TREC2005 Enterprise Track Experiments at BUPT

Zhao Ru, Yuehua Chen, Weiran Xu, Jun Guo
(Pattern Recognition and Intelligent System Lab,
Beijing University of Posts and Telecommunications, Beijing, China, 100876)
rudjao@hotmail.com, chenyh_pris@hotmail.com

Abstract. This paper introduces and analyzes some experiments to find valid methods and features in enterprise search. For this purpose, two main experiments have been done. One is to retrieve some emails which contain the required information in all the emails of an enterprise, and the other is to try to find some experts who are helpful in a particular fields. Some features of the intranet dataset, such as the subject, the author, the date and the thread, are proved to be useful when searching an email. A new two-stage rank method which is different from traditional IR is introduced for expert search.

1. Introduction

As the Web search engines being widely used in people's life, in these years, a new area that IR in the Web pages of an enterprise attracts many researchers' eyes. However, the methods to search the internet are usually unconformable to intranet search. There are some new challenges in enterprise search, for example, the definition of an appropriate test collection, the effective search of email, the usage of intranet features, etc [2].

Enterprise track is a new track of TREC. It is to study the issues that arise when searching the documents of an enterprise. This year's main tasks include email search and expert search, with focus on the retrieval of particular emails and of experts on given topics respectively. The corpus is a real enterprise collection of the W3C web site.

This is the first year for BUPT to participate in TREC. We participated in both tasks of the enterprise track. Efforts have been made on two directions: finding features of the enterprise data which can improve the retrieval performance, and experimenting on new ranking methods. All our experiments are based on the 3.1version of Lemur Toolkit[1], which is developed by the University of Massachusetts and the Carnegie Mellon University.

2. Analysis of Email Search Task

In email search task we are concentrating on the emails in W3C collection as key sources. There are two important parts in the email search task: known item experiment and discussion experiment.

Our goal in this task is to find some useful features of the email, which would help to improve the retrieval performance. In this task, we used some common retrieval methods provided in the Lemur Toolkit and made a few modifications on them. Results show that some are effective and some are not.

[1] http://www.cs.cmu.edu/~lemur

2.1 Known Item Experiment

In this task we try to find an email which is known to exist. Different from the ad-hoc task, this task is to find a named page from emails which have a special form [6]. So we can make use of the form to improve the retrieval performance.

2.1.1 Document Structure

One obvious character is that each email has a fixed structure: a title, an author name and a date. Since each of the known item search queries is corresponding to a particular email, it always contains some information of these three components, which can often differentiate the email from others. To show an emphasis of the corresponding query words, these important components are given an additional weight:

$$\omega_t^{'} = \omega_t + T(idf_t) \tag{1}$$

where $T(idf_t)$ is the additional weight to the term of the three components if the term also occurs in the query, and we think it is relative to the term's inverted document frequency. Results indicate that for different term weight computing methods the improvements are all obvious, shown in Table1.

Table 1: results with and without additional weight to the structure information

	Tfidf	Logtf	Avtf	BM25
Ad hoc	0.409	0.457	0.422	0.413
Additional weight	0.512	0.520	0.513	0.551

2.1.2 Word Correlation

Since user often explains his goal explicitly, the query words in this task are more important than that be in ad-hoc task. People usually quote some strings of words they remember directly from the target document for a description. We think the strings of the query words are also important and give an additional weight to the words which are conjunctive in the document if they are also next to each other in the query. To the documents which have these strings, their final score are calculated as:

$$score_{new}(D,Q) = score(D,Q) \cdot (1 + \frac{\alpha}{n_{tp}} \cdot \sum_{p} \frac{k^{pn}}{pn}) \tag{2}$$

where pn is the count of terms in a string. n_{tp} is the total count of terms and strings in the document. α and k are parameters. In our experiments, α is set to 1/8 and k is set to 3. The results are not as good as we expected, shown in Table2.

Table 2: results with and without additional weight to the correlation of words

	Tfidf	Logtf	Avtf	BM25
With correlation	0.479	0.499	0.518	0.543
Without correlation	0.512	0.520	0.513	0.551

2.1.3 Anchor Text

There are also some anchor texts in the email pages, which indicate the emails before or after them by arrival time. Originally the anchor texts are treated as part of the email content, but sometimes they are not correlative with the context. It is expected that the results would be better if the anchor texts are removed, because the modified documents are more precise than before. But it seems invalid. Table3 shows the results.

Table 3: results with and without removing the anchor texts

	Tfidf	Logtf	Avtf	BM25
With anchor	0.512	0.520	0.513	0.551
Without anchor	0.536	0.555	0.510	0.535

We assume that the reason is as this: There are also some anchor texts which indicate the emails before or after the current email in thread. These anchor texts are usually correlative with the email documents and can be contained as a special emphasis on the email topic to improve the corresponding results.

2.2 Discussion Experiment

The goal of this task is to find some emails with pros and cons of an argument among several persons. Our goal in this experiment is to make use of the group information to help our retrieval performance.

2.2.1 Discussion Group

As it says, the queries are restricted to the emails which belong to a discussion. The emails on one discussing topic make up a discussion group, and all the discussion groups make up the retrieval data set. Each email except the first one in a discussion group would be a reply to a previous one, which is indicated by some anchor texts in the email document. All the emails are threaded in this way and a discussion group usually contains several email threads. Since a discussion group always focuses on one topic, the information of it can be used to find the relative emails through the threads [3].

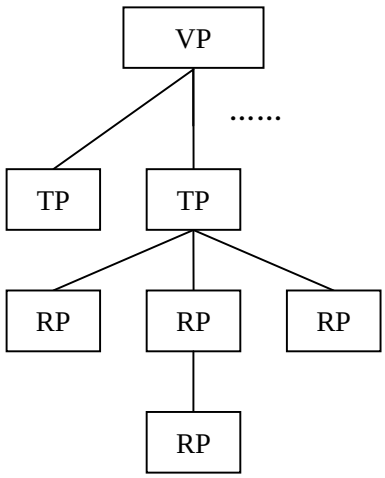

Figure 1: Email tree

For this purpose, we selected the emails which belongs to a discussion group according to their anchor texts and made all the emails into an email tree (see Figure 1), whose root is virtual and every branch to the root is a discussion group.

2.2.2 Ranking model

Two variations of a simple linear combination model are used by given the different features of the information in a group. One is to use the information of the entire discussion group; the other is to use the information of the first email in a discussion group.

With the email tree built, we combined the emails in a discussion group to form a new document. When doing retrieval we both retrieved the emails and the new documents of the groups. The final score of an email is the combination of its score and the score of the discussion group it belongs to:

$$score_{final}(D,Q) = score(G,Q) + score(D,Q) \quad (3)$$

In a discussion group there may be some emails which do not mention the discussing topic of this group. When we used the information of the entire group, these emails would have opposite effect to the results. But normally the first email in the thread of a discussion group, which is usually a sponsor and replied to by other emails, is correlative with the discussing topic. So as a variation we used the first email of a discussion group instead of the entire group to retrieve and combined the scores:

$$score_{final}(D,Q) = score(D_{top},Q) + score(D,Q) \quad (4)$$

We used the first variation in the run PRISDS1 and PRISDS5, and the second in the others. The queries in this task contain two parts: a title tagged with <query> and a narrative tagged with <narrative>. In the run PRISDS4 we only used the title and in the others we used both. The Table4 below shows the results of the five runs.

Table 4: results of discussion search in TREC-2005

	PRISDS1	PRISDS2	PRISDS3	PRISDS4	PRISDS5
MAP	0.3077	0.1288	0.2199	0.2603	0.0976
R-prec	0.3393	0.2500	0.2599	0.2947	0.1192
bpref	0.3204	0.1875	0.2320	0.2852	0.1973
recip-rank	0.6617	0.5000	0.4885	0.5835	0.3029

3. Analysis of Expert Search Task

The scenario of this task is the users input a topical query and the system retrieves a ranked list of people who are experts on the topic. The training and test topics are the team names of the W3C, and the returned candidates should be ranked by their relevance to the team or how professional they are in the team.

To treat this strange requirement, Conrad and Utt [8] linked the name with features. A method with Latent Semantic Indexing was introduced by Dumais and Nielsen [9]. Maybury, et al. [10] made the experts into a network, then used clustering and network analysis techniques to find experts. For this task, our effort focused on 3 aspects:

(1) How to find the relevant experts and how to rank them?

(2) Is it necessary to use the full corpus? Or could we get a better performance by using part of the corpus?

(3) How to deal with the homonymy?

3.1 Two-Stage Rank

To find experts relevant to a query, a natural way is to retrieve documents relevant to the query first and then find experts within those documents. This two-stage rank is defined as:

$$Score(E_j, Q) = \sum_{i=1}^{N_r} (Score(D_i, Q) \times Score(E_j, D_i)) \quad (5)$$

Here query-document similarity scores are combined with document-candidate similarity scores to calculate how relevant an expert to a query. N_r is a parameter which means the amount of documents taken from the document rank result for calculating $Score(E_j, Q)$. In our experiments we set N_r to 100 and took from the top rank.

We used the BM25 weight and a language-model based on KL-divergence to rank documents, both with a pseudo feedback. Two factors are considered to find relevant experts: the name frequency (nf) and the inverted document frequency (idf). Candidates who appear in a document are more relevant to it than those who don't appear. However, if a candidate appears far and wide in the corpus, s/he is not so relevant to a particular document. We write the expert rank formula as:

$$Score(E_j, D_i) = nf_{ij} \times \log \frac{N+1}{df_j + 0.5} \quad (6)$$

where D_i is a document. E_j is a candidate. nf_{ij} is the count of the name and email [7] of X_j in D_i. N is the count of documents in the corpus. df_j is the count of documents in which E_j 's name or email occurs.

3.2 Corpus Refinement

There are some documents which do not contain any information of the candidates. According to the expert rank method mentioned above, if the top N_r relevant documents do not refer to any candidates, no expert will be ranked. A representative example is the dev category: to query EX47, 99% relevant documents in top N_r belong to the dev and no candidates returned in the result.

We refined the corpus by removing the dev sub-corpus. The results for both training and test are shown in Table 5.

Table 5: Sub-corpus experiments

	Average Precision		Improvement
	Full corpus	Without dev	
Training	0.4164	0.4004	-3.84%
Test	0.1498	0.1833	22.36%

According to Table 5, the average precision decreases on training query and increases on test query. Observing the query-document ranking results, we found that in the top 100 documents, 0.4 documents per query on training results and 6.1 documents per query on test results are from dev category. It can be assumed that only those dev-related queries would be impacted by cutting dev category from the corpus.

3.3 Name Disambiguation

According to the statistics on the candidate list of 1092 people, there are 218 groups of homonymy. The sizes of the groups are between 2 and 28. In the documents of a corporation, names seldom appear in the complete form. Two forms that usually appear are only the first name or a family name with a prefix like Mr., Mrs., Dr. and so on.

To distinguish homonymy during indexing, we used a set of rules in our experiments. The rules were designed according to people's reading custom, listed as below:

(1) If a name or email can be affirmed, it is counted as one occurrence.

(2) Assign A as one's first name, B as his/her surname, C as possible prefix.

(3) If a complete name A B occurred in position p1, and a name other than A, B, A B or C B occurs in position p2, all the A or C B between p1 and p2 is affirmed as A B.

(4) If a B without prefix follows the A B, it cannot be affirmed, because it must be another one's first name.

(5) If a name A B has no homonymy according to the candidate list, a single A or a C B anywhere can be affirmed.

3.4 Official Runs

The descriptions of our five runs are listed as follows and the results are shown in Table 6:

Run ID	document rank method	expert rank method	corpus
PRISEX1	kl + feedback	Expert Rank	corpus without dev
PRISEX2	BM25+feedback	raw nf	corpus without dev
PRISEX3	BM25+feedback	Expert Rank	corpus without dev
PRISEX4	kl + feedback	Expert Rank	full corpus
PRISEX5	kl + feedback + dir	Expert Rank	full corpus

Table 6: Expert search official runs

	PRISEX1	PRISEX5	PRISEX4	PRISEX2	PRISEX3
map	0.0560	0.0589	0.0597	0.1483	0.1833
R-prec	0.0668	0.0664	0.0694	0.1951	0.2269
bpref	0.4268	0.4344	0.4221	0.3614	0.4182
recip-rank	0.1097	0.0951	0.1233	0.4017	0.5614

4. Conclusion

Although the enterprise track is mutated from the web track, it differs from the web track in several ways such as known email finding, threads in discussion group and people ranking. Our experiments are designed corresponding to those features of the enterprise track. The results indicate that some methods can improve performance obviously like email structure information, the discussion group information and the expert rank formula, and some methods are not very stable like anchor text and sub corpus.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No.60475007), Key Project of Foundation of Ministry of Education of China (Grant No.02029), and Cross-Century Talents Foundation.

References

[1] Ricardo Baeza-Yates, Berthier Ribeiro-Neto. *Modern Information Retrieval.* New York: ACM Press, 1999.

[2] David Hawking. *Challenges in Enterprise Search.* Dunedin: Proceedings of the Australasian Database Conference, 2004.

[3] Wensi Xi, Jesper Lind, Eric Brill. *Learning Effective Ranking Functions for Newsgroup Search*. Sheffield: SIGIR 2004: 394-401, 2004.

[4] Kenricj Mock. *An experimental framework for email categorization and management.* New Orleans: Poster in SIGIR 2001: 392-393, 2001.

[5] Steve Whittaker, Candace Sidner. *Email overload: exploring personal information management of email*. Proceedings of the SIGCHI conference on Human factors in computing systems: 276—283, 1996.

[6] Paul Ogilvie, Jamie Callan. *Combining Document Representations for Known Item search.* Toronto: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval: 143-150, 2003.

[7] Byron Dom, Iris Eiron, Alex Cozzi, Yi Zhang. *Graph-based Ranking Algorithms for E-mail Expertise Analysis*. San Diego: DMKD03, 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003.

[8] Jack G. Conrad, Mary Hunter Utt. *A System for Discovering Relationships by Feature Extraction from Text Databases*. SIGIR 1994: 260-270, 1994.

[9] Susan T. Dumais, Jakob Nielsen. *Automating the Assignment of Submitted Manuscripts to Reviewers*. Denmark: Proceedings of the ACM SIGIR'92 15th International Conference on Research and Development in Information Retrieval, 233-244, 1992.

[10] Mark Maybury, Ray D'Amore, David House. *Expert Finding for Collaborative Virtual Environments*. Communications of the ACM 14(12): 55- 56, 2001.

[11] David W. McDonald, Mark S. Ackerman. *Just Talk to Me: A Field Study of Expertise Location.* Seattle: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW '98), 1998.

[12] David Hawking, Francis Crimmins, Nick Craswell, Trystan Upstill. *How valuable is external link evidence when searching enterprise webs?* Dunedin: Proceedings of the Australasian Databases Conference ADC2004, 2004.

[13] J.H. Lee. *Combining multiple evidence from different properties of weighting schemes.* Seattle: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval: 180-188, 1995.

[14] David W. McDonald, Mark S. Ackerman. *Expertise recommender: A flexible recommendation and Architecture.* Philadelphia: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW '00), 2000.

[15] Nick Craswell, David Hawking, Anne-Marie Vercoustre, Peter Wilkins. *P@NOPTIC Expert: Searching for Experts not just for Documents.* Poster in Proceedings of AusWeb'01, 2001.

[16] Dave Mattox, Mark Maybury, Daryl Morey. *Enterprise Expert and Knowledge Discovery.* München: International Conference on Human Computer International (HCI'99): 303-307, 1999.