BUPT at TREC 2006: Enterprise Track

Zhao Ru, Qian Li, Weiran Xu, Jun Guo
(Pattern Recognition and Intelligent System Lab,
Beijing University of Posts and Telecommunications, Beijing, China, 100876)
rudjao@hotmail.com, liqian4280@gmail.com

1. Introduction

This is the second time that Pattern Recognition and Intelligent System Lab (PRIS) participate in TREC. In enterprise track, our efforts have been focused on the expert search task this year. The goal is to develop more elaborate model for expert searching and find effective support providing method for new request.

2. Task Analysis

This year, the expert search task requires a list of support documents provided for each expert. The change implied that support documents for the potential experts should be found before getting the experts themselves, which is one of the natural ways for expert search. The two-stage ranking method we used last year was just following this way.

We develop an expert experience model using window-based method this year, in which our efforts were focused on the combination of using local content for evidence and quoting entire document for support. We also tried to treat some important types of data particularly both in the corpus and in a document. Finally the headings in every page were given a high weight. Each email author was given an additional weight for the confidence of their relationship with the email content.

All our experiments were based on the 4.2version of Lemur Toolkit[1], in which language model with Bayesian smoothing was used for relevance computing. For candidate location, the candidate list and the name disambiguation rules[1] used last year were still working this time. But we found there were some problems in encoding which would cause missing match for a few candidates. We accepted several encoding representation in our system. The detail of the expert experience model and some improvements are in the following analysis.

3. Expert Experience Model

Although our two-stage ranking method followed a reasonable way, it is found to have a disadvantage that the results are not related to the detail in a document. This year, we intended to use window-based method for expertise evidence extraction. We see the context of expertise evidence which is the window of text around a candidate occurrence as the expert experience. The experience of a candidate in a document was assumed to be about one topic. We call it an *experience molecule*. The probabilistic formula to compute candidate expertise according to the

[1]http://www.lemurproject.org/

experience molecule is shown as:

$$P(E/Q) = P(\bigcup E_d / Q) \quad (1)$$

Here E_d is the experience molecule of candidate E in document d. The expertise of a candidate is a set of all her/his experience molecules which, in our assumption, are mutually exclusive. Then the probability can be written as:

$$P(E/Q) = \sum_d P(E_d / Q) \quad (2)$$

Here the probability of a candidate being an expert of a given topic becomes the sum of the probabilities of her/his experience molecules in the condition. It is also obvious that the experience molecules can be ordered for their relevance according the probabilities. As an experience molecule represents the expertise of a candidate in one document, it is the relation between the candidate and the document. The more relevant the experience molecule is, the stronger support the document can give. So the documents can be ranked according their support.

As a convenience, we chose to use fixed-length window which contained 150 words. We trained our model using last year's topics. The performance was not as good as we expected. We found it was due to the huge number of experience molecules by which the effect of relevance value was badly affected. So it was reasonable to use the relevant molecule only. We chose the top 2000 experience molecules retrieved which were assumed to be relevant for that each query requires 100 experts and each expert requires 20 support documents. Table 1 shows the performance, compared with the two-stage ranking method last year.

Table 1. Effect of the Expert Experience Model

	MAP	Bpref	P@10
Two-stage Ranking	0.1833	0.4182	0.3080
Expert Experience	0.2160	0.5180	0.3400

4. Headword

There are two kinds of headword referred in a page: One is the words in headings enclosing a candidate name, the other is a candidate name in the headline. For the first one, the words must be contained in the experience molecules of the candidates nearby. A higher weight is given to those words for the reason that they are considered to be good at representing expertise. For the second one, all the words in the precinct of the headline are seen as relating to the headwords, i.e., the candidate the headwords represented. So the corresponding experience molecule is composed of all the words in that region, which seems more reasonable than the basic fixed-window-length region.

In our experiments, the words tagged with <TITLE>, <H1>, <H2> and <H3> were considered as headwords. When the name of a candidate was found in a heading, the new experience molecule was used instead. And when there was no candidate in a heading, the weight

of the headwords contained in an experience molecule was set to 3. Experiments had been done to test the validity of the treatment. Results are shown in Table 2.

Table 2. Effect of the Treatment to headwords

	MAP	Bpref	P@10
Baseline	0.2160	0.5180	0.3400
Baseline + Headword	0.2238	0.5312	0.3480

5. Model Improvement

The assumption that the experience molecules of a candidate in different documents are mutually exclusive is not well-founded. Because some documents in the enterprise data usually share more or less the same contents, and even some have duplications. So the problem becomes the over-completeness of the information for some candidates that too much repeated information in their experience molecules and the incompleteness for some others that their experience molecules can not wholly represent their expertise. A coefficient is introduced to balance the information of different candidates. Then the formula becomes:

$$P(E/Q) \propto \phi(N_f) \times \sum_d P(E_d/Q) \tag{3}$$

Here N_f is the number of experience molecules that candidate E has. $\phi(N_f)$ is represented as:

$$\phi(N_f) = \frac{R_f + \lambda}{N_f} \tag{4}$$

where R_f is the number of experience molecules that E has in the top 2000 retrieved. λ is a parameter.

In our experiments, we tested the performance of the refined model in different values of λ. Then a compare between the refined model and the original one was experimented.

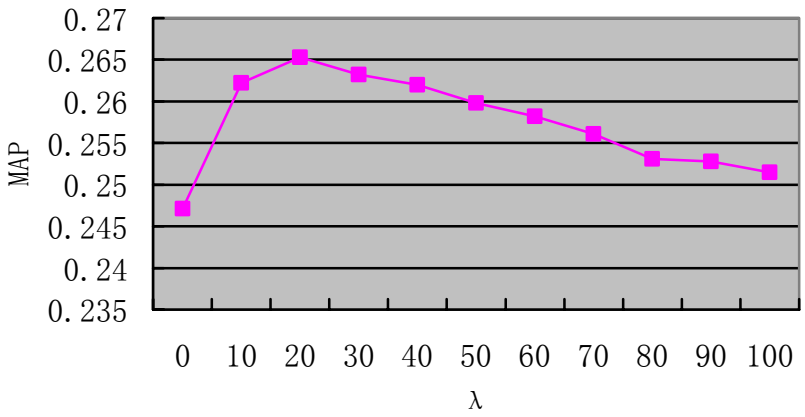

Figure 1. The Effect of Different λ

Figure 1 shows how the average precision changes according to different values of λ. It can be seen that the performance increases with higher value of λ, reaching its maximum around 20 and then decreases slowly. So we set λ to 20, and got the results shown in Table 3.

Table 3. Effect of Refined Model

	MAP	Bpref	P@10
Window-based Model	0.2238	0.5312	0.3480
Refine Model	0.2653	0.5537	0.4040

6. Email Author

The success of email expertise analysis[2] illuminates that it is valid to find expertise in email. The information extracted from email is more credible in dealing with candidates than that from the simple context. For that reason, the email content was added to the corresponding candidates as a special experience molecule, which also increased its weight as a support document. We tested the performance of the three conditions that the email as an additional experience molecule was given to the sender, the recipients, and the both. Table 4 shows the results. Finally we chose to give an additional experience molecule to the email author.

Table 4. Results of Adding Weight to Email

	MAP	Bpref	P@10
Baseline	0.2653	0.5537	0.4040
Baseline + Sender	0.2672	0.5586	0.3940
Baseline + Recipients	0.2667	0.5567	0.3960
Baseline + Both	0.2671	0.5599	0.3940

7. Submitted Runs

Four runs had been submitted, all of which were obtained using the expert experience model described in Section 3 with special treatment of headwords described in Section 4. The relevance computing method and the candidate finding method used were described in Section 2. The details of the four runs are listed as follows.

PRISEXB: Using the topic words in the <title> fields and the <desc> fields in the proportion of 5:1 as the queries.

PRISEXR: Using the same method as PRISEXB but refining the basic model as the description in Section 5.

PRISEXRM: Using the same method as PRISEXR but adding the email for experience molecules of the corresponding candidates.

PRISEXRMT: Using the same method as PRISEXRM except using the queries that are only the texts in the <title> fields.

Table 5. Results without Support Document

Run id	PRISEXB	PRISEXR	PRISEXRM	PRISEXRMT
MAP	0.5564	0.4724	0.4855	0.4991
bpref	0.5614	0.4766	0.4875	0.4942
P@10	0.6653	0.5551	0.5776	0.5776

Table 6. Results with Support Document

Run id	PRISEXB	PRISEXR	PRISEXRM	PRISEXRMT
MAP	0.3345	0.2877	0.3077	0.3133
bpref	0.4228	0.3705	0.3875	0.3892
P@10	0.4571	0.3918	0.4224	0.4245

The results in Table 5 show that the expert experience model is promising. But the refinement of the model which improved the performance on last year's topics gave a reverse effect this time. It is surprising. We can only assume that the refinement is topic-relating. Run PRISEXRM achieved better results than PRISEXR. We can see that evidence in email is more credible than in other types of documents. Run PRISEXRMT achieved even higher results, which indicates that our model performs better on short queries. Table 6 shows the results considering support documents. We can see the performance drops badly compared with those in Table 5. It must be due to the support documents were selected from the relevant experience molecules. We chose only 2000 relevant experience molecules so that the support documents were not enough.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No.60475007,

60675001), Key Project of Foundation of Ministry of Education of China (Grant No.02029), and Cross-Century Talents Foundation.

References

[1] Zhao Ru, Yuehua Chen, Weiran Xu, Jun Guo. *TREC 2005 Enterprise Track Experiments at BUPT*. In proceedings of TREC-2005, 2005.

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