Language Models for Expert Finding –UIUC TREC 2006 Enterprise Track Experiments

Hui Fang, Lixin Zhou, ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
USA

Abstract

In this paper, we report our experiments in the TREC 2006 Enterprise Track. Our focus is to study a language model for expert finding. We extend an existing language model for expert retrieval in three aspects. First, we model the document-expert association using a mixure model instead of name matching heuristics as in the existing work; such a mixture model allows us to put different weights on email matching and name matching. Second, we propose to model the prior of an expert based on the counts of email matches in the supporting documents instead of using uniform prior as in the previous work. Finally, we perform topic expansion and generalize the model from computing the likelihood to computing the cross entropy. Our experiments show that the first two extensions are more effective than the third extension, though when optimized, they all seem to be effective.

1 Introduction

The problem of *expert finding* is concerned with finding experts on a specified topic. The enterprise track [4] of TREC [9] provides a common platform for expert finding, which includes the following three components: (1) a supporting document collection; (2) a list of expert candidates, each of whom is described with name and email; (3) a set of topics (i.e., descriptions of expertise). The task is to rank the candidate experts for a given topic based on the information from a data collection.

Based on such a problem setting, many participants tried various methods. For example, Fu et. al. [5], Azzopardi et. al. [1] and Macdonald et. al. [6] built an expertise profile for each candidate, and rank the candidate experts based on the relevance score of their profiles to the given topic. Some heuristics have been used to identify mentions of candidate experts, and create expertise profiles. Cao et. al. [3] and Balog et. al. [2] proposed some probabilistic models for expert finding, but their approaches require some heuristics to improve the performance, such as specific name matching techniques.

We develop a new language model for expert finding by extending the model 2 proposed in [2] in three aspects. First, we propose a better way to model the document-expert association. Instead of using name matching heuristics [2], we use a mixture model to put different weights on name matching and email matching. Intuitively email matching should be weighted more than name matching. Second, we estimate the prior probability that a candidate is an expert based on the counts of email matches in the colletion. Finally, we perform topic expansion and generatlize the language model from computing the likelihood to computing the cross entropy.

Experiment results show that When optimized, these expansions seem to be all effective, but the email count-based prior and the weighting of email matching and name matching seem to be more effective than topic expansion.

2 Language Models for Expert Finding

2.1 Basic setup

The problem of expert finding is similar to the traditional ad hoc retrieval problem in the sense that both attempt to find "relevant" information items for a given query. The main difference is that the "information items" are persons in the case of expert finding, rather than documents as in traditional retrieval. A person is relevant to a topic query if and only if the person has the expertise on the topic specified in the query.

Following the probabilistic ranking principle [8], we can rank persons according to the probability that the person is relevant to the query. Thus, the key challenge is how to compute the probability that an expert is "relevant" to a topic (i.e., the expertise specified in the query).

Formally, suppose $S = \{d_1, ..., d_{|S|}\}$ is a collection of supporting documents. Let $t = t_1 t_2 ... t_m$ be the description of a topic, where t_i is a term in the description. Let c be an expert candidate whose email and name are denoted as $e(c)$, $n(c)$, respectively. Given a query t and an expert candidate c, we are interested in estimating the conditional probability $p(c|t)$, i.e., the probability that candidate c is an expert given the topic t. After applying Bayes' Theorem, we have

$$p(c|t) = \frac{p(t|c) \times p(c)}{p(t)}.$$

Since $p(t)$ is the same for all the experts, it can be ignored for the purpose of ranking experts. Thus, we have

$$p(c|t) \propto p(t|c) \times p(c), \quad (1)$$

where $p(t|c)$ is the probability that the topic t reflects the expertise of candidate c, and $p(c)$ is the prior probability that the candidate expert c is an expert.

Now the remaining questions are how to estimate $p(t|c)$ and how to estimate $p(c)$.

2.2 Setting the candidate prior $p(c)$

We first consider how to estimate $p(c)$, which is the prior probability of a candidate being an expert. In most existing work [3, 2], this probability is assumed to be uniform. However, as shown in Section 3, reasonable prior can help improve the retrieval accuracy. In this paper, we assume such a prior probability is related to the occurrences of the person's email, and model the email frequency in the same way as the term frequency in Okapi [7]. Specifically, this prior probability is assumed to be given by

$$p(c) \propto \frac{c(e(c), S)}{c(e(c), S) + \beta}, \quad (2)$$

where $c(e(c), S)$ is the count of mentions of the email of candidate c in the collection S, and β is a parameter to control the skewness of the prior. A larger β would reduce the skewness of the prior (i.e., leading to a weaker prior), thus it can be interpreted as being inversely proportional to our confidence in this prior.

2.3 Modeling candidate-topic relationship $p(t|c)$

Second, we discuss how to estimate $p(t|c)$, the probability that t describes the expertise of candidate c. Similar to what other participants have been doing, we also exploit the co-occurrences of the topic terms and candidate mentions in the supporting documents and estimate $p(t|c)$ based on the supporting documents. Specifically, we use supporting documents as a "bridge" to connect t and c in the following way:

$$p(t|c) = \sum_{d \in S} p(t|d) \times p(d|c)$$

That is, we assume that a topic t can be "generated" from a candidate c by first "generating" a supporting document d from c, and then "generating" t from the document d. Intuitively, this formula is quite reasonable: $p(d|c)$ allows us to model the probability that a supporting document d mentions candidate c, while $p(t|d)$ allows us to model the probability that d matches t. A document d with high values for both $p(d|c)$ and $p(t|d)$ would contribute the most to our estimate of $p(t|c)$ which intuitively makes sense.

$p(t|d)$ can be computed efficiently by treating t as a query and using the standard query likelihood retrieval method. In order to compute $p(d|c)$, we may further rewrite the equation above as follows:

$$p(t|c) = \sum_{d \in S} p(t|d) \times p(d|c)$$

$$= \sum_{d \in S} p(t|d) \times \frac{p(c|d)p(d)}{p(c)}$$
$$= \sum_{d \in S} p(t|d) \times \frac{p(c|d)p(d)}{\sum_{d' \in S} p(c|d')p(d')}$$

The rewriting of $p(d|c)$ in terms of $p(c|d)$ allows us to treat the name and email of c as a query and use the standard query likelihood retrieval method to compute $p(c|d)$ efficiently.

If we have some prior knowledge about the supporting documents in S, we can exploit $p(d)$ to favor a certain type of documents in S (e.g., email messages). Here we simply assume that $p(d)$ is uniform, which leads to

$$p(t|c) \approx \sum_{d \in S} p(t|d) \times \frac{p(c|d)}{\sum_{d' \in S} p(c|d')} \quad (3)$$

Equation 3 is our basic expert retrieval formula. It has two components – $p(c|d)$ and $p(d|t)$, which we now describe how to compute.

2.4 Modeling candidate mentions $p(c|d)$

In general, $p(c|d)$ can be computed by treating the description of the candidate c (e.g., name and/or email) as a query and using a standard query likelihood retrieval method to score the document d.

The simplest method would be to concatenate the email $e(c)$ and the name $n(c)$ to form a query to represent candidate c, and $p(c|d)$ would thus be computed as

$$p(c|d) = p(\text{“}e(c), n(c)''|d) \quad (4)$$

However, intuitively, a name and an email have different characteristics. For example, using email alone to identify an expert could generate high-precision results, but using name alone to identify an expert could cause high-recall results. Thus, to capture this intuition, we propose to model $p(c|d)$ using a mixture model involving both $p(e(c)|d)$ and $p(n(c)|d)$. That is, $p(c|d)$ can be computed as

$$p(c|d) = \lambda_e \cdot p(e(c)|d) + (1 - \lambda_e) \cdot p(n(c)|d) \quad (5)$$

where λ_e is the weight of the email model. Since $e(c)$ and $n(c)$ are both text, they can be computed using the query likelihood retrieval model with Dirichlet prior smoothing as described in [11].

2.5 Incorporating topic expansion

Since t is a piece of text, $p(t|d)$ can be computed using the query likelihood retrieval method with Dirichlet prior smoothing [11]. However, the original topic description t tends to be quite short, so it may not be informative. We thus propose to use some pseudo feedback method (e.g., the model-based feedback method proposed in [10]) to estimate an enriched query model θ_t, and incorporate this query model into our expert finding model through generalizing the topic likelihood $p(t|d)$ as the cross entropy of the query model θ_t and the document model θ_d estimated based on d using Dirichlet prior smoothing.

That is, $p(t|d) \propto exp(\sum_w p(w|\theta_t) \log p(w|\theta_d))$. Clearly, if we set θ_t to the empirical word distribution in t, this would be equivalent to the original topic likelihood.

2.6 Parameters

There are five parameters in the proposed model: μ_t, μ_e, μ_n, λ_e, and β. μ_t, μ_e, and μ_n are the Dirichlet prior smoothing parameter involved in computing $p(t|d)$, $p(e(c)|d)$, and $p(n(c)|d)$, respectively. λ_e controls the relative weight on email matching as compared with name matching. β is a prior confidence parameter defined earlier in this section.

3 Experiments

3.1 Official runs

We submitted four runs on expert finding – UIUCe1, UIUCe2, UIUCeFB1, and UIUCeFB2. These runs differ in that in UIUCe1 and UIUCe2, we used the original topic description while in UIUCeFB1 and UIUCeFB2, we expanded the original topic description by using the model-based feedback approach [10] to add 50 terms extracted from the top 10 documents. In UIUCe1 and UIUCeFB1, $\beta = 2$ whereas in UIUCe2 and UIUCeFB2, $\beta = 5$. In all these experiments, we set $\mu_t = \mu_e = \mu_n = 100$ and $\lambda_e = 0.9$.

We use the entire corpus, and only the titles of the topic description. We have done minimal preprocessing, where we apply stemming with a Porter stemmer and no stop word is removed.

Table 1. Comparison of four official runs

Run	Regular Expert Ranking			Supported Expert Ranking		
	MAP	BPref	Prec@5doc	MAP	BPref	Prec@5doc
UIUCe1	0.3043	0.3110	0.4512	0.1489	0.2420	0.2531
UIUCe2	0.3364	0.3388	0.5388	0.1650	0.2582	0.3143
UIUCeFB1	0.3185	0.3364	0.5143	0.1521	0.2518	0.3061
UIUCeFB2	0.3114	0.3450	0.5020	0.1382	0.2467	0.2898

In Table 1, we compare these four official runs. We see that the results seem to be mixed. For unexpanded runs, a larger β value is clearly helpful, but for expanded runs, a smaller β is more beneficial. Overall, the unexpanded run with $\beta = 5$ performs the best.

We now report some performance from our preliminary and diagnostic runs. The collection used in Enterprise track of 2005 will be labeled as "Ent05", while the collection used in Enterprise track of 2006 will be labeled as "Ent06". We evaluate the methods with mean average precision (MAP).

3.2 Preliminary Experiments on Enterprise 2005

Table 2. Performance Comparison on Ent05

		query	expanded query
BL	priorBL	0.151	0.166
	prior	0.155	0.170
Mixture	priorBL	0.172	0.172
	prior	0.196	0.204

In our preliminary experiments, we evaluated the proposed models, and compared them with the baseline models over Enterprise 2005 collection. The optimal performance is shown in Table 2. "Mixture" means that we estimate $p(c|d)$ as a mixture model as in Equation 5 while "BL" means that $p(c|d)$ is estimated as $p(\text{“}e(c), n(c)''|d)$. "prior" means that $p(c)$ is estimated using Equation 2, and "priorBL" means that $p(c)$ is same for all the experts. "query" means that $p(t|d)$ is computed with query generation model described in [11], and "expanded query" means that $p(t|d)$ is computed with expanded query model using the model-based feedback method described in [10].

From Table 2, we can make the following observations. First, the mixture model in Equation 5 is a better model for $p(c|d)$. Second, the proposed estimate of prior in Equation 2 is better than the uniform prior. Third, the performance is always improved when $p(t|d)$ is estimated using expanded topic instead of query likelihood. Finally, the proposed model outperforms the model 2 proposed in [2] where optimal MAP is 0.1880.

3.3 Parameter sensitivity

We examined the parameter sensitivity for the three Dirichlet prior smoothing parameters (i.e., μ_t, μ_e, μ_n) in Figure (1). In every case, we change the value of one parameter while fixing the values of the other parameters. The figures show that the performance is relatively more stable w.r.t. the change of μ_e compared with the change of μ_t and μ_n. In addition, the optimal values of μs is around 500, which is much smaller than 2000, the recommended value in traditional ad hoc IR [11].

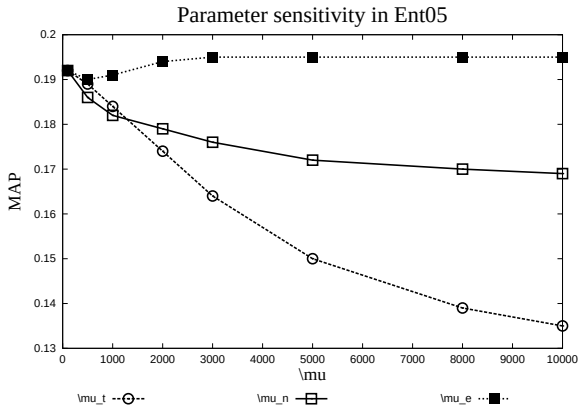

Figure 1. Performance Sensitivity in Ent05

Another paramter in topic generation model is λ_e introduced in Equation 5. Figure 2 shows that the performance is relatively stable when $\lambda < 0.9$.

Finally, there is one more paramter β to examine. The parameter sensitivity curve is shown in Figure 3.

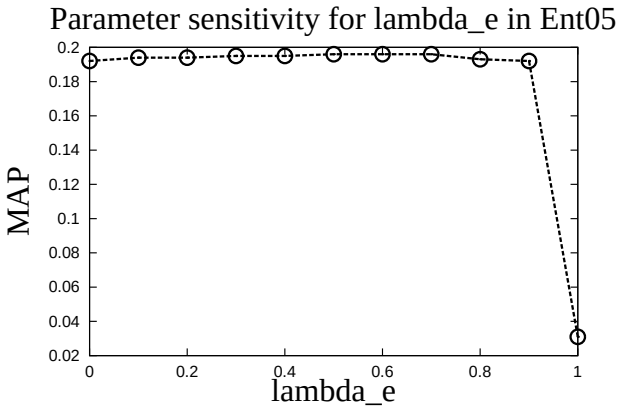

Figure 2. Performance Sensitivity (λ_e)

This figure shows that the performance is sensitive to the value of the paramter β.

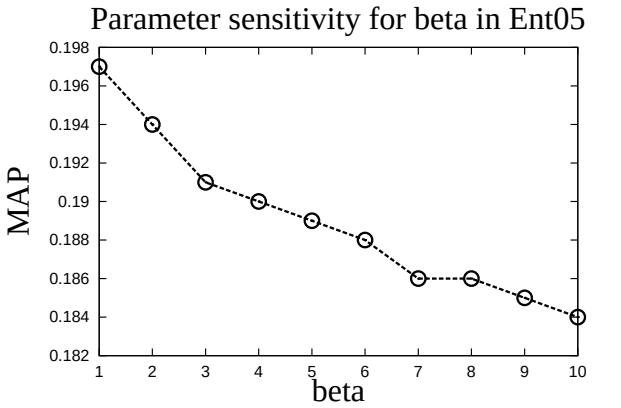

Figure 3. Performance Sensitivity of β in Ent05

3.4 Experiments on Enterprise 2006

Table 2 has demonstrated that the effectiveness of our estimation for expert prior probability (i.e., $p(c)$) and probability that an expert is relevant to a document (i.e., $p(c|d)$). In fact, we can also make similar observations from the results over Ent06.

In Table 4. "Ent05-Param" corresponds to the performance of using the parameter values trained over Enterprise 2005 collection. "Ent06-Param" corresponds to the optimal performance over Enterprise 2006 collection. The optimal parameter values trained over two collections are shown in Table 3.

Table 3. Optimal Parameter Values

Parameters	Ent05	Ent06
μ_t	100	100
μ_n	100	100
μ_e	100	100
λ_e	0.6	0.9
β	2.5	5

Table 4. Performance Comparison of Topic Generation Models on Ent06

		query	expanded query
Ent05-Param.	priorBL	0. 204	0.227
	prior	0.318	0.336
Ent06-Param.	priorBL	0.205	0.234
	prior	0.334	0.359

Table 3 and 4 show that the optimal paramter values are similar for both collections. And Table 4 also demonstrates that when optimized, the proposed three extensions, i.e., estimation of document-expert association, prior of an expert and topic expansion, seem to be all effective.

4 Summary

In our experiments with the TREC 06 Enterprise Track expert finding task, we evaluated some ideas of extending an existing language model, including assigning different weights to email matching and name matching, defining an email count-based prior, and incorporating topic expansion. When optimized, these expansions seem to be all effective, but the email count-based prior and the weighting of email matching and name matching seem to be more effective than topic expansion.

References

[1] L. Azzopardi, K. Balog, and M. de Rijke. Language modeling approaches for enterprise tasks. In *Proceedings of TREC-05*, 2006.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of SIGIR-06*, 2006.

[3] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of trec2005. In *Proceedings of TREC-05*, 2006.

[4] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec-2005 enterprise track. In *Proceedings of TREC-05*, 2006.

[5] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, and S. Ma. Thuir at trec 2005: Enterprise track. In *Proceedings of TREC-05*, 2006.

[6] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of glasgow at trec 2005: Experiments in terabyte and enterprise tracks with terrier. In *Proceedings of TREC-05*, 2006.

[7] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of SIGIR'94*, 1994.

[8] S. E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, Dec. 1977.

[9] E. Voorhees and D. Harman, editors. *Proceedings of Text REtrieval Conference (TREC1-9)*. NIST Special Publications, 2001. http://trec.nist.gov/pubs.html.

[10] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM-01*, 2001.

[11] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR-01*, 2001.