Using Document Weight Combining Method for Enterprise Expert Mining

Haozhen Zhao, Wei Lu
Center for Studies of Information Resources
Wuhan University
Wuhan, People's Republic of China
zhaohaozhen@gamil.com，reedwhu@gmail.com

***Abstract*—Locating people with specific expertise on a given topic is crucial for the success of projects in enterprise. Rather than using the common method which needs to generate expert profiles, in this paper we present an alternative method, document weight combing, for the expert mining task. This method is much easier to implement and more flexible for the dynamic data sets than the common method. Our experiment and evaluation on the effectiveness of our method by using W3C corpus show that this method is practical and promising.**

Keywords- Enterprise Retrieval Expert mining Document weight combining

I. INTRODUCTION

Expertise mining plays an increasingly important role in enterprises, given the multiplication of enterprise digital information resources and proliferation of their types. A report from IDC gave a quantitatively description of the economic loss caused by the low capability of search in enterprises. [1] According to CSIRO's survey on many Australian enterprises, it is rather common that many enterprises are low in search capability and have scant knowledge of its importance.

The main task of expert mining is to find experts on specific topics in enterprise based on the support of all relevant resources, e.g. email, report, article, web page etc.. In this paper, we propose a novel method, document weight combining, for the enterprise expert mining. In the following section, an overview on some related work done in this field is given. In section 3, we illustrate our method and weighing model for expert mining in enterprise. We then use the W3C data collection as an experimental set, and evaluate our results based on the general paradigm of information retrieval evaluation. Finally, we discuss our findings and limitations.

II. LITERATURE REVIEW

As it is crucial to identify people with appropriate knowledge and skills to ensure the success of the projects in enterprises, people have been trying to develop tools which can find experts in enterprises. Among early methods, databases which contain the description of expertise of people in organization were adopted. [2] However, it is laborious and costly to explicate the expertise information for each individual. And the static nature of database makes this information easily incomplete and outdated. Moreover, the query format in database tends to be fixed and specific while description of expertise tends to be generic. [3] To solve these problems many systems have been proposed to automatically discover timely expertise information from secondary sources. For example, Campbell et al [4] have tried to mine experts in email communication through analyzing the link structure defined by authors and receivers of emails using a modified version of the Hyperlink-Induced Topic Search (HITS) algorithm. But, objects processed in these methods are of specific types, which is not exactly the same as the real situation. Because date type in today's enterprises is a compound of emails, reports, databases files and web pages.

The creation of TREC enterprise track was to address the aforementioned shortcomings by providing a public platform for empirical evaluation of expert finding methods and technologies. One common method for the expert search task in TREC 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 [5, 6, 7, 8, 9]. Balog et al [10] presented two general strategies to expert searching given a document collection which is formalized using generative probabilistic models. The first one directly models an expert's knowledge based on the documents that they are associated with, while the second locates documents on topic, and then finds the associated expert. According to their results, the second strategy consistently outperforms the first. In TREC 2006, we adopted a window-based method to build descriptions of experts. [11] That is, we use a window around occurrences of an expert name or email address to create a profile for the expert. In this paper, we will try another method to implement expert mining in enterprises. This method needs not to build expert profiles in advance, but just combines the weight scores of relevant documents for each expert candidate. See details of this in section 3.

III. OUR METHODOLOGY AND MODEL

A. Research Method

Different from the common expert mining method, we adopt a document weight combining method to find experts related to particular topics by utilizing the enterprise data collection. The method is illustrated as in Figure 1. Firstly, we index the data collection and extract information about expert candidates. Secondly, we get result sets relevant to each topic by retrieving the indexed results using the topics as query topics. Meanwhile, we use the feature evidences of expert

candidates, e.g. names and email addresses, to extract the enterprise data collection and get relevant result sets for each expert candidate. Thirdly, we cross these two result sets and combine the documents sets, generating relevant result sets for each expert candidate related to specific topics. For each topic, we then calculate the weight of all documents related to the expert candidates and combine them. Finally, we rank the weight scores for each expert, generating an expert list for each topic.

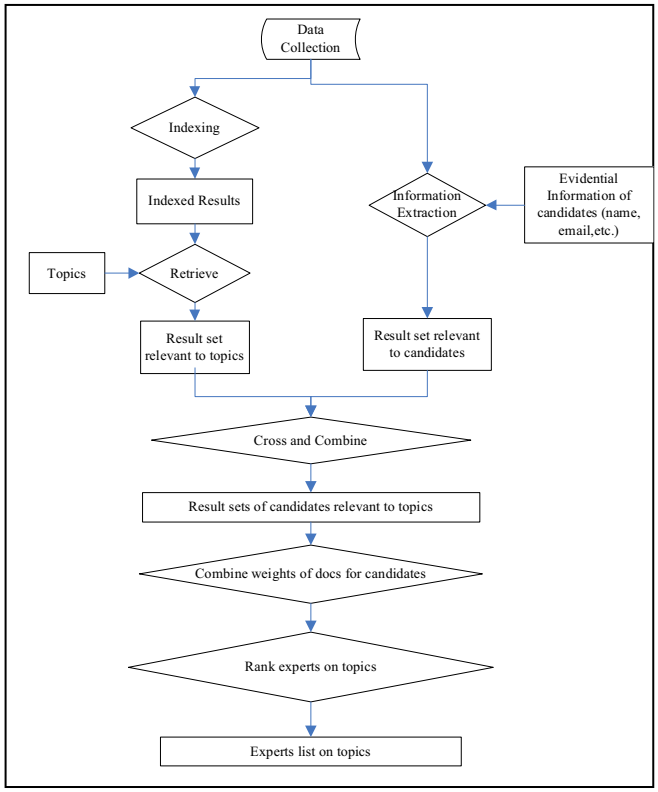

Figure 1. Document Weight Combining Method for Expert Mining

B. *BM25 Model*

In our experiments, we use the BM25 as the core retrieval model. BM25 is a series of probabilistic models derived by Robertson et al [12] for document level retrieval. The formula used in our experiment is as follows:

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

where C denotes the document collection, *tfj* is the term frequency of the *j*th term in document $\bar{d}$, *dfj* 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:

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

IV. DATA COLLECTION

Data collection is an important issue for system evaluation. It is hard to find an appropriate date set since most organizations are unwilling to open its intranet to public distribution, even for the purpose of research. Therefore we choose the publicized data collection of World Wide Web Consortium (W3C). The collection is a crawl of the public W3C (*.w3.org) sites in June 2004. It comprises 331,037 documents, of which date types cover form html, text, pdf, word, rtf to ppt and so on. Some details of the corpus are in Table I.

As for query topics and related result sets, we adopt the 55 topics and result sets in TREC 2006 Enterprise Track Expert Search Sub-track. More details about these topics and result sets see [13].

TABLE I. DETAILS OF W3C CORPUS [7]

Type	Scope *	Size(GB)	Docs	avdocsize(KB)
Email	lists	1.855	198,394	9.8
Code	dev	2.578	62,509	43.2
Web	www	1.043	45,975	23.8
Wiki Web	esw	0.181	19,605	9.7
Misc	Other	0.047	3,538	14.1
Web	people	0.003	1,016	3.6
All		5.7	331,037	18.1

* Scope is the name of the sub collection and also the hostname where the pages were found, for example lists.w3.org. The exception is the sub collection 'other' which contains several small hosts.

V. RESULTS AND EVALUATION

Our experiment is largely conducted on Okapi 2.51 in a Linux environment (using Red Hat 9). The evaluation measure used here are TREC official metrics: Map, R-Prec, B-pref, Recip-Rank, P@10. [14] For exploring the effects of document length, we tune parameter b from 0 to 1. The results are shown in Figure 2. From this figure we can see that there is not much difference of using different value of b, which implies that our method is not sensitive to length of documents. This is very different to the window-based method in our previous work [11].

To get a comparable performance of our method, we implemented the common expert mining method, and the result on metric P@10 is shown in Figure 3. We can see from this figure that our method get nearly the same effectiveness as the

common method and it's much stable than the latter one. We got nearly the same results for most of the other metrics.

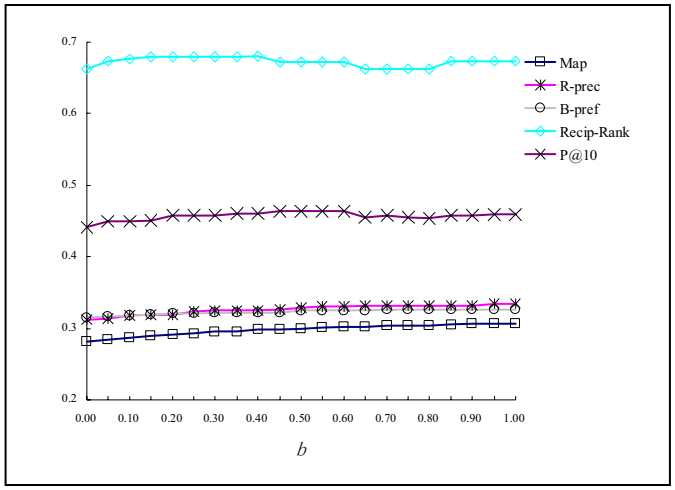

Figure 2. Experimental Result given different values of parameter b

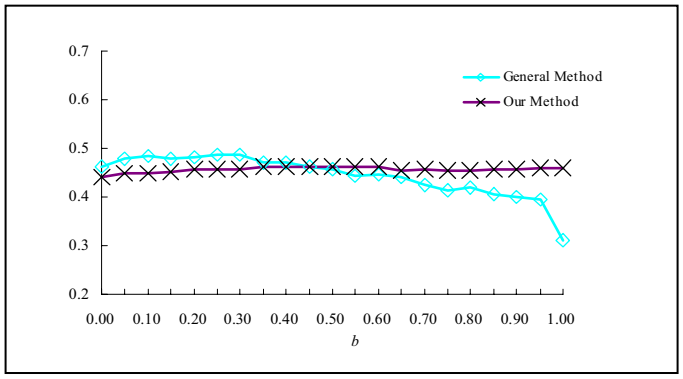

Figure 3. Comparison between two methods

VI. Conclusions and Future Work

We implemented an alternative expert mining method, Document Weight Combining, on the W3C corpus by using the Okapi BM25 model. This method needn't generate profiles of expert candidates; therefore it is easy to be implemented and has a good flexibility and extendibility especially for dynamic data collections. Compared with general method of expert mining, our method is pretty much the same in overall performance but excels in stability. The limitation of our method is that we adopt simple string matching method to identify expert which might get incomplete result sets relevant to experts. In the future, we will do more research on name entity recognition technology to improve the overall performance and explore this method in the real web environment.

Acknowledgment

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

References

[1] S. Feldman, C. Sherman. The high cost of not finding information. Technical Report #29127, IDC, April 2003.

[2] D. Yimam-Seid and A. Kobsa. Expert finding systems for organizations: Problem and domain analysis and the demoir approach. Journal of Organizational Computing and Electronic Commerce, 13(1):1–24, 2003.

[3] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), pages 3–9, 1996.

[4] C. S. Campbell, P. P. Maglio, A. Cozzi, and B. Dom. Expertise identification using email communications. In CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management, pages 528–531. ACM Press, 2003.

[5] N. Craswell, A. P. Vries, I. Soboroff. Overview of the TREC-2005 Enterprise Track. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[6] 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.

[7] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, S. Ma. THUIR at TREC 2005: Enterprise Track. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[8] W. Zhu, M. Song, R. B. Allen. TREC 2005 Enterprise Track Results from Drexel. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005

[9] L. Azzopardi, K. Balog, M. de Rijke. Language Modeling Approaches for Enterprise Tasks. In Proceedings of the 14th Text REtrieval Conference (TREC 2005), Gaithersburg, MD, USA, 2005.

[10] K. Balog, L. Azzopardi, M. de Rijke. Formal Models for Expert Finding in Enterprise Corpora. SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

[11] W. Lu, S. E. Robertson, A. Macfarlane, H. Zhao. Window-based Enterprise Expert Search. In Proceedings of the Ffteenth Text REtrieval Conference (TREC 2006), Gaithersburg, MD, USA, 2006

[12] S. E. Robertson, S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. 1994, 345-354.

[13]]http://trec.nist.gov/data/t15_enterprise.html [Visited 2006-4-1]

[14] http://trec.nist.gov/pubs.html [Visited 2006-4-1]