Window-based Enterprise Expert Search

Wei Lu[1], Stephen Robertson[2,3], Andrew Macfarlane[3], Haozhen Zhao[1]

[1] Center for Studies of Information Resources, School of Information Management
Wuhan University, China and City University
{sa713@soi.city.ac.uk, zhaohaozhen@gmail.com}
[2] Microsoft Research, Cambridge, U.K. and City University
ser@microsoft.com
[3] Centre for Interactive Systems Research, Department of Information Science
City University London
andym@soi.city.ac.uk

Abstract. This is the first year for the participation of the City University Centre of Interactive System Research (CISR) in the Expert Search Task. In this paper, we describe an expert search experiment based on window-based techniques, that is, we build profile for each expert by using information around the expert's name and email address in the documents. We then use the traditional IR techniques to search and rank experts. Our experiment is done on Okapi and BM25 is used as the ranking model. Results show that parameter *b* does have an effect on the retrieval effectiveness and using a smaller value for *b* produces better results.

1. Introduction

This is the second year for the Enterprise Expert Search task. One of the common methods for this task is to create a profile for each expert and then apply normal IR techniques to index and search these profiles, using the topics as queries [1, 2, 3, 4, 5]. The key issue for this is how to generate profiles by collecting various expertise evidences from the enterprise collections. Some work has been done using this method in TREC 2005, e.g. Macdonald et al [2] generate profiles by using weighted occurrences of person in corpus, personal website and email threads. Fu et al [3] developed a novel method called document reorganization which collects and combines related information from different media formats to organize a document for an expert candidate. Zhu et al [4] represented each name extracted from corpus with a collection of documents (for instance, all the emails the person had sent) and then used different information retrieval models (Vector Space (VS) model and Latent Semantic Indexing (LSI) model) to measure the relevance between the collections of documents and the topics. Azzopardi et al [5] use various expert name and email match methods to extract possible expert in-

formation and then build expert profile based on this. Their experiments show that the performance depends crucially on the ability to recognize names of experts.

In this paper, a window-based method is adopted to build descriptions of experts. That is, we use a window around occurrences of an expert name or email address to create a profile for the expert. The basic idea of our approach is that the information around the expert name and email address should have more association with the expert, than other textual information. Some past research such as [6,7,8] have shown that using this method is effective for document retrieval. We hope this could also be applied to enterprise expert search, although the effectiveness still needs to be investigated.

In the next section we briefly describe the preliminary search completed for the expert search challenge in order to help the community to understand relevance assessments for this track. This gives some motivation for our approach. We then briefly introduce the retrieval model BM25 used in our experiment in section 3. We then describe our experiment in section 4 and explore the evaluation results in section 5. A conclusion is given at the end.

2. Expert Search Challenge

In order to give participants in the track some common experience in judging relevance for the expert search task, a challenge was set to find experts in the field of "Scalable Vector Graphics animation". The expert identified should have had significant knowledge in the area of animation in SVG, general knowledge of SVG was regarded as being insufficient. Fig 1 lists the results of our exploratory search:

candidate-0163 Jon Ferraiolo http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0751 David Duce http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0979 Jerry Evans http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0497 Vincent Hardy http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0553 Lofton Henderson http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0500 Dean Jackson http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0983 Christophe Jolif http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-1062 Kelvin Lawrence http://www.w3.org/TR/2000/CR-SVG-20000802/
candidate-0044 Chris Lilley http://www.w3.org/TR/2000/CR-SVG-20000802/

Figure 1. Results of search for expert on SVG animation

The search undertaken was simple and rushed, very typical of the type of search end users undertake. The search on the W3C site led to one particular page on Scalable Vector Graphics which was found directly from the hitlist and was linked to via other links on the hitlist. Most of the retrieved links dealt with accessibility,

and we did not feel that any people associated with this knowledge would necessarily know about SVG animation. This is why the choice of candidates is more restricted than others who completed the expert search challenge.

One issue which was difficult to resolve, was that the authors associated with a specification where not differentiated with respect to the components they had worked on – that is, a specification usually has a single list of authors. The experts identified in figure 1 could be wrong as some of the candidates chosen may not know that much about graphics – they may be experts in other parts of the specification. It would appear that using a single source of evidence to identify an expert is therefore problematic. We hope that the window method put forward in this paper, will in part deal with this issue.

3. Modelling

In our experiments, we use the BM25 as the core retrieval model. BM25 is a series of probabilistic models derived by Robertson et al [9] for document level retrieval. The formula used in our experiment is as follows:

$$w_j(d,C) = \frac{(k_1+1)tf_j}{k_1((1-b)+b\frac{dl}{avdl})+tf_j}\log\frac{N-df_j+0.5}{df_j+0.5} \qquad (1)$$

where

C denotes the document collection,

tf_j is the term frequency of the jth term in document d,

df_j is the document frequency of term j,

dl is the document length, $avdl$ is the average document length across the collection,

and k_1 and b are tuning parameters which normalize the term frequency and element length.

Then the document score is obtained by term weights of terms matching the query q:

$$W(d,q,c) = \sum_j w_j(d,c)\cdot q_j \qquad (2)$$

Due to the huge variety of the generated expert profile length and the number of documents containing the expert name and email address, we use various k_1 and b for submitting the runs. These will be discussed in section 4.

4. Experiment

Our experiment is largely conducted on Okapi 2.51 in a Linux environment (using Red Hat 9). The experimental procedure is divided into four steps: the first step is the expert recognition and profile creation; the second step is the profile indexing and the original document collection indexing; the third step is the retrieval and ranking of experts; and the last step is the retrieval and ranking for the supporting document. The details are as follows:

Expert recognition and profile creation. As mentioned above, the key issue for expert search is to generate an expert profile. These need technique such as name entity recognition to extract expert name and email address. Due to the time limitations, we used naive string match algorithm to extract expert full name and email addresses, and then used a fixed window around the expert name or email address to build the expert profile. In our experiment, the fixed window size is 2000 characters length which is about 150-250 words.

Profile and the original document collection indexing. This year's expert search task required participants to submit both ranked experts and supporting documents. Both the expert profiles and the original document collection were indexed. Due to the huge variety length of generated profiles (from several KB to 110MB), we modified Okapi slightly to support large document record indexing. At the same time, we also built an index for the original document collection.

Retrieval and ranking of the experts. Based on the indexed expert profiles, we submit queries and rank experts accordingly based on BM25. The only issue which needs to be mentioned with respect to the ranking formulae is that we use various k_1 and b for submitting the runs due to the huge variety of the expert profiles' length and associated document numbers. The values of parameters $\{k_1, b\}$ used for the 4 submitted runs are {1.2, 0.35}, {1.2, 0.55}, {1.2, 0.75} and {1.8, 0.75}. These represent typical values found to be effective in document search.

Retrieval and ranking of the supported documents. For each expert, the associated documents were ranked to illustrate their support of the corresponding expert. We firstly retrieved all the documents relating to a specific query, and then we use the association between documents and experts to filter out those documents which are not pertinent to the expert. The remaining documents are then ranked as supporting evidence.

5. Evaluation

As mentioned above, we submitted 4 runs by using different k_1 and b values. The results of these runs without taking support into account are listed in Table 1 and the results of those taking support into account are listed in Table 2.

From the tables we can see that parameter b has more effect than k_1. The runs using the smallest value of b have the best results for most of the metrics. This suggests that the length of profiles is not a very important feature in ranking. More specifically, we should not normalise *tf* values too strongly. A query term which appears one or more times in the profile is a strong indicator of relevance, irrespective of profile length. This result is somewhat similar to results obtained using anchor text in web search – good b values for anchor text are often lower than for body text. To put it another way, it seems that if a profile is long and contains many terms, this is evidence that the expert is indeed expert in many topics. However, from our limited experiments, varying k_1 has little effect. This may indicate that we simply do not often get high *tf* values in our profiles.

Runs	k_1	b	**Map**	**R-prec**	**B-pref**	**Recip-Rank**	**P@10**
Ex3512	1.2	0.35	0.3158	0.3425	0.3299	0.7912	0.4612
Ex5512	1.2	0.55	0.2950	0.3308	0.3151	0.7222	0.4551
Ex7512	1.2	0.75	0.2718	0.3167	0.2973	0.6506	0.4143
Ex5518	1.8	0.55	0.2984	0.3345	0.3166	0.7226	0.4531

Table 1: Results without taking support into account

Runs	k_1	b	**Map**	**R-prec**	**B-pref**	**Recip-Rank**	**P@10**
Ex3512	1.2	0.35	0.2031	0.2466	0.2724	0.6481	0.3286
Ex5512	1.2	0.55	0.1905	0.2396	0.2642	0.5893	0.3347
Ex7512	1.2	0.75	0.1783	0.2312	0.2531	0.5719	0.3082
Ex5518	1.8	0.55	0.1927	0.2399	0.2646	0.5897	0.3327

Table 2: Results taking support into account

These results suggested that we should try more b values around the lower end. For a fuller investigation of this after the conference, we tuned b from 0 to 1

jumping by 0.05; the results are shown in Figure 2. The implication of Figure 2 seems to be that we should turn the b parameter (which controls the extent of document length normalization) right down to zero in this application. This is an interesting conclusion, and diverges from most of our other experiences.

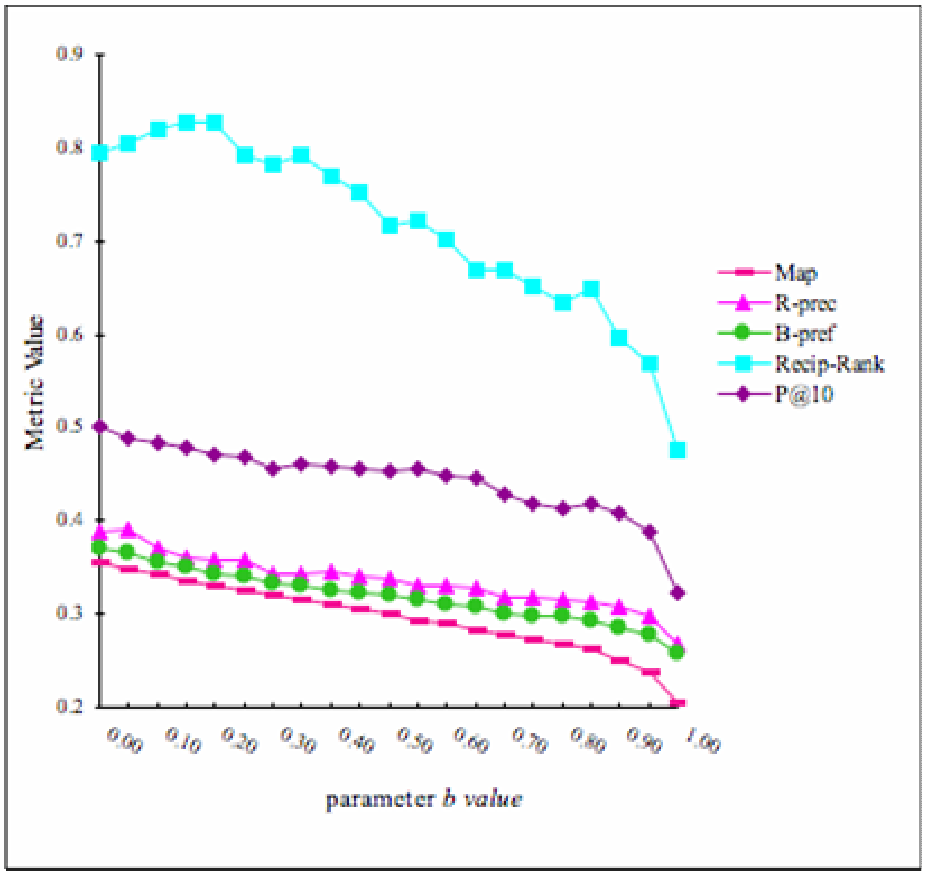

Figure 2. Evaluation results on all measures by tuning the b parameter

A possible hypothesis as to why this is so is as follows. Document length normalization is usually necessary because a scattering of occurrences of a query term over a longer document provides weaker evidence of relevance than the same number of occurrences concentrated in a shorter document. But in this case the 'document' (actually user profile) is constructed from fixed length windows from other documents; so the variation in length is primarily due to the number of such windows observed, i.e. to the number of mentions of the identified expert in the database. It appears that each window provides independent evidence of relevance; a lot of other windows indicating other expertise areas of this expert do not

in any way reduce the evidence gathered from some windows about expertise in the domain of the query. A similar effect, although not quite so strong, is observed in web search using anchor text.

Note that there is a slightly complex interaction with the k_1 parameter which controls the *tf* effect, which we have not yet explored.

6 Conclusion

We have tried a simple window-based method for enterprise expert search. Due to the time limitation, we only submitted runs with various k_1 and b values. The window size is fixed to 2000 character-length. In the future work, we will exploit the effectiveness of this method by using different window sizes. And we also need to use more sophisticated techniques to extract expert name and email address, so that we can build more concrete profiles for the expert.

Acknowledgements

This work is supported in part by National Social Science Foundation of China 06CTQ006.

References

[1] Nick Craswell, Arjen P. de Vries, Ian Soboroff. Overview of the TREC-2005 Enterprise Track. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[2] Craig Macdonald, Ben He, Vassilis Plachouras, Iadh Ounis. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of the 14th Text REtrieval Conference* (TREC 2005), Gaithersburg, MD, USA, 2005.

[3] Yupeng Fu, Wei Yu, Yize Li, Yiqun Liu, Min Zhang, Shaoping Ma. THUIR at TREC 2005: Enterprise Track. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[4] Weizhong Zhu, Min Song, Robert B. Allen. TREC 2005 Enterprise Track Results from Drexel. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[5] Leif Azzopardi, Krisztian Balog, Maarten de Rijke. Language Modeling Approaches for Enterprise Tasks. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[6] Wensi Xi, Xurong Richard, Khoo Christopers S. G., Lim Ee-Peng. Incorporating window-based passage-level evidence in document retrieval. Journal of Information Science. vol. 27, pp. 73-80, 2001.

[7] Alistair Moffat, Ron Sacks-Davis, Ross Wilkinson, Justin Zobel. Retrieval of Partial Document. In Proceedings of the Second Text Retrieval onference (TREC-2), 181-190. 1993

[8] Mittendorf, E., and Schauble, P. (1994). Document and passage retrieval based on hidden Markov models. In Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 318-327. New York. 1994.

[9] Stephen Robertson, Steve Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In W B Croft and C J van Rijsbergen, editors, SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Springer-Verlag, 1994.