolate the contribution each difference makes to differences in evaluation outcomes. The intermediate sets of ground thruth are: **GT2**: The 239 self-selected profiles of participants in the assessment experiment; **GT3**: For each self-selected profile of an assessor, we only use knowledge areas that were in the system-generated profile. This means that knowledge areas that are not in the system-generated profile are treated as irrelevant; **GT4**: The knowledge areas judged relevant during the assessment experiment. We only consider binary relevance; if a knowledge area was selected it is considered as relevant, otherwise it is taken to be irrelevant. We report Kendall's τ correlation between system rankings using consecutive sets of ground truth. We rank the eight systems that contributed to the generated profile, but leave out the algorithm that combined them. In this abstract, we focus on system rankings computed with nDCG@100. With eight systems, Kendall's τ correlations of 0.79 or higher are significant at the $\alpha = 0.01$ level. Correlating **GT1-GT2**, we find that evaluating on a subset of experts does not change system ranking much: $\tau = 0.86$. Correlating **GT2-GT3**, we find that regarding non-pooled knowledge areas as irrelevant does not rank our eight systems very differently: $\tau = 0.86$. Correlating **GT3-GT4** we find that new knowledge areas judged relevant during the assessment do change system ranking: $\tau = 0.56$. Contrasting **GT4-GT5** we find that considering the grade of relevance does not change system ranking: $\tau = 1.00$.

Pairwise significant differences. The final analysis we conduct concerns a high-level perspective: the sensitivity of our evaluation methodology. The measurement that serves as a rough estimate here is the average number of systems each system differs from; we compute this for each of the five sets of assessments **GT1-5**, and focus here on nDCG@100. We use Fisher's pairwise randomization test ($\alpha = 0.001$). For **GT1** we get 4.75. For **GT2** we observe 3.00, the decrease is not surprising as **GT2** has much less experts. Regarding non-pooled knowledge areas as irrelevant does not affect sensitivity much (**GT3**: 2.75). The sensitivity increases again when we evaluate with the more complete judged system-generated knowledge areas (**GT4**:3.50). Taking into account the level of expertise indicated, we see another small increase (**GT5:**4.00).

4. CONCLUSION

We released, described and analyzed the TU expert collection for assessing automatic expert profiling systems. In an error analysis of system-generated profiles, we found that it is easier to generate profiles for experts that have a research description. A content analysis of expert feedback revealed that there is room for improvement in the expert profiling task, and that an interesting direction for future work is to consider diversity in profiles. Contrasting using the self-selected profiles or using the judged system-generated profiles for evaluation, we find that the latter profiles are more complete. The two sets of ground truth rank systems somewhat differently.

Acknowledgments. This research was partially 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, the PROMISE Network of Excellence co-funded by the 7th Framework Programme of the European Commission, grant agreement no. 258191, 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 Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 380-70-011, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-nl program, and under COMMIT project Infiniti.

References

[1] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *SIGIR'07*, pages 551–558. ACM, 2007.

[2] K. Balog, I. Soboroff, P. Thomas, N. Craswell, A. P. de Vries, and P. Bailey. Overview of the TREC 2008 Enterprise Track. In *TREC 2008 Proceedings*. NIST, 2009. Special Publication.

[3] R. Berendsen, K. Balog, T. Bogers, A. van den Bosch, and M. de Rijke. On the assessment of expertise profiles. *JASIST*, To appear.

Towards Optimum Query Segmentation: In Doubt Without
(Extended Abstract)*

Matthias Hagen Martin Potthast Anna Beyer Benno Stein

Bauhaus-Universität Weimar
99421 Weimar, Germany
<first name>.<last name>@uni-weimar.de

ABSTRACT

Query segmentation is the problem of identifying compound concepts or phrases in a query. We conduct the first large-scale study of human segmentation behavior, introduce robust accuracy measures, and develop a hybrid algorithmic segmentation approach based on the idea that, in cases of doubt, it is often better to (partially) leave queries without any segmentation.

1. INTRODUCTION

Keyword queries are the predominant way of expressing information needs on the web. Search engines nowadays rely on tools that help them to interpret, correct, classify, and reformulate every submitted query in a split second before the actual document retrieval begins. We study one such tool that identifies indivisible sequences of keywords in a query (e.g., `new york times`) that users could have included in double quotes—the task of query segmentation.

Our contributions include the first large-scale analysis of human segmentation behavior (50 000 queries, each segmented by 10 annotators) showing that different segmentation strategies should be applied to different types of queries. In particular, a good strategy often is to refrain from segmenting too many keywords (i.e., in doubt without segmentation).

2. NOTATION AND RELATED WORK

A query q is a sequence $(w_1, \ldots, w_k)$ of k keywords. Every contiguous subsequence of q forms a potential segment. A valid segmentation for q consists of disjunct segments whose concatenation yields q again. The problem of query segmentation is the automatic identification of the "best" valid segmentation, where "best" refers to segmentations that humans would choose or that maximize retrieval performance. Note that a valid segmentation determines for each pair $\langle w, w' \rangle$ of consecutive keywords in q whether or not there should be a segment break between w and w'. Hence, there are 2^{k-1} valid segmentations for a k-keyword query and $k(k-1)/2$ potential segments with at least two keywords.

Risvik et al. [5] were the first to propose an algorithm for query segmentation based on pointwise mutual information. Later on, more sophisticated approaches like the supervised learning method by Bergsma and Wang [1] combined many features (web and query log frequencies, POS tags, etc.). Recently, efficiency issues become more important [3] and evaluation moves away from simple accuracy against human segmentations towards retrieval impact analyses [4].

3. HOW HUMANS SEGMENT

Our study of human segmentation behavior is based on the Webis-QSeC-10 corpus [3] consisting of 53 437 web queries (3–10 keywords) with at least 10 different annotators per query. One of our intentions is to compare human quoting on noun phrase queries with that on other queries. As automatic POS-tagging in short queries is a difficult task, we restrict our analysis to *strict noun phrases* (SNP) composed of only nouns, numbers, adjectives, and articles. These parts-of-speech can be identified reliably using for instance Qtag.[1] About 47% of the queries are tagged as SNP queries.

Our study of how humans quote queries results in the following major findings. (1) SNP queries are segmented more often than others, (2) in segmented SNP queries more keywords are contained in segments, and (3) annotators agree more on short queries but unanimity is an exception (many queries even do not have a segmentation supported by an absolute majority of annotators). These findings suggest that algorithms aiming at accuracy against human segmentations should take into account the query type. The second implication is to carefully reconsider the traditional accuracy measures (some based on annotator unanimity).

4. ACCURACY MEASURES REVISITED

Segmentation accuracy is typically measured against a corpus of human segmentations on three levels: query accuracy (ratio of correctly segmented queries), segment accuracy (precision and recall of the computed segments), and break accuracy (ratio of correct decisions between pairs of consecutive words). The crucial point is the choice of the reference segmentation from the corpus. Traditionally, the reference is the segmentation that best fits the computed one (i.e., the one with highest break accuracy) without any further considerations. We argue that for corpora with many annotators per query (e.g., the Webis-QSeC-10) this is an oversimplification and scoring references from a set of weighted alternatives should be an integral part of accuracy measuring.

Given a query q, and a list of m reference segmentations $(S_1, \ldots, S_m)$ from m different annotators, we propose the following two strategies to select a reference segmentation. (1) Weighted Best Fit: select the S_i chosen by an absolute majority of annotators if there is one. Otherwise select the S_i as the traditional best fit strategy (i.e., the S_i maximizing break accuracy). But then, the obtained accuracy values are weighted by the ratio of votes allotted to S_i compared to the maximum number of votes on any segmentation in $(S_1, \ldots, S_m)$. (2) Break Fusion: instead of selecting a reference segmentation from $(S_1, \ldots, S_m)$, fuse them into one. For each pair of consecutive words in q: if at least half of the annotators inserted a segment break, so does this strategy. If not, no break is inserted.

*Original paper with all the omitted details in CIKM 2012 [2].

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

[1] http://phrasys.net/uob/om/software

To demonstrate the impact of the new reference schemes, we apply them in a comparison of the segmentation algorithms from the literature (results in the full paper). With our new schemes many of the relative accuracy differences between segmentation algorithms increase and more of these differences become statistically significant. Hence, the new reference selectors provide a more robust means to evaluate segmentation accuracy.

5. HYBRID QUERY SEGMENTATION

The decision whether or not to introduce segments into a query is a risky one: a bad segmentation leads to bad search results or none at all, whereas a good one improves them. Since keeping users safe from algorithm error is a core principle at most search engines, and since even a small error probability yields millions of failed searches given billions of searches per day, a risk-averse strategy is the way to go. In doubt, it is always safer to do without any query segmentation. This observation suggests to use a hybrid strategy that treats different types of queries in different ways. One of the main findings on human segmentation behavior is to distinguish SNP queries from others. As potential strategies for either type, we consider algorithms from the literature and two newly developed baselines that only segment Wikipedia titles (WT) or only Wikipedia titles and SNPs (WT+SNP) following our dictionary based scheme [3].

6. EVALUATION

In our evaluation, we compare instances of hybrid query segmentation to traditional approaches with respect to three performance measures. (1) We measure segmentation accuracy using the Webis-QSeC-10. (2) We measure retrieval performance in a TREC setting using the commercial search engine Bing and the Indri ClueWeb09 search engine hosted at Carnegie Mellon University.[2] (3) We measure runtime performance and memory footprint.

We have systematically combined traditional segmentation algorithms (including the option “none” of not segmenting) to form instances of hybrid segmentation. As expected, there is no one-fits-all combination which maximizes performance with respect to all of the above measures. The following table shows the best performing combinations.

Query type	**Hybrid segmentation instance**		
	HYB-A (accuracy)	HYB-B (Bing)	HYB-I (Indri)
SNP	[3] (= WT+SNP)	None	None
other	WT	WT	[3]

In what follows, we give brief descriptions of the experimental results (more details in the full paper). An explanation for the variant HYB-A can be found in our analysis of human quoting behavior. There, it is shown that accuracy-oriented algorithms should segment SNP queries more aggressively (more keywords in segments) than other queries, which in turn should be segmented conservatively (less keywords in segments). This is exactly the strategy of HYB-A. On SNP queries, the algorithm [3] aggressively segments all phrases that appear at least 40 times on the web, whereas the WT baseline on the other queries conservatively segments only Wikipedia titles.

With respect to retrieval performance we evaluate on the TREC topics in the Web tracks 2009–2011 and the Million Query track 2009 with at least one document being judged as relevant and at least 3 keywords (61 topics from the Web tracks, 294 from the Million query track). Our results suggest that different search engines (i.e., retrieval models) each require specifically tailored hybrid segmentation algorithms. Otherwise, query segmentation may not improve significantly over not segmenting at all.

The main findings of evaluating accuracy and retrieval performance are the following: (1) better accuracy not necessarily improves retrieval performance, (2) SNP queries can often be left unsegmented in terms of retrieval performance. However, there is a grain of salt: our TREC experiments are small-scale compared to the number of queries that went into measuring accuracy. The retrieval performance experiments should be scaled up significantly in order to draw more reliable conclusions. In any case, our experiments have shown that the decision of when to segment at all is an important one.

Besides accuracy and retrieval performance, also runtime and memory consumption are crucial criteria to judge the applicability of a segmentation algorithm in a real-world setting. Runtime is typically measured as throughput of queries per second while memory consumption concerns the data needed for operation. Regarding throughput, a pointwise mutual information baseline is by far the fastest approach (with bad accuracy and retrieval performance). The WT and WT+SNP baselines are faster than [3] since they sum up fewer weights of potential segments. The hybrid approaches are slowest due to the POS tagging step. With respect to memory consumption the WT baseline needs an order of magnitude less data than mutual information or WT+SNP which in turn need much less than [3]. Taking into account the rumored monthly throughput of major search engines of about 100 billion queries (i.e., about 40 000 queries per second), all segmentation approaches can easily handle such a load when run on a small cluster of standard PCs.

7. CONCLUSION AND OUTLOOK

Our study of human query segmentation behavior inspired a new hybrid framework that treats SNP queries different than other queries and that can be tailored to mimic human query quoting better than the state-of-the-art algorithms. However, an important and somewhat unexpected outcome of complementary TREC style evaluation is that maximizing segmentation accuracy not necessarily maximizes retrieval performance as well. Nevertheless, we show the flexibility of the hybrid framework and optimize it for two retrieval models. There, not segmenting SNP queries at all is best, opposing our finding that humans quote SNP queries more aggressively.

We hypothesize that query segmentation is especially beneficial on long non-SNP queries, which currently are underrepresented in the TREC corpora. Hence, scaling up retrieval performance evaluation with a broad range of retrieval models is an important future direction. This could shed light on the question of why SNP queries apparently are better off without any segmentation. One starting point could be an analysis of the best segmentations for different retrieval models in order to better understand what differentiates a “perfect” retrieval-oriented segmentation from those of the algorithms developed so far.

8. REFERENCES

[1] S. Bergsma and Q. Wang. Learning noun phrase query segmentation. In *EMNLP-CoNLL 2007*, pp. 819–826.

[2] M. Hagen, M. Potthast, A. Beyer, and B. Stein. Towards optimum query segmentation: in doubt without. In *CIKM 2012*, pp. 1015–1024.

[3] M. Hagen, M. Potthast, B. Stein, and C. Bräutigam. Query segmentation revisited. In *WWW 2011*, pp. 97–106.

[4] Y. Li, B.-J. P. Hsu, C. Zhai, and K. Wang. Unsupervised query segmentation using clickthrough for information retrieval. In *SIGIR 2011*, pp. 285–294.

[5] K. Risvik, T. Mikolajewski, and P. Boros. Query segmentation for web search. In *WWW 2003 (Posters)*.

[2] http://boston.lti.cs.cmu.edu/Services/batchquery

Reusing Historical Interaction Data for Faster Online Learning to Rank for IR (Abstract)

Katja Hofmann
k.hofmann@uva.nl

Anne Schuth
a.g.schuth@uva.nl

Shimon Whiteson
s.a.whiteson@uva.nl

Maarten de Rijke
derijke@uva.nl

ISLA, University of Amsterdam

ABSTRACT

We summarize the findings from Hofmann et al. [6]. Online learning to rank for information retrieval (IR) holds promise for allowing the development of "self-learning" search engines that can automatically adjust to their users. With the large amount of e.g., click data that can be collected in web search settings, such techniques could enable highly scalable ranking optimization. However, feedback obtained from user interactions is noisy, and developing approaches that can learn from this feedback quickly and reliably is a major challenge. In this paper we investigate whether and how previously collected (historical) interaction data can be used to speed up learning in online learning to rank for IR. We devise the first two methods that can utilize historical data (1) to make feedback available during learning more reliable and (2) to preselect candidate ranking functions to be evaluated in interactions with users of the retrieval system. We evaluate both approaches on 9 learning to rank data sets and find that historical data can speed up learning, leading to substantially and significantly higher online performance. In particular, our pre-selection method proves highly effective at compensating for noise in user feedback. Our results show that historical data can be used to make online learning to rank for IR much more effective than previously possible, especially when feedback is noisy.

1. INTRODUCTION

In recent years, learning to rank methods have become popular in information retrieval (IR) as a means of tuning retrieval systems. However, most current approaches work *offline*, meaning that manually annotated data needs to be collected beforehand, and that, once deployed, the system cannot continue to adjust to user needs, unless it is retrained with additional data. An alternative setting is *online* learning to rank, where the system learns directly from interactions with its users. These approaches are typically based on reinforcement learning techniques, meaning that the system tries out new ranking functions (also called rankers), and learns from feedback inferred from users' interactions with the presented rankings. In contrast to offline learning to rank approaches, online approaches do not require any initial training material, but rather automatically improve rankers while they are being used.

A main challenge that online learning to rank for IR approaches have to address is to learn as quickly as possible from the limited quality and quantity of feedback that can be inferred from user interactions. In this paper we address this challenge by proposing the first two online learning to rank algorithms that can reuse previously collected (historical) interaction data to make online learning more reliable and faster.

DIR 2013, April 26, 2013, Delft, The Netherlands.

2. METHOD

We model online learning to rank for IR as a cycle of interactions between users and retrieval system. Users submit queries to which the system responds with ranked result lists. The user interacts with the result lists, and these interactions allow the search engine to update its ranking model to improve performance over time. We address the problem of learning a ranking function that generalizes over queries and documents, and assume that queries are independent of each other, and of previously presented results.

Learning in this setting is implemented as stochastic gradient descent to learn a weight vector $\mathbf{w}$ for a linear combination of ranking features. Ranking features $\mathbf{X}$ encode the relationship between a query and the documents in a document collection (e.g., tf-idf, PageRank, etc.). Given a weight vector $\mathbf{w}$, and ranking features $\mathbf{X}$ candidate documents are scored using $\mathbf{s} = \mathbf{w}\mathbf{X}$. Sorting the documents by these scores results in a result list for the given $\mathbf{w}$. Our baseline method learns weight vectors using the *dueling bandit gradient descent* (DBGD, [8]) algorithm. This algorithm maintains a current best weight vector, and learns by generating candidate weight vectors that are compared to the current best. When a candidate is found to improve over the current best weight vector, the weights are updated.

User feedback is interpreted using interleaved comparison methods [7]. These methods can infer unbiased relative feedback about ranker quality from implicit feedback, such as user clicks. In particular, they combine the result lists produced by the two rankers into one result ranking, which is then shown to the user. Clicks on the documents contributed by each ranker can then be interpreted as votes for that ranker. Our baseline interleaved comparison methods are Balanced Interleave (BI) and Team Draft (TD) [7]. Our extensions for reusing historical data are enabled by Probabilistic Interleave (PI) [4].

Based on DBGD and PI, we can now define our two approaches for reusing historical data to speed up online learning to rank.

Reliable Historical Comparison (RHC). RHC is based on the intuition that repeating comparisons on historical data should provide additional information to complement live comparisons, which can make estimates of relative performance more reliable. This is expected to reduce noise and lead to faster learning. Reusing historical interaction data for additional comparisons is possible using PI, but estimates may be biased. To remove bias, we use importance sampling as proposed in [5]. We combine the resulting historical estimates with the original live estimate using the Graybill-Deal estimator [2]. This combined estimator weights the two estimates by the ratio of their variances.

Candidate Pre-Selection (CPS). Our second approach for reusing historical data to speed up online learning to rank for IR uses historical data to improve candidate generation. Instead of randomly

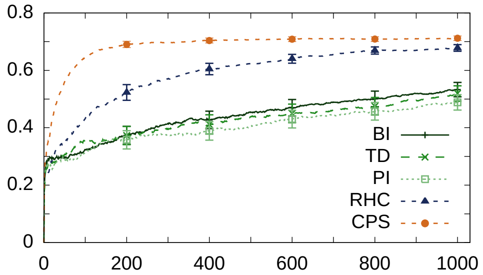


Figure 1: Offline performance in NDCG (vertical axis, computed on held-out test queries after each learning step) on *NP-2003* data set, for the *informational* click model over 1K queries.

generating a candidate ranker to test in each comparison, it generates a pool of candidate rankers, and selects the most promising one using historical data. We hypothesize that historical data can be used to identify promising rankers, and that the increased quality of candidate rankers can speed up learning.

3. EXPERIMENTS AND RESULTS

Our experiments are designed to investigate whether online learning to rank for IR can be sped up by using historical data. They are based on an existing simulation framework, which combines fully annotated learning to rank data sets with probabilistic user models to simulate user interactions with a search engine that learns online.

We conduct our experiments on the 9 data sets provided as LETOR 3.0 and 4.0. These data sets implement retrieval tasks that range from navigational (e.g., home page finding) to informational (e.g., literature search). They range in size from 50 to 1700 queries, 45 to 64 features, and up to 1000 judged documents per topic. Starting a data set, we simulate user queries by uniform sampling from the provided queries. After the retrieval system returns a ranked result list, user feedback is generated using the Dependent Click Model (DCM) [3], an extension of the Cascade Model [1] that has been shown to be effective in explaining users' click behavior in web search. We instantiate the user model with three levels of noise. The *perfect* click model provides reliable feedback. The *navigational* and *informational* model reflect two types of search tasks. Our experiments compare and contrast three baseline runs (BI: *balanced interleave*, TD: *team draft*, and PI: *probabilistic interleave*) and our proposed methods for reusing historical interaction data, RHC and CPS. Over all data sets, we find that the performance of the baseline methods substantially degrades with noise as expected. Comparing the performance of these baseline methods to that of RHC and CPS answers our research question of whether reusing historical interaction data can compensate for this noise.

Performance for our method RHC confirms our hypothesis. Under perfect user feedback, the method's performance is equivalent to that of the baseline methods that use live data only. However, its relative performance improves with increased noise. Under the informational click model, the method performs significantly better than the best baseline method on five of the nine data sets. Performance is still equivalent on two data sets, and decreases on the remaining two. Best performance under all click models is achieved by our CPS method. While we expected performance improvements under noisy click feedback, this method achieves significant improvements over the baseline methods even when click feedback is perfect. We attribute this improvement to the more exhaustive local exploration enabled by this approach. Performance improvements are highest under noisy feedback. An example is shown in Figure 1. This graph shows the offline performance in terms of NDCG on the held-out test folds for the data set NP2003 over the number of iterations (queries). We see that the baseline methods BI, TD, and PI learn slowly as the amount of available feedback increases. RHC, learning is significantly and substantially faster, because complementing comparisons with historical data makes feedback for learning more reliable. Finally, CPS is able to compensate for most of the noise in user feedback, leading to significantly faster learning.

4. CONCLUSION

In this paper, we investigated whether and how historical data can be reused to speed up online learning to rank for IR. We proposed the first two online learning to rank approaches that can reuse historical interaction data. RHC uses historical interaction data to make feedback inferred from user interactions more reliable. CPS uses this data to preselect candidate rankers so that the quality of the rankers compared in live interactions is improved.

We found that both proposed methods can improve the reliability of online learning to rank for IR under noisy user feedback. Best performance was observed using the CPS method, which can outperform all other methods significantly and substantially under all levels of noise. Performance gains of CPS were particularly high when click feedback was noisy. This result demonstrates that CPS is effective in compensating for noise in click feedback.

This work is the first to show that historical data can be used to significantly and substantially improve online performance in online learning to rank for IR. These methods are expected to make online learning with noisy feedback more reliable and therefore more widely applicable.

Acknowledgements. This research was partially 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, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP and BILAND projects funded by the CLARIN-nl program, the Dutch national program COMMIT, by the ESF Research Network Program ELIAS, and the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences.

References

[1] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM '08*, pages 87–94, 2008.

[2] F. Graybill and R. Deal. Combining unbiased estimators. *Biometrics*, 15(4):543–550, 1959.

[3] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *WSDM '09*, pages 124–131, 2009.

[4] K. Hofmann, S. Whiteson, and M. de Rijke. A probabilistic method for inferring preferences from clicks. In *CIKM '11*, pages 249–258, 2011.

[5] K. Hofmann, S. Whiteson, and M. de Rijke. Estimating interleaved comparison outcomes from historical click data. In *CIKM '12*, 2012.

[6] K. Hofmann, A. Schuth, S. Whiteson, and M. de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In *WSDM '13*, pages 183–192, 2013.

[7] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *CIKM '08*, 2008.

[8] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *ICML'09*, pages 1201–1208, 2009.

Reliability and Validity of Query Intent Assessments

Compressed version of paper accepted for publication in JASIST

Suzan Verberne
s.verberne@cs.ru.nl

Maarten van der Heijden
m.vanderheijden@cs.ru.nl

Max Hinne
mhinne@cs.ru.nl

Maya Sappelli
m.sappelli@cs.ru.nl

Saskia Koldijk
saskia.koldijk@tno.nl

Eduard Hoenkamp
hoenkamp@acm.org

Wessel Kraaij
w.kraaij@cs.ru.nl

Keywords

Query intent classification, User studies, Data collection, Validation

1. INTRODUCTION

The quality of a search engine critically depends on the ability to present results that are an adequate response to the user's query and intent. If the intent (or the most likely intent) behind a query is known, a search engine can improve retrieval results by adapting the presented results to the more specific intent instead of the — underspecified — query [6]. Several studies have proposed classification schemes for query intent. Broder [3] suggested that the intent of a query can be either informational, navigational or transactional. He estimated percentages for each of the categories by presenting Altavista users a brief questionnaire about the purpose of their search after submitting their query. After manual classification of 1,000 queries he warned that "inferring the user intent from the query is at best an inexact science, but usually a wild guess." Later, many expansions and alternative schemes have been proposed, and more dimensions were added.

In many existing intent recognition studies, training and test data for automatic intent recognition have been created in the form of annotations by external assessors who are not the searchers themselves [2, 1, 4]. Post-hoc intent annotation by external assessors is not ideal; nevertheless, intent annotations from external judges are widely used in the community for evaluation or training purposes. Therefore it is important for the field to get a better understanding of the quality of this process as an approximation for first-hand annotation by searchers themselves. Some annotation studies have investigated the *reliability* of query intent annotations by measuring the agreement between two external assessors on the same query set [1, 4]. What these studies do not measure, is the *validity* of the judgments.

In this paper, we aim to measure the validity of query intent assessments, i.e. how well an external assessor can estimate the underlying intent of a searcher's query. We use a classification scheme to describe search intent.

2. OUR INTENT CLASSIFICATION SCHEME

We introduce a multi-dimensional classification scheme of query intent that is inspired by and uses aspects from [3], [2], [4] and [5]. Our classification scheme consists of the following dimensions of search intent.

1. Topic: categorical, fixed set of categories from the well-known Open Directory Project (ODP), giving a general idea of what the query is about.
2. Action type: categorical, consisting of: *informational*, *navigational* and *transactional*. This is the categorisation by Broder.
3. Modus: categorical, consisting of: *image*, *video*, *map*, *text* and *other*. This dimension is based on [5].
4. *source authority sensitivity*: 4-point ordinal scale (high sensitivity: relevance strongly depends on authority of source).
5. *spatial sensitivity*: 4-point ordinal scale (high sensitivity: relevance strongly depends on location).
6. *time sensitivity*: 4-point ordinal scale (high sensitivity: relevance strongly depends on time/date).
7. *specificity*: 4-point ordinal scale (high specificity: very specific results desired; low specificity: explorative goal).

3. EXPERIMENTS

In order to obtain labeled queries from search engine users, we created a plugin for the Mozilla Firefox web browser. After installation by the user, the plugin locally logs all queries submitted to Google. We asked colleagues (all academic scientists and PhD students) to participate in our experiment. Participants were asked to occasionally (at a self-chosen moment) annotate the queries they submitted in the last 48 hours, using a form that presented our intent classification scheme. To guarantee that no sensitive information was involuntarily submitted, participants were allowed to skip any query they did not want to submit.

In total, 11 participants enrolled in the experiment. Together, they annotated 605 queries with their query intent, of which 135 duplicates. On average, each searcher annotated 55 queries (standard deviation=73). The three topic

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

Table 1: Reliability and validity of query intent assessments in terms of Cohen's Kappa, averaged over the assessor pairs. Boldface indicates moderate agreement ($\kappa >= 0.4$) or higher.

Dimension	Reliability (stdev)	Validity (stdev)
Topic	**0.56** (0.19)	**0.42** (0.16)
Action type	0.29 (0.20)	0.09 (0.08)
Modus	**0.41** (0.14)	0.22 (0.10)
Source authority sensitivity	0.05 (0.05)	0.10 (0.03)
Time sensitivity	**0.48** (0.08)	0.14 (0.04)
Spatial sensitivity	**0.69** (0.07)	**0.41** (0.04)
Specificity	0.26 (0.10)	0.05 (0.09)

categories that were used most frequently in the set of annotated queries were *computer*, *science* and *recreation*.

To obtain labels from external assessors we used the same form as was used by the participants. Four of the authors acted as external assessors; all queries were assessed by at least two assessors.

4. RESULTS

In order to answer the question "How *reliable* is our intent classification scheme as an instrument for measuring search intent?", we calculated the interobserver reliability as the agreement between the external assessors using Cohen's κ. The middle column of Table 1 shows the average agreement over the assessor pairs for each dimension. For only one of the seven dimensions from our classification scheme) substantial agreement (0.6 or higher) was reached. For four of the seven, at least moderate agreement (0.4 or higher) was reached: least moderately reliable query intent classification is possible for the dimensions topic, modus, time sensitivity and spatial sensitivity.

In order to answer the question, "How *valid* are the intent classifications by external assessors?", we compared the intent classifications by the external assessors to the intent classifications by the searchers themselves. We calculated κ-scores per dimension for each assessor–searcher pair. The rightmost column of Table 1 shows the average agreement over the assessor–searcher pairs. The table shows that moderately valid query intent classification is possible on two of the seven dimensions from our classification scheme: topic and spatial sensitivity. The difference between the inter–assessor agreement and the assessor–searcher agreement was significant on all dimensions.

Our experiments suggest that classification of queries into Topic categories can be done reliably, even though we had 17 different topics to choose from. This is good news for a future implementation of automatic query classification because topic plays an important role in query disambiguation and personalisation. The second reliable dimension, Spatial sensitivity, is an important dimension for local search: every web search takes place at a physical location, and there are types of queries for which this location is relevant (e.g. the search for restaurants or events). The finding that external assessors can reach a moderate agreement with the searcher on this dimension shows the feasibility of recognizing that a query is sensitive to location. The search engine can respond by promoting search results that match with the location.

For the implementation of intent classification in a search engine, training data is needed: The features are the query terms (the textual content of the query) and the labels are the values for the dimensions in the classification scheme. Analysis of the queries shows that for many intent dimensions, there is no direct connection between words in the query and the intent of the query. For example, in the 33 queries that were annotated by the searcher with the *image* modus (e.g. "photosynthesis"; "coen swijnenberg") there were no occurrences of words such as 'image' or 'picture', and only 2 of the 90 queries that were annotated with a high temporal sensitivity contained a time-related query word. This means that for automatic classification, it is difficult to generalize over queries. However, the most likely intent can still be learned for individual queries by following the diversification approach in the ranking of the search results: The engine can learn the probability of intents for specific queries by counting clicks on different types of results. This approach requires a huge amount of clicks to be recorded (which is possible for large search engines such as Google) and the long tail of low-frequency queries will not be served.

5. CONCLUSIONS

We found that four of the seven dimensions in our classification scheme could be annotated moderately reliably ($\kappa > 0.4$): topic, modus, time sensitivity and spatial sensitivity. An important finding is that queries could not reliably be classified according to the dimension 'action type', which is the original Broder classification. Of the four reliable dimensions, only the annotations on the topic and spatial sensitivity dimensions were valid ($\kappa > 0.4$) when compared to the searcher's annotations. This shows that the agreement between external assessors is not a good estimator of the validity of the intent classifications.

In conclusion, we showed that Broder was correct with his warning that "inferring the user intent from the query is at best an inexact science, but usually a wild guess". Therefore, we encourage the research community to consider - where possible - using query intent classifications by the searchers themselves as test data.

6. REFERENCES

[1] A. Ashkan, C. Clarke, E. Agichtein, and Q. Guo. Classifying and characterizing query intent. *Advances in Information Retrieval*, pages 578–586, 2009.

[2] R. Baeza-Yates, L. Calderón-Benavides, and C. González-Caro. The Intention Behind Web Queries. In F. Crestani, P. Ferragina, and M. Sanderson, editors, *String Processing and Information Retrieval*, LNCS 4209, pages 98–109, Berlin Heidelberg, 2006. Springer-Verlag.

[3] A. Broder. A taxonomy of web search. In *ACM SIGIR forum*, volume 36, pages 3–10. ACM, 2002.

[4] C. González-Caro, L. Calderón-Benavides, R. Baeza-Yates, L. Tansini, and D. Dubhashi. Web Queries: the Tip of the Iceberg of the User's Intent. In *Workshop on User Modeling for Web Applications, WSDM 2011*, 2011.

[5] S. Sushmita, B. Piwowarski, and M. Lalmas. Dynamics of genre and domain intents. *Information Retrieval Technology*, pages 399–409, 2010.

[6] R. White, P. Bennett, and S. Dumais. Predicting short-term interests using activity-based search context. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1009–1018. ACM, 2010.

What Snippets Say About Pages (Abstract)

T. Demeester
Ghent University
tdmeeste@ugent.be

D. Nguyen
University of Twente
d.nguyen@utwente.nl

D. Trieschnigg
University of Twente
d.trieschnigg@utwente.nl

C. Develder
Ghent University
cdvelder@ugent.be

D. Hiemstra
University of Twente
d.hiemstra@utwente.nl

ABSTRACT

We summarize findings from [1]. What is the likelihood that a Web page is considered relevant to a query, given the relevance assessment of the corresponding snippet? Using a new Federated Web Search test collection that contains search results from over a hundred search engines on the internet, we are able to investigate such research questions from a global perspective. Our test collection covers the main Web search engines like Google, Yahoo!, and Bing, as well as smaller search engines dedicated to multimedia, shopping, etc., and as such reflects a realistic Web environment. Using a large set of relevance assessments, we are able to investigate the connection between snippet quality and page relevance. The dataset is strongly heterogeneous, and care is required when comparing resources. To this end, a number of probabilistic variables, based on snippet and page relevance, are introduced and discussed.

1. INTRODUCTION

Finding our way around among the vast quantities of data on the Web would be unthinkable without the use of Web search engines. Apart from a limited number of very large search engines that constantly crawl the Web for publicly available data, a large amount of smaller and more focused search engines exist, specialized in specific information goals or data types (e.g., online shopping, news, multimedia, social media). In order to promote research on Federated Web Search, we created a large dataset containing sampled results from 108 search engines on the internet, and containing relevance judgments for the top 10 results (both snippets and pages) from all of these resources for 50 test topics (from the TREC 2010 Web Track). The relevance judgements are particularly interesting for analysis, partly because they originate from very diverse collections (both in size and in scope, whereby the relevance judgments are done in a generic way), and partly because we not only judged the result pages, but also, independently, the original snippets. Our analysis deals with ranked result lists from diverse retrieval algorithms, and with snippets from various snippet generation strategies, as they are currently in use on the Web.

This abstract is based on [1], which has the following scope. First, after an overview of related work, the relevance judgments for the new dataset are discussed at length, with emphasis on the assessors' consistency. Second, a number of potential difficulties in Federated Web Search and especially in the evaluation of relevance are discussed, related to the heterogeneous character of the resources. Finally, a probabilistic analysis of the relationship between the indicative snippet relevance and the actual page relevance is presented (where by 'page' we denote a result item like a web page, a video, scientific paper... as returned by the included search engines). In a further contribution [2], it is shown that the information carried by an average snippet can be used to make a reasonable prediction of the relevance of the result page itself. Within the limits of this abstract, we will primarily focus on the question of why the user's snippet-based prior estimation of the page relevance is of paramount importance for the overall performance of the search service. Using the relevance judgments for the dataset presented in [3], the relevant concepts are illustrated for the specific case of large general web search engines.

DIR 2013, April 26, 2013, Delft, The Netherlands.

2. SNIPPET VS. PAGE RELEVANCE

The intuition behind this paper is simple: a search engine can only exploit the full potential of its retrieval algorithm if the result snippets reflect the relevance of the corresponding pages as well as possible. This means that a highly relevant result should be presented to the user by a very promising snippet, and a less relevant result page by a less interesting snippet. If there is a mismatch between what the user estimates from a result snippet and the actual result page, the overall performance of the system degrades.

For a more formal analysis, we introduce the snippet relevance variable S, and the page relevance variable P. As for the specific relevance levels, the snippet relevance S ranges from No, over Unlikely and Maybe, to Sure, indicating how likely the assessor estimates the result page behind the snippet to be relevant. The levels for P, the page relevance, are Non, Rel (containing minimal relevant information), HRel (highly relevant), Key (worthy of being a top result), and Nav (for navigational queries). In this paper we will either indicate the considered relevance level explicitly, such as S = Sure (i.e., considering only snippets with the label Sure), or define binary relevance levels, such as P $\geq$ HRel (indicating page relevance levels of HRel, Key, or Nav).

Table 1: Overview of the relationship between page and snippet judgments, for different types of resources, and based on the page relevance level P≥HRel.

	S=Unlikely	S=Maybe	S=Sure		
	$\mathcal{P}(P\|S)$	$\mathcal{P}(P\|S)$	$\mathcal{P}(P\|S)$	$\mathcal{P}(P,S)$	$\mathcal{P}(P)$
General Web search	0.20	0.40	0.65	0.26	0.34
Multimedia	0.09	0.23	0.48	0.06	0.09
News	0.09	0.19	0.42	0.02	0.03
Shopping	0.06	0.10	0.21	0.01	0.03
Encyclopedia/Dict	0.05	0.23	0.58	0.11	0.14
Books	0.12	0.10	0.18	0.02	0.05
Blogs	0.12	0.23	0.40	0.05	0.07

Table 2: Comparison of the largest general Web search engines

		P≥HRel and S=Sure			P≥Key and S=Sure		
	$\mathcal{P}$(S=Sure)	$\mathcal{P}(P\|S)$	$\mathcal{P}(P,S)$	$\mathcal{P}(P)$	$\mathcal{P}(P\|S)$	$\mathcal{P}(P,S)$	$\mathcal{P}(P)$
Google	0.42	0.68	0.28	0.38	0.39	0.16	0.19
Yahoo!	**0.47**	0.69	**0.32**	**0.44**	0.38	0.18	**0.22**
Bing	0.41	0.60	0.24	0.28	0.30	0.12	0.13
Baidu	0.21	0.43	0.09	0.12	0.23	0.05	0.06
Mamma.com	0.43	**0.73**	0.31	0.41	**0.44**	**0.19**	**0.22**

Retrieval systems are typically being evaluated based on the probability of relevance of the result page, written $\mathcal{P}(P)$. If however the access to that page also depends on the user's estimate of a snippet, the actual measure to consider should be $\mathcal{P}(P,S)$, the mutual probability of relevance for both the snippet and the page. Note that it can be written as $\mathcal{P}(S)\mathcal{P}(P|S)$, in which $\mathcal{P}(P|S)$ is the conditional probability of the page label, given the snippet label. Studying $\mathcal{P}(P|S)$ is especially instructive, for instance to find out how often a relevant page remains hidden behind a non-convincing snippet.

For several resource categories, table 1 gives empirical estimates of such probabilities for binary page relevance P≥HRel, based on our relevance judgements. Comparing $\mathcal{P}(P|S)$ for the snippet labels Maybe and Sure shows that a relatively large amount of HRel pages are behind snippets which were judged only Maybe, especially for the general search engines. This shows that often a HRel page's snippet cannot convince the user that the page is indeed highly relevant. We also observe that for the snippet label S=Sure, e.g., the News resources display a relatively high $\mathcal{P}(P|S)$, against a very low $\mathcal{P}(P,S)$. In other words, these resources returned only very few relevant results for our test topics, but if a snippet was found relevant, 4 out of 10 times it points to one of those few relevant results.

As the test topics are best suited for the general Web search engines, we can explicitly compare the performance of four of the largest general Web search engines in our collection, i.e., Google, Yahoo!, Bing, and Baidu, as well as Mamma.com, which is actually a metasearch engine. Table 2 presents the results. It appears that for the snippet label S=Sure and two page relevance levels (P≥HRel and P≥Key), $\mathcal{P}(P,S)$ is consistently lower than $\mathcal{P}(P)$, which is actually the averaged precision@10 of page relevance, and does not take into account the fact that the snippet is not always as promising as the page is relevant. The metasearch engine outperforms the others, as it aggregates results from a number of resources, such as Google, Yahoo!, and Bing. We want to stress that the considered test topics are still no representative collection of, for example, popular Web queries, and therefore we cannot draw any further conclusions about these search engines beyond the scope of our test collection. Yet, here is another example of how the table might be interpreted, with that in mind. Considering only Key results, we could compare Yahoo! and Bing. Yahoo! seems to score higher for all reported parameters, so either Bing's collection contains a smaller number of relevant results, or Yahoo!'s retrieval algorithms are better tuned for our topics. The lower value of $\mathcal{P}(P|S)$ for Bing shows that it has a slightly increased chance that the page for a promising snippet appears less relevant. However, the ratio of $\mathcal{P}(P,S)$ and $\mathcal{P}(P)$ is higher for Bing than for Yahoo!, indicating that for Yahoo!, its own recall on Key pages will be decreased more due to the quality of the snippets, than for Bing. In fact, we found that $\mathcal{P}$(S=Sure|P≥Key) is 79% for Yahoo!, but 91% for Bing.

3. CONCLUSIONS

Analyzing the relationship between the relevance of snippets from a large amount of on-line search engines and the relevance of the corresponding result pages, clearly shows that in the evaluation of and comparison between different resources, the snippets cannot be left out.

4. ACKNOWLEDGMENTS

This research was partly supported by the Netherlands Organization for Scientific Research, NWO, grants 639.022.809 and 640.005.002, and partly by iMinds in Flanders.

5. REFERENCES

[1] T. Demeester, D. Nguyen, D. Trieschnigg, C. Develder, and D. Hiemstra. What Snippets Say about Pages in Federated Web Search. In *AIRS*, 2012.

[2] T. Demeester, D. Nguyen, D. Trieschnigg, C. Develder, and D. Hiemstra. Snippet-Based Relevance Predictions for Federated Web Search. In *ECIR*, 2013.

[3] D. Nguyen, T. Demeester, D. Trieschnigg, and D. Hiemstra. Federated Search in the Wild: the Combined Power of over a Hundred Search Engines. In *CIKM*, 2012.

Cognitive Temporal Document Priors (Abstract)*

Maria-Hendrike Peetz and Maarten de Rijke
ISLA, University of Amsterdam
{M.H.Peetz, derijke}@uva.nl

1. INTRODUCTION

Every moment of our life we retrieve information from our brain: we remember. We remember items to a certain degree: for a mentally healthy human being retrieving very recent memories is virtually effortless, while retrieving untraumatic memories from the past is more difficult [4]. Early research in psychology was interested in the rate at which people forget single items, such as numbers. Psychology researchers have also studied how people retrieve events. Chessa and Murre [1] record events and hits of web pages related to an event and fit models of how people remember, the so-called *retention function*. Modeling the retention of memory has a long history in psychology, resulting in a range of proposed retention functions. In information retrieval (IR), the relevance of a document depends on many factors. If we request recent documents, then how much we remember is bound to have an influence on the relevance of documents. Can we use the psychologists' models of the retention of memory as (temporal) document priors? Previous work in temporal IR has incorporated priors based on the exponential function into the ranking function [2, 3]—this happens to be one of the earliest functions used to model the retention of memory. Many other such functions have been considered by psychologists to model the retention of memory—what about the potential of other retention functions as temporal document priors?

Inspired by the cognitive psychology literature on human memory and on retention functions in particular, we consider seven temporal document priors. We propose a framework for assessing them, building on four key notions: *performance*, *parameter sensitivity*, *efficiency*, and *cognitive plausibility*, and then use this framework to assess those seven document priors. We show that on several data sets (newspaper and microblog), with different retrieval models, the exponential function as a document prior should not be the first choice. Overall, other functions, like the Weibull function, score better within our proposed framework.

2. METHODS

We introduce basic notation and then describe several retention functions serving as temporal document priors.

We say that document D in document collection $\mathcal{D}$ has time *time*(D) and text *text*(D). A query q has time *time*(q) and text *text*(q). We write $\delta_g(q,D)$ as the time difference between *time*(q) and *time*(D) with the granularity g.

We introduce a series of retention functions. The *memory chain models* ((1) and (2)) build on the assumptions that there are different memories. The Weibull functions ((3) and (4)) are of interest to psychologists because they fit human retention behavior well. In contrast, the retention functions *linear* and *hyperbolic* ((6) and (7)) have little cognitive background.

Memory Chain Model. The memory chain model [1] assumes a multi-store system of different levels of memory. The probability to store an item in one memory being μ,

$$f_{\text{MCM-1}}(D,q,g) = \mu e^{-a\delta_g(q,D)}. \tag{1}$$

The parameter a indicates how items are being forgotten. The function $f_{\text{MCM-1}}(D,q,g)$ is equivalent to the exponential decay in [2] when the two parameters (μ and a) are equal. In the two-store system, an item is first remembered in short term memory with a strong memory decay, and later copied to long term memory. Each memory has a different decay parameter, so the item decays in both memories, at different rates. The overall retention function is

$$f_{\text{MCM-2}}(D,q,g) = 1 - e^{-\mu_1\left(e^{-a_1\delta_g(q,D)} + \frac{\mu_2}{a_2-a_1}(e^{-a_2\delta_g(q,D)} - e^{-a_1\delta_g(q,D)})\right)}, \tag{2}$$

where an overall exponential memory decay is assumed. The parameter μ_1 and μ_2 are the likelihood that the items are initially saved in short and long term memory, whereas a_1 and a_2 indicate the forgetting of the items. Again, t is the time bin.

One can also consider the Weibull function

$$f_{\text{BW}}(D,q,g) = \left(e^{-\frac{a\delta_g(D,q)}{d}^{d}}\right), \tag{3}$$

and its extension

$$f_{\text{EW}}(D,q,g) = b + (1-b)\mu e^{\left(-\frac{a\delta_g(D,q)}{d}\right)^{d}}. \tag{4}$$

Here, a and d indicate how long the item is being remembered: a indicates the overall volume of what can potentially be remembered, d determines the steepness of the forgetting function; μ determines the likelihood of initially storing an item, and b denotes an asymptote parameter.

The power function is ill-behaved between 0 and 1 and usual approximations start at 1. The *amended power function* is

$$f_{\text{AP}}(D,q,g) = b + (1-b)\mu(\delta_g(D,q)+1)^{a}, \tag{5}$$

where a, b, and μ are the decay, an asymptote, and the initial learning performance.

A very intuitive baseline is given by the linear function,

$$f_{\text{L}}(D,q,g) = \frac{-(a\cdot\delta_g(q,D)+b)}{b}, \tag{6}$$

where a is the gradient and b is $\delta_g(q, \operatorname{argmax}_{D'\in\mathcal{D}}\delta_g(q,D'))$. Its range is between 0 and 1 for all documents in $\mathcal{D}$.

The hyperbolic discounting functionhas been used to model how humans value rewards: the later the reward the less they consider

*The full version of this paper appeared in ECIR 2013 [5].

Table 1: Assessing temporal document priors; # improved queries is w.r.t. MCM-1.

Condition	MCM-1	MCM-2	BW	EW	AP	L	HD
# impr. queries (temp.)	n/a	14 (58%)	5 (20%)	16 (67%)	5 (20%)	2 (8%)	6 (25%)
# impr. queries (non-temp.)	n/a	27 (35%)	35 (46%)	26 (34%)	38 (50%)	36 (47 %)	33 (43%)
# impr. queries (Tweets2011)	n/a	16 (32%)	17 (34%)	22 (44%)	0 (0%)	17 (34 %)	21 (42%)
MAP	+	–	+	0	0	–	0
P10	–	–	0	–	0	0	0
Rprec	0	±	+	±	0	0	0
MRR	0	0	+	0	+	+	+
Sensitivity of parameters	–	–	+	–	+	+	+
Efficiency: # parameters	2	4	2	4	3	2	1
Plausibility: fits human behav.	+	++	+	++	+	n/a	n/a
Plausibility: neurobiol. expl.	+	+	–	+	–	–	–

the reward worth. Here,

$$f_{\mathrm{HD}}(D,q,g) = \frac{1}{-(1+k*\delta_g(q,D))}, \qquad (7)$$

where k is the discounting factor.

3. EXPERIMENTS

We propose a set of three criteria for assessing temporal document priors and we determine whether the priors meet the criteria.

A framework for assessing temporal document priors.

Performance. A document prior should improve the performance on a set of test queries for a collection of time-aware documents. A well-performing document prior improves on the standard evaluation measures across different collections and across different query sets. We use the *number of improved queries* as well as the *stability of effectiveness* with respect to different evaluation measures as an assessment for performance, where stability refers to that improved or non-decreasing performance over several test collections.

Sensitivity of parameters. A well-performing document prior is not overly sensitive with respect to parameter selection: the best parameter values for a prior are in a *region* of the parameter space and not a single value.

Efficiency. Query runtime efficiency is of little importance when it comes to distinguishing between document priors: if the parameters are known, all document priors boil down to simple look-ups. We use the *number of parameters* as a way of assessing the efficiency of a prior.

Cognitive plausibility. We define the cognitive plausibility of a document prior (derived from a retention function) with the goodness of fit in large scale human experiments [4]. This conveys an experimental, but objective, view on cognitive plausibility. We also use a more subjective definition of plausibility in terms of *neurobiological background* and how far the retention function has a biological explanation.

Discussion. To ensure comparability with previous work, we use different models for different datasets: TREC-2 and TREC-{6,7,8} for news and Tweets2011 for social media. On the news data set, we analyse the effect of different temporal priors on the performance of the baseline, query likelihood with Dirichlet smoothing [2]. We optimize parameters for different priors on TREC-6 using grid search. On the Tweets2011 data set, we analyse the effect of different temporal priors incorporated in the query modeling [3].

Table 1 gives an overview of the assessment of different document priors. We find that all but BW, AP, and L are stable in the parameter optimisation. Of those functions, BW and L have only few parameters, and BW performs best.

4. CONCLUSION

We have proposed a new perspective on functions used for temporal document priors used for retrieving recent documents. We showed how functions with a cognitive moti- vation yield similar, if not significantly better results than others on news and microblog datasets. In particular, the Weibull function is stable, easy to optimize, and motivated by psychological experiments.

Acknowledgments. We thank Jessika Reissland for her inspiration. This research was partially 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, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the BILAND project funded by the CLARIN-nl program, the Dutch national program COMMIT, by the ESF Research Network Program ELIAS, and the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences.

References

[1] A. G. Chessa and J. M. Murre. A memory model for internet hits after media exposure. *Physica A Statistical Mechanics and its Applications*, 2004.

[2] X. Li and W. B. Croft. Time-Based Language Models. In *CIKM '03*, 2003.

[3] K. Massoudi, E. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating Query Expansion and Quality Indicators in Searching Microblog Posts. In *ECIR 2011*, 2011.

[4] M. Meeter, J. M. J. Murre, and S. M. J. Janssen. Remembering the news: modeling retention data from a study with 14,000 participants. *Memory & Cognition*, 33:793–810, 2005.

[5] M.-H. Peetz and M. de Rijke. Cognitive temporal document priors. In *34th European Conference on Information Retrieval (ECIR'13)*, 2013.

Social Book Search: The Impact of Professional and User-Generated Content on Book Suggestions

Marijn Koolen[1] Jaap Kamps[1] Gabriella Kazai[2]

[1] University of Amsterdam, The Netherlands
[2] Microsoft Research, Cambridge UK,

ABSTRACT

The Web and social media give us access to a wealth of information, not only different in quantity but also in character—traditional descriptions from professionals are now supplemented with user generated content. This challenges modern search systems based on the classical model of topical relevance and ad hoc search. We compare classical IR with social book search in the context of the LibraryThing discussion forums where members ask for book suggestions. This paper is an compressed version of [2].

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process)*

General Terms: Experimentation, Measurement, Performance

Keywords: Book search, User-generated content, Evaluation

1. INTRODUCTION

The web gives access to a wealth of information that is different from traditional collections both in quantity and in character. Especially through social media, there is more subjective and opinionated data, which gives rise to different tasks where users are looking not only for facts but also views and interpretations, which may require different notions of relevance. In this paper we look at how search has changed by directly comparing classical IR and social search in the context of the LibraryThing (LT) discussion forums, where members ask for book suggestions. We use a large collection of book descriptions from Amazon and LT, which contain both professional metadata and user-generated content (UGC), and compare book suggestions on the forum with Mechanical Turk judgements on topical relevance and recommendation for evaluation of retrieval systems. Searchers not only consider the topical relevance of a book, but also care about how interesting, well-written, recent, fun, educational or popular it is. Such affective aspects may be mentioned in reviews, but Amazon, LT and many similar sites do not include UGC in the main search index. Our main research question is:

- How does social book search compare to traditional search tasks?

For this study, we set up the Social Search for Best Books (SB) task as part of the INEX 2011 Books and Social Search Track.[1] We want to find out whether the suggestions are complete and reliable enough for retrieval evaluation and how social book search is related to traditional search tasks. We also want to know if users prefer professional or UGC for judging topical relevance and for recommendation, and how standard IR models cope with UGC.

2. SOCIAL SEARCH FOR BEST BOOKS

In this section we detail collection and the LT forum topics.

Collection The Amazon/LT collection [1] consists of 2.8 million book records from Amazon, identified by ISBN, extended with social metadata from LT, marked up in XML. These records contain title information, Dewey classification codes and Subject headings supplied by Amazon. The reviews and tags were limited to the first 50 reviews and 100 tags respectively during crawling. The professional metadata is more evenly distributed than the UGC. Books have a single classification code and most have one or two subject headings, although a small fraction has no professional metadata. Typical of UGC, popular books have many tags and reviews while many others have few or none. The median number of reviews and tags are 0 and 5 respectively. That is, the majority has no reviews but at least a handful of tags.

Topics LibraryThing users discuss their books in forums dedicated to certain topics. Many of the topic threads are started with a request from a member for interesting, fun new books to read. Other members often reply with links to works catalogued on LT, which we connected to books in our collection through their ISBN. These requests for recommendations are natural expressions of information needs for a large collection of online book records, and the book suggestions are human recommendations from members interested in the same topic. For the Social Search for Best Books task we selected a set of 211 topics, some focused on fiction and some on non-fiction books. For the Mechanical Turk experiment we focus on a subset of 24 topics.

MTurk Judgements We compare the LT forum suggestions against traditional judgements of topical relevance, as well as against recommendation judgements. We set up an experiment on Amazon Mechanical Turk to obtain judgements on document pools based on top-10 pooling of the 22 runs submitted by the 4 participating groups. We designed a task to ask Mechanical Turk workers to judge the relevance of 10 books for a given book request. Apart from a question on topical relevance, we also asked whether they would recommend a book to the requester and which part of the metadata—curated or user-generated—was more useful for determining the topical relevance and for recommendation. We included some quality assurance and control measure to deter spammers and sloppy workers. Averaged over workers the LT agreement is 0.52.

3. SYSTEM-CENTERED ANALYSIS

We compare system rankings of the 22 official runs based on the forum suggestions and on the MTurk relevance judgements. The Kendall's τ system ranking correlation between the forum sugges-

[1] https://inex.mmci.uni-saarland.de/tracks/books/

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

Table 1: MTurk and LT Forum evaluation (nDCG@10 and recall@1000) of runs over different index fields

	MTurk							
	Rel		Rec		Rel&Rec		LT-Sug	
Field	nDCG	recall	nDCG	recall	nDCG	recall	nDCG	recall
Title	0.212	0.601	0.260	0.545	0.172	0.591	0.055	0.350
Dewey	0.000	0.009	0.003	0.007	0.000	0.005	0.001	0.022
Subject	0.016	0.008	0.021	0.010	0.016	0.009	0.003	0.009
Review	**0.579**	0.720	**0.786**	**0.756**	**0.542**	**0.783**	**0.251**	**0.680**
Tag	0.368	0.694	0.435	0.665	0.320	0.718	0.216	0.602

tions for 211 topics and the MTurk judgements on the 24 topics is 0.36. This is not due to the difference between the 211 topics of the forum suggestions and the subset of 24 topics selected for MTurk, as the correlation between the forum suggestions of the 211 and 24 topic sets is $\tau = 0.90$. It could be that the forum suggestions are highly incomplete. Most topics have few suggestions (median is 7). If the suggestions are a small fraction of all relevant books, good and bad systems will perform poorly as the chances of ranking the few suggested books above other relevant books is small. However, the highest MRR score among the 22 runs is 0.481. This means that on average, over 211 topics, this system returns a suggested book in the top 2. If this only occurs for a few topics, it could be ascribed to mere coincidence, but over 211 topics, such a high average is unlikely due to chance. Based on this, we argue the forum suggestions are relatively complete but represent a different task from the ad hoc task modelled by the topical relevance judgements from MTurk. In [2] we also show that the forum suggestions behave differently from known-item topics.

Next, we created a number of our runs to compare the forum suggestions against the MTurk judgements. For indexing we use Indri, Language Model, with Krovetz stemming, stopword removal and default smoothing (Dirichlet, μ=2,500). The titles of the forum topics are used as queries. In our base index, each xml element is indexed in a separate field, to allow search on individual fields.

Generally, systems perform better on recommendation judgements (MTurk-Rec in Table 1) than on topical relevance judgments (MTurk-Rel), and their combination (MTurk-Rel&Rec) and worst on the forum suggestions (LT-Sug). The suggestions seem harder to retrieve than books that are topically relevant. The Title field is the most effective of the non-UGC fields. It gives better precision and recall than the Dewey and Subject fields across all sets of judgements. The Review field is more effective than the Tag field. Note that all runs use the same queries. Even though book titles alone provide little information about books, with the Title field the majority of the judged topically relevant books can be found in the top 1,000, but only a third of the suggestions. The review and tag fields have high R@1000 scores for all four sets of judgements. There is something about suggestions that goes beyond topical relevance, which the UGC fields are better able to capture. Furthermore, the retrieval system is a standard language model, which was developed to capture topical relevance. Apparently these models can also deal with other aspects of relevance. It also shows how ineffective book search systems are if they ignore reviews. Even though there are many short, vague and unhelpful reviews, there seems to be enough useful content to substantially improve retrieval. This is different from general web search, where low quality and spam documents need to be dealt with.

4. USER-CENTERED ANALYSIS

The MTurk workers answered questions on which part of the metadata is more useful to determine topical relevance and which part to determine whether to recommend a book. Workers could indicate the description does not have enough information to answer questions Q1 (topical relevance) and Q3 (recommendation). We see in Table 2 the fraction of books for which workers did not have enough information split over the descriptions with no reviews (column 2), at least one review (column 3), no tags (column 4) and at least 10 distinct tags (column 5). First, without reviews, workers indicate they do not have enough information to determine whether a book is topically relevant in 37% of the cases, and label the book as relevant in 30% of the cases. When there is at least one review, in only 1% of the cases do workers have too little information to determine topical relevance, but in 54% of the cases they label the book as relevant. Reviews contain important information for topical relevance. The presence of tags seems to have no effect, as the fractions are stable across books with different numbers of tags. We see a similar pattern for the recommendation question (Q3).

Table 2: Impact of presence of reviews and tags on judgements

		Reviews		Tags	
		0 rev.	≥1 rev.	0 tags	≥10 tags
Top. Rel. (Q1)	Not enough info.	0.37	0.01	0.09	0.09
	Relevant	0.30	0.54	0.49	0.48
Recommend. (Q3)	Not enough info.	0.53	0.01	0.14	0.12
	Rel. + Rec.	0.22	0.51	0.46	0.45

In summary, the presence of reviews is important for both topical relevance and recommendation, while the presence and quantity of tags plays almost no role.

5. CONCLUSIONS

In this paper we ventured into unknown territory by studying the domain of social book search with traditional metadata complemented by a wealth of user generated descriptions. We also focused on requests and recommendations that users post in real life based on the social recommendations of the forums. We observe that the forum suggestions are complete enough to be used as evaluation, but they are different in nature than traditional judgements for known-item, ad hoc and recommendation tasks. Even though most online book search systems ignore UGC, our experiments show that this content can improve both traditional ad hoc retrieval effectiveness and book suggestions and that standard language models seem to deal well with this type of data.

Our results highlight the relative importance of professional metadata and UGC, both for traditional known-item and ad hoc search as well as for book suggestions.

Acknowledgments

This research was supported by the Netherlands Organization for Scientific Research (NWO projects # 612.066.513, 639.072.601, and 640.005.001) and by the European Community's Seventh Framework Program (FP7 2007/2013, Grant Agreement 270404).

REFERENCES

[1] T. Beckers, N. Fuhr, N. Pharo, R. Nordlie, and K. N. Fachry. Overview and Results of the INEX 2009 Interactive Track. In *ECDL*, volume 6273 of *LNCS*, pages 409–412. Springer, 2010.

[2] M. Koolen, J. Kamps, and G. Kazai. Social book search: Comparing topical relevance judgements and book suggestions for evaluation. In *Proceedings of the 21st ACM Conference on Information and Knowledge Management (CIKM 2012)*. ACM Press, New York NY, 2012.

Exploiting Social Networks in Recommendation: a Multi-Domain Comparison

Alejandro Bellogín[a,b], Iván Cantador[a], Pablo Castells[a], Fernando Díez[a]
[a]Information Retrieval Group, Department of Computer Science, Universidad Autónoma de Madrid
[b]Information Access, Centrum Wiskunde & Informatica
{alejandro.bellogin, ivan.cantador, pablo.castells, fernando.diez }@uam.es[a],
alejandro.bellogin@cwi.nl[b]

ABSTRACT

Recommender Systems aim at automatically finding the most useful products or services for a particular user, providing a personalised list of items according to different input and attributes of users and items. State-of-the-art recommender systems are usually based on ratings and implicit feedback given by users about the items. Recently, due to the large number of social systems appearing in the so called Web 2.0, where friendship relations between people are explicit, *social contexts* exploitation has started to receive significant interest. In particular, social recommenders have started to be investigated that exploit social links between users in a community to suggest interesting items. In this paper we compare a series of experiments developed in recent years with different datasets where standard collaborative and social filtering techniques were analysed. We show that social filtering techniques achieve very high performance in the three domains discussed (bookmarks, music, and movies), although they may have lower coverage than traditional collaborative filtering algorithms.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Experimentation, Performance

Keywords

Recommender systems, Social Networks, Evaluation

1. INTRODUCTION

With the advent of the Social Web, a variety of new recommendation approaches have been proposed in the literature [1]. Most of these approaches are based on the exploitation of social tagging information and explicit friendship relations between users (social filtering recommenders) [5, 8]. Commonly, algorithms dealing with social context attempt to exploit the social connections of an active user. For example, Shepitsen et al. [10] employs a personalisation algorithm for recommendation in folksonomies that relies on hierarchical tag clusters, which are used to recommend the most similar items to the user's closest cluster, by using the cosine similarity measure. Other works focus on graph-based techniques for finding the most relevant items for a particular user, inspired by algorithms from quite different areas, successfully bringing them to social recommendation [6].

In this paper, we compare the performance of social filtering methods with standard collaborative filtering (CF) baselines using four different datasets on three domains (bookmarks, music, and movies). With this goal in mind, in the next section we present the methods evaluated in this paper, then, in Section 3 we discuss the datasets used. After that, in Section 4 we present the results obtained.

DIR 2013, April 26, 2013, Delft, The Netherlands.

2. SOCIAL FILTERING RECOMMENDERS

Inspired by the approach presented in Liu & Lee [8], we analyse a pure social recommender that incorporates social information into the user-based CF model, named as **friends-based** (FB). Standard user-based CF typically computes predictions by performing a weighted sum over a set of similar users (usually called neighbours) as follows [1]: $s(u,i) = C\sum_{v\in N(u)} sim(u,v)r(v,i)$, where $r(v,i)$ denotes the rating given by user v to item i, and $sim(u,v)$ is the similarity between the two users. In this context, FB makes use of the same formula as the user-based CF technique, but replaces the set of nearest neighbours ($N(u)$) with the active user's (explicit) friends.

In [3] we propose a **social popularity** recommender (SocPop), where the algorithm suggests those items that are more popular among the set of the active user's friends. A third social recommender is evaluated where explicit distances between users in the social graph are integrated in the prediction formula: $s(u,i) = \sum_{v\in X(u,L)} K^{-d(u,v)}r(v,i)$. This approach was originally proposed in [5] and named as **personal-social** (PerSoc), where the authors use the Breadth-First Search algorithm in order to build a social tree for each user (denoted as $X(u,L)$), where L is the maximum number of levels taken into consideration in the algorithm, and K is an attenuation coefficient of the social network that determines the extent of the effect of distance $d(u,v)$ (we use Dijkstra's algorithm, $K=2$ and $L=6$).

Besides these pure social recommenders, hybrid social recommenders are useful not only for exploiting the social context of a user, but for providing higher coverage in extreme situations (such as the social or rating cold start, where no social context or ratings are available for a particular user). In this paper we analyse the performance of a combination between the friends-based method described above and the classic user-based CF method, where all the active user's friends along with the set of most similar nearest neighbours are used to produce recommendations. We name this method **user-and-friends-based** (UFB). Alternatively, more complex hybrid recommenders can be defined based on random walks [6] and linear combinations of the predictions from several recommenders [4], but we leave the comparison of these methods across several domains as future work (some initial insights can be found in [3]).

3. A MULTI-DOMAIN PERSPECTIVE

We report results using four different datasets on three domains. The first one was gathered from the social music website *Last.fm*. As described in [2], we built our dataset aiming to obtain a representative set of users, covering all music genres, and forming a dense social network. This dataset contains 1.9K users, 17.6K artists (17.0K of them tagged), 186.5K tag assignments (98.6 per user), and 25.4K friend relations (13.4 per user).

The second dataset was obtained from *Delicious*, a social bookmarking site for Web pages. Also described in [2], we built this dataset with the same goal in mind as the one stated for Last.fm dataset: to cover a broad range of document's topics, and obtain a dense social network. In this case, the dataset contains 1.9K users, 69.2K bookmarked Web pages, 437.6K tag assignments, and 15.3K friend relations. On average, each user profile has 56.1 bookmarks, 234.4 tag assignments, and 8.2 friends.

The third dataset used was provided in the social track of the CAMRa Challenge [9]. This dataset was gathered by the Filmtipset community, and contains social links between users, movie ratings, movie comments, and other attributes of users and movies. However, in such dataset every test user has a social network, which is not a realistic scenario, since in many social media applications such as Delicious or Last.fm the social network coverage is only partial. Because of this, we create a fourth dataset where we incorporate a number of users with no friends in the new test set used in our experiments, more specifically, such number corresponds to the number of test users contained in the original test set (439 users). We denote the former dataset as CAMRa-Social (CAMRa-S) and the latter as CAMRa-Collaborative (CAMRa-C).

4. PERFORMANCE COMPARISON

Table 1 shows the performance results of the four social filtering recommenders presented before on the four datasets already described. We also use a standard user-based CF method with 15 neighbours and Pearson's similarity [1] (UB) and a matrix factorisation approach in which the rating matrix is factorised into 50 dimensions [7] (MF) as baselines.

We observe that the best performing approach is the PerSoc strategy, which adapts the well-known CF formula by weighting the similarity between the user's and her neighbours' rating-based profiles with the users' distances in the social graph. These results thus provide empiric evidence that combining CF and social networking information produces better recommendations than CF alone. Very interestingly, the FB strategy, which recommends items liked by explicit friends, obtains acceptable precision values. As concluded by Konstas and colleagues [6] for Last.fm, recommendations generated from the users' social networks represent a good alternative to rating-based methods; here, we extend such conclusion to other domains like bookmarks and movies. Merging this strategy with CF (UFB), nonetheless, does not improve the results obtained by the approaches separately except in the movie domain, where the CF algorithm shows better performance than in the other contexts.

Table 1: Obtained performance values for different datasets (reported metric is P@10). Best value for each dataset in bold.

Method	Last.fm	Delicious	CAMRa-S	CAMRa-C
UB	0.009	0.008	0.072	0.052
MF	0.025	0.003	0.038	0.026
FB	0.043	0.023	0.057	0.050
SocPop	0.021	0.011	0.001	0.001
PerSoc	**0.085**	**0.054**	**0.344**	**0.342**
UFB	0.014	0.008	0.077	0.053

Additionally, when considering alternative evaluation metrics, we found in [2] that social filtering methods have lower coverage and novelty than traditional CF and content-based recommenders; however, their diversity is higher, as measured using α-nDCG. These negative aspects could be improved by building hybrid recommenders, where the performance accuracy is slightly degraded at the expenses of better coverage and novelty [3, 2].

5. ACKNOWLEDGMENTS

This work is supported by the Spanish Government (TIN2011-28538-C02-01) and the Government of Madrid (S2009TIC-1542).

6. REFERENCES

[1] Adomavicius, G., and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng. 17*, 6 (2005), 734–749.

[2] Bellogín, A., Cantador, I., and Castells, P. A comparative study of heterogeneous item recommendations in social systems. *Inf. Sci. 221* (2013), 142–169.

[3] Bellogín, A., Cantador, I., Díez, F., Castells, P., and Chavarriaga, E. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM TIST 4*, 1 (2013), 14.

[4] Bellogín, A., Castells, P., and Cantador, I. Self-adjusting hybrid recommenders based on social network analysis. In *SIGIR* (2011), W.-Y. Ma, J.-Y. Nie, R. A. Baeza-Yates, T.-S. Chua, and W. B. Croft, Eds., ACM, pp. 1147–1148.

[5] Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisels, A., Shani, G., and Naamani, L. Recommender system from personal social networks. In *AWIC* (2007), K. Wegrzyn-Wolska and P. S. Szczepaniak, Eds., vol. 43 of *Advances in Soft Computing*, Springer, pp. 47–55.

[6] Konstas, I., Stathopoulos, V., and Jose, J. M. On social networks and collaborative recommendation. In *SIGIR* (2009), J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, and J. Zobel, Eds., ACM, pp. 195–202.

[7] Koren, Y., Bell, R. M., and Volinsky, C. Matrix factorization techniques for recommender systems. *IEEE Computer 42*, 8 (2009), 30–37.

[8] Liu, F., and Lee, H. J. Use of social network information to enhance collaborative filtering performance. *Expert Syst. Appl. 37*, 7 (2010), 4772–4778.

[9] Said, A., Berkovsky, S., and Luca, E. W. D. Introduction to special section on camra2010: Movie recommendation in context. *ACM TIST 4*, 1 (2013), 13.

[10] Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R. D. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys* (2008), P. Pu, D. G. Bridge, B. Mobasher, and F. Ricci, Eds., ACM, pp. 259–266.

How Much Data Resides in a Web Collection: How to Estimate Size of a Web Collection

Mohammadreza Khelghati, Djoerd Hiemstra, Maurice Van Keulen

1. INTRODUCTION

With increasing amount of data in deep web sources (hidden from general search engines behind web forms), accessing this data has gained more attention. In the algorithms applied for this purpose, it is the knowledge of a data source size that enables the algorithms to make accurate decisions in stopping crawling or sampling processes which can be so costly in some cases [4]. The tendency to know the sizes of data sources is increased by the competition among businesses on the Web in which the data coverage is critical. In the context of quality assessment of search engines [2], search engine selection in the federated search engines, and in the resource/collection selection in the distributed search field [6], this information is also helpful. In addition, it can give an insight over some useful statistics for public sectors like governments. In any of these mentioned scenarios, in case of facing a non-cooperative collection which does not publish its information, the size has to be estimated [5]. In this paper, the approaches in literature are categorized and reviewed. The most recent approaches are implemented and compared in a real environment. Finally, four methods based on the modification of the available techniques are introduced and evaluated. In one of the modifications, the estimations from other approaches could be improved ranging from 35 to 65 percent.

Contributions. As the first contribution, an experimental comparison among a number of size estimation approaches is performed. Having applied these techniques on a number of real search engines, it is shown which technique can provide more promising results. As the second contribution, a number of modifications to the available approaches are suggested (Table 1 [3]).

2. THE SUGGESTED APPROACH

In this work, Heterogeneous and Ranked Model (Mhr), Multiple Capture Recapture (MCR), MCR Regression, Capture History (CH), CH Regression, Generalized Capture Recapture (G-MCR) and Bar-Yossef et al. approaches from the literature are implemented. Having studied these approaches, a number of ideas are suggested to improve their accuracy.

In the approaches like MCR and CH which are based on creating samples and the number of duplicates among them, the idea of considering only the different samples is applied. This can test if different samples can provide more information on the collection size. The similarity of samples is considered as the basic modification idea for MCR and CH.

Different nature of Bar-Yossef et al. needs a different improvement idea. Bar-Yossef et al. is based on a predefined query pool. The number of queries in this pool which cover the collection data is estimated and this number directly affects the collection size estimation. In our experiments over Bar Yossef et al., it was noticed that defining the query pool can highly affect the estimation process. Based on this observation, a different query pool selection method is suggested. In this suggested approach, queries are divided into different query pools based on their frequencies. These pools are indexed and easily accessible by the approach. By sending queries and investigating their results, it is decided if the pool is appropriate or not for the collection. This helps choosing the most appropriate query pool for the collection.

3. RESULTS

Having applied the Mhr, MCR, MCR-Regression, CH, CH-Regression and G-MCR approaches on the test set, the results are illustrated in the Figure 1 [3]. These websites are chosen in a way to cover different subjects and have different sizes. In this figure, to be able to compare the performance of the approaches on different data collections of different sizes, the results are normalized by using the Relative Bias metric. If an approach could estimate half of the actual size of a data collection, the corresponding relative bias for that approach is -0.5 which is related to -50 percent in the figure.

However, it is important to mention that the Bar-Yossef et al. approach implemented in this work was so costly in most of the cases that caused stopping the estimation process. This problem is introduced by the choices of the query pools made during the implementation phase of this approach. Among two pools suggested by Bar-Yossef et al. [1], the one aimed at real cases and not designed for training purposes is implemented. Therefore, the results for Bar-Yossef et al. approach are missing in this part.

DIR 2013, April 26, 2013, Delft, The Netherlands.

Table 1: Improvements Resulting From Modifications

	Mhr	MCR	MCR-Reg	CH	CH-Reg	G-MCR
M-Bar-Yossef	36.25	63.67	67.36	44.74	54.70	62.77
M-MCR	-19.1	8.27	11.96	-10.6	-0.7	7.37
M-MCR-Reg	-24.1	3.25	6.94	-15.6	-5.7	2.34
M-CH-1	1.35	28.77	32.46	9.84	19.79	27.86
M-CH-1-Reg	2.50	29.92	33.60	10.98	20.94	29.01
M-CH-2	0.81	28.23	31.92	9.30	19.26	27.33
M-CH-2-Reg	2.77	30.19	33.87	11.25	21.21	29.28

Note: This table provides the percentage of improvements that the modified approaches could result regarding the previously available approaches; considering the average of all the performances on all the tested real data collections on the Web.

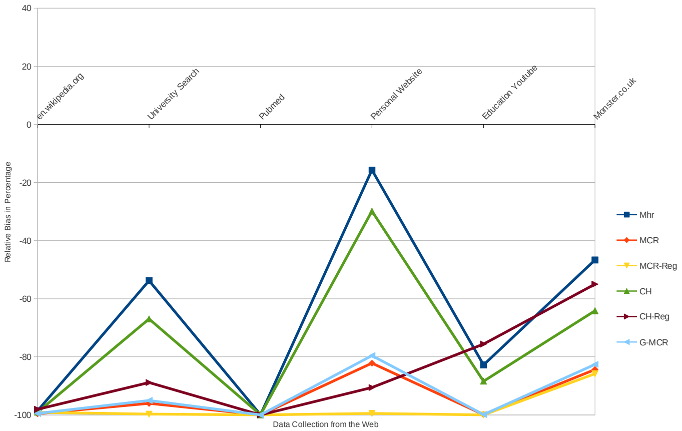


Figure 1: The Performance of the Approaches on the Real Data Collections from the Web

Note: The lines are added only to provide more readability of the graph.

4. CONCLUSION

Having studied the state-of-the-art in size estimation of non-cooperative websites, the most recent approaches introduced in the literature are implemented in this work. Hence, the MCR, CH, G-MCR, Bar Yossef et al. and regression-based approaches are selected to be studied and compared. To provide an appropriate comparison setting, two issues were regarded highly important. First, the test collection is definrd as a set of websites on the Web from different domains (such as job vacancies, wikis, articles, and personal websites) with different sizes. The second issue was the information available for each approach. The number of sampling events and the samples sizes were set to be the same for all the approaches. Although this test environment could be improved by adding more real deep websites, it is believed that it could provide an appropriate basis for comparing the available size estimation approaches.

Among all the studied approaches, the modified version of Bar-Yossef et al. could provide 35 to 65 percent better estimations on size of the tested deep websites. However, the M-Bar-Yossef et al. approach could not be implemented for the websites which do not provide the access to the content of the search results. In the case of facing such websites, the Mhr approach, both modified versions of the CH approach (M-CH-1 and M-CH-2) and their regressions (M-CH-1-Regression and M-CH-2-Regression) could be among the options to be applied. These approaches had close estimations considering the average performances on all the tested websites.

As future work, we aim at research on the most appropriate time to stop the sampling in estimation process. The alternative approaches could be continuing as far as the limitations or to study questions like what is the adequate number of samples and the most appropriate sample size to provide the most accurate estimation. As another future work, the potential further improvements could be mentioned. As an example, in the selection of pools in the M-Bar-Yossef et al. approach, the selection procedure could be based on the queries from different domains. This classification might lead to higher accuracy of the size estimations.

5. ACKNOWLEDGEMENT

This publication was supported by the Dutch national program COMMIT.

6. REFERENCES

[1] Bar-Yossef, Z., and Gurevich, M. Efficient search engine measurements. *ACM Trans. Web 5*, 4 (Oct. 2011), 18:1–18:48.

[2] Broder, A. Z., Fontoura, M., Josifovski, V., Kumar, R., Motwani, R., Nabar, S. U., Panigrahy, R., Tomkins, A., and Xu, Y. Estimating corpus size via queries. In *CIKM* (2006), pp. 594–603.

[3] Khelghati, M., Hiemstra, D., and van Keulen, M. Size estimation of non-cooperative data collections. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications &#38; Services* (New York, NY, USA, 2012), IIWAS '12, ACM, pp. 239–246.

[4] Lu, J. Ranking bias in deep web size estimation using capture recapture method. *Data Knowl. Eng. 69*, 8 (Aug. 2010), 866–879.

[5] Shokouhi, M., Zobel, J., Scholer, F., and Tahaghoghi, S. M. M. Capturing collection size for distributed non-cooperative retrieval. In *SIGIR* (2006), pp. 316–323.

[6] Xu, J., Wu, S., and Li, X. Estimating collection size with logistic regression. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 2007), SIGIR '07, ACM, pp. 789–790.

Using Intent Information to Model User Behavior in Diversified Search (Abstract)*

Aleksandr Chuklin[1,2] Pavel Serdyukov[1] Maarten de Rijke[2]

[1] Yandex, Moscow, Russia
[2] ISLA, University of Amsterdam, The Netherlands

chuklin@yandex-team.ru, pavser@yandex-team.ru, derijke@uva.nl

ABSTRACT

A result page of a modern commercial search engine often contains documents of different types targeted to satisfy different user intents (news, blogs, multimedia). When evaluating system performance and making design decisions we need to better understand user behavior on such result pages. To address this problem various click models have previously been proposed. In this paper we focus on result pages containing fresh results and propose a way to model user intent distribution and bias due to different document presentation types. To the best of our knowledge this is the first work that successfully uses intent and layout information to improve existing click models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models

General Terms

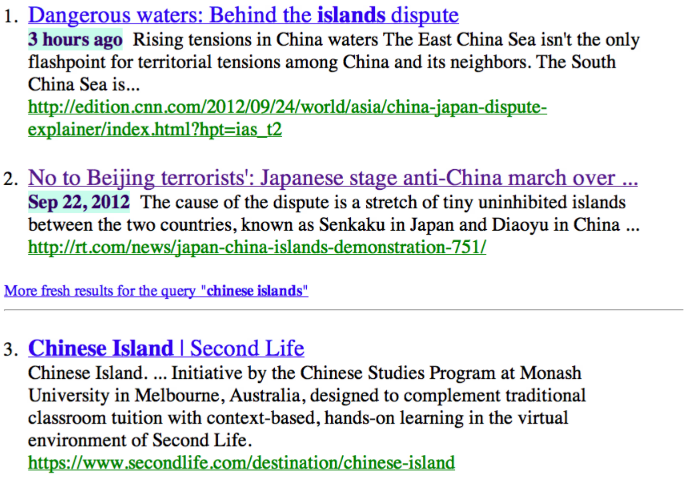
Algorithms, Experiment, Theory

Keywords

Click models, Diversity, User Behavior

1. INTRODUCTION

The idea of search result diversification appeared several years ago in the work by Radlinski and Dumais [8]. Since then all major commercial search engines addressed the problem of ambiguous queries either by the technique called *federated* / *vertical* search (see, e.g., [2]) or by making result diversification a part of the ranking process [1, 9]. In this work we focus on one particular vertical: *fresh* results, i.e., recently published webpages (news, blogs, etc.). Fig. 1 shows part of a search engine result page (SERP) in which fresh results are mixed with ordinary results in response to the query "Chinese islands". We say that every document has a *presentation type*, in our example "fresh" (the first two documents in the figure) or "web" (the third, ordinary search result item). We will further refer to the list of presentation types for the current result page as a *layout*. We assume that each query has a number of *categories* or *intents* associated with it. In our case these will be "fresh" and "web".



1. Dangerous waters: Behind the **islands** dispute
3 hours ago Rising tensions in China waters The East China Sea isn't the only flashpoint for territorial tensions among China and its neighbors. The South China Sea is...
http://edition.cnn.com/2012/09/24/world/asia/china-japan-dispute-explainer/index.html?hpt=ias_t2

2. No to Beijing terrorists': Japanese stage anti-China march over ...
Sep 22, 2012 The cause of the dispute is a stretch of tiny uninhibited islands between the two countries, known as Senkaku in Japan and Diaoyu in China ...
http://rt.com/news/japan-china-islands-demonstration-751/

More fresh results for the query "**chinese islands**"

3. **Chinese Island** | Second Life
Chinese Island. ... Initiative by the Chinese Studies Program at Monash University in Melbourne, Australia, designed to complement traditional classroom tuition with context-based, hands-on learning in the virtual environment of Second Life.
https://www.secondlife.com/destination/chinese-island

Figure 1: Group of fresh results at the top followed by an ordinary search result item.

The main problem that we address in this paper is the problem of modeling user behavior in the presence of vertical results. In order to better understand user behavior in a multi-intent environment we propose to exploit intent and layout information in a click model so as to improve its performance. Unlike previous click models our proposed model uses additional information that is already available to search engines. We assume that the system already knows the probability distribution of intents / categories corresponding to the query. This is a typical setup for the TREC diversity track as well as for commercial search systems. We also know the presentation type of each document. We argue that this presentation may lead to some sort of bias in user behavior and taking it into account may improve the click model's performance.

2. CLICK MODELS

Click data has always been an important source of information for web search engines. It is an *implicit* signal because we do not always understand how user behavior correlates with user satisfaction: user's clicks are biased. Following Joachims et al. [7], who conducted eye-tracking experiments, there was a series of papers that model user behavior using probabilistic graphical models. The most influential works in this area include the UBM model by Dupret and Piwowarski [6], the Cascade Model by Craswell et al. [5] and the DBN model by Chapelle and Zhang [3].

A *click model* can be described as follows. When a user submits a query q to a search engine she gets back 10 results: $u_1, \ldots,$

*The full version of this paper appears in *ECIR 2013* [4].

DIR'13, Delft, The Netherlands.

u_{10}. Given a query q we denote a *session* to be a set of events experienced by the user since issuing the query until abandoning the result page or issuing another query. Note that one session corresponds to exactly one query. The minimal set of random variables used in all models to describe user behavior are: *examination* of the k-th document (E_k) and *click* on the k-th document (C_k):

- E_k indicates whether the user looked at the document at rank k (hidden variables).
- C_k indicates whether the user clicked on the k-th document (observed variables).

In order to define a click model we need to denote dependencies between these variables. For example, for the UBM model we define

$$P(E_k = 1 \mid C_1, \ldots, C_{k-1}) = \gamma_{kd} \quad (1)$$

$$E_k = 0 \Rightarrow C_k = 0 \quad (2)$$

$$P(C_k = 1 \mid E_k = 1) = a_{u_k}, \quad (3)$$

where γ_{kd} is a function of two integer parameters: the current position k and the distance to the rank of previous click $d = k - PrevClick = k - \max\{j \mid 0 \leq j < k \;\&\; C_j = 1\}$ (we assume $C_0 = 1$). Furthermore, a_{u_k} is a variable responsible for the attractiveness of the document u_k for the query q. If we know the a and γ parameters, we can predict click events. The better we predict clicks the better the click model is.

We propose a modification to existing click models that exploits information about user intent and the result page layout. As a basic model to modify we use the UBM click model by Dupret and Piwowarski [6]. However, our extensions can equally well be applied to other click models. We focus on HTML results that look very similar to the standard 10 blue links. We do not know beforehand that the user notices any differences between special (vertical) results and ordinary ones.

We add one hidden variable I and a set of observed variables $\{G_k\}$ to the two sets of variables $\{E_k\}$ and $\{C_k\}$ commonly used in click models:

- $I = i$ indicates that the user performing the session has *intent* i, i.e., relevance with respect to the category i is much more important for the user.
- $G_k = l$ indicates that the result at position k uses a presentation specific to the results with dominating intent l. For example, for the result page shown in Fig. 1 we have $G_1 = \mathit{fresh}$, $G_2 = \mathit{fresh}$, $G_3 = \mathit{web}$. We will further refer to a list of presentation types $\{G_1, \ldots, G_{10}\}$ for a current session as a *layout*.

A typical user scenario can be described as follows. First, the user looks at the whole result page and decides whether to examine the k-th document or not. We assume that the examination probability $P(E_k)$ does not depend on the document itself, but depends on the user intent, her previous interaction with other results, the document rank k and the SERP layout. If she decides to examine the document (if $E_k = 1$) we assume that she is *focused* on that particular document. It implies that the probability of the click $P(C_k = 1|E_k = 1)$ depends only on the user intent I and the document relevance / attractiveness of the current document, but neither on the layout nor on the document position k. After clicking (or not clicking) the document the user moves to another document following the same "examine-then-click" scenario.

3. RESULTS

We used the UBM model as our baseline and ran experiments in order to answer the following research questions:

- How do intent and layout information help in building click models? How does the performance change when we use only one type of information or both of them?
- How does the best variation of our model compare to other existing click models?

The main contribution of our work is a framework of intent-aware click models, which incorporates both layout and intent information. Our intent-aware modification can be applied to any click model to improve its perplexity. One interesting feature of an intent aware click model is that it allows us to infer separate relevances for different intents from clicks. These relevances can be further used as features for specific vertical ranking formulas. Another important property of intent-aware additions to click models is that by analyzing examination probabilities we can see how user patience depends on his/her intent and the search engine result page layout. Put differently, it allows us to use a click model as an ad-hoc analytic tool.

As to future work, we see a number of directions, especially concerning specific verticals in order to check that our method is also applicable to other verticals/intents. For instance, the mobile arena provides interesting research opportunities.

Sometimes, intents are very unique, like for instance for the query "jaguar" there are at least two intents: finding information about *cars* and finding information about *animals*. It is very unlikely that a search engine has a special vertical for these intents. However, we believe that knowledge of the user's intent can still be used in order to better understand his/her behavior. Applying our ideas to these minor intents is an interesting direction for future work.

4. REFERENCES

[1] Agrawal, R., Gollapudi, S., Halverson, A., Ieong, S.: Diversifying search results. In: WSDM. p. 5. ACM (2009)

[2] Arguello, J., Diaz, F., Callan, J., Crespo, J.: Sources of evidence for vertical selection. In: SIGIR. pp. 315–322. ACM (2009)

[3] Chapelle, O., Zhang, Y.: A dynamic bayesian network click model for web search ranking. In: WWW. ACM (2009)

[4] Chuklin, A., Seryukov, P., de Rijke, M.: Using Intent Information to Model User Behavior in Diversified Search. In: ECIR. Springer (2013)

[5] Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: WSDM. p. 87. ACM (2008)

[6] Dupret, G., Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In: SIGIR. pp. 331–338. SIGIR '08, ACM (2008)

[7] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: SIGIR. p. 154. ACM (2005)

[8] Radlinski, F., Dumais, S.: Improving personalized web search using result diversification. In: SIGIR. ACM (2006)

[9] Styskin, A., Romanenko, F., Vorobyev, F., Serdyukov, P.: Recency ranking by diversification of result set. In: CIKM. pp. 1949–1952. ACM (2011)

Using Wikipedia's Category Structure for Entity Search

Rianne Kaptein[1] Jaap Kamps[2]

[1] TNO, Delft, The Netherlands*
[2] University of Amsterdam, Amsterdam, The Netherlands

ABSTRACT

In this paper we investigate how the category structure of Wikipedia can be exploited for Entity Ranking. In the last decade, the Web has not only grown in size, but also changed its character, due to collaborative content creation and an increasing amount of structure. Current Search Engines find Web pages rather than information or knowledge, and leave it to the searchers to locate the sought information within the Web page. A considerable fraction of Web searches contains named entities. We focus on how the Wikipedia structure can help rank relevant entities directly in response to a search request, rather than retrieve an unorganized list of Web pages with relevant but also potentially redundant information about these entities. Our results demonstrate the benefits of using topical and link structure over the use of shallow statistics. This paper is a compressed version of [1].

1. INTRODUCTION

Searchers looking for entities are better served by presenting a ranked list of entities directly, rather than an unorganized list of Web pages with relevant but also potentially redundant information about these entities. The goal of the entity ranking task is to return entities instead of documents or text as are returned for most common search tasks. Entities can be for example persons, organizations, books, or movies.

A resource that is large enough to generate meaningful statistics, and contains interpretable semantic structure is Wikipedia. The nature and structure of Wikipedia presents new opportunities to solve problems that were thought to require deep understanding capabilities and where bottlenecks such as high cost and scalability where applicable in the past. Combining the benefits of the structured information and the large scale of Wikipedia, creating the opportunity to use probabilistic methods, we can now efficiently process all of the information contained in Wikipedia.

In this paper is motivated by the following main research question: *How can we exploit the structure of Wikipedia to retrieve entities?* We start by looking at how we can retrieve entities inside Wikipedia, which is also the task in the INEX entity ranking track. INEX[1](Initiative for the Evaluation of XML retrieval) is an information retrieval evaluation forum that provides an IR test collection to evaluate the task of entity ranking using Wikipedia as its document collection. Our first research question is: *How can we exploit category and link information for entity ranking in Wikipedia?*

Since a requirement for a relevant result in entity ranking is to retrieve the correct entity type, category information is of great importance for entity ranking. Category information can also be regarded in a more general fashion, as extra context for your query, which could be exploited for ad hoc retrieval. Our second research question is therefore: *How can we use entity ranking techniques that use category information for ad hoc retrieval?*

Since usually ad hoc queries do not have target categories assigned to them, and providing target categories for entity ranking is an extra burden for users, we also examine ways to assign target categories to queries. Our third research question is: *How can we automatically assign target categories to ad hoc and entity ranking queries?*

2. RETRIEVAL MODEL

In this section we describe our retrieval model, how we use category information for entity ranking, how we combine these sources of information, and how we assign categories to query topics automatically.

Exploiting Category Information Although for each entity ranking topic one or a few target categories are provided, relevant entities are not necessarily associated with these provided target categories. Relevant entities can also be associated with descendants of the target category or other similar categories. Therefore, simply filtering on the target categories is not sufficient. multiple categories, not all categories of an answer entity will be similar to the target category. We calculate for each target category the distances to the categories assigned to the answer entity. To calculate the distance between two categories, we tried three options. The first option (*binary distance*) is a very simple method: the distance is 0 if two categories are the same, and 1 otherwise. The second option (*contents distance*) calculates distances according to the contents of each category, and the third option (*title distance*) calculates a distance according to the category titles. We use KL-divergence to calculate distances between categories, and calculate a category score that is high when the distance is small.

Combining information Finally, we have to combine our different sources of information. Our first source of information is a standard language model for retrieval, which calculates the probabilities of occurrence of the query terms

*Work done while at the University of Amsterdam.

[1] `https://inex.mmci.uni-saarland.de/`

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

Table 1: 2007 ER Topics using Category Information

Category representation	Weight	MAP	P10
Baseline		0.1840	0.1920
Binary	0.1	0.2145^{-}	0.1880^{-}
Contents	0.1	0.2481°	0.2320°
Title	0.1	0.2509°	0.2360°
Contents	0.05	**0.2618**$^{\circ}$	**0.2480**$^{\circ}$
Title	0.05		

Table 2: Ad Hoc vs. Entity Ranking results in MAP

	Query	Category	Combi.	Best Score	
Set (M/A)	$\mu = 0.0$	$\mu = 1.0$	$\mu = 0.1$	μ	
ER07a M	0.2804	0.2547^{-}	0.3848$^{\bullet}$	0.2	0.4039$^{\bullet}$
ER07a A	0.2804	0.2671^{-}	0.3607°	0.1	0.3607°
ER07b M	0.1840	0.1231^{-}	0.2481°	0.1	0.2481°
ER07b A	0.1840	0.1779^{-}	0.2308°	0.2	0.2221°
AH07a M	0.3653	0.2067°	0.4308°	0.1	0.4308°
AH07b M	0.3031	0.1761$^{\bullet}$	0.3297°	0.05	0.3327$^{\bullet}$

in a document. This standard language model also serves as our baseline retrieval model. We explore two possibilities to combine information. First, we make a linear combination of the document, link and category score. All scores and probabilities are calculated in the log space, and then a weighted addition is made.

Alternatively, we can use a two step model. Relevance propagation takes as input initial probabilities as calculated by the baseline document model score. Instead of the baseline probability, we can use the scores of the run that combines the baseline score with the category information.

Target Category Assignment Besides using the target categories provided with the entity ranking query topics, we also look at the possibility of automatically assigning target categories to entity ranking and ad hoc topics. From our baseline run we take the top N results, and look at the T most frequently occurring categories belonging to these documents, while requiring categories to occur at least twice. These categories are assigned as target categories to the query topic.

3. EXPERIMENTS

In this section we describe our experiments with entity ranking and ad hoc retrieval in Wikipedia.

Experimental Set-up We experiment with two different tasks. First of all we experiment with the entity ranking task as defined by INEX. We will make runs on the topic sets from 2007 to 2009. Secondly, we experiment with ad hoc retrieval using category information on the ad hoc topic sets from 2007 and compare automatic and manual category assignment for ad hoc and entity ranking topics.

Entity Ranking Results The results on the 2007 entity ranking topic set (ER07b, 19 topics) are summarized in Table 1. The weight of the baseline score is 1.0 minus the weight of the category information. For all three distances, a weight of 0.1 gives the best results. In addition to these combinations, we also made a run that combines the original score, the contents distance and the title distance. When a single distance is used, the title distance gives the best results. The combination of contents and title distance gives the best results overall. For the 2008 and 2009 entity ranking topic sets (not shown here), also significant improvements are achieved when category information is used. Additional improvements to the approach are to rerank the top 2500 documents from the baseline retrieval run, instead of the top 500, which have been reranked for the 2007 runs. Normalizing the scores before combining shows improvements for the 2009 topics.

Ad Hoc Retrieval Results A selection of 19 topics in the ad hoc topic set (AH07a) was transformed into an additional entity ranking topics (set ER07a). There are 80 more judged ad hoc topics (set AH07b). Results for 2007 entity ranking and ad hoc topics expressed in MAP are summarized in Table 2, where "M" stands for manually assigned categories, and "A" for automatically assigned categories.

From the four topic sets, the baseline scores of the ad hoc topic sets are higher. There is quite a big difference between the two entity ranking topic sets, where the topics derived from the ad hoc topics are easier than the genuine entity ranking topics. The entity ranking topics benefit greatly from using the category information with significant MAP increases of 44% and 35% for topic sets ER07a and ER07b respectively. When we use the category information for the ad hoc topics with manually assigned categories improvements are smaller than the improvements on the entity ranking topics, but still significant. Comparing manual and automatic assignments of target categories, manually assigned target categories perform somewhat better than the automatically assigned categories. However, for both topic sets using the automatically assigned categories leads to significant improvements over the baseline.

4. CONCLUSION

In this paper we have experimented with retrieving entities from Wikipedia exploiting its category structure. First, we examined whether Wikipedia category and link structure can be used to retrieve entities inside Wikipedia as is the goal of the INEX Entity Ranking task. Category information proves to be a highly effective source of information, leading to large and significant improvements in retrieval performance on all data sets. Secondly, we studied how we can use category information to retrieve documents for ad hoc retrieval topics in Wikipedia. Considering retrieval performance, also on ad hoc retrieval topics we achieved significantly better results by exploiting the category information. Finally, we examined whether we can automatically assign target categories to ad hoc and entity ranking queries. Guessed categories lead to performance improvements that are not as large as when the categories are assigned manually, but they are still significant. Our main conclusion is that the category structure of Wikipedia can be effectively exploited, in fact not only for entity ranking, but also for ad hoc retrieval, and with manually assigned as well as automatically assigned target categories.

Thanks This research was funded by NWO (# 612.066.513).

REFERENCES

[1] R. Kaptein and J. Kamps. Exploiting the Category Structure of Wikipedia for Entity Ranking, In *Artificial Intelligence*, Volume 194, January 2013, Pages 111-129.

Feeding the Second Screen: Semantic Linking based on Subtitles (Abstract) *

Daan Odijk
d.odijk@uva.nl

Edgar Meij
edgar.meij@uva.nl

Maarten de Rijke
derijke@uva.nl

ISLA, University of Amsterdam
Amsterdam

ABSTRACT

Television broadcasts are increasingly consumed on an interactive device or with such a device in the vicinity. Around 70% of tablet and smartphone owners use their devices while watching television [11]. This allows broadcasters to provide consumers with additional background information that they may bookmark for later consumption in applications such as depicted in Figure 1.

For live television, edited broadcast-specific content to be used on second screens is hard to prepare in advance. We present an approach for automatically generating links to background information in real-time, to be used on second screens. We base our semantic linking approach for television broadcasts on subtitles and Wikipedia, thereby effectively casting the task as one of identifying and generating links for elements in the stream of subtitles.

The process of automatically generating links to Wikipedia is commonly known as *semantic linking* and has received much attention in recent years [3, 6, 7, 9, 10]. Such links are typically explanatory, enriching the link source with definitions or background information [2, 4]. Recent work has considered semantic linking for short texts such as queries and microblogs [6–8]. The performance of generic methods for semantic linking deteriorates in such settings, as language usage is creative and context virtually absent.

While link generation has received considerable attention in recent years, our task has unique demands that require an approach that needs to (i) be high-precision oriented, (ii) perform in real-time, (iii) work in a streaming setting, and (iv) typically, with a very limited context.

We propose a learning to rerank approach to improve upon a strong baseline retrieval model for generating links from streaming text. In addition, we model context using a graph-based approach. This approach is particularly appropriate in our setting as it allows us to combine a number of context-based signals in streaming text and capture the core topics relevant for a broadcast, while allowing real-time updates to reflect the progression of topics being dealt with in the broadcast. Our graph-based context model is highly accurate, fast, allows us to disambiguate between candidate links and capture the context as it is being built up.

Our main contribution is a set of effective feature-based methods for performing real-time semantic linking. We show how a learning to rerank approach for semantic linking performs on the task of real-time semantic linking, in terms of effectiveness and efficiency. We extend this approach with a graph-based method to keep track of context in a textual stream and show how this can further improve effectiveness. By investigating the effectiveness and efficiency of individual features we provide insight in how to improve effectiveness while maintaining efficiency for this task. Additional contributions include a formulation of a new task: semantic linking of a textual stream, and the release of a dataset[1] for this new task, including ground truth.

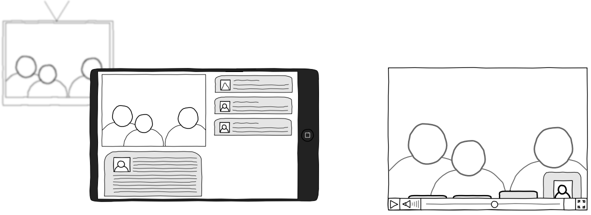


Figure 1: Sketches of a second screen (left) and an interactive video player (right) showing links to background information, synchronized with a television broadcast. Links pop up briefly when relevant and are available for bookmarking or exploring.

Real-Time Semantic Linking. Our approach to real-time semantic linking consists of a retrieval model that is based on how links between Wikipedia articles are created. Our method for real-time link generation consists of three steps: link candidate finding, ranking and reranking. In this retrieval model, each Wikipedia article is represented by the anchors that are used to link to it within Wikipedia. The first, recall-oriented step is aimed at finding as many link candidates as possible. Here, we produce a set of link candidates that each link to a Wikipedia article. To this end, we perform lexical matching in the subtitles of each constituent n-gram with the anchor texts found in Wikipedia.

The second step is to rank the link candidates in L. In particular, we can use statistics on the anchor text usage. We consider the prior probability that anchor text a links to Wikipedia article w:

$$COMMONNESS(a, w) = \frac{|L_{a,w}|}{\sum_{w' \in W} |L_{a,w'}|}, \tag{1}$$

where $L_{a,w}$ denotes the set of all links with anchor text a and target w. The intuition is that link candidates with anchors that always link to the same target are more likely to be a correct representation than those where anchor text is used more often to link to other targets. We consider these first two steps our baseline retrieval model.

The third step is aimed at improving precision using a learning to rerank approach, that was effective on similar tasks [5, 8, 10]. For link candidates many ranking criteria are in play, making learning to rerank particularly appropriate. We use a set of lightweight features (based on [8]), that can be computed online. These 26 features

*The full version of this paper will appear in OAIR 2013 [12].

[1] The dataset will be shared upon publication of [12]; it consists of subtitles for 50 video segments, with more than 1,500 manually annotated links.

DIR 2013, April 26, 2013, Delft, The Netherlands.

Table 1: Semantic linking results with classification time. Significant differences, tested using a two-tailed paired t-test, are indicated $^{\blacktriangle}$ ($p < 0.01$); the position indicates whether the comparison is against line 1 (left most) or line 2 (right most).

	Average classification time per line (in ms)	R-Prec	MAP
1. Baseline retrieval model	54	0.5753	0.6235
2. Learning to rerank approach	99	0.7177$^{\blacktriangle}$	0.7884$^{\blacktriangle}$
3. Learning to rerank + context	108	0.7454$^{\blacktriangle\blacktriangle}$	0.8219$^{\blacktriangle\blacktriangle}$

are organized in four groups based on their source: textual anchor, target Wikipedia article and anchor+target. This set includes simple textual features, link probability measures and visitor statistics for a Wikipedia article. The full set of features is listed in [12].

We use a decision tree based approach as it has outperformed Naive Bayes and Support Vector Machines on similar tasks [8, 10]. We choose Random Forests [1] as it is robust, efficient and easily parallelizable.

Modeling Context. Link generation methods that rely on an entire document are not suited for use in a streaming text context as such methods are computationally expensive. What we need, instead, is a method to model context that can be incrementally updated and allows for easily computing features for link candidates.

We model the context of a textual stream as an undirected graph. The graph reflects the content of the textual stream and encodes the structure. This results in a smaller distance for things mentioned together. Furthermore, nodes for Wikipedia articles that are mentioned more often, will have more anchors connecting to them, making them more central and thus more important in the graph.

To feed our learning to rerank approach with information from the context graph we compute a number of features for each link candidate. First, we compute the degree of the target Wikipedia article in this graph. To measure how closely connected a target is, we compute degree centrality. Finally, we measure the importance of a target by computing its PageRank [13].

Experimental evaluation. To measure the effectiveness and efficiency of our proposed approach to semantic linking, we use the subtitles of six episodes of a live daily talk show. The subtitles are generated during live broadcast by a professional and are intended for the hearing impaired. From these subtitles, video segments are identified, each covering a single item of the talk show. Our data set consists of 5,173 lines in 50 video segments, with 6.97 terms per line. The broadcast time of all video segments combined is 6 hours, 3 minutes and 41 seconds.

In order to train the supervised machine learning methods and evaluate the end result, we need to establish a gold standard. We have asked a trained human annotator to manually identify links that are relevant for a wide audience. A total of 1,596 links have been identified, 150 with a NIL target and 1,446 with a target Wikipedia article, linking to 897 unique articles, around 17.94 unique articles per video segment and 2.47 unique articles per minute.

Results and Discussion. An overview of the results is shown in Table 1. First, we consider the performance of our baseline retrieval model. Line 1 in Table 1 shows the scores for the ranking baseline. The recall oriented link candidate finding step produces 120,223 links with 42,265 target articles, including 771 known targets that are in the ground truth (a recall of 0.8595). With this many link candidates, there is a clear need for ranking. The ranking baseline achieves reasonable effectiveness scores; these numbers are comparable to the literature, while leaving room for improvement.

The results for our learning to rerank approach (Line 2) show that it can be highly effective and significantly improve over the retrieval baseline. We can achieve this high effectiveness at an average online classification time of less than 100 milliseconds, making the learning to rerank approach efficient and suited for usage in real time. The results for the learning to rerank runs with context features added are listed in line 3. Compared to the learning to rerank approach, we are able to achieve significantly higher performance.

Conclusion. Motivated by the rise in so-called second screen applications we introduced a new task: real-time semantic linking of streaming text. We have created a dataset for this task. We have shown that learning to rerank can be applied to significantly improve an already competitive retrieval baseline and that this can be done in real-time. Additionally, we have shown that by modeling context as a graph we can significantly improve the effectiveness of this learning to rerank approach. This graph-based method to keep track of context is especially well-suited for the streaming text, as we can incrementally update the context model.

Acknowledgments

This research was partially 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, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.-802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the BILAND project funded by the CLARIN-nl program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), and the Netherlands eScience Center under project number 027.012.105.

REFERENCES

[1] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[2] M. Bron, B. Huurnink, and M. de Rijke. Linking archives using document enrichment and term selection. In *TPDL '11*, pages 360–371. Springer, 2011.

[3] P. Ferragina and U. Scaiella. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *CIKM '10*, pages 1625–1628. ACM, 2010.

[4] J. He, M. de Rijke, M. Sevenster, R. van Ommering, and Y. Qian. Automatic link generation with Wikipedia: A case study in annotating radiology reports. In *CIKM '11*, pages 1867–1876. ACM, 2011.

[5] M. Larson, E. Newman, and G. Jones. Overview of VideoCLEF 2009. In *CLEF '09*, pages 354–368. Springer, 2010.

[6] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Learning semantic query suggestions. In *ISWC '09*, pages 424–440. Springer, 2009.

[7] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Mapping queries to the linking open data cloud: A case study using DBpedia. *J. Web Semantics*, 9(4):418–433, 2011.

[8] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM 2012*, pages 563–572. ACM, 2012.

[9] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM '07*, pages 233–242. ACM, 2007.

[10] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *CIKM '08*, pages 509–518. ACM, 2008.

[11] Nielsen. In the U.S., tablets are TV buddies while ereaders make great bedfellows, May 2012. `http://bit.ly/L41f9E` [Online; accessed May 2012].

[12] D. Odijk, E. Meij, and M. de Rijke. Feeding the Second Screen: Semantic Linking based on Subtitles. In *Open research Areas in Information Retrieval (OAIR 2013)*, Lisbon, Portugal, 2013.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Summarization and Expansion of Search Facets[*]

Aparna Nurani Venkitasubramanian Marie-Francine Moens
Department of Computer Science
Katholieke Universiteit Leuven
Leuven, Belgium
{aparna.nuranivenkitasubramanian,sien.moens}@cs.kuleuven.be

ABSTRACT

We present a novel method for summarization and expansion of search facets. To dynamically extract key facets, the ranked list of search results generated from a keyword search is coupled with the spatial distribution of relevant documents in a hierarchical taxonomy of subject classes. An evaluation of the method based on the relevance and diversity of the produced facets indicates its effectiveness for both summarization and expansion.

Keywords

Selection of search facets, Expansion, Summarization

1. INTRODUCTION

The combination of a 'keyword' and a 'faceted' search has the potential to enhance user experience by providing a better arrangement of search results and aiding further search exploration. However, such a framework poses two key problems: 1) a given query may cover several facets, requiring an aggregation or summarization of the most relevant ones; and 2) a query may cover too few facets necessitating an expansion to include additional facets. We exploit the spatial distribution of topics relevant to a query in a hierarchy together with the relevance ranking of the documents for the query, in order to select search facets that optimize diversity and relevance.

2. SELECTING SEARCH FACETS

We assume that the search results of a query are annotated with subject classes (here *facets* or *nodes*) obtained from a hierarchical taxonomy. In the experiments below, the DMOZ[*] hierarchy is used. For each query, we define: a set of *activated nodes* that have documents relevant to the query and a set of *presentation nodes* that will be presented to the user as facets relevant for the query.

When the user presents a query, the DMOZ facets associated with the result of the query are first extracted, i.e., the *activated nodes* are identified. Next, if the number of *activated* facets associated with the query is larger than k[†], the set of *activated* nodes or facets is summarized by picking the best k candidates. If the number of *activated* facets is less than k, then the set is expanded by adding related facets. The summarization and expansion are carried out using the 'Subtree density' model (Section 3) which takes as input a set of *activated* nodes and produces the *presentation* nodes. For some queries, DMOZ activates not only the lowest level facets, but also some of their ancestors. In such a case to ensure presentation of as many distinct facets as possible, the summarization uses only the descendants, while the expansion uses only the ancestors.

3. SUBTREE DENSITY MODEL

This model finds nodes which represent dense clusters of facets, each having many search results important for the query. First, the subtrees associated with the relevant set of activated nodes are extracted. The subtree $\mathbf{S}$ for a node v comprises the node and its descendants (children, grand children etc. until the last level).

Then, one possible candidate to represent a subtree is the medoid identified as the node with the minimum average distance to all the other nodes of the subtree. The distances between nodes in the subtree are computed using a distance metric that captures semantic distances between topics in a hierarchy. Since the basic relations in the taxonomy are the parent-child relations, distance between any two nodes is represented using the connection weights between the parent-child pairs associated. In taxonomy $\mathbf{T}$ with root at level 0, the connection weight D between node v_i at level l and its child v_j at level $l+1$ is as follows:

$$D(v_i, v_j) = 2^{-l} \tag{1}$$

Using this metric, the distance between any two nodes v_m and v_n in $\mathbf{T}$ is defined as the sum of connection weights between all nodes v_x spanning the path between v_m and v_n.

Once the medoids of the subtree have been identified, we must rank them to identify the best k medoids that will be presented. This is done using a score computed in Eq. 2

$$score(m) = \frac{density(\mathbf{S})}{distance(m, \mathbf{S})} \tag{2}$$

where $density(\mathbf{S})$ is given by

$$density(\mathbf{S}) = \frac{\sum_{v \in S} importance(v, \mathbf{R})}{|\mathbf{S}|} \tag{3}$$

where $|\mathbf{S}|$ is the size of the subtree in terms of number of nodes $v \in \mathbf{S}$ and $importance(v, \mathbf{R})$ is computed using the Discounted Cumulative Gain (DCG) [2] over the retrieved Web pages assigned to facet v.

[*] Full paper published at CORIA 2013 [3]

[*] http://www.dmoz.org

[†] k is chosen based on the size of the interface medium and the cognitive load acceptable for a user

DIR 2013, April 26, 2013, Delft, The Netherlands.

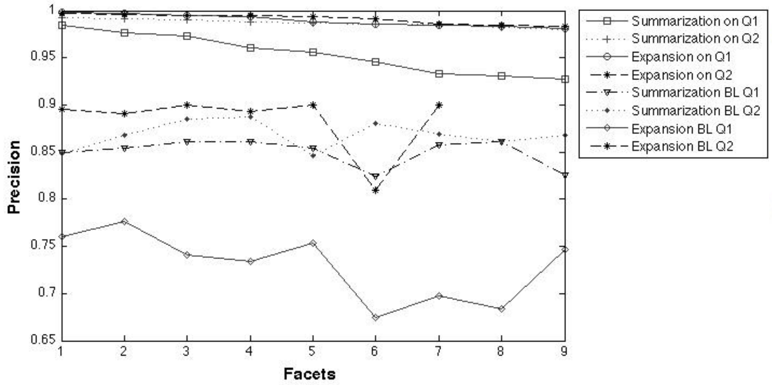
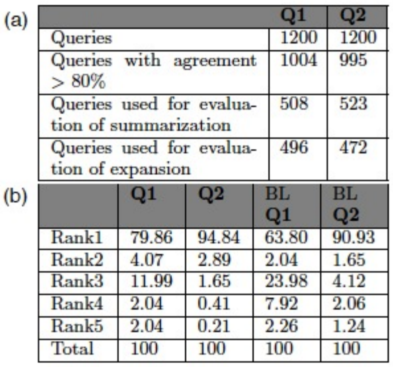


Figure 1: Precision of the summarization and expansion for the nine highest ranked facets for query sets Q1 and Q2



(a)

	Q1	Q2
Queries	1200	1200
Queries with agreement > 80%	1004	995
Queries used for evaluation of summarization	508	523
Queries used for evaluation of expansion	496	472

(b)

	Q1	Q2	BL Q1	BL Q2
Rank1	79.86	94.84	63.80	90.93
Rank2	4.07	2.89	2.04	1.65
Rank3	11.99	1.65	23.98	4.12
Rank4	2.04	0.41	7.92	2.06
Rank5	2.04	0.21	2.26	1.24
Total	100	100	100	100

Table 1: (a) Statistics of the query sets (b) Diversity of facets produced by summarization (% of facet clusters at rank 1..5)

$$importance(v, \mathbf{R}) = rel_1 + \sum_{i=rank(d), i>1, d\in\mathbf{R}} \frac{rel_i}{log_2(i)} \quad (4)$$

where **R** is a ranked list of documents retrieved for the query obtained from a search engine, i is the position of the retrieved document in the list, and $rel_i = 1$ if the ith document belongs to facet v and 0 otherwise.

The idea of this score is as follows:

- A node that has lesser distance from every other node of the subtree is a better representative of the subtree;
- A subtree that has a higher density is an important one for the query.

4. EXPERIMENTS AND RESULTS

Two sets of queries have been used for evaluation. The first query set **Q1** contains titles of English Wikipedia articles. The second query set **Q2** comprises real user queries collected by Torres et al. [1]. The queries were submitted to the Bing search engine, restricting the search results to the Web pages from the DMOZ Kids and Teens subdirectory. The subtree density model has been benchmarked against two baseline (BL) models, one for summarization and the other for expansion. The baseline model for summarization uses the top k distinct activated nodes from the ranked results from a search engine, while the baseline model for expansion uses the siblings of the activated nodes for presentation.

The evaluation is based on two aspects- relevance and diversity. First, the facets selected by the model for each query of the two query sets were presented to five Crowdflower[‡] evaluators, who were asked to judge whether the facets produced were relevant to the query. Next, to evaluate diversity of the summarization, we put together two clusters of related facets (that were judged relevant by Crowdflower evaluators)- one for each summarization model, per query for the queries in **Q1** and **Q2**. Then, Crowdflower evalutors were asked to rank these clusters on a scale of 1 to 5 based on the diversity of the facets in the clusters, with rank 1 corresponding to 'Very diverse'. For both relevance and diversity evaluations, only queries for which the agreement among Crowdflower evaluators was over 80% (as reported by Crowdflower) were retained.

The number of queries used for evaluation, the precision and diversity of the model have been indicated in Table 1 and Figure 1. From Figure 1, it is evident that the subtree density model performs better than the baselines in terms of precision (measured by relevance), both in summarization and expansion. Table 1b indicates that the subtree density model also outperforms the baseline (based on ranked results) in terms of diversity. These results are explained by the fact that in our model, important facets come from dense clusters of search results in the taxonomy.

5. CONCLUSIONS

In this paper we have presented the subtree density model for summarizing and expanding search results mapped to a subject taxonomy. Evaluation of the method using human evaluators indicates that it is effective as it optimizes both relevance and diversity. A next step in our research is to develop navigation models for interactive browsing consisting of the presented facets and their corresponding Web pages.

Acknowledgements This research was funded by the Puppy IR project EU FP7 231507.

6. REFERENCES

[1] S. Duarte Torres, D. Hiemstra, and P. Serdyukov. An analysis of queries intended to search information for children. In *Third Symposium on Information Interaction in Context*. ACM, 2010.

[2] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2000.

[3] A. Nurani Venkitasubramanian and M.-F. Moens. Selection of search facets. In *CORIA 2013 - 10th Francophone Information Retrieval Conference*, 2013.

[‡] http://crowdflower.com/

The Digital Archiving of Historical Political Cartoons: An Introduction

Junte Zhang
Meertens Institute & NIOD
Institute for War, Holocaust
and Genocide Studies

Kees Ribbens
Erasmus University Rotterdam
& NIOD Institute for War,
Holocaust and Genocide
Studies

Rob Zeeman
Meertens Institute

Royal Netherlands Academy of Arts and Sciences

ABSTRACT

Political (editorial) cartoons often capture the *Zeitgeist* of society and convey a message. Increasingly, historians study them to understand commentaries of past events or personalities. Visual culture as an academic subject could be greatly enhanced if this information can be digitally archived. We employ crowdsourcing to obtain valuable metadata by guiding volunteers' feedback using an online survey with 31 targeted questions. We provide intellectual access to a set of about 300 cartoons of a single creator spanning over multiple years in a highly interactive search engine.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process; H.3.7 [**Digital Libraries**]: Systems issues, user issues; H.5.2 [**Information interfaces and presentation**]: Graphical user interfaces (GUI)

General Terms

Design, Human Factors

Keywords

metadata, crowdsourcing, e-Humanities, cartoons

1. INTRODUCTION

Newspapers often have political (editorial) cartoons that contain a commentary about events or personalities [4] which is being disseminated. For historians, these capture the *Zeitgeist* of the period of time of their study, and become an invaluable source of information. These print newspapers are stored in libraries and get digitally archived – for example by the National Library of the Netherlands – for long-term preservation to continued access. Digital archiving is the management of the life cycle of digital assets (records) [2], from preservation to continued use.

In the Radical Political Representation project, we aim to digitally archive historical political cartoons created by a single cartoonist and published before and during the Second World War, so we gain insight into different points of view and support the study of visual culture using a computational approach. This is made possible because newspaper pages have been digitized as images, which contain cartoons. These cartoons are not yet machine-readable, therefore providing intellectual access is the best option. It has been proposed in [5] to detect the text lines in cartoons using OCR, but this is difficult because it involves handwritten texts. In [3] it is pointed out that "more descriptive areas by which images might be accessed are largely neglected," and argued that subject indexing as a field of academic work is *aboutness* – and VRA Core 4.0 is referred to as a metadata schema to record bibliographic information.

Our aim is to transcribe a cartoon, and move beyond standard bibliographic information by comprehensively capturing its meaning(s) for historical research by eliciting user feedback using crowdsourcing. So we address the following question: How can we provide intellectual access to, and allow for, advanced use of these cartoons?

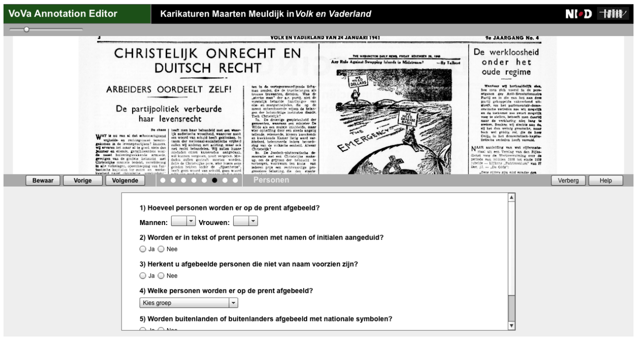
2. CROWDSOURCING OF CARTOONS

The objects of our study are so far 286 cartoons published by Maarten Meuldijk in the weekly *Volk en Vaderland* (VoVa) of the National Socialist Movement in the Netherlands from 1937 to 1942. Pages on which they occur have been digitized by the National Library. To obtain descriptions about the cartoons, we experiment with crowdsourcing to see whether crowdsourcing is applicable in our context.

The search tasks that we have in mind are more complex, therefore we created a comprehensive survey that captures the questions historians typically would ask about a cartoon. This also requires more contextual knowledge. Fig.1(a) shows the VoVa Annotation Editor developed in Adobe Flex, where we guide users through a set of 31 targeted questions in 8 stages, and aid them by offering answers of these questions with pre-defined multiple choice answers in combination with open answers. Users can zoom in/out on a cartoon, but also read contextual information related to the cartoon in the articles on the page – a strategy used by a number of users. There are no time limits and a cartoon is randomly assigned and stays assigned to a user until completion. To control for the completion of a cartoon description, we validate all questions for at least 1 given answer.

We invited interested volunteers online and in printed national media. In total 189 users registered, where eventually 83 volunteers participated with at least 1 completed description of a cartoon and with 5 users completing more than 10.

DIR 2013, April 26, 2013, Delft, The Netherlands.


VoVa Annotation Editor
Karikaturen Maarten Meuldijk in *Volk en Vaderland*
NIOD
VOLK EN VADERLAND VAN 24 JANUARI 1941
9e JAARGANG No. 4
CHRISTELIJK ONRECHT EN DUITSCH RECHT
ARBEIDERS OORDEELT ZELF!
De partijpolitiek verbeurde haar levensrecht
De werkloosheid onder het oude regime
Any Rule Against Swapping Islands in Midstream?
WE NEED DOLLARS
WE NEED DEFENSE BASES
THE EMERGENCY
Bewaar
Vorige
Volgende
Personen
Verberg
Help

1) Hoeveel personen worden er op de prent afgebeeld?

Mannen: Vrouwen:

2) Worden er in tekst of prent personen met namen of initialen aangeduid?

○ Ja ○ Nee

3) Herkent u afgebeelde personen die niet van naam voorzien zijn?

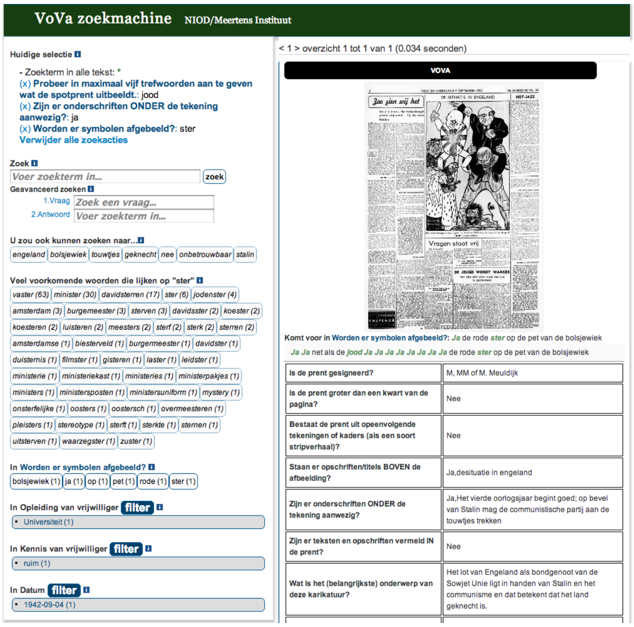
○ Ja ○ Nee

4) Welke personen worden er op de prent afgebeeld?

Kies groep

5) Worden buitenlanden of buitenlanders afgebeeld met nationale symbolen?

(a) The VoVa Annotation Editor, where volunteers can provide valuable metadata about the cartoon, ranging from plain descriptions to their opinion of a cartoon.



Huidige selectie

- Zoekterm in alle tekst: *
(x) Probeer in maximaal vijf trefwoorden aan te geven wat de spotprent uitbeeldt.: jood
(x) Zijn er onderschriften ONDER de tekening aanwezig?: ja
(x) Worden er symbolen afgebeeld?: ster
Verwijder alle zoekacties

Zoek

Voer zoekterm in... zoek

Geavanceerd zoeken

1.Vraag Zoek een vraag...
2.Antwoord Voer zoekterm in...

U zou ook kunnen zoeken naar...

engeland | bolsjewiek | touwtjes | geknecht | nee | onbetrouwbaar | stalin

Veel voorkomende woorden die lijken op "ster"

vaster (63) | minister (30) | davidsterren (17) | ster (6) | jodenster (4) | amsterdam (3) | burgemeester (3) | sterven (3) | davidsster (2) | koester (2) | koesteren (2) | luisteren (2) | meesters (2) | sterf (2) | sterk (2) | sterren (2) | amsterdamse (1) | biesterveld (1) | burgermeester (1) | davidster (1) | duisternis (1) | filmster (1) | gisteren (1) | laster (1) | leidster (1) | ministerie (1) | ministeriekast (1) | ministeries (1) | ministerpakjes (1) | ministers (1) | ministersposten (1) | ministersuniform (1) | mystery (1) | onsterfelijke (1) | oosters (1) | oostersch (1) | overmeesteren (1) | pleisters (1) | stereotype (1) | sterft (1) | sterkte (1) | sternen (1) | uitsterven (1) | waarzegster (1) | zuster (1)

In Worden er symbolen afgebeeld?

bolsjewiek (1) | ja (1) | op (1) | pet (1) | rode (1) | ster (1)

In Opleiding van vrijwilliger filter

- Universiteit (1)

In Kennis van vrijwilliger filter

- ruim (1)

In Datum filter

- 1942-09-04 (1)

< 1 > overzicht 1 tot 1 van 1 (0.034 seconden)

Komt voor in Worden er symbolen afgebeeld?: Ja de rode ster op de pet van de bolsjewiek

Ja Ja net als de jood Ja Ja Ja Ja Ja Ja Ja Ja de rode ster op de pet van de bolsjewiek

Is de prent gesigneerd?	M, MM of M. Meuldijk
Is de prent groter dan een kwart van de pagina?	Nee
Bestaat de prent uit opeenvolgende tekeningen of kaders (als een soort stripverhaal)?	Nee
Staan er opschriften/titels BOVEN de afbeelding?	Ja,desituatie in engeland
Zijn er onderschriften ONDER de tekening aanwezig?	Ja,Het vierde oorlogsjaar begint goed; op bevel van Stalin mag de communistische partij aan de touwtjes trekken
Zijn er teksten en opschriften vermeld IN de prent?	Nee
Wat is het (belangrijkste) onderwerp van deze karikatuur?	Het lot van Engeland als bondgenoot van de Sowjet Unie ligt in handen van Stalin en het communisme en dat betekent dat het land geknecht is.

(b) The VoVA Search Engine, which is used to gain intellectual (advanced) access to the cartoons.

Figure 1: The digital archiving of cartoons with the VoVa Annotation Editor and Search Engine.

3. SERENDIPITY IN CONTEXT

Having obtained the metadata, we want to use it. Since the search engine should serve historians, we design it to support serendipitous search and be highly interactive in order to focus on a high recall (rather than precision). The system has been designed to maximize the user's ability to explore. We have proposed search features to support serendipitous and focused access in [6], and these features have been reimplemented here. The search features primarily deal with query expansion, recommendation, and interactive visualizations of aggregated results. The former is based on using ternary search trees for spellchecking, returning the top term vectors related to the original query, and returning the top terms that have the original query as substring. The latter is based on charts, maps and word clouds.

A user can improve the searching in a session by effectively reducing the information space step by step, i.e. incrementally combining questions. This confirms with the Berrypicking model of [1] – queries are not static, but rather evolve, and users "gather information in bits and pieces instead of in one grand best retrieved set." These steps are stored as part of the search trail, so the overview is kept. The user interface of the system is depicted in Fig 1(b). In this example, someone looked for a cartoon about a "Jood" (*Jew*) used as a main keyword to describe a cartoon, with captions under it, and a "ster" (*star*) depicted as a symbol. The search engine treats the questions asked in the survey as facets, and is therefore a straight-forward question-answering system. Facets that always appear are the date of publication of a cartoon, and the education and knowledge levels of the volunteers who provided the descriptions. We show in [7] that making the credibility of the source transparent gives users greater confidence in their selection. We think historians will be aided with this part of the search process.

There are different search strategies possible. Users can search by full-text or focused (within the answers of questions). The query gets highlighted in context given the full-text and the survey question. A dynamic word cloud widget that supports query expansion is not activated, unless the autocompletion is used. Using the Advanced Search option, users can look up a question and then enter a keyword also with the autocompletion feature. Wildcard (empty) queries can be used to obtain the distribution of words of the answers given a question in a word cloud for a quick summary.

4. CONCLUSIONS

We have presented – in a compressed version – the mission statement and some results of the Radical Political Representation project. We completed the first phase of crowdsourcing, and pending further releases of data by the National Library, we can further digitally archive the complete series of Meuldijk cartoons. The technical infrastructure to digitally archive political cartoons has been set-up.

This means we can expand our scope to other cartoonists in different times – there is no shortage of cartoons. We can refine our survey to allow for more different information needs of historians, or embed our survey as part or extension of a formal metadata schema like VRA Core. We will improve the UI and further implement useful information visualization of results, and evaluate the search engine. It can be used at `www.meertens.knaw.nl/vova/search`.

5. REFERENCES

[1] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989.

[2] G. M. Hodge. An information life-cycle approach : Best practices for digital archiving. *Journal of Electronic Publishing*, 5(4):1–14, 2000.

[3] C. Landbeck. Issues in subject analysis and description of political cartoons. *Advances in Classification Research Online*, 19(1), 2008.

[4] C. Sterling. *Encyclopedia of Journalism*. Sage, 2009.

[5] Y. Wu. Searching digital political cartoons. In *Proceedings of the 2010 IEEE International Conference on Granular Computing*, GRC '10, pages 541–545, Washington, DC, USA, 2010. IEEE Computer Society.

[6] J. Zhang. Supporting serendipitous and focused search. In *EuroHCIR*, volume 909 of *CEUR Workshop Proceedings*, pages 79–82, 2012.

[7] J. Zhang, A. Amin, H. S. M. Cramer, V. Evers, and L. Hardman. Improving user confidence in cultural heritage aggregated results. In *SIGIR*, pages 702–703, 2009.

PoliticalMashup Ngramviewer

Tracking who said what and when in parliament

Bart de Goede
Dispectu
University of Amsterdam
bart@dispectu.com

Justin van Wees
Dispectu
University of Amsterdam
justin@dispectu.com

Maarten Marx
PoliticalMashup
University of Amsterdam
maartenmarx@uva.nl

ABSTRACT

The PoliticalMashup Ngramviewer is an application that allows a user to visualise the use of terms and phrases in the "Tweede Kamer" (the Dutch parliament). Inspired by the Google Books Ngramviewer[1], the PoliticalMashup Ngramviewer additionally allows for *faceting* on politicians and parties, providing a more detailed insight in the use of certain terms and phrases by politicians and parties with different points of view.

Categories and Subject Descriptors

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

1. INTRODUCTION

The Google Books Ngramviewer [2] allows a user to query for phrases consisting of up to 5 terms. The application visualises the relative occurrence of these phrases in a corpus of digitised books written in a specific language over time.

Inspired by the Google Books Ngramviewer, the PoliticalMashup Ngramviewer[2] allows the user to query phrases consisting of up to 7 terms spoken in the Dutch parliament between 1815 and 2012, and visualise the occurrence of those phrases over time. Additionally, the PoliticalMashup Ngramviewer allows the user to facet on politicians and parties, allowing for comparison of the use of phrases through time by parties with different ideologies.

In this demonstration paper we describe the data used in this application, the approach taken with regard to analysing and indexing that data, and examples of how the application could be used in research on agenda setting and linguistics.

2. NGRAMVIEWER

2.1 Data

The PoliticalMashup project [1] aims to make large quantities of political data, such as the proceedings of the Dutch parliament, available and searchable. In addition, a goal of the project is to combine (or *mash up*) political data from different sources, in order to provide for *semantic search*, such as queries for events or persons.

This Ngramviewer is an example of why linking raw text to entities such as persons or parties can be useful: for each word ever uttered in the Dutch parliament, we know who said it, when it was said, to which party that person belonged at that time, and which role that person had at that point in the debate. By linking text to speakers, faceting on persons and parties is enabled.

The data this application uses originates from three sources: Staten-Generaal Digitaal[3], Officiële Bekendmakingen[4] and Parlementair Documentatiecentrum Leiden[5]. PoliticalMashup collected, analysed and transformed data from these sources, determining which speaker said what when, and to which party that speaker belonged at the time. This dataset is freely available via DANS EASY[6].

n-gram	unique terms	without hapaxes
1-grams	2,773,826	992,291
2-grams	38,811,679	12,852,501
3-grams	170,314,738	38,648,440
4-grams	358,360,166	48,621,948
5-grams	498,848,849	36,838,184
6-grams	573,197,917	22,737,318
7-grams	606,867,133	13,655,460
total	2,249,174,308	174,346,142

Table 1: Distribution of unique n-grams in the Ngramviewer corpus for all terms, and with all *hapaxes* (terms that occur only once in the corpus) removed.

DIR 2013, April 26, 2013, Delft, The Netherlands.

[1] http://books.google.com/ngrams

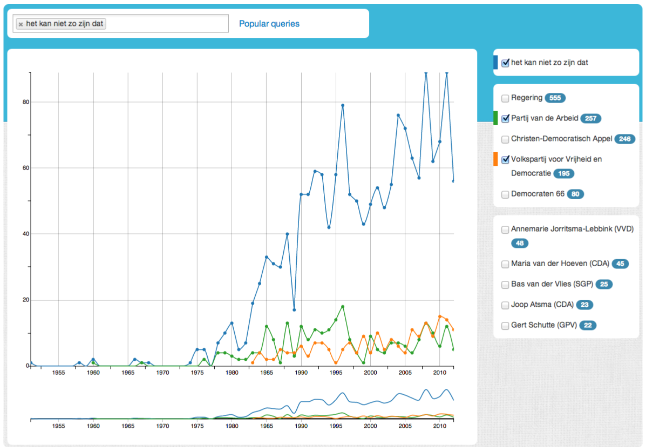
[2] http://ngram.politicalmashup.nl

[3] Project of the Koninklijke Bibliotheek (http://kb.nl/en/), digitising all Dutch parliamentary proceedings between 1814 and 1995 (http://statengeneraaldigitaal.nl/overdezesite).

[4] Portal of the Dutch government, providing a search interface to all govermental proclamations, including parliamentary proceedings since 1995 (https://zoek.officielebekendmakingen.nl/).

[5] Biographical information on politicians and parties (http://www.parlement.com/).

[6] http://www.persistent-identifier.nl/urn:nbn:nl:ui:13-k2g8-5h



☑ het kan niet zo zijn dat

☐ Regering 555
☑ Partij van de Arbeid 257
☐ Christen-Democratisch Appel 246
☑ Volkspartij voor Vrijheid en Democratie 195
☐ Democraten 66 80

☐ Annemarie Jorritsma-Lebbink (VVD) 48
☐ Maria van der Hoeven (CDA) 45
☐ Bas van der Vlies (SGP) 25
☐ Joop Atsma (CDA) 23
☐ Gert Schutte (GPV) 22

Figure 1: The PoliticalMashup Ngramviewer interface showing results for "het kan niet zo zijn dat", with facets on PvdA and VVD, illustrating the rise of the phrase since the eighties.

2.2 Indexing

The PoliticalMashup Ngramviewer is built on top of an Apache Lucene[7] index. We defined a document as *every word* of a specific *politician* spoken on a *particular day*. This allows for comparison of term frequencies per person, per day, which can be aggregated to words spoken by all members of a particular party in a particular time period (week, month, year, etcetera).

We used standard tokenisation and analysis on these documents; lowercasing, character folding and removal of punctuation, but *keeping* stopwords, in order to facilitate search on phrases containing common words such as articles or determiners. Additionally, we constructed word n-grams ($1 \leq n \leq 7$), respecting sentence boundaries.

The index contains data from 4 April 1815 to 9 September 2012, with 326,315 documents (where a document is all the text one person said on one day), 18,572 days for which there are documents, for in total 3,085 politicians which are members of 119 parties or the government. Table 1 shows the distribution of n-grams in the corpus. The second column shows the distribution of n-grams that occur more than once in the corpus, yielding a reduction of the vocabulary size of one order of magnitude. This is partly due to OCR errors (all proceedings predating 1995 are scans of paper archives).

2.3 Architecture

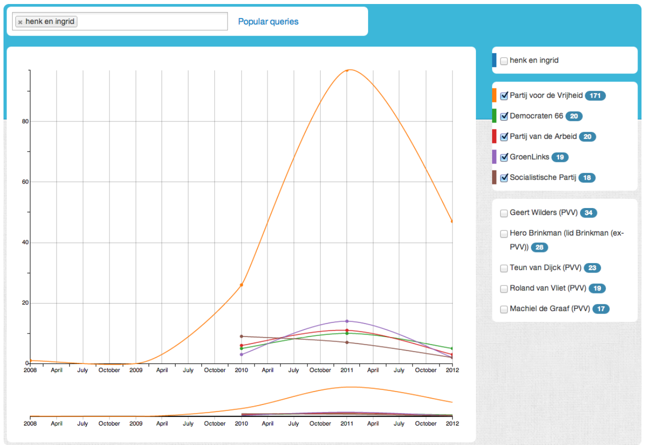
We constructed an inverted index in Lucene, storing the document frequency for each n-gram, and the term frequency for each document that n-gram occurs in.

Additionally, each document has attributes, such as the date the terms of that document were spoken, and identifiers that resolve to politicians and parties[8].

At query time, these identifiers are used to obtain information on persons and parties, which are subsequently cached in a Redis key-value store. This Redis store is also used to cache query results and keep track of popular queries. Also, date frequencies are aggregated to frequencies per year at query time.

[7] http://lucene.apache.org/core/

[8] PoliticalMashup maintains a resolver that maps identifiers to persons parties and proceedings.



☐ henk en ingrid

☑ Partij voor de Vrijheid 171
☑ Democraten 66 20
☑ Partij van de Arbeid 20
☑ GroenLinks 19
☑ Socialistische Partij 18

☐ Geert Wilders (PVV) 34
☐ Hero Brinkman (lid Brinkman (ex-PVV)) 28
☐ Teun van Dijck (PVV) 23
☐ Roland van Vliet (PVV) 19
☐ Machiel de Graaf (PVV) 17

Figure 2: The PoliticalMashup Ngramviewer interface showing results for "Henk en Ingrid", with facets on parties, showing the introduction of the term in 2008, no use in 2009, and that the term is picked up by other parties in 2010.

2.4 Examples

"Het kan niet zo zijn dat"[9] is a popular phrase used by (Dutch) politicians, lending their statement a more urgent feeling, (unconsciously) trying to manipulate their audience, while the person is just ventilating an opinion. Figure 1 shows the rapid increase in use since the eighties, and the use of the Ngramviewer for linguistic research.

"Henk en Ingrid" are a fictional couple, conceived by the Dutch politician Geert Wilders[10], representing the average Dutch family. Figure 2 shows how Wilders' party introduced the phrase in 2008, but was left unused until 2010, when other parties picked up the phrase as well. This example shows the use of the Ngramviewer for agenda-setting.

3. DEMONSTRATION

The demonstration will show how the PoliticalMashup Ngramviewer can be used, displaying a graph of how often the entered phrases occur over time in the proceedings of the Dutch parliament. Also, it will demonstrate faceting on politicians and parties, showing the occurrence of the entered phrases over time for specific politicians and parties.

4. ACKNOWLEDGMENTS

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 380-52-005 (PoliticalMashup).

5. REFERENCES

[1] M. Marx. Politicalmashup. Retrieved March, 2013 from http://politicalmashup.nl/over-political-mashup/.

[2] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[9] In English: "It is unacceptable that . . . "

[10] http://en.wikipedia.org/wiki/Geert_wilders

Traitor: Associating Concepts using the World Wide Web

On-line demonstrator at `http://evilgeniuses.ophanus.net`

Wanno Drijfhout Oliver Jundt Lesley Wevers Djoerd Hiemstra
University of Twente
{wrc.drijfhout,oliver.jundt}@gmail.com {l.wevers,d.hiemstra}@utwente.nl

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing; H.3.3 [**Information Search and Retrieval**]: Query formulation

General Terms

Querying, Visualization, Experimentation

Keywords

Information Extraction, Question Answering, MapReduce

ABSTRACT

We use Common Crawl's 25TB data set of web pages to construct a database of associated concepts using Hadoop. The database can be queried through a web application with two query interfaces. A textual interface allows searching for similarities and differences between multiple concepts using a query language similar to set notation, and a graphical interface allows users to visualize similarity relationships of concepts in a force directed graph.

1. INTRODUCTION

What do Apple and Samsung have in common? What does Java have that PHP does not have? Which terms could refine the search query "`lance armstrong`"? What is the context of the sentence "*Food, games and shelter are just as important as health*"? You may know the answers to these questions or know where to look for them—but can a computer answer these questions for you too? This paper discusses Traitor, a tool that creates a database of associated concepts, and answers questions similar to our examples. Traitor was created for submission to the Norvig Web Data Science Award[1], a research challenge organized by Common Crawl and SURFsara. We won the award.

2. MINING CONCEPT ASSOCIATIONS

We use Common Crawl's 25TB data set of web pages to mine associated concepts. Other corpora (e.g., search logs[4], USENET[6] and the British National Corpus[2]) have been used for similar purposes before. To do this mining, we have implemented a map-reduce program in Java with Hadoop. For simplicity, the program assumes that one word represents one concept (and vice versa), and for every pair of words that co-occur in a sentence, the concepts they represent are associated.

In a few steps our program transforms the raw HTML responses from the Common Crawl data set to a list of key value pairs (k, v), where the key k is a word pair (w_1, w_2) that co-occurred in some sentence on the page, and the count v denotes the number of pages on which this co-occurrence has been found. We consider the count of co-occurring word pairs a measure of the association's *strength*. A more sophisticated approach such as a probabilistic[7] association measure[1, 8], based on the concept of *mutual information*[3] could have been used instead—the limited time to submit our solution to the Norvig Award jury pressured us to use simpler methods.

We chose sentences as the semantic unit (rather than the same paragraph, document or a sliding window[6]) for generating word pairs for two reasons. A technical reason is to constrain the result size. Pairing every non-trivial word with every other produces results in the order $O(n^2)$. The second reason is based on human language semantics[5]. We suppose that words within a sentence are more likely to represent actual concept associations than words that are far apart in a document (or in different sentences).

2.1 Implementation

In the mapping phase we extract distinct word pairs from documents. We extract "interesting" text from the raw HTML documents using the BoilerPipe library; i.e., text in navigation menus, headers and footers is omitted. We then split each sentence in the text into a set of words and normalize these words by converting them to lowercase ASCII characters (e.g., `á` and `Á` become `a`). This reduces the number of generated word pairs and compensates for small notation differences between words. Moreover, we discard words from non-English sentences[2], words containing non-alphabetic characters, and stop-words such as "the" and "that". For each normalized and filtered sentence S, the mapper creates a word pair p for each[3] pair of words (w_1, w_2) where $w_1, w_2 \in S$, $w_1 < w_2$ (lexicographically). Finally, for every web page, the mapper emits tuples $(p, 1)$ for each distinct word pair p on that page.

In the reduction phase, we sum the counts of the word pairs produced by the mapper. Because the distribution of words follows a Zipf distribution, we find that the majority of the resulting pairs have a very low count. To reduce

[1] `http://norvigaward.github.com`

[2] Our heuristic is rather crude: we check if a sentence contains at least two English 'stop-words'.

[3] We limit the number of pairs produced for excessively long sentences.

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

storage costs of the final output, we can discard pairs with a count less than some N; e.g., if $N = 2$, we discard all pairs that only occur once. Essentially, this allows us to reduce the output to any size that is practical for use in the presentation layer. Unfortunately, pairs containing rare words are cut indiscriminately due to this policy, even if these co-occurring words are still usable associations.

3. QUERYING CONCEPT ASSOCIATIONS

The resulting tuples from the map-reduce step are imported into a database which can be queried through a web application with two query interfaces.

3.1 Textual interface

Users of the textual interface can search for similarities and differences between multiple concepts using a query language similar to set notation or boolean algebra. For each search term in the query, associated words and co-occurrence counts are fetched from the database, and a score is assigned to each associated word based on the structure of the query expression.

A sequence of words separated by whitespace denotes a conjunction and yields the words that are associated with *all* words in the sequence. A sequence of words separated by plus-symbols, denotes a disjunction and yields the words that are associated with *any* word in the sequence. A word preceded by a minus-symbol denotes the complement. To illustrate: the query `(a + b) -c` yields all words that are associated with `a` *or* `b`, *and not* with `c`.

3.2 Graphical interface

A graphical interface allows users to visualize similarity relationships of concepts in a *force directed graph* using D3.js. Users can enter a list of words which become labeled nodes in the graph. The graphical size of the nodes indicates the number of associations a concept has; the more associations a concept has, the bigger the node. For each word in the query the system retrieves the 50 strongest related concepts, which can be interpreted as an estimate of the concept's context. For each word pair we apply the RBO metric[9] to estimate the similarity. A link is created between two nodes if the similarity is more than 5%. The length and thickness of the link indicate the similarity. If two nodes are connected by a short thick line the corresponding concepts share a very similar top 50 ranking. Conversely, distant nodes and nodes without any link between them have very few (or no) concepts in common.

4. RESULTS

Using our map-reduce program, we populated an association database with over 43 million distinct word pairs. We attempted to assess the quality of the Traitor's query results by answering questions from this paper's introduction (and others). For the sake of brevity, table 1 shows a few query results produced by Traitor. Additionally, to illustrate the disjunctive operator, we could 'deduce the context' of a sentence; the query "`food + games + and + shelter + are + just + as + important + as + health`" produces the union of each word's associations: *care*, *insurance*, *information*, *play*, *good*. The reader is encouraged to try Traitor on-line for more example queries (see the About-page) and for a live demonstration of the visualization interface.

5. CONCLUSIONS AND FUTURE WORK

In about 8 hours, the SURFsara Hadoop cluster of circa 66 nodes reduced 25 terabytes of Common Crawl corpus data to about 10 gigabytes of uncompressed word associations by aggressive filtering of the input, and dropping all pairs with a count less than 100. By means of a query language and a simple scoring algorithm, we can express and answer queries about the concepts and associations stored in this database. A visualization interface allows for comparison of concepts by the similarity of their associations.

`apple`	`samsung`	`java`
iphone	phone	code
ipod	galaxy	application(s)
ipad	mobile	games
store	battery	software
mac	tv	programming
`apple samsung`	`lance armstrong`	`political party`
phone	tour	information
tv	cancer	parties
mobile	france	policy
battery	foundation	third
iphone	team	government
`java -php`	`java coffee -php`	`wii -xbox`
applet(s)	cup	balance
alden	bean	kart
jvm	beans	nunchuk
coffee	tea	resort
marketplace	espresso	motionplus

Table 1: Query results produced by Traitor.

Despite the simplistic methods used, Traitor can provide reasonable results thanks to the large data corpus. As discussed in Section 2, we expect that a probabilistic scoring model can further improve the quality of our results. Moreover, Traitor only supports 'concepts' described by single words; one could extract n-grams from sentences to identify concepts described by multiple words. Future work can include further normalization of words; e.g., equating plural and singular words, or applying word stemming.

6. REFERENCES

[1] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1), 1990.

[2] Stefan Evert. *The statistics of word cooccurrences.* PhD thesis, Dissertation, Stuttgart University, 2005.

[3] Robert M. Fano. *Transmission of Information: A Statistical Theory of Communication.* The MIT Press, March 1961.

[4] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng. Mining query subtopics from search log data. In *Proc. of the 35th int. ACM SIGIR conf. on IR*, 2012.

[5] C. Lioma, B. Larsen, and W. Lu. Rhetorical relations for information retrieval. In *Proc. of the 35th int. ACM SIGIR conf. on IR*, 2012.

[6] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2), June 1996.

[7] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01), 2004.

[8] T. Sugimachi, A. Ishino, M. Takeda, and F. Matsuo. A method of extracting related words using standardized mutual information. In *Discovery Science*, 2003.

[9] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4), November 2010.

xTAS and ThemeStreams

Extendable Text Analysis Service and its Usage in a Topic Monitoring Tool

Ork de Rooij
University of Amsterdam
Science Park 904
1098 XH Amsterdam, The Netherlands
orooij@uva.nl

Tom Kenter
University of Amsterdam
Science Park 904
1098 XH Amsterdam, The Netherlands
tom.kenter@uva.nl

Maarten de Rijke
University of Amsterdam
Science Park 904
1098 XH Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

xTAS is an extendable multi-user text analysis service for large scale multi-lingual document analysis developed at the University of Amsterdam. It can process large amounts of documents in a timely manner through a web interface that can be used by multiple users at once. In this demonstration paper we present recent additions which include semanticization, on the fly TF-IDF model generation and on the fly co-occurrence metrics. Furthermore, we demonstrate ThemeStreams, a novel topic monitoring tool built on top of xTAS.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text Analysis

General Terms

Algorithms, Performance, Experimentation

Keywords

text analysis, web service, distributed processing, microblog visualization

1. INTRODUCTION

xTAS[1] is an integrated set of text analysis services for processing documents in a timely manner. It is available through a web API that can be used by multiple users at once. xTAS includes tools for stemming, tokenization, named entity recognition, part–of–speech tagging, sentiment analysis and various types of aggregation on top of this. The purpose of xTAS is to run text processing tasks as fast as possible, without concerning users about databases, storage or result caching.

The software can run multiple tasks in parallel, possibly on different machines (nodes). xTAS is built solely with open source software. It uses Celery [2] to distribute tasks between nodes. By default MongoDB [4] is used to store documents and results though other options are available as well.

The software is extendable. Additional functionality can easily be added through a plugin architecture.

In what follows we describe recent additions to xTAS and we present ThemeStreams, a novel topic monitoring tool built on top of xTAS.

2. XTAS

Recent additions and improvements to xTAS include:

- Semanticization[2]

  xTAS can semantically enrich texts by linking entities mentioned in it to their Wikipedia article.

- On the fly TF-IDF model generation and application

  TF-IDF models based on a user selected series of documents can be trained on the fly. The models can be used to provide TF-IDF statistics for words in new documents.

- Co-occurrence metric calculation

  A variety of co-occurrence metric calculation methods were added to xTAS, including maximum likelihood estimate, point wise mutual information, log likelihood ratio and χ^2. This enables users to calculate the co-occurrence of entities in a set of documents.

- Automatic language identification

  If the language of a document is not supplied xTAS can automatically determine it. Currently this is implemented by using TextCat [6].

- Support for multiple document stores

  Besides mongoDB [4], xTAS can communicate directly with Apache Solr [1] or ElasticSearch [3]. These stores can be used as a document repository as well as a result cache.

- Response time improvements

  Analysis of xTAS usage over time shows that named entity recognition is a frequently requested and time consuming analysis. In order to keep response times to

[1] See http://xtas.net

[2] Semanticization, the process of linking mentions of concepts in a text to the articles in an external knowledge base they denote, is also referred to as entity linking or Wikification.

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

near-real time speeds xTAS keeps several NER models (for all supported languages) in memory on each xTAS node.

3. THEMESTREAMS

ThemeStreams[3] is a visual interface that helps answer the question *"Who is talking about what?"*. It does so for topics in the Dutch political landscape by showing the ebb and flow of conversations about particular themes trough time. While there are many topic monitoring tools available, the novelty of ThemeStreams lies in its ability to present the user with a quick overview of the relative frequency of posts a particular group of users issued on a certain subject. ThemeStreams is based on tweets posted on Twitter by four groups of people:

- politicians (ministers, members of parliament, but also the local ranks of politicians in municipalities and provinces)
- political journalists (news paper journalists as well as talk show hosts of political television shows)
- lobbyists (people pushing the people who are active in politics)
- other influencers (these include (satirical) columnists, politically engaged celebrities and stand-up comedians)

The harvesting of these tweets started late 2011. At the time of writing, we follow about 1400 individual users, who, together with all people participating in conversations with these *inner circle* users yield a set of just over 3.9M tweets.

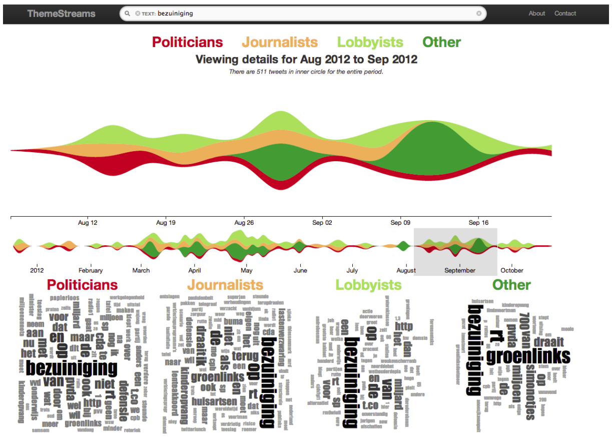
The interactive visual interface is aimed at giving insight into the ownership and dynamics of themes being discussed. It enables users to answer questions such as *Who put this issue on the map?*, *Who picked up on this topic?*, *Is this topic gaining momentum?* ThemeStreams allows users to explore streams of tweets either from a fixed set of predefined themes or through a search box. It uses stream graphs [5] to indicate how the four influence groups discuss a specified theme, thereby depicting the volume, the "aliveness" and ownership of a topic.

The interface indicates the time a tweet was posted, the influence group the poster belongs to and the number of people which reacted to a statement (which can be used to estimate the "size" and "lifetime" of statement). Initially a combined word cloud is shown with words colorized by the group they originate from. Users can zoom in to parts of the stream for more detail. Doing so results in individual word clouds being displayed per influence groups during the selected period.

Initial usability studies were carried out with university staff members and media analysts working for a communication agent. We found that ThemeStreams was intuitive to understand and it was easy to inspect parts of a tweet stream in detail. The combined clouds proved to be insightful for a fast overview of data. The individual clouds proved to be useful for inspecting relative word usage between groups. We also found a need for depicting the most represented speakers within a group.

4. FUTURE WORK

xTAS is actively being used in a number of research and production environments. As such, work on xTAS is ongoing and features are being deployed in close collaboration with end users. Currently, we focus on adding support for temporal tagging and for easier deployment on large clusters.



Viewing details for Aug 2012 to Sep 2012

There are 511 tweets in inner circle for the entire period.

Figure 1: ThemeStreams - A visual interface that answers the question *Who is talking about what?*. Tweets are shown in a stream graph, categorized by their authors and weighted by their conversational influence. Parts of the stream can be selected and detailed word clouds per group pop up to show what was being said by whom during that period in time.

A more detailed user study of ThemeStreams is currently in progress. Also we are looking into additional application scenarios for ThemeStreams, like discourse analysis over time in other domains such as news paper archives.

5. ACKNOWLEDGEMENTS

This research was partially 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, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 727.-011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the BILAND project funded by the CLARIN-nl program, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), and the Netherlands eScience Center under project number 027.012.105.

6. REFERENCES

[1] Apache solr. http://lucene.apache.org/solr/.

[2] Celery: Distributed Task Queue. http://celeryproject.org/.

[3] elasticsearch. http://www.elasticsearch.org/.

[4] MongoDB. http://www.mongodb.org/.

[5] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1245–1252, 2008.

[6] W. B. Cavnar, J. M. Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.

[3] See an online demo at http://themestreams.xtas.net/

A semantic wiki for novelty search on documents

Michael Färber*
Karlsruhe Institute of Technology (KIT)
Institute AIFB
76131 Karlsruhe
michael.faerber@kit.edu

Achim Rettinger
Karlsruhe Institute of Technology (KIT)
Institute AIFB
76131 Karlsruhe
rettinger@kit.edu

ABSTRACT

Technology-oriented companies are typically interested in monitoring developments concerning their technologies. However, most companies, especially SMEs, don't have an efficient process how this is achieved. If at all, efforts are mostly limited to uncoordinated keyword queries on web resources. Here, we present a semi-automatic approach that allows for structured and continuous detection of relevant, novel and domain specific documents appearing on the Web. Our system is based on a semantic wiki where the domain expert is able (i) to store all relevant information in an adequate knowledge base with the ability for monitoring and trend mining and (ii) to import detected novel items such as future technologies and their properties to the knowledge base in a continuos fashion. The latter is achieved by generating a structured query based on the user context and by representing found documents as semantic graphs. In this way, novel items can be found easier and in a semi-automatic fashion.

Categories and Subject Descriptors

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

General Terms

Algorithms, Economics

Keywords

semantic wiki, novelty detection, document ranking, ontology-supported information extraction.

1. MOTIVATION

Technology forecast and trend detection are indispendable tasks for technology companies in order to be informed about market developments and inventions in their fields. With the advent of more and more documents on the Web, companies face the task of extracting relevant and novel information for this purpose. Currently, this has to be done usually purely manually and without any structured background data making it a very time-consuming task. Therefore, we provide a semi-automatic process for trend detection and monitoring services. We present a semantic wiki-based application which is based on ontology-based information extraction (OBIE) where ontologies are used within the information extraction (IE) process. Since usually appropriate ontologies regarding technologies and their properties are missing or are too small, we focus our work on the crucial task of how to efficiently find new textual information which is relevant to the domain expert, but has not been stored in the knowledge base (KB) and, therefore, has been made usable in some sense.

2. RELATED WORK

Within the TREC "novelty track" in 2002–2004 [2], systems for detecting novelty were designed. However, the task took place on sentence level, was limited to event and opinion detection, and was aligned for non-domain specific texts such as news. *Newsjunkie* [1] is also geared to detecting novelty by comparing a new document against an existing document collection. Contrary to such systems, we face domain-specific documents like technical reports and patents, and therefore do not have to deal with the problem of analysing huge amounts of articles in a very short time period, known as "burst of novelty". Instead of purely statistical measures, our approach is based on semantic technologies.

3. DOCUMENT RANKING AND ONTOLOGY POPULATION

Figure 1 gives an overview of the interplay between an ontology and documents with potentially novel information: Given our own KB with instances and schema, our goal is to search for documents and to rank them, so that the documents most novel to the KB and relevant to both the query and the KB have the highest ranking. In a second step the user is able to import phrases marked in the document into his/her KB as property values.

Concerning the first part, Semantic MediaWiki[1] as an instance of a semantic wiki is assigned the central role: The user is able to create new wiki pages (within the semi-automatic process or just manually) and to add

*This work is supported by the German Federal Ministry of Education and Research (BMBF) under grant 02PJ1002 (SyncTech).

DIR 2013, April 26, 2013, Delft, The Netherlands.

[1] http://semantic-mediawiki.org/

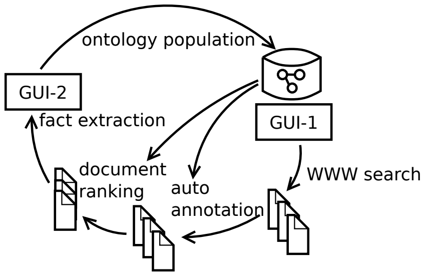


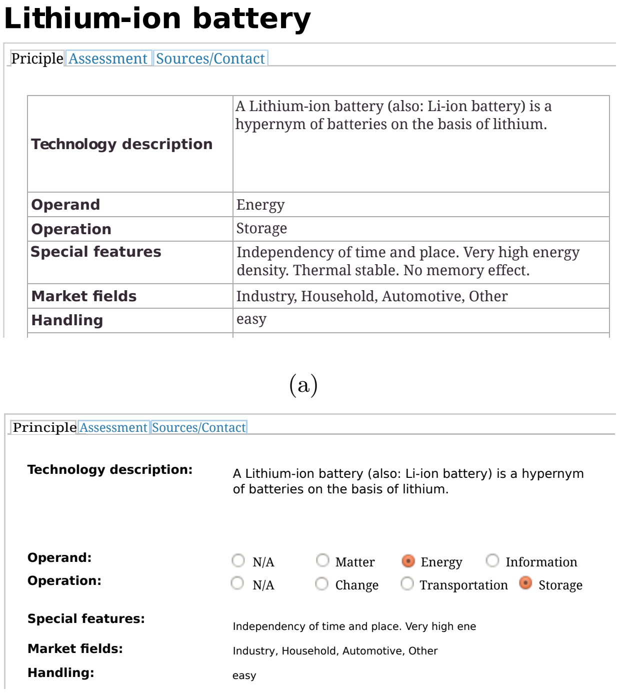
Figure 1: According to a user's context a structured query is generated with the help of an underlying ontology. Afterwards, ranking is performed using annotated document corpus. In the last step, annotations are verified by the user and used for populating the ontology. In succeeding search rounds search is based on the enriched ontology.

appropriate properties with the help of a class-specific form (see figure 2). Internally, all data is stored in a structured way. The wiki allows the user to create a search query out of the context by taking instances and property values (from the KB) as well as search keywords written by the user. After an optional expanding of the query graph with neighbouring entities, we can generate the final query graph. Since all documents are annotated with the help of named entity recognition tools[2], we can compare the generated query graph with all document entity graphs (generated from extracted named entities). Ranking of the documents is facilitated by weights which were assigned to every relation in the KB schema graph. We can use implicit user feedback in the following way: If a user imports some novel item as a new property or instance, the weights in the KB schema graph are adapted. By this means, we can defer to the personal views what relationships between certain classes and properties (or other classes) are of great significance and should be reinforced for next search sessions.

Our focused use cases are determined by our use case partners[3] which are medium-sized technology companies. Hence, the lightweight ontologies we used consist of classes like *technology*, *institution*, and *product*. As document corpus, web documents retrieved by search engine requests are considered. In addition, trend detection in conjunction with patents can be enabled by using the patent database Espacenet[4], where access to over 70 million patent documents and their meta data is provided.

4. CONCLUSION

Existing processes and tools for trend mining and technology watch are often only rudimentary implemented, especially in SMEs. We have presented a semantic wiki for storing and displaying structured information about a specific domain (industrial technology field) and for generating a context-aware semantic search query. With the help of a new proposed ranking schema, the more relevant and potentially novel information a document contains, the higher it is ranked and, hence, more likely to be worth reading and used for ontology population. Due to the use of structured information and approriate background data the way of doing trend mining can be changed towards a semi-automatic process with better search and monitoring capabilities.



Lithium-ion battery

Technology description	A Lithium-ion battery (also: Li-ion battery) is a hypernym of batteries on the basis of lithium.
Operand	Energy
Operation	Storage
Special features	Independency of time and place. Very high energy density. Thermal stable. No memory effect.
Market fields	Industry, Household, Automotive, Other
Handling	easy

(a)

Technology description: A Lithium-ion battery (also: Li-ion battery) is a hypernym of batteries on the basis of lithium.

Operand: ○ N/A ○ Matter ● Energy ○ Information

Operation: ○ N/A ○ Change ○ Transportation ● Storage

Special features: Independency of time and place. Very high ene

Market fields: Industry, Household, Automotive, Other

Handling: easy

(b)

Figure 2: Screenshots of a Semantic MediaWiki: (a) displaying technology property values within a wiki page (b) edit functionality using form.

5. REFERENCES

[1] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 482–490, New York, NY, USA, 2004. ACM.

[2] Ian Soboroff and Donna Harman. Novelty detection: the TREC experience. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 105–112, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[2] One of these tools is the wikify service of the Wikipedia Miner (http://wikipedia-miner.cms.waikato.ac.nz/) which we adapt by using the content of our domain specific semantic-based wiki. In order to detect also new entities, property values, and relationships, we use GATE (http://gate.ac.uk), a well-established rule-based framework.

[3] Industry partners within the German research project *syncTech* (http://synctech-innovation.de).

[4] http://worldwide.espacenet.com

Site Search Using Profile-Based Document Summarisation

Azhar Alhindi
University of Essex,
Colchester, UK
ahalhi@essex.ac.uk

Udo Kruschwitz
University of Essex,
Colchester, UK
udo@essex.ac.uk

Chris Fox
University of Essex,
Colchester, UK
foxcj@essex.ac.uk

ABSTRACT

Text summarisation is the process of distilling the most important information from a source to produce an abridged version for a particular user or task. This demo presents the use of profile-based summarisation to provide contextualisation and interactive support for site search and enterprise search. We employ log analysis to acquire continuously updated profiles to provide profile-based summarisations of search results. These profiles could be capturing an individual's interests or those of a group of users. Here we look at acquiring profiles for groups of users.

1. MOTIVATION

Summarisation is a broad area of research [8]. The sort of information contained in a summary differs according to the mechanism used in the summarisation process: It may highlight the basic idea (generic summarisation), or it may highlight the specific user's individual area of interest (personalised summarisation). One of the techniques used to achieve personalisation is user profiling. User profiles may include the preferences or interests of a single user or a group of users and may also include demographic information [4]. Normally, a user profile contains topics of interest to that single user. We are interested in capturing profiles not of single but groups of users.

We utilise query and click logs to acquire a profile reflecting the population's search patterns and this profile is being automatically updated in a continuous learning cycle. We are then applying the acquired profiles in the summarisation process to support users searching a document collection. The potential of personalised summarisation over generic summaries has already been demonstrated, e.g. [3], but summarisation of Web documents is typically based on the query rather than a full profile, e.g. [11, 9]. Our specific interest lies in enterprise search which is different from Web search and has attracted less attention [5]. The benefit of this context is that we can expect a more homogeneous population of searchers who are likely to share interests and information needs. Our hypothesis is that profile-based summarisation can help a user in this process and guide the user to the right documents more easily (e.g. by presenting the summaries instead of or alongside snippets).

2. METHODS AND EXAMPLES

The demo presents an integrated Solr-based search system applying a number of different methods for building summaries for search results. The first two algorithms were designed for traditional (generic) summarisation, and they represent widely used baselines, e.g. [12]. The other three are all variations of an approach that has been proposed in the literature for building an adaptive community profile/domain model, a *"biologically inspired model based on ant colony optimisation applied to query logs as an adaptive learning process"* [1]. The approach is simple to implement, the idea here is that query logs are segmented into sessions and then turned into a graph structure. Figure 1 gives an example of part of the profile as it has been derived from our query logs. We used the log files collected on the existing search engine over a period of three years[1] to bootstrap this ant colony optimisation (ACO) model, i.e. our profile. The example illustrates the domain-specific nature of the derived profiles, e.g. the University library is named after Albert Sloman.

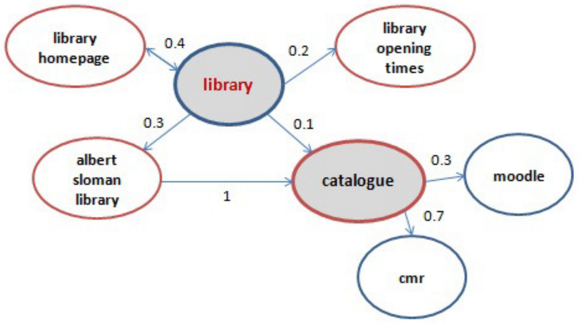


Figure 1: Partial profile derived from query logs.

A profile-based (extractive) summary of a document is then generated by turning the profile into a flat list of terms (we use three different methods to do this as explained further down) and selecting those sentences from the document

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

[1] More than 1.5 million queries, described in more detail elsewhere [6]

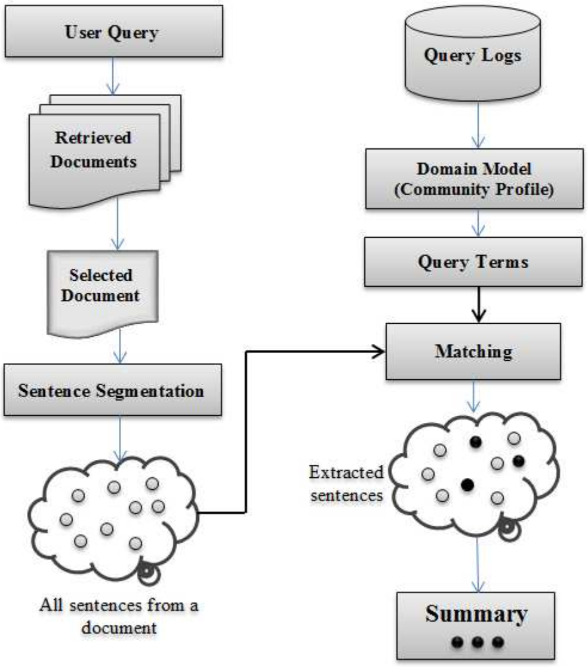


Figure 2: Architecture of profile-based single-document summariser.

that are most similar to the profile using cosine similarity. Figure 2 shows an architectural diagram for our profile-based summarisation system. Following DUC 2002 convention we select 100-word abstracts [7]. This gives us the following five methods:

1. **Random:** Selects sentences from the document randomly [12].
2. **Centroid:** A centroid-based approach to summarisation [10]. This algorithm takes into account first-sentence overlap and positional value, which are then used to generate a coherent summary.
3. **ACO:** A query graph built by processing the log data according to [1]. The entire model is turned into a flat list of terms for summarisation.
4. **ACO trimmed:** Starting with *ACO* we trim all those edges whose weights fall below the overall average weight of an edge. The remaining model is turned into a flat list of terms for summarisation.
5. **ACO query refinements:** The list of terms used for summarisation are all those that are directly linked to the query node in the ACO model.

Note that **ACO** and **ACO trimmed** are query-independent as they are using the *entire* model to generate the summary, whereas **ACO query refinements** is query-specific (for the frequently submitted query *"library"* the selected terms are *library*, *albert sloman library*, *library homepage*, *library opening times* and *catalogue*, see Figure 1).

To illustrate the different summaries obtained using three of the summarisation methods we apply the methods to the University of Essex Library homepage[2] and get the following summaries:

- **Random:** *We provide you with first class library facilities to complement and assist your studies. The Library allows you to access more than one million books, periodicals and microfilms. Our collections covering Latin America, Russia and Eastern Europe are of national significance. Viewing facilities for DVDs and videos are also available.*
- **Centroid:** *We provide you with first class library facilities to complement and assist your studies. You can find out more by accessing our University library services. Our Albert Sloman Library is just a few minutes' walk from teaching buildings and student accommodation at our Essex Campus.*
- **ACO query refinements:** *In addition, 110 networked PCs and terminals provide access to over 47,000 online journals, databases, e-books and library catalogues. Students at our Essex Campus can visit the Albert Sloman Library or borrow books from its collection via a daily dispatch service. The Albert Sloman Library has long opening hours, a total of 84 hours over seven days a week during term and 42.5 to 84 hours in vacations.*

Obviously, the actual usefulness of such summaries can only be assessed in a realistic search setting. In a pilot study we found that the ACO-based summaries have the potential of outperforming the different baselines [2]. A task-based evaluation using TREC Interactive Track guidelines is currently being conducted. As the immediate next step, we are interested in investigating how the profile can be integrated into multi-document summarisation.

3. REFERENCES

[1] M-D. Albakour, U. Kruschwitz, N. Nanas, D. Song, M. Fasli, and A. De Roeck. Exploring ant colony optimisation for adaptive interactive search. In *Proceedings of ICTIR*, pages 213–224. Springer, 2011.

[2] A. Alhindi, U. Kruschwitz, and C. Fox. A pilot study on using profile-based summarisation for interactive search assistance. In *Proceedings of ECIR*, pages 672–675, 2013.

[3] A. Díaz and P. Gervás. User-model based personalized summarization. *Information Processing & Management*, 43(6):1715–1734, 2007.

[4] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. *The Adaptive Web*, pages 54–89, 2007.

[5] D. Hawking. Enterprise Search. In R. Baeza-Yates and B. Ribeiro-Neto, editors, *Modern Information Retrieval*, pages 641–683. Addison-Wesley, 2nd edition, 2011.

[6] U. Kruschwitz, D. Lungley, M-D. Albakour, and D. Song. Deriving Query Suggestions for Site Search. *JASIST*, 2013. Forthcoming.

[7] C.Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL*, pages 71–78. ACL, 2003.

[8] A. Nenkova and K. McKeown. *Automatic summarization*. Now Publishers, 2011.

[9] S. Park. Personalized summarization agent using non-negative matrix factorization. *PRICAI 2008: Trends in Artificial Intelligence*, pages 1034–1038, 2008.

[10] D.R. Radev, H. Jing, M. Stys, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[11] C. Wang, F. Jing, L. Zhang, and H.J. Zhang. Learning query-biased web page summarization. In *Proceedings of CIKM*, 2007.

[12] R. Yan, J.Y. Nie, and X. Li. Summarize what you are interested in: An optimization framework for interactive personalized summarization. In *In Proceedings of EMNLP*, pages 1342–1351, 2011.

[2] http://www.essex.ac.uk/life/library/

AVResearcher: Exploring Audiovisual Metadata

Bouke Huurnink
Nederlands Instituut voor
Beeld en Geluid
Sumatralaan 45
Hilversum, The Netherlands

Amit Bronner
Nederlands Instituut voor
Beeld en Geluid
Sumatralaan 45
Hilversum, The Netherlands

Marc Bron
University of Amsterdam
Science Park 904
Amsterdam
The Netherlands

Jasmijn van Gorp
TViT, Utrecht University
Muntstraat 2A
Utrecht
The Netherlands

Bart de Goede
Dispectu
Julianaweg 61
Wijk aan Zee
The Netherlands

Justin Wees
Dispectu
Julianaweg 61
Wijk aan Zee
The Netherlands

{bhuurnink, abronner}@beeldengeluid.nl, m.m.bron@uva.nl
j.vangorp@uu.nl, {bart, justin}@dispectu.com

ABSTRACT

In this demonstration we present AVResearcher, a prototype aimed at allowing media researchers to explore metadata associated with large numbers of audiovisual broadcasts. It allows them to compare and contrast the characteristics of search results for two topics, across time and in terms of content. Broadcasts can be searched and compared not only on the basis of traditional catalog descriptions, but also in terms of spoken content (subtitles), and social chatter (tweets associated with broadcasts). AVResearcher is a new and ongoing valorisation project at the Netherlands Institute for Sound and Vision.

1. INTRODUCTION

In this demonstration we present AVResearcher, a prototype aimed at allowing media researchers to explore the professional, content-based, and social metadata associated with a collection of hundreds of thousands of broadcasts. With the continuous online production and storage of audiovisual broadcasts, a challenge for media researchers has arisen. There is a large amount of archival metadata about broadcasts becoming available. In addition, metadata from additional sources is becoming available. For example, the Netherlands Institute for Sound and Vision has over 960,000 catalog entries, and has an archive of subtitles for a subset of television broadcasts going back to 1989. In addition, members of the public write about broadcasts on Twitter, in the Netherlands sometimes amounting to tens of thousand of tweets for an individual program. Our prototype addresses this challenge, allowing media researchers to examine the metadata characteristics of sets of broadcast results.

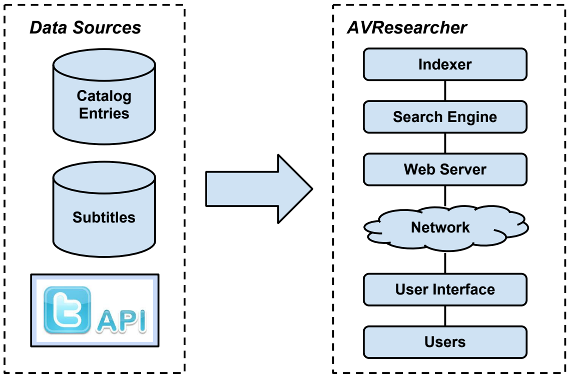


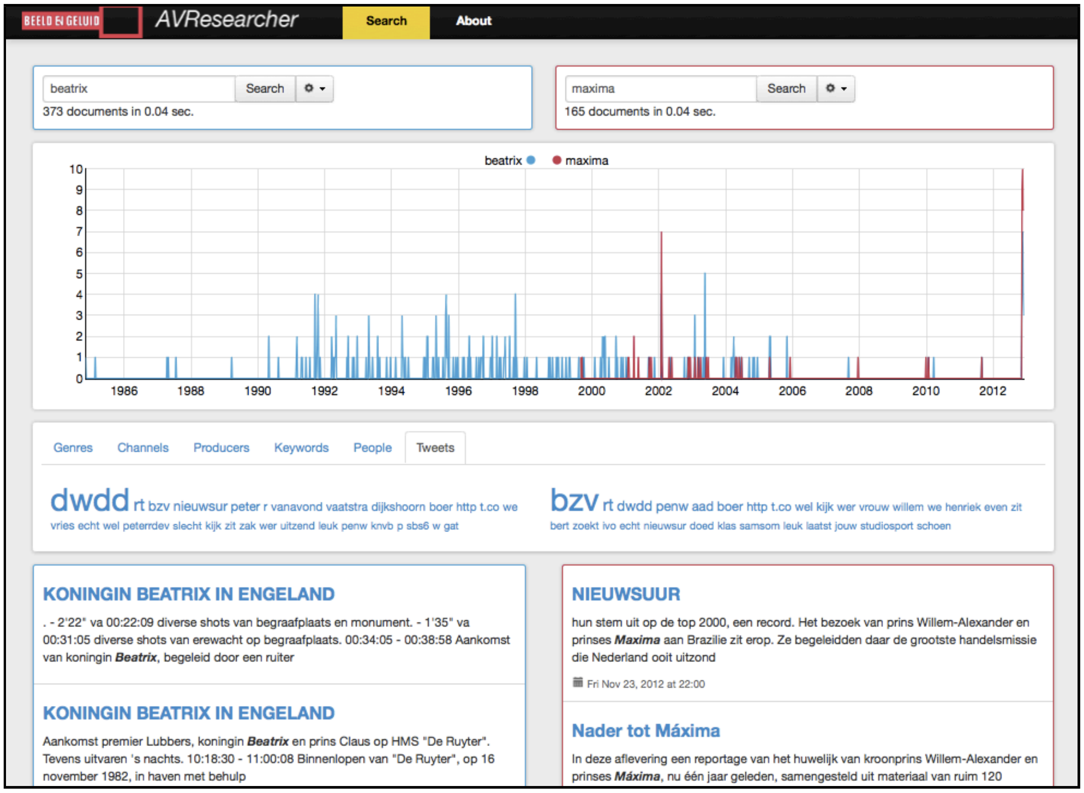
Figure 1: AVResearcher system overview.

AVResearcher is based on the Media Researchers Data Exploration Suite (MeRDES) [1], which was developed specifically to support media studies researchers to explore audiovisual catalog entries. In addition to the professional catalog entries supported by MeRDES, AVResearcher allows media researchers to explore social chatter in the form of tweets, and spoken content in the form of subtitles. In addition, the code of AVResearcher has been completely rewritten for improved speed and scalability. It is a new valorisation project at the Netherlands Institute for Sound and Vision, and as such is under active development. It is undergoing iterative development using Agile methods: user feedback is used to determine the requirements and their prioritisation for each iteration. After the second iteration has been accepted the prototype will be made available to media professionals through an online portal of the archive. At DIR 2013 we will present the current version of the software.

2. AVRESEARCHER SYSTEM

An overview of the AVResearcher system is given in Figure 1. Here we briefly summarize the system in terms of the

DIR 2013, April 26, 2013, Delft, The Netherlands.


BEELD EN GELUID
AVResearcher
Search
About
beatrix
Search
373 documents in 0.04 sec.
maxima
Search
165 documents in 0.04 sec.
beatrix ●
● maxima
0
1
2
3
4
5
6
7
8
9
10
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
Genres
Channels
Producers
Keywords
People
Tweets
dwdd rt bzv nieuwsur peter r vanavond vaatstra dijkshoorn boer http t.co we vries echt wel peterrdev slecht kijk zit zak wer uitzend leuk penw knvb p sbs6 w gat
bzv rt dwdd penw aad boer http t.co wel kijk wer vrouw willem we henriek even zit bert zoekt ivo echt nieuwsur doed klas samsom leuk laatst jouw studiosport schoen

KONINGIN BEATRIX IN ENGELAND

. - 2'22" va 00:22:09 diverse shots van begraafplaats en monument. - 1'35" va 00:31:05 diverse shots van erewacht op begraafplaats. 00:34:05 - 00:38:58 Aankomst van koningin ***Beatrix***, begeleid door een ruiter

KONINGIN BEATRIX IN ENGELAND

Aankomst premier Lubbers, koningin ***Beatrix*** en prins Claus op HMS "De Ruyter". Tevens uitvaren 's nachts. 10:18:30 - 11:00:08 Binnenlopen van "De Ruyter", op 16 november 1982, in haven met behulp

NIEUWSUUR

hun stem uit op de top 2000, een record. Het bezoek van prins Willem-Alexander en prinses ***Maxima*** aan Brazilie zit erop. Ze begeleidden daar de grootste handelsmissie die Nederland ooit uitzond

Fri Nov 23, 2012 at 22:00

Nader tot Máxima

In deze aflevering een reportage van het huwelijk van kroonprins Willem-Alexander en prinses ***Máxima***, nu één jaar geleden, samengesteld uit materiaal van ruim 120

Figure 2: Main result exploration screen of AVResearcher, with two queries compared side-by-side.

underlying data set, architecture, and visualization.

Data Set Catalog descriptions of the broadcasts are obtained from the archive of the Netherlands Institute for Sound and Vision: at the time of writing the collection consists of just over 960,000 broadcasts. Subtitles are obtained through an agreement with the Netherlands public broadcasters from November 2012 onwards. In the future we also plan to incorporate a legacy database of subtitles dating back to December 1989. Tweets about programs also date from November 2012 onwards. They are obtained using the Twitter Streaming API[1]: we monitor a collection of official hashtags for 25 Dutch television shows, obtained from the website `http://hekjeplekje.nl`. If a tweet occurs during a television broadcast, it is associated with that broadcast.

Architecture Data for television broadcasts is collected from the three different sources: catalog entries maintained by the archive, subtitles obtained from the Netherlands Public Broadcasting system, and tweets from Twitter. The data is stored and indexed for use by an open-source search system.[2] The user interface is made available on the web-server. Users can interact with the interface over a secure network connection.

Visualization The AVResearcher interface, shown in Figure 2, allows users to issue two search queries and compare the results side-by-side. For each query the user can view:

- The number of broadcasts containing the query terms on a timeline. The hits for each query are visualised on the same timeline, and given a different color. This allows researchers to see how two given topics (represented by queries) have evolved over time.
- Term clouds of frequently occurring terms in the results, divided into facets from the catalog entries (genres, channels, producers, keywords, and people), as well as words frequently occurring in subtitles and tweets.
- The list of search results used to generate the timeline and term cloud. When users click a search result they can see more details for that particular broadcast.

3. CONCLUSION

AVResearcher is a prototype that addresses the problem of exploring different kinds of broadcast metadata on a large scale. It allows media studies researchers to explore and compare metadata for two different topics in a collection of hundreds of thousands of broadcasts. It includes subtitles and tweets, as well as professional catalog data, and in this way allows media studies researchers to explore spoken content and social chatter about broadcasts. The system is under active development, and will be used to perform user studies aimed at improving archival access. At DIR 2013 we will present the current version of the prototype.

4. REFERENCES

[1] M. Bron, J. van Gorp, F. Nack, M. de Rijke, and S. de Leeuw. A subjunctive exploratory search interface to support media studies researchers. In *SIGIR '12: 35th international ACM SIGIR conference on Research and development in information retrieval,*, pages 425–434, Portland, Oregon, 2012. ACM, ACM.

[1] `https://dev.twitter.com/docs/api`

[2] We use the ElasticSearch search engine, `http://elasticsearch.org`, which scales to our needs.

Readability of the Web: A study on 1 billion web pages.

Marije de Heus
University of Twente
m.deheus@student.utwente.nl

Djoerd Hiemstra
University of Twente
hiemstra@cs.utwente.nl

ABSTRACT

We have performed a readability study on more than 1 billion web pages. The Automated Readability Index was used to determine the average grade level required to easily comprehend a website. Some of the results are that a 16-year-old can easily understand 50% of the web and an 18-year old can easily understand 77% of the web. This information can be used in a search engine to filter websites that are likely to be incomprehensible for younger users.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Selection process; H.1.2 [**User/Machine Systems**]: Human information processing

General Terms

Algorithms, Measurement, Human Factors

Keywords

Readability, ARI, Code Crawl, MapReduce

1. INTRODUCTION

The internet has users of all ages. Some texts are more easily readable by young users than others. In general, texts that have longer sentences with longer words which contain more syllables, are less likely to be easily understood by young users than texts with shorter sentences that consist of short words. This paper analyzes the readability of the web, as part of the Norvig Web Data Science Award[1].

There are several measures to compute the readability of a text, such as Flesch-Kincaid readability[10], Gunning Fog index[9], Dale-Chall readability[5], Coleman-Liau index[6], SMOG[11] and the Automated Readability Index[12]. Most of these use a formula that requires counting the number of syllables. Deciding where a syllable begins and ends is a difficult problem, depending on the language. Therefore we chose to use the Automated Readability Index, which was designed for real-time computation of readability on electronic typewriters and does not use the number of syllables. Instead it uses the average number of characters per word and the average number of words per sentence. The outcome represents the US grade level that is needed to easily comprehend the text.

DIR 2013, April 26, 2013, Delft, The Netherlands.
.

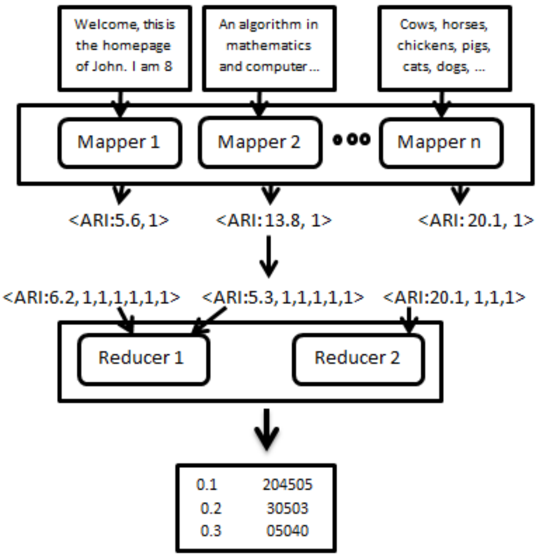


Figure 1: Visual overview of the MapReduce program

The ARI formula[12] is shown below.

$$ARI = 4.71 * \frac{characters}{words} + 0.5 * \frac{words}{sentences} - 21.43 \quad (1)$$

So far most of the research regarding readability of websites has focused on legal documents and health documents [2][8][3]. No previous experiments with readability large numbers of websites have been found. The goal of our research is to examine the readability of the web. For this purpose, we ran a MapReduce program on more than a billion webpages. The Common Crawl dataset consists among others of 61 million domains, 92 million PDF Docs and 7 million Word Docs. More than 60% of the data came from .com TLD's, with .org and .net on second and third place. Thereafter came .de, co.uk, .ru, .info, .pl, .nl et cetera[1]. We did not filter non-English websites.

2. IMPLEMENTATION

The program was implemented using MapReduce[7] on Hadoop[4]. Figure 2 provides a visual overview of our program. The mapper takes the text of a website without html tags. It computes the ARI of the text. It then emits this ARI and a count of 1. The reducer receives an ARI score

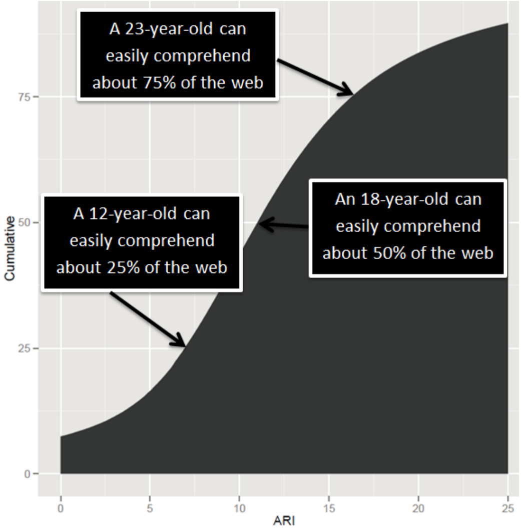


Figure 2: Cumulative results

and a number of counts. It sums the counts and writes the ARI and the sum to one line of the output file.

3. RESULTS

Figure 3 shows the cumulative results. This graph answers questions such as ´how much of the web can a 12-year-old (grade 6) easily comprehend?' (answer: about 20%).

4. DISCUSSION, CONCLUSION AND FUTURE WORK

4.1 Discussion

Very low results.

7% of the websites received a score below 0. 1.7% of the websites was empty. These results cannot be interpreted in terms of a US grade level. However, we can infer that these websites are probably easily readable for all ages, because such websites must have very short sentences and very short words.

Very high results.

13.3% of the websites received a score higher than 22. This means that a person would need more than 22 years of education to easily comprehend the website. Some of these even got scores above 100. A lot of these websites consist of enumerations of items, dates, addresses et cetera, which are not stripped. It is not clear what effect such items have on the readability. Maybe they should be ignored when computing the readability, or maybe they do influence readability. Some of these enumerations may be detected by certain html list tags, while others may not be removed as easily.

Non-English Languages.

In our analysis, we did not filter non-English websites. Automated Readability Index was not designed for English specifically, but Smith and Senter [12] only experimented with the English languages. We did not find studies on how accurate ARI is for other languages.

4.2 Conclusion

This paper presented an anlysis of the readability of the web using ARI and MapReduce. The results (presented in figure 3) depend on the reliability of ARI for web pages of different languages and can be used in a search engine to adjust search results to a user's education level.

4.3 Future Work

ARI for non-English texts.

We did not find literature on the accuracy of ARI for non-Enlgih languages. This needs to determined before ARI can be used in (multilingual) practice.

Readability of web pages.

Some of the high ARI scores may be due to the structure of some websites, e.g. long enumerations and lists of items. A readability measure like ARI may not be reliable on such websites. More research can be done on how the readability of a web page can be accurately determined.

5. REFERENCES

[1] Norvig Web Data Science Award. `http://norvigaward.github.io/index.html`, 2013.

[2] G.K. Berland, M.N. Elliott, L.S. Morales, J.I. Algazy, R.L. Kravitz, M.S. Broder, D.E. Kanouse, J.A. Muñoz, J.A. Puyol, M. Lara, et al. Health information on the internet. *JAMA: the journal of the American Medical Association*, 285(20):2612–2621, 2001.

[3] E.V. Bernstam, D.M. Shelton, M. Walji, and F. Meric-Bernstam. Instruments to assess the quality of health information on the world wide web: what can our patients actually use? *International journal of medical informatics*, 74(1):13–20, 2005.

[4] D. Borthakur. The hadoop distributed file system: Architecture and design. *Hadoop Project Website*, 11:21, 2007.

[5] J.S. Chall. *Readability revisited: The new Dale-Chall readability formula*, volume 118. Brookline Books Cambridge, MA, 1995.

[6] M. Coleman and T.L. Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283, 1975.

[7] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[8] M.A. Graber, C.M. Roller, B. Kaeble, et al. Readability levels of patient education material on the world wide web. *The Journal of family practice*, 48(1):58, 1999.

[9] R. Gunning. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13, 1969.

[10] J.P. Kincaid, R.P. Fishburne Jr, R.L. Rogers, and B.S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document, 1975.

[11] G.H. Mc Laughlin. Smog grading-a new readability formula. *Journal of reading*, pages 639–646, 1969.

[12] R.J. Senter and E.A. Smith. Automated readability index. Technical report, DTIC Document, 1967.

Relating Political Party Mentions on Twitter with Polls and Election Results

Eric Sanders
CLS/CLST, Radboud University Nijmegen
Erasmusplein 1
6525 HT Nijmegen
+31 24 3616087
e.sanders@let.ru.nl

Antal van den Bosch
CLS, Radboud University Nijmegen
Erasmusplein 1
6525 HT Nijmegen
+31 24 3611647
a.vandenbosch@let.ru.nl

ABSTRACT

In each of the last ten days preceding the parliamentary elections of 2012 in the Netherlands at least one election poll was published. Throughout the same period close to 170 thousand Dutch microtext messages with references to political parties were posted on Twitter, the microblogging platform. In this study we investigate whether these tweets can serve as an addition to, or even an alternative for the traditional polls as predictors of the election outcomes. We show that counts of mentions of political party names are strongly correlated with the polls and the election results. While polls remain more accurate as a predictor of the outcome (a mean absolute error of 1.1% and a correlation of about 0.98 with the actual percentage of votes cast for all parties), the Twitter statistics show a mean absolute error of 1.9% when aggregated over a number of days, and display a high correlation with elections and polls (in both cases, r≈0.95). We conclude that tweet mention counts form a good complementary basis for predicting election results.

Categories and Subject Descriptors

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

General Terms

Human Factors, Languages.

Keywords

Twitter, Elections, Polls.

1. INTRODUCTION

With a current average of about a half billion messages posted daily, Twitter hosts a massive amount of accessible messages, which in turn harbor vast amounts of information. Tweets are often related to personal affairs, but may also refer to popular events. One of the interesting uses of the information in tweets is to try to determine people's opinions about certain matters. Politics is an attractive subject to try to get opinions about from tweets. In terms of events, political elections typically evoke the posting of tweets containing political views.

A conventional way of assessing average opinions about politics during election periods is polling. The standard polling method is to ask a small but representative part of the population what party or person one is planning to vote for. On Twitter people give this information without being prompted. It would be an interesting addition to (or even alternative to) polls if we could extract this information from tweets. The most challenging part of it is to gather a balanced representation from the tweets of the people participating in the elections. In essence this is impossible; while the legal voting age in the Netherlands is 18, many users on Twitter have not reached that age, but demographic information regarding individual users is not available in any trustworthy way on Twitter. The sheer magnitude of data available on Twitter may compensate for this partly unrepresentative information.

In this paper a comparison between the predictive potential of tweets and polls with respect to the outcome of the Dutch parliament elections of 12 September 2012 is presented. The number of times a political party is mentioned in a Dutch tweet is compared to the polls and the election results without normalization. This was done for all eleven parties that won at least one seat in the parliament. The next sections discuss related work, describe the data, explain the experiment, discuss the results, and draw conclusions.

2. RELATED WORK

Work that has focused on predicting election outcomes through social media mining offers a mixed bag of results. Tumasjan *et al* [1] show that for the six biggest parties in the election of the German parliament held on 27 September 2009, the percentage of tweets in which a party is mentioned between 13 August and 19 September 2009 highly correlates with the election result of that party. Their particular selection of parties and the period over which counts were gathered is questioned in a responding paper by Jungherr *et al* [2]. They claim that the choices made by Tumasjan et al give overly optimistic results on badly grounded heuristics.

O'Connor *et al* [3] compare the sentiment ratio of tweets containing 'obama' with presidential job approval polls in 2009 and presidential election polls in 2008. The ratio correlates well with the first poll but does not with the latter. Marchetti-Bowick and Chambers [4] build on the work of O'Connor et al. and use distant supervision for both topic identification and sentiment analysis. The comparison of the results with Obama's job approval poll gives better correlation than earlier work.

Tjong Kim Sang and Bos [5] compare tweet mentions and election results for the Dutch senate elections of 2011. Beyond raw counts of tweets they test and compare the predictive power of four alternative counting methods, but they do not find large improvements with these methods.

The novelty of the work described in this paper is that it is based on a relatively large number of consecutive polls on each of the ten days before the elections.

Gayo-Avello [6] pinpoints a couple of problems with predicting elections based on tweets and gives some suggestions. Apart from those addressed in this paper, he indicates that only good results are published and analyzing afterwards is not predicting.

3. DATA

The Twitter data used in the experiments is taken from a substantial archive of Dutch tweets collected within the TwiNL project (ifarm.nl/erikt/twinl). The FAQ of the related search website twiqs.nl states that an estimated 40% of all Dutch tweets are collected since December 16, 2010. The present study makes use of all tweets gathered between September 2 to September 12, 2012, for which between 2.0 and 2.4 million tweets per day have been archived.

The poll data is taken from the website Alle Politieke Peilingen (www.allepeilingen.com) that has saved the poll results from 2000 onwards of the six most cited polling institutes in the Netherlands. These are: peil.nl, TNS NIPO, de politieke barometer, buzzpeil.nl, de Stemming and NOS Peilingwijzer. All these polls try to predict the result of the elections (if the elections were held on the day of the poll).

4. EXPERIMENT

For the eleven parties that won one or more seats in parliament we counted how often the party name was mentioned in a tweet in the ten days before the elections and on election day, 12 September 2012. This was done with a basic pattern match. First it was investigated by which names parties are mentioned in the tweets. Most parties are almost exclusively mentioned by their abbreviation and rarely by their full name. Most full names are therefore ignored. For instance, the acronym of the VVD occurs over thousand times more often than its full name, ‘Volkspartij voor Vrijheid en Democratie’. However, two parties are often mentioned by their full name: GroenLinks and ChristenUnie. Their respective abbreviations can also have other meanings: GL being a typical English shorthand for ‘good luck’ and CU for ‘see you’, but a manual inspection revealed that these abbreviations are rarely used in these meanings.

We needed to generate several specific pattern-matching expressions. Three parties have ‘van de’ (‘of the’) or ‘voor de’ (‘for the’) in their full name which can be expressed in many ways, e.g. ‘vd’, ‘v.d.’, ‘v/d’, ‘van de’, ‘v d’, which are all represented in the search pattern that was used. Matching is case-insensitive, so ‘SGP’, ‘sgp’, ‘Sgp’ etc. are all recognised. No effort was made to find misspelled party names. The party names can be preceded by ‘@’ (Twitter account names) or ‘#’ (Twitter hashtags) and preceded or followed by punctuation.

Table 1 lists the resulting regular expressions for the parties.

During the period of ten days before the election, for each day and each party, the percentage of tweets in which a party is mentioned is compared to the result of the average of all polls that came out that day. This was done to investigate how much the percentage of party mentions in tweets resembles the polls. Subsequently, the results of the averaged polls on the day before the election and the election results are compared to each other and to the tweet mentions of (1) election day, (2) the day before election day, (3) an aggregate of all tweets during the 10-day period before the elections, and (4) an aggregate over a 5-day period before the elections.

Table 1. The regular expressions that were used to detect the party names in the tweets

<table>
<tr><th>Party</th><th>Regular Expression Pattern</th></tr>
<tr><td>VVD</td><td>"vvd"</td></tr>
<tr><td>PVDA</td><td>"pvda","partij\s+v(oor\s+|an\s+|.)?d(e|.)?\s+arbeid"</td></tr>
<tr><td>SP</td><td>“sp”</td></tr>
<tr><td>PVV</td><td>"pvv","partij\s+v(oor\s+|an\s+|.)?d(e|.)?\s+vrijheid"</td></tr>
<tr><td>CDA</td><td>“cda”</td></tr>
<tr><td>D66</td><td>"d\'?66"</td></tr>
<tr><td>GL</td><td>"gl","groen.?links"</td></tr>
<tr><td>CU</td><td>"cu","christen.?unie"</td></tr>
<tr><td>SGP</td><td>“sgp”</td></tr>
<tr><td>PVDD</td><td>"pvdd","partij\s+v(oor\s+|an\s+|.)?d(e|.)?\s+dieren"</td></tr>
<tr><td>50PLUS</td><td>"50[^\d]?(\+|plus)"</td></tr>
</table>

An average of 0.7% of all daily tweets posted throughout the last ten days before the election mentions at least one political party. Table 2 shows that these nearly 170 thousand tweets are not uniformly divided over the eleven days; about one third of all tweets is posted on election day, and more tweets are posted closer to election day.

5. RESULTS

First, comparisons are shown in three figures (Figures 1, 2, and 3) between daily percentages of Twitter mentions and daily poll results of selections of two or three parties during the ten days before the elections.

The daily percentage of Twitter mentions for a particular party is computed as follows:

$$Perc = 100 * \#mentions_p / \sum_p \#mentions_p$$

where $\#mentions_p$ is the number of mentions of a particular party, and $\sum_p \#mentions_p$ is the total number of mentions of all eleven parties. The counts thus represent mentions, not tweets: if in a tweet two parties are mentioned, the tweet is counted twice.

The percentages of poll results are computed from the predicted number of parliament seats, which is the statistic by which they are reported and stored. As there are 150 seats in the Dutch parliament, each seat stands for 0.67%. The percentage used here is the mean percentage of the predicted number of seats of all polling institutes that released a prediction that day. For some days there is a poll of only one institute. The predictions of the polling institutes differ slightly. The largest difference between two predictions from poll estimates for the same party on the same day is 4.7%. On the day before the elections, 11 September, all polling institutes published results.

Table 2. Number of tweets with at least one political party mentioned in the 10 days before elections and on election day (0 days)

Days before election	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
#tweets	56,580	24,004	17,224	8,498	12,011	10,178	9,373	9,062	10,317	8,048	4,700

5.1 Twitter vs Polls Correlation

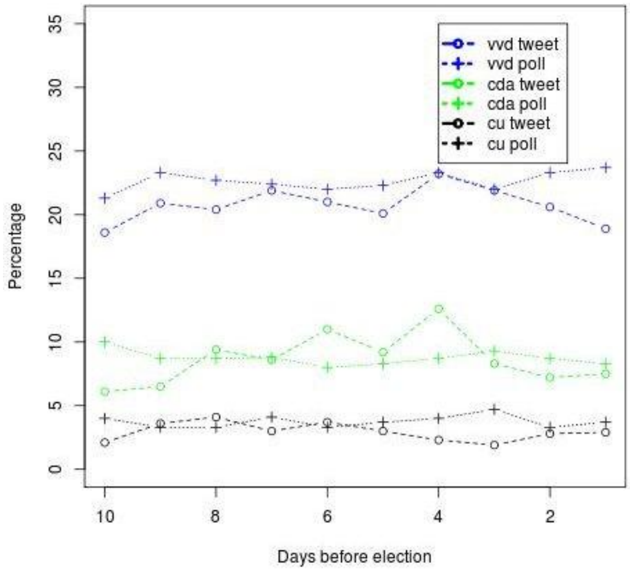


Figure 1. Twitter mentions and poll results for VVD, CDA and CU

Figure 1 displays the results for VVD, the party that won the elections, CDA, a middle party, and ChristenUnie (CU), a small party. The figure exemplifies the fairly strong correlation of the percentages of Twitter mentions and poll results during the whole period.

5.2 Twitter vs Polls Outliers

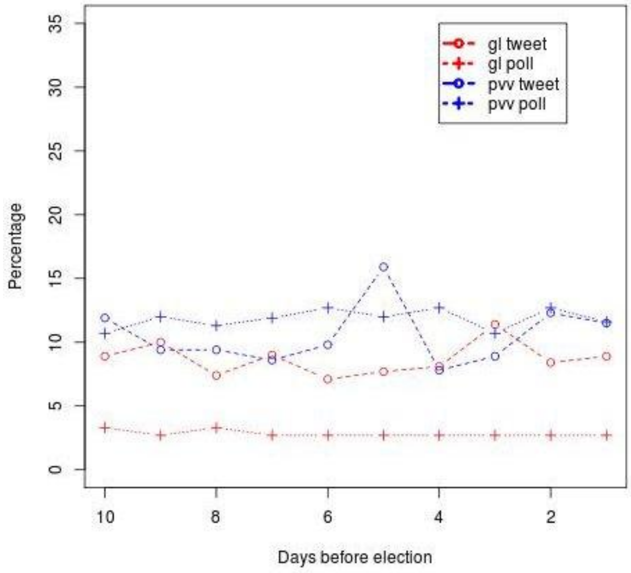


Figure 2. Twitter mentions and poll results for PVV and GL

This trend is typical for all but one party, GroenLinks (GL), as shown in Figure 2. For comparison, the GroenLinks estimates are compared against the predictions for the PVV. The figure displays an unexpected difference between the Twitter mentions and poll results for GroenLinks. This party is well known for its above-average use of and presence on social media in their campaign [7]. As an aside, the figure also shows a relatively high peak in the Twitter mentions of the PVV five days before the elections. This may be explained by the news that day that the PVV had falsely declared money from the European Union, while their campaign was outspokenly anti-Europe.

5.3 Twitter vs Polls Trend

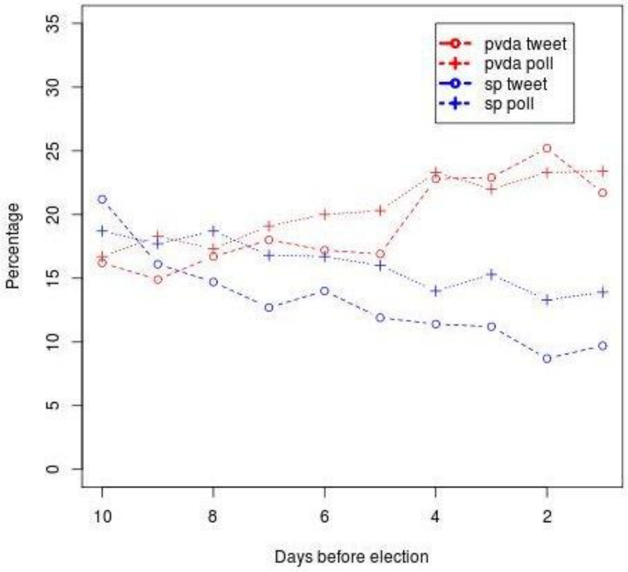


Figure 3. Twitter mentions and poll results for PVDA and SP

Figure 3 shows how both in the Twitter mentions as in the poll results the PvdA, one of the socialist parties and runner-up in the election results, was gaining in the last ten days before the elections while the SP, another socialist party, was losing voters. The SP started out popular, but in the debates the PvdA leader was doing well, while the SP leader’s debating was considered disappointing.

5.4 Twitter vs Polls vs Election

Table 3 shows for all parties the difference between the election results on 12 September, the mean result of all polls on the day before the elections, and the relative percentage of tweets the party was mentioned on (1) election day, (2) the day before, (3) during all ten days and (4) during five days before the elections. The fourth and second rows from below list the mean absolute error (MAE) of the column with the election results (2nd column) and with the polls of the pre-election day (3rd column). The third last and final row show the correlation and the 95% confidence interval with the election and poll results.

The MAE of the polls with the election results is smaller than the MAE of the tweet mentions with the election results in all cases, meaning that polls are a better predictor of the election results than raw counts of party names in tweets. The table also shows that tweet mentions of a time span of several days (five or ten) before the elections are closer to the election results than the tweet mentions on one specific day (election day or the day before). Tweet mentions gathered during five days before the elections are closer to the election results than all tweet mentions from ten days before the election results. Finally, the correlation coefficient and the confidence interval show the same trend as the MAE, and are very high in all cases; 0.93 or higher.

Table 3. Comparison between election results, polls and tweets from different time slots in %

Party	**Election 12 Sep**	**Polls 11 Sep**	**Tweet 12 Sep**	**Tweet 11 Sep**	**Tweet 2-11 Sep**	**Tweet 7-11 Sep**
VVD	*26.8*	23.7	24.6	18.9	20.7	20.6
PVDA	*25.1*	23.4	18.5	21.7	20.2	22.2
PVV	*10.2*	11.6	13.6	11.5	10.7	11.4
SP	*9.8*	13.9	8.7	9.7	12.0	10.3
CDA	*8.6*	8.3	6.0	7.5	8.6	8.6
D66	*8.1*	7.9	9.8	9.7	9.0	8.5
CU	*3.2*	3.7	2.6	2.9	3.0	2.7
GL	*2.4*	2.7	7.0	8.9	8.6	8.8
SGP	*2.1*	1.7	3.2	4.4	2.9	2.8
PVDD	*2.0*	1.8	3.6	3.5	3.2	3.2
50PLUS	*1.9*	1.7	2.4	1.3	1.1	1.1
MAE elections		*1.1*	*2.4*	*2.4*	*2.2*	*1.9*
Corr elections		*0.98* *(0.93 -1.0)*	*0.95* *(0.82-0.99)*	*0.94* *(0.78-0.98)*	*0.95* *(0.83-0.99)*	*0.96* *(0.84-0.99)*
MAE poll	*1.1*		*2.4*	*2.3*	*2.0*	*1.7*
Corr poll	*0.98* *(0.93-1.0)*		*0.93* *(0.76-0.98)*	*0.94* *(0.78-0.98)*	*0.96* *(0.87-0.99)*	*0.96* *(0.83-0.99)*

6. DISCUSSION

The results of our comparative study on the 2012 Dutch parliament elections provide case-based evidence that tweets are a good basis for predicting election results. Purely on the basis of raw counts of party name mentions (with flexible pattern matching rules), without further domain knowledge, a strong correlation with the poll results can be observed (around 0.95). In a number of cases the difference between the Twitter mentions and the polls is larger than 5%, but the difference between the various polls is also almost 5% in a few cases. Although the polls more accurately predict the election outcome, the correlation between tweet-based estimates and the outcome is observed to be as high as 0.96, with a mean absolute error of only 1.9% (the polls attain 1.1%), provided that the tweet counts are aggregated over a number of days.

As Gayo-Avello rightly points out in his paper [6] our kind of approach lacks information that could improve the prediction of election outcomes or poll results based on Twitter. First, who is tweeting? If the Twitter account is from a party member or official the tweet could be filtered out as it may be used to steer social media opinions or even statistics. However, it is hard to ascertain whether a Twitter account is from a party member. Automatic profiling based on machine learning and text classification may help in this respect. Second, is the tweet polar or neutral? A Twitter user who will vote for a party is likely to compose positive tweets about that party. Automatic sentiment analysis (perhaps trained on political opinions to capture domain-specific sentiment markers) might be used to reweight counts.

Negation and hedging may be a third factor that could partially be determined automatically and improve estimates. A tweet such 'I will not vote for partyX' could then be left out of the count for partyX. This is a very challenging task, though. Morante and Daelemans [8] provide pointers on how this may be addressed. Fourth, can we account for factors that cause an increase in the number of tweets of a certain party? The detection of other events involving entities that also play a role in the focus event (such as the PVV scandal mentioned in the discussion of Figure 2) may be used to discount tweets about this event.

Finally, we observed that estimates based on counts aggregated over several days better approximated the election results than the counts on a specific day; five days seem to represent a reasonable aggregation window. A further study could be carried out to see whether an optimal time window can be found for events similar to the single case studied here.

We do not share Gayo-Avello's conclusion that elections cannot be predicted with Twitter, but acknowledge that further research has to be carried out before we say Yes we can! (predict elections with Twitter).

7. ACKNOWLEDGMENTS

We thank Ruut Brandsma from www.allepeilingen.com for providing the data from the polling institutes.

8. REFERENCES

[1] Tumasjan, A., Sprenger, T., Sander, P., Welpe, I. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. *Proceedings of the fourth International AAAI Conference on Weblogs and Social Media, Washington D.C., USA, 2010*

[2] Jungherr, A., Jürgens, P., Schoen, H. (2012). Why the Pirate Party Won the German Election of 2009 or The Trouble With Predictions: A Response to Tumasjan, A., Sprenger, T.O., Sander, P.G., Welpe, I.M. "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment", *Social Science Computer Review*

[3] O'Connor, B., Balasubramanyan, E., Routledge, B., Smith, N. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. *Proceedings of the fourth International AAAI Conference on Weblogs and Social Media, Washington D.C., USA, 2010*

[4] Marchetti-Bowick, M., Chambers, N. (2012). Learning for Microblogs with Distant Supervision: Political Forecasting with Twitter. *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, 2012*

[5] Tjong Kim Sang, E. and Bos. J. (2012). Predicting the 2011 Dutch Senate Election Results with Twitter, *Proceedings of SASN 2012, the EACL 2012 Workshop on Semantic Analysis in Social Networks, Avignon, France, 2012*

[6] Gayo-Avello, D. (2012). No, You Cannot Predict Elections with Twitter, *IEEE Internet Computing, vol. 16, no 6. pp. 91-94, Nov.-Dec. 2012*

[7] http://www.sax.nu/Portals/615/docs/Rapport_Saxion_Social_Media_Politiek_2012.pdf

[8] Morante, R., and Daelemans, W. (2009). A metalearning approach to processing the scope of negation. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pp. 21-2), 2009*

When is the Structural Context Effective?

Muhammad Ali Norozi
Dept. of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
mnorozi@idi.ntnu.no

Paavo Arvola
School of Information Sciences
University of Tampere
Tampere, Finland
paavo.arvola@uta.fi

ABSTRACT

Structural context surrounding the relevant information is intuitively and empirically considered important in information retrieval. Utilizing this context in scoring has improved the retrieval effectiveness. In this study we will objectively look into the significance of the *structural context* in contextualization process, and try to answer the core question of under which circumstances do we need to deal with the such types of context?

Categories and Subject Descriptors

H.3.3 [**Info. Search and Retrieval**]: Search process

1. INTRODUCTION

Document parts, referred to as elements, have both a hierarchical and a sequential relationship with each other. The hierarchical relationship is a partial order of the elements, which can be represented with a directed acyclic graph, or more precisely, a tree. In the hierarchy of a document, the upper elements form the context of the lower ones. In addition to the hierarchical order, the sequential relationship corresponds to the order of the running text. From this perspective, the context covers the surroundings of an element. An implicit chronological order of a document's text is formed, when the document is read by a user.

In focused retrieval, the use of context is a driving force to alleviate or "un-bias" the retrieval of items with varying length. Namely, information retrieval is based on evidence of the retrievable units at hand, and longer text units have indeed more textual evidence. This has led to a play-safe strategy where the larger elements are favoured by retrieval systems. How effective the context is to neutralize the side-effects or bias because of size or length (smaller elements with less textual evidence gets same opportunity to satisfy the users need), has been reported experimentally in many studies [1–3, 6, 9, 10, 8, 7]. The question asked here is: why the structural context is important in the retrieval of focused items? In addition, we also ask if the use of context, under certain circumstances (worst-case), would harm the retrieval. This means if the context is poor or even misleading.

2. CONTEXT

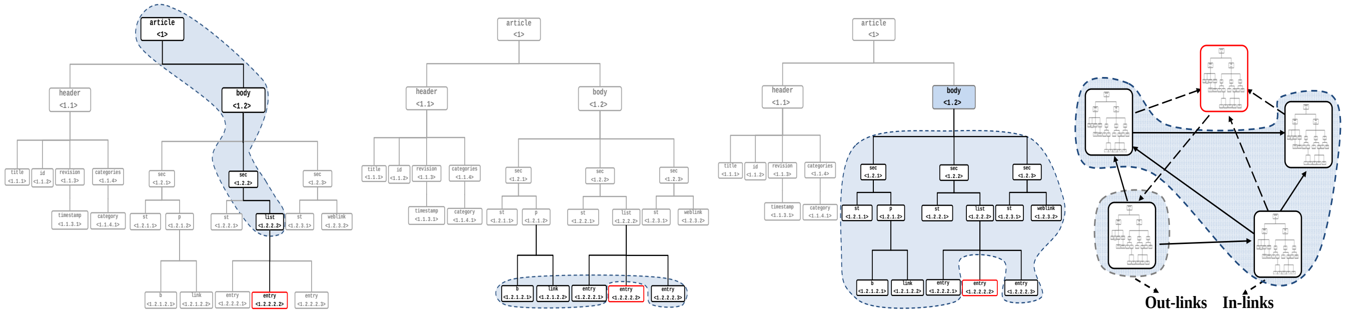
In semi-structured documents, context of an element covers everything in the document excluding the element itself. The surrounding items or elements of the relevant information is the *context*. The representation of the semi-structured documents aims to follow the established structure of documents, i.e., an academic book is typically composed of ⟨`chapters`⟩, ⟨`sections`⟩, ⟨`subsections`⟩ etc., structures. ⟨`chapter1`⟩ is followed by ⟨`chapter2`⟩ and within ⟨`chapter1`⟩, ⟨`section1`⟩ is followed by ⟨`section2`⟩. Elements ⟨`section1`⟩ and ⟨`section2`⟩ are siblings, and hence most likely, semantically related. The following element takes the concepts further from the preceding elements, and the preceding elements provide the basics or foundation for the following elements. Therefore, together in the document order, the *preceding* and *following* elements form a strong and connected perspective (the kinship structural context), surrounding the relevant information. Two general types of context can be distinguished based on the standard relationships. Hierarchical context, for one, refers to the ancestors, whereas horizontal refers to the preceding and following elements [3]. In existing studies, context has been referred to *externally* as the hyperlink structure of the elements as well. The context is *internal* when it is considered from within the document, and it is external when it is considered outside the document(s).

Contextualization [3] is a re-scoring model, where the basic score, usually obtained from a full-text retrieval model, of a contextualized document or element is re-enforced by the weighted scores of the contextualizing documents or elements (elements in the sub-tree of interest or structural context). In this section, we will formalize the context from in and outside the document using contextualization model.

2.1 Structural Context

Structural context is the *sub-tree of interest* from the hierarchical tree structure of the semi-structured document. *Internally*, in *hierarchical contextualization* [3], the intrinsic tree structure within the XML document is employed. Structural context in hierarchical or vertical contextualization is the context based on parent-child relationship in document's hierarchical structure. An element's parent or ancestors are accounted to be the structural context, while contextualizing the element itself. The sub-tree of interest is shown in Figure 1(a). *Horizontal contextualization* [3] takes into account the sibling elements in the document's hierarchical structure as the structural context. If we visualize the document's hierarchically tree structure, horizontal structural context is horizontal, as it is based on one level (the same level as the element to be contextualized) of the tree at a time (see Figure 1(b)). The most recent form of contextualization, the *Kinship contextualization* [7], is both horizontal (siblings) and vertical (ancestors & descendants

DIR 2013, April 26, 2013, Delft, The Netherlands.



(a) Hierarchical (b) Horizontal (c) Kinship (d) Citation

Figure 1: Structural context, the sub-tree of interest, example taken from [7]

elements) but intrinsically non-hierarchical perspective of the hierarchical information. Structural context is hence both vertical and horizontal in the document's hierarchical form, Figure 1(c).

And *externally*, in *citation contextualization* [8], the document's hyperlink structure is taken in to account. The structural context here is based on the hyperlinks' graph of documents hyper-linking (connecting) one another in form of inlinks (indegree) and outlinks (outdegree). In this case, the sub-graphs instead of tree of interest are the out-links graph and the in-links graphs (see Figure 1(d)).

2.2 Why Structural Context?

Structural context is the essential component of the Contextualization model [1]. With contextualization model, using the structural context, the aim is to rank higher an element in a good context (strong evidence in the structural context) than an identical element in a not so good context (less or no evidence in the structural context) within the document. And therefore, retrieve elements independent of their sizes. A small element, in term of size, can be viewed and hence scored in relation to its structural context, and its smaller size (which means having less evidence in total) doesn't stop it from being selected as one of the best results.

In order to cope up with the "biasness" issue (described earlier), in contextualization model, the weight of a relevant element is adjusted by the basic weights of the elements in the structural context (its contextualizing elements). In addition to basic weights, each element in the structural context of the contextualized element, should possess an *impact* factor. An higher impact factor shows the importance of the contextualizing element and vice versa. The role and relation of elements in the structural context are operationalized by giving the element a contextualizing weight. A contextualization vector is defined to capture the impact factor of each contextualizing element, and this contextualization vector is represented by a g function in Equation 1.

2.3 Contextualization and Random Walks

Random walk principle is employed, for contextualization, to induce a similarity structure over the documents based on the containment and reverse-containment relationships (element, sub-element and vice versa). Hence, these relationships affect the weight each element, in the structural context, has in contextualization.

The premise is that *good structural context* (identified by random walk and the contextualization model [7]) provides evidence that an element in focused retrieval is a good candidate to satisfy the user's need and therefore, the elements should be contextualized by the elements in the sub-tree of interest. Hence, the good structural context contains a strong likelihood factor that should be used to deduce that the contextualized element is a good candidate for the posed query.

The tree-structure of the XML document (Figure 1) is assumed to be a graph. In order for the structural context to take part in the contextualization process, each of the nodes in the sub-tree of interest should possess an impact factor. Conceptually, the impact factor is produced in the following manner: Myriad of random surfers traverse the XML graphs. In particular, at any time step a random surfer is found at an element and either (a) makes a next move to the sub-element of the existing element by traversing the containment edge, or (b) makes a move to the parent-element of the existing element, or (c) jumps randomly to another element in the XML graph. As the time goes on (the number iterations), the expected percentage of surfer at each node converges to a limit, the dominant eigenvector of the XML graph. This limit provides the impact or strength of each element in the structural context of the element to be contextualized, in the form of g function. All the elements, in the structural context of the contextualized element, are considered for contextualization; where the contextualization vector g identifies the importance of each of the unit of the structural context (Equation 1).

2.4 Generalized Combination Functions

The generalized re-ranking combination function based on the random walk principle, which also captures the structural context, can be formally defined as follows:

$$CR(x, f, C_x, g^k) = (1-f) \cdot BS(x) + f \cdot \frac{\sum_{y \in C_x} BS(y) \cdot g^k(y)}{\sum_{y \in C_x} g^k(y)} \qquad (1)$$

where

- $BS(x)$ is the basic score of contextualized element x (text-based score, e.g., $tf \cdot ief$)
- f is a parameter which determines the weight of the context in the overall scoring.
- C_x is the kinship context surrounding the contextualizing element x, i.e., $C_x \subseteq structural\_context(x)$, $\subseteq$, because only the structural context containing the query terms are considered.

- $g^k(y)$ is the generalized contextualization vector based on random walk, which gives the authority weight (the impact) of y, the contextualizing elements (elements in structural context) of x in the sub-tree of interest.

3. EFFECTS OF CONTEXTUALIZATION ON DIFFERENT TEST COLLECTIONS

Structural context in the contextualization framework, is independent of the basic weighting scheme of the elements and it could be applied on the top of any query language, retrieval systems and test collections. The effects of contextualization on different test collections have been observed in the existing studies. Contextualization model has been applied on the top of different and competitive baseline systems using a diverse set of test collections, e.g., semantically annotated Wikipedia collection from INEX 2009[1], IEEE collection, and iSearch scientific collection [3, 7, 8]. In order to get the best possible baseline system, a data fusion was performed based on sum of normalized scores (CombSUM) [11] and Reciprocal Ranking [4] of INEX 2009 submitted runs.

In the experimental evaluation the retrieval effectiveness at different granularity levels were observed. Mainly, retrieval effectiveness at paragraph, article and INEX's focused retrieval level selection has been observed. The approaches were evaluated using the evaluation framework provided by TREC and INEX evaluation initiatives. The reported results were shown to be promising using both TREC and INEX evaluation framework [3, 7].

The focused task in INEX ad-hoc track is to retrieve most focused elements satisfying an information need without overlapping elements. An overlapping result list means that the elements in the result list may have a descendant relationship with each other and share the same text content. For instance, in Figure 1 the ⟨entry⟩ element ⟨1.2.2.2.1⟩ and the ⟨sec⟩ element ⟨1.2.2⟩ are overlapping. In the existing studies, in the focused retrieval task, the INEXs' focused approach is followed, considering a result list where only one of the overlapping elements from each branch is selected. This means that including the ⟨sec⟩ element in the results would mean excluding the entry element in the results or vice versa.

Contextualization and the fusion approach as scoring methods, however, do not take any stand on which elements should be selected from each branch. Thus a structural fusion has been performed, where the element level selection is taken from the baseline run and subsequently re-rank the elements of the baseline run.

3.1 Test Settings

The hierarchical structure of XML documents in the Wikipedia 2009 collection, are captured using the dewey encoding scheme (as shown in Figure 1). This way each element in the document possess a unique index within the document, and together with document's unique id, this becomes unique for the entire collection. The tree structure of XML documents are converted into a matrix, and random walk is performed on this matrix at indexing time, as it is described in detail, in our earlier work [7]. The contextualization vector g^k from Equation 1 is computed off-line for each and every XML document in the Wikipedia collection. This suggests that computing g^k vector is feasible for a reasonably large XML document collections. At the query time, the scores from g^k vector and the basic scores are combined to produce an overall ranking score, using Equation 1.

In the generalized combination function given (Equation 1), the contextualization force has to be parametrized. In our earlier work [7], the contextualization force was tuned and reported the values leading to best overall performance. In the parametrization process it was found that the optimal values of contextualization force f (from Equation 1) lies in the range, ($f \in \{.25,..., 2.50\}$). These optimal values for f are obtained by using cross-validation technique. A 68-fold[2] cross-validation (or complete cross-validation) technique has been performed - by randomly partitioning the collection into 68 training and test samples based on the number of assessed topics. Of the 68 samples, a single sample is retained as the validation set for testing, and remaining 67 samples are used as training set. The cross-validation process is repeated 68 times (for each fold), with each of 68 samples used exactly only once as validation set. These 68 independent or unseen samples are then combined to produce a single or a set of estimations for parameter f.

3.2 Query Term Probabilities

If a relevant element does not contain any of the query term(s), it does not match to the query. Hence, in order to retrieve such elements, some expansive methods, such as contextualization, ought to be used. It seems obvious that, in a relevant small element, the probability of occurrence of a query term is smaller than in a larger element. In order to demonstrate this lack of evidence on small elements, we calculated some posteriori probabilities for query term occurrences in a relevant document (R_d) and in a relevant paragraph (R_p, i.e., the relevant ⟨p⟩ elements from the XML graph), based on INEX 2009, 68 topics (title field) and their relevance assessments. The probabilities are calculated as the fraction of relevant elements containing any query term, or all query terms over all relevant elements of same kind. The probability of occurrence of any query term (from the query Q) in a R_p and in a R_d respectively are:

$$P\left(\bigcup_{q\in Q} q \middle| R_p\right) = 0.847, \quad P\left(\bigcup_{q\in Q} q \middle| R_d\right) = 0.995$$

This means that the probability of occurrence of none of the query terms in R_p and a R_d is 0.153 and 0.005 respectively[3]. Accordingly, the probabilities of occurrence of all the query terms in R_p and R_d, respectively are:

$$\prod_{q\in Q} P\left(q|R_p\right) = 0.127, \quad \prod_{q\in Q} P\left(q|R_d\right) = 0.469$$

The difference in the amount of evidence at different granularity levels become even more obvious, when we draw the frequencies of the query terms in this picture. A query term occurs on average 3.4 times in a R_p and 45.4 times in a R_d.

4. WORST CASE ANALYSIS

Worst-case for a document d, in contextualization models, means when structural context of element x is chosen such that:

$$structural\_context(x) \notin elements_y(d) \qquad (2)$$
$$(\forall \text{ elements } y \text{ in document } d \quad \text{where } x \text{ and } y \in d)$$

[1] Wikipedia collection containing 2.66 million semantically annotated XML documents (50, 7 Gb) and 68 related topics provided by the INEX 2009 ad-hoc track [5].

[2] 68, because of the 68 topics from INEX 2009.

[3] Test is performed without stemming or stop-word removal

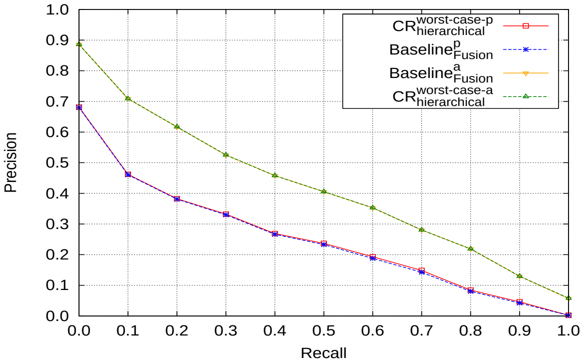


Figure 2: Precision - recall, worst-case scenario at article (a) and paragraph (p) granulation and the fusion baseline systems.

The *non-structural context* (Equation 2), should theoretically expose the worst-case effects of the contextualization model. Non-structural context is structural by definition, but physically not in the structural context of element x. How should we interpret the non-structural context, in order to experimentally visualize the worst-case scenario? Instead of taking the actual and true structural context, we randomly select the structural context from another non-relevant but retrieved document. Such a document (retrieved but not relevant) would have misleading evidence (false positive) and hence best suited for the worst-case evaluation. Randomly selecting a document with zero basic score would be trivial and not suitable for our purposes.

By applying this simplistic approach on every element to be contextualized, we can formulate the worst-case scenario. We have used the reciprocal rank fusion approach (fusing 98 INEX 2009 runs) as the baseline system, for worst-case analysis, which has been used before in our earlier work, find further details from [8]:

$$RRScore(e, q) = \sum_{r \in R} \frac{1}{k + rank(r, e, q)} \tag{3}$$

where

- R is the set of runs (rankings)
- and $rank(r, e, q)$ returns the rank of element e as a result of query q in run r.
- If e is not in the ranking, $rank(r, e, q)$ is not defined and the outcome of $\frac{1}{k+rank(r,e,q)}$ is 0.
- The parameter k is for tuning.

Figure 2 reveals the worst-case depiction of the contextualization model. Not unexpectedly, the worst-case scenario is as good as the baseline system, slightly better but not significant enough to be visible statistically. We can claim here that, when the structural context is chosen randomly (haphazardly), in the worst-case, the contextualization method will not be worse than the basic scoring method.

5. CONCLUSIONS AND FUTURE WORK

Structural context is the sub-tree of interest, utilized in conjunction with contextualization model, improves the retrieval effectiveness. We have presented an exploratory and theoretical study into the use of structural context from elements in the hierarchical structure of information, to improve retrieval performance. We looked into the structural context from document's hierarchical structure internally, and hyperlinks structure externally. We looked theoretically into the hypothesis that structural context gathered from within the document, "horizontally" and "vertically" using the hierarchical tree structure of document, and from outside, using the hyperlinks graph structure of documents referencing each other, influences the retrieval effectiveness. Worst-case experiments also support the theoretical soundness of contextualization, i.e., if we apply contextualization blindly, in the worst case, we would have as good result as the basic scoring method. The results obtained in this study are in-line with the earlier work on contextualization [1, 3, 6, 7, 9, 10]. In this study we have experimented with semi-artificial data, in the sense that we muddled the context for the worst-case analysis. However, in real data the quality of context varies as well. For example in Wikipedia there are different kinds of pages ranging from listings to topically very coherent documents. In order to get the best results in retrieval, analysing the quality and topical coherency of context would be of great benefit. The analysis of context may be topic dependent, since some queries may have contextual parts. For instance a query: "Losses Belgium in WW2", crave for answers about *Belgium* in the context of *WW2*.

6. REFERENCES

[1] P. Arvola, M. Junkkari, and J. Kekäläinen. Generalized Contextualization Method for XML Information Retrieval. In *Proc. of the 14th ACM CIKM*, pages 20–27. ACM, 2005.

[2] P. Arvola, J. Kekäläinen, and M. Junkkari. The Effect of Contextualization at Different Granularity Levels in Content-oriented XML Retrieval. In *Proc. of the 17th ACM CIKM*, pages 1491–1492. ACM, 2008.

[3] P. Arvola, J. Kekäläinen, and M. Junkkari. Contextualization Models for XML Retrieval. *Info. Processing & Management*, pages 1–15, 2011.

[4] G. Cormack, C. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759. ACM, 2009.

[5] S. Geva, J. Kamps, M. Lethonen, R. Schenkel, J. Thom, and A. Trotman. Overview of the INEX 2009 ad-hoc track. *Focused Ret. and Evaluation*, pages 4–25, 2010.

[6] Y. Mass and M. Mandelbrod. Component Ranking and Automatic Query Refinement for XML Retrieval. *Advances in XML IR*, pages 1–18, 2005.

[7] M. A. Norozi, P. Arvola, and A. P. de Vries. Contextualization using hyperlinks and internal hierarchical structure of wikipedia documents. In *Proc. of the 21st ACM CIKM*, pages 734–743. ACM, 2012.

[8] M. A. Norozi, A. P. de Vries, and P. Arvola. Contextualization from the Bibliographic Structure. In *Proc. of the ECIR 2012 Workshop on Task-Based and Aggregated Search (TBAS2012)*, page 9, 2012.

[9] P. Ogilvie and J. Callan. Hierarchical Language Models for XML Component Retrieval. *Advances in XML IR*, pages 269–285, 2005.

[10] G. Ramirez Camps. *Structural Features in XML Retrieval.* PhD thesis, SIKS, the Dutch Research School for Information and Knowledge Systems., 2007.

[11] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *The 2nd TREC.* Citeseer, 1994.

Towards a grammar formalism for retrieving information on the Semantic Web

Eunhye Shin
Korea university
dc20005245@korea.ac.kr

Sujin Yoo
Korea university
mynameislydia@korea.ac.kr

Seongbin Park*
Korea university
hyperspace@korea.ac.kr

ABSTRACT

In this paper, we report our ongoing research on a grammar formalism for retrieving information on the Semantic Web. We view the Semantic Web as a collection of databases and use a two-level grammar by which one can specify context-sensitive constraints that need to be satisfied. To retrieve information, a user can simply specify keywords and our system can show a result that is a string derived using the two-level grammar.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation; D.4 [**Information Systems Applications**]: Miscellaneous; H.5.4 [**Hypertext/Hypermedia**]: Architectures

1. INTRODUCTION

The Semantic Web is an environment where information is represented in a machine understandable way. One way to view the Semantic Web is that it consists of databases that can help computational agents perform various kinds of tasks [1].

In this paper, we propose an approach to write context-sensitive constraints using a grammar in order to retrieve information on the Semantic Web. To this end, we use a two-level grammar that is a 6-tuple (M, V, T, R_M, R_V, S), where M is a finite set of metanotions, V is a finite set of syntactic variables such that $M \cap V = \emptyset$, T is a finite subset of V^+, R_M is a finite set of metarules $X \to Y$, where $X \in M$, $Y \in (M \cup V)^*$ or Y is a regular expression, and for all $W \in M$, (M, V, R_M, W) is a collection of context-free grammar and regular expression rules, R_V is a finite set of hyperrules of the form $H_0 \to H_1, H_2, \cdots, H_m$, where $m \geq 1$ and $H_0 \in (M \cup V)^+$, $H_i \in (M \cup V)^*$ for $i \geq 1$, H_i is a hypernotion, and S is a string of positive length over $M \cup V$, respectively [2].

The motivation of using a two-level grammar is that it allows to specify context-sensitive information in an intuitive way. Our approach allows a user to express facts that contain certain parameters which reflect structures of databases that constitute the Semantic Web. In other words, the formalism allows users to specify certain facts together with placeholders that can be instantiated using the data stored in databases.

The structure of this paper is as follows. Section 2 describes related research works. Section 3 explains the idea behind our approach using illustartive examples. Section 4 describes the structure of the system that we implemented. Section 5 concludes the paper and discusses research directions.

2. RELATED WORKS

There are two research areas that are related to our research. One is research about two-level grammar and the other is retrieving information on the Semantic Web. Two level grammar was introduced to define the syntax of ALGOL 68 by van Wijngaarden [3]. There are two types of rules in a two-level grammar. One is a metarule and the other is a hyperrule. A metarule is a context-free production rule and it can provide possible values for a metanotion in a hyperrule. A hyperrule can describe context-sensitive conditions and this is a machanism by which we can model constraints in a database or between databases. A two-level grammar has been used in specifying the syntax of a natural language which reflects grammatical constraints [4]. It has been also applied to define a programming language [2, 5]. One way to retrieve information on the Semantic Web is to use a query language, but traditional database query languages are not appropriate and users need a semantic query language such as SPARQL [6, 7]. In the mean time, for end users, a system such as SPARK [8] that can convert keyword queries into SPARQL queries can be helpful.

3. ILLUSTRATIVE EXAMPLES

In this section, we show how we can describe context-sensitive information using a two-level grammar. A string that can be derived using the grammar corresponds to certain information that can result from combining data that exist in the databases. There are two types of examples. The first example shows the case where we use one database that contains some number of tables. The second example shows how we can use two databases.

3.1 Example 1

*Corresponding author

DIR 2013, April 26, 2013, Delft, The Netherlands.

Figure 1 shows the ER-Diagram of an example database (Travel database). There are five tables, where PK refers to a primary key and FK refers to a foreign key, respectively.

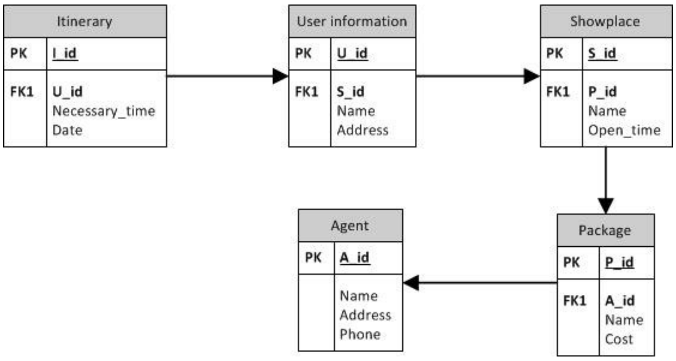


Figure 1: Travel database

Tables 1 to 5 show data stored in the database tables.

Table 1: Itinerary Table

I_id	U_id	Neccessary_time	date
350	1002	2 hours	2012-8-1
353	1001	5 hours	2012-7-2

Table 2: User information Table

U_id	S_id	Name	Address
1001	420	Sujin	Yangcheon-gu, Seoul
1002	401	Bob	Gangnam-gu, Seoul

Given this database, we can write metarules and hyperrules so that a string whose structure looks '( ), ( ) travels ( ) with ( ) package in ( ) agency.' can be derived as follows, where ( ) is a placeholder that can be instantiated with the data stored in the database tables.

The metarule is as follows.

```
I  ::  DATE,  U_ID
U  ::  U_ID U_NAME travels S_ID
S  ::  S_ID S_NAME with P_ID
P  ::  P_ID P_NAME package in A_ID
A  ::  A_ID A_NAME agency.
```

The hyperrule is as follows.

```
START  :  I  U  S  P  A
where U_ID is U_ID : true
where S_ID is S_ID : true
where P_ID is P_ID : true
where A_ID is A_ID : true
```

Assuming that the tables contain data shown in table 1 to table 5, a possible derivation looks as follows.

$START \Rightarrow I\ U\ S\ P\ A \Rightarrow DATE,\ U\_ID\ U\ S\ P\ A$
$\Rightarrow 2012-7-2,\ 1001\ U\ S\ P\ A$
$\Rightarrow 2012-7-2,\ 1001\ U\_ID\ U\_NAME\ travels\ S\_ID\ S\ P\ A$

Table 3: Showplace Table

S_id	P_id	Name	Open_time
401	300	Buckingham Palace	11pm(10pm, Sun)
420	302	Hokkaido	11pm

Table 4: Package Table

P_id	A_id	Name	cost
300	204	15 days in West Europe	$2700
302	200	Hot Spring in Tokyo	$1400

$\Rightarrow 2012-7-2,\ 1001\ 1001\ Sujin\ travels\ S\_ID\ S\ P\ A$
$\Rightarrow 2012-7-2,\ Sujin\ travels\ S\_ID\ S\ P\ A$
$\Rightarrow \cdots \Rightarrow 2012-7-2,\ Sujin\ travels\ Hokkadio\ with\ Hot\ Spring\ in\ Tokyo\ package\ in\ Hana\ tour\ agency.$

Figure 2 shows how parameters in hyperrules can be instantiated and our system shows the string at the end of the derivation. The derivation starts by using the first hyperrule, START : I U S P A, where each of I U S P A is replaced by the corresponding metarule; i.e., I is replaced by DATE, U_ID, U is replaced by U_ID U_NAME travels S_ID, etc. A hyperrule which starts with "where" is applied in order to check context-sensitivity. For example, the U_ID from I (i.e., DATE, U_ID) and the U_ID from U (i.e., U_ID U_NAME travels S_ID) disappear when the hyperrule, "where U_ID is U_ID :true" is applied.

3.2 Example 2

In order to show how the same approach can be used with multiple databases, we added a university database that consists of two tables. In addition, we modified the travel database. Customers in the travel database are students in the university database. Buy table contains the information about package purchasing for each customer and user name of travel database corresponds to student name of university database (figure 3).

Tables 6 to 8 show the data contained in the database tables. Given these databases, we can write metarules and hyperrules so that a string whose structure looks '( ) is a Korea University student, majors in ( ) and takes a trip with ( ) package.' can be dervied as follows, where ( ) is a placeholder that can be instantiated with the data stored in the database tables.

The metarule is as follows.

```
RELATION :: USER STUDENT
UNI :: UNI.NAME DEPART
TRAVEL :: TRAVEL.NAME PACKAGE
```

Table 5: Agent Table

A_id	Name	Address	Phone_num
200	Hana tour	Seocho-gu, Seoul	02-993-2941
204	E agent	Bucheon, Gyeonggi	031-424-4421

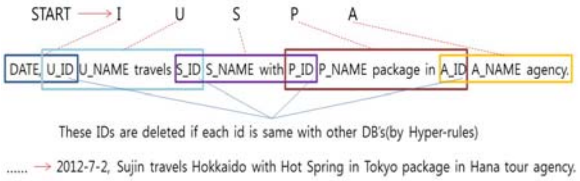

START → I U S P A
DATE U_ID U_NAME travels S_ID S_NAME with P_ID P_NAME package in A_ID A_NAME agency.
These IDs are deleted if each id is same with other DB's(by Hyper-rules)
...... → 2012-7-2, Sujin travels Hokkaido with Hot Spring in Tokyo package in Hana tour agency.

Figure 2: Derivation of a string

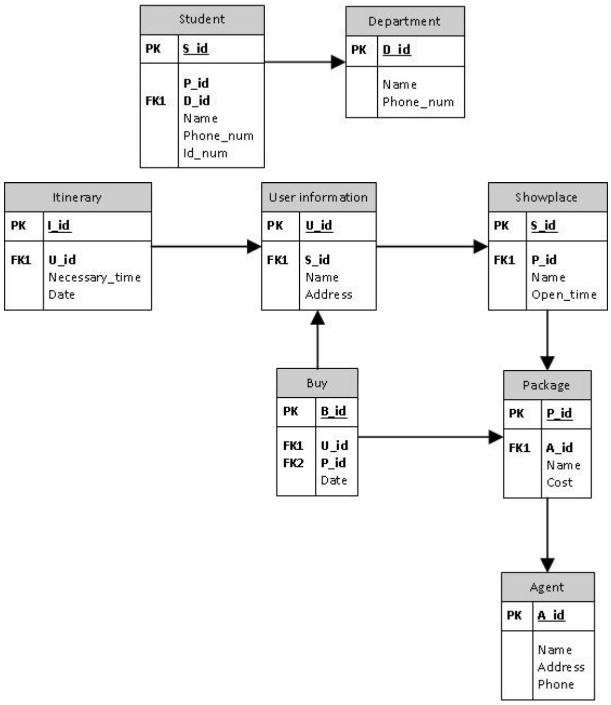


Figure 3: University database and modified Travel database

```
STUDENT :: S_Name
USER :: U_Name
UNI.NAME :: S_Name is a Korea University
            student, majors in D_ID
DEPART :: D_ID D_Name
TRAVEL.NAME :: U_NAME and takes a trip with
               U_ID BUY P_ID
BUY :: U_ID P_ID
PACKAGE :: P_ID P_NAME package.
```

There are five types of hyperrules.

(1) The following hyperrule is the start rule.

```
START : UNI RELATION TRAVEL
```

(2) The following hyperrules verify whether student name and user name are the same or not.

```
S_NAME of UNIV.NAME is same S_NAME of STUDENT
 of RELATION, U_NAME of TRAVEL.NAME is same
U_NAME of USER of RELATION, S_NAME of STUDENT
 is same U_NAME of USER, S_NAME of STUDENT,
U_NAME of TRAVEL.NAME and U_NAME of USER
 : true
```

Table 6: Buy Table

B_id	U_id	P_id	Date
465	1001	300	2011.11.01
274	1002	302	2011.10.10

Table 7: Student Table

S_id	P_id	D_id	Name	Phone_num	ID_num
1	150	100	Sujin Yoo	243-5678	051901
2	150	100	Bob	234-5784	011901

(3) The following hyperrules verify whether data is connected correctly in Univ DB.

```
D_ID of UNI.NAME is same D_ID of DEPART is
same, D_ID of UNI.NAME and D_ID of DEPART
 : true
```

(4) The following hyperrules verify whether data is connected correctly in Travel DB.

```
P_ID of TRAVEL.NAME is same P_ID of PACKAGE,
P_ID of BUY is same P_ID of PACKAGE, P_ID of
TRAVEL.NAME, P_ID of PACKAGE and P_ID of BUY
 : true
```

(5) The following hyperrule verifies whether purchasing data is connected correctly in Travel DB.

```
where U_ID is U_ID : true
```

Figure 4 shows how parameters in hyperrules can be instantiated and our system shows the string at the end of this figure.

4. TLG SYSTEM

Our system (TLG sysetm) has been implemented using Java and HSQLDB system [9]. The operation starts by taking a keyword from a user. The Searching scale setting module assigns a column according to the keyword. The Result creating module matches the keyword against data in the assigned column. Finally, the Result printing module shows the result from result creating module as a string. Figure 5 shows the structure of the TLG system, where numbers inside small circles correspond to steps involved in sequence.

5. CONCLUSIONS

Table 8: Department Table

D_id	Name	Phone_num
100	Computer Science Education	32-1234
102	Mathematics	32-3456

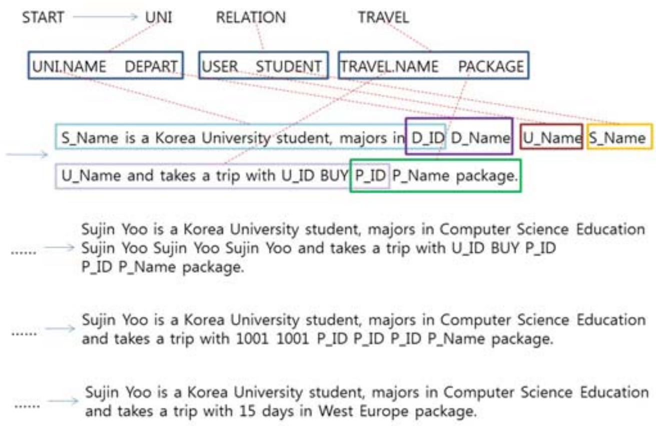

START
UNI
RELATION
TRAVEL
UNI.NAME DEPART
USER STUDENT
TRAVEL.NAME PACKAGE
S_Name is a Korea University student, majors in D_ID D_Name U_Name S_Name
U_Name and takes a trip with U_ID BUY P_ID P_Name package.
Sujin Yoo is a Korea University student, majors in Computer Science Education Sujin Yoo Sujin Yoo Sujin Yoo and takes a trip with U_ID BUY P_ID P_ID P_Name package.
Sujin Yoo is a Korea University student, majors in Computer Science Education and takes a trip with 1001 1001 P_ID P_ID P_ID P_Name package.
Sujin Yoo is a Korea University student, majors in Computer Science Education and takes a trip with 15 days in West Europe package.

Figure 4: Derivation of a string

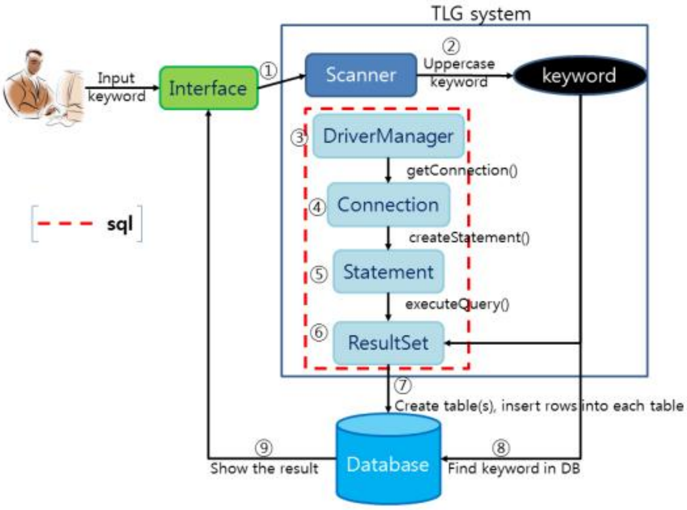


Figure 5: The structure of the system

In this paper, we report our ongoing research on how a two-level grammar can be used to retrieve information on the Semantic Web. We view the Semantic Web as a collection of databases and constraints existing in the colecction can be specified using hyperrules of the formalism that we proposed.

The motivation of the current research is that once we have a declarative description about the Semantic Web using a formal grammar, it becomes possible to *process* the Semantic Web. In other words, a part of the Semantic Web can be fed into a computer program as an input and the program can parse the input and perform some task. This is in line with the goal of utilizing the Semantic Web as a representation medium for computations [10].

Currently, we are implementing a system that parses expressions defined using a two-level grammar and shows derivation results. We are also working on ways by which the information on the Semantic Web can be exploited using a Semantic Web browser [11] and how to extend the idea of a Semantic Web expression [12] in the context of linked data [13].

6. REFERENCES

[1] C.C. Marshall and F.M. Shipman. Which Semantic Web?. In *Proceedings of the 14th ACM Conference on hypertext and hypermedia*, 2003.

[2] B. Edupuganty and B.R. Bryant. Two-level Grammar as a Functional Programming Language. In *The computer journal*, vol 32, no 1, 1989.

[3] A.van Wijngaarden. Report on the Algorithmic language ALGOL 68. Numer 14: 79-218, 1969.

[4] B.R. Bryant, D, Johnson, and B. Edupuganty. Formal specification of natural language syntax using two-level grammar. In *Proceedings of the 11th International Conference on Computational Lnguistics*, 1986.

[5] J. Maluszynski. Towards a Programming Language Based on the Notion of Two-Level Grammar In *Theoretical Computer Science*, 13-43, 1984.

[6] R. Fikes, P. Hayes, and I. Horrocks. OWL-QL-a language for deductive query answering on the Semantic Web In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web.* vol 2, issue 1, 19-29, 2004.

[7] M. Arenas, C. Gutierrez, D.P. Miranker, J. Pérez, and J.F. Sequeda. Querying Semantic Data on the Web. In *SIGMOD Record*, vol. 41, no. 4, 2012.

[8] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y .Yu. SPARK: adapting keyword query to semantic search In *Proceedings og the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, 694-707, 2007.

[9] HSQLDB *www.hsqldb.org*

[10] M.A. Rodriguez and J. Bollen. Modelling Computations in a Semantic Network. `http://arxiv.org/abs/0706.0022`, 2007

[11] Y. Kim, S, Yoo, and S. Park. A Semantic Web browser for novice users. In *Proceedings of the 6th International Conference on Complex, Intelligent, and Software Intensitve Systems*, 2012.

[12] E. Shin and S. Park. Two level grammar and the Semantic Web. In *Proceedings of the International Conference on Applied and Theoretical Information Systems Research*, 2012.

[13] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. In *International Jounral on Semantic Web and Information Systems*, 2009.