Language Models for Enterprise Search: Query Expansion and Combination of Evidence

Krisztian Balog **Edgar Meij** **Maarten de Rijke**

ISLA, University of Amsterdam
http://ilps.science.uva.nl/

Abstract: We describe our participation in the TREC 2006 Enterprise track. We provide a detailed account of the ideas underlying our language modeling approaches to both the discussion search and expert search tasks. For discussion search, our focus was on query expansion techniques, using additional information from the topic statement and from message threads; while the former was generally helpful, the latter mostly hurt performance. In expert search our main experiments concerned query expansion as well as combinations of expert finding and expert profiling techniques.

1 Introduction

Our aim for the discussion search task at TREC 2006 was to experiment with query expansion techniques. Our first method employs blind relevance feedback, by including content from message threads. Our second method enriches the query by using additional information from the topic statement. Additionally, we experiment with combining the results from the two different methods.

In expert search our main goal was to evaluate the methods that we have been developing recently within the TREC setting, on unseen data. Our baseline method calculates the probability of a candidate being an expert given the query topic, by iterating over all documents that are associated with the given person. We introduce the topical profile of an individual, which reflects the person's competency on a set of knowledge areas. Our experiments concern query expansion as well as combinations of expert finding and expert profiling techniques.

The rest of the paper is organized as follows. In two largely independent sections we first discuss our work on the discussion search task (Section 2) and then our work on the expert search task (Section 3). We conclude in Section 4.

2 Discussion Search

The aim of the discussion search task is to retrieve email messages that contain a discussion about a given topic, where highly relevant documents should introduce a new point to the discussion (such as pro or con given the topic). This year, our aim for the discussion search task was to experiment with various query expansion techniques. First, we employed blind relevance feedback, but instead of simply using the top ranked documents, we also included the contents of the accompanying threads. Next, we enriched the query by adding noun phrases from the description and narrative fields. In addition, we experimented with combining the outcomes of the different approaches.

2.1 Collection Processing

For the discussion search task, we used a cleaned version of the corpus of email forum documents [7]. We stemmed the collection (using Porter's stemmer) and used a standard stopword list (containing 457 terms).

2.2 Modeling

We addresssed the discussion search task using a language modeling approach. The standard query likelihood approach computes the probability of a query q being generated from a document model θ_d on behalf of the document d as follows:

$$p(q|\theta_d) = \prod_{t \in q} \left\{(1-\lambda)p(t|d) + \lambda p(t)\right\}^{n(t,q)}, \quad (1)$$

where $p(t|d)$ is the maximum likelihood estimate of term t in document d, $p(t)$ is the unconditional probability of t (also determined using the maximum likelihood estimate), $n(t,q)$ is the number of times term t occurs in query q, and λ is the smoothing parameter. If λ is set to $\frac{\beta}{n(d)+\beta}$, where $n(d)$ is the size of the document, Bayes Smoothing with a Dirichlet prior of the document model is obtained (instead of Jelinek-Mercer Smoothing) [13].

2.3 Query Expansion

We considered two ways of expanding queries, as detailed below.

2.3.1 Thread-Based Query Expansion

We experimented with the use of query expansion, using e-mail thread-based relevance feedback (tQE). Regular blind relevance feedback, as described by Ponte [10], adds new terms to the original query, based on an initial retrieval run. Terms that are indicative for these top-n ranked documents are selected, based on a comparison of their language model with the language model of the collection, and added to the original query.

As is well-known, including a large number of additional terms in a query may result in higher recall but also in more noise. We propose an alternative approach, based on the thread structure of the e-mail messages in the collection. Instead of using only the top ranked documents, we also include the contents of the accompanying threads to build a language model. The intuition is that these documents share the same subject and are thus more indicative of the information need than individual documents. Our method resembles the ideas behind Local Context Analysis [4, 12], which is theoratically appealing but has yielded mixed results. Table 1 shows some example topics with terms added by tQE.

Topic		Added term
68.	assistive technology evaluation tools	Kynn
90.	P3P vocabulary problems	policies
100.	intellectual property	ipr

Table 1: Examples of expansion terms generated by tQE

2.3.2 POS tagging

While evaluating last year's results, we noticed that relevant but not retrieved documents often included synonyms of query terms, but not the query terms themselves. These synonyms were included in the ⟨`narrative`⟩ fields of the topics, which we did not use. This year, we intended to tackle this by looking specifically at the ⟨`desc`⟩ and ⟨`narr`⟩ fields. We expected that indiscriminately adding the entire contents of these fields to the original ⟨`title`⟩ query would lead to too much query drift. We therefore used a Part of Speech (POS) tagger to identify noun phrases from the additional fields, and added only the noun phrases to the original query. We used a simple filter to exclude non-content bearing phrases that occur in almost every ⟨`narr`⟩ field; e.g., "documents."

2.4 Combining Results

Experiments on last year's topics suggested that the two query expansion methods outlined above would yield distinct sets of relevant results. We therefore decided to use a linear interpolation method to combine the results from two different methods, effectively blending in document likelihood probabilities from these distinct approaches [6, 9]:

$$p_{final}(q|\theta_d) = \mu \cdot p_1(q|\theta_d) + (1-\mu) \cdot p_2(q|\theta_d). \quad (2)$$

2.5 Runs

We submitted the following runs, all of which were automatic. After training on last year's data, we found the optimal value of β (in Equation 1) to be 120.

`UAmsBase` Baseline run using only the *title* field.

`UAmsThreadQE` Same as `UAmsBase`, but we expand the original query using tQE. We add 1 additional term from the top 3 returned threads.

`UAmsPOSBase` Linear combination of the `UAmsBase` run with the results of an expanded run based on the POS-tagged query terms from the ⟨`desc`⟩ and ⟨`narr`⟩ fields, using equal weights.

`UAmsPOStQE` Linear combination of the `UAmsThreadQE` run with the results of the POS expanded run, $\mu = 0.6$.

2.6 Results

The results displayed in Table 2 are computed based on non-argumented messages, whereas Table 3 displays the results when only argumented messages are considered relevant (best scores in boldface). We use the sign test to look for improvements over the baseline (one-tailed) at significance levels 0.95 (*) and 0.999 (**) [8].

Run id	rel_ret	MAP	bpref	P@10	r_rank
`Base`	6353	0.371	0.382	0.568	0.724
`ThreadQE`	6237	0.366	0.390	0.552**	0.703**
`POSBase`	**6441**	**0.375**	0.394	**0.592****	**0.778****
`POStQE`	6271	0.372	**0.408**	0.564*	0.685**

Table 2: Results for Discussion Search – Relevance Level 0

Run id	rel_ret	MAP	bpref	P@10	r_rank
`Base`	3815	0.251	0.261	**0.385**	0.565
`ThreadQE`	3741	0.249	0.273	0.376**	0.542**
`POSBase`	**3832**	**0.259**	0.274	0.383**	**0.603****
`POStQE`	3711	0.250	**0.277**	0.361**	0.541*

Table 3: Results for Discussion Search – Relevance Level 1

Incorporating terms from the ⟨`desc`⟩ and ⟨`narr`⟩ fields has a clear beneficial effect on retrieval effectiveness. The proposed tQE method, however, does not deliver the expected results. At both relevance levels, query drift occurs

due to the addition of non-relevant terms into the query. While on some topics postive results are clearly noticable, on others tQE hurts retrieval performance. This clearly leaves room for future optimization, such as detecting if and when query expansion is neccessary—a topic of ongoing research in itself [5, 11].

3 Expert Search

The Expert Search task presents the following scenario: Given the document repositories of the organization, find the experts on a particular topic, or in a particular field or area.

3.1 Modeling

We model the expert finding task as follows: *what is the probability of a candidate ca being an expert given the query topic q?* Instead of computing this probability $p(ca|q)$ directly, we can use Bayes' Theorem and estimate:

$$p(ca|q) = \frac{p(q|ca)p(ca)}{p(q)}, \tag{3}$$

where $p(ca)$ is the probability of a candidate, and $p(q)$ is the probability of a query. Since $p(q)$ is a constant, it can be ignored for the purpose of ranking. The *apriori* belief that candidate ca is an expert, $p(ca)$, is assumed to be uniform. Thus, we rank candidates in proportion to $p(q|ca)$, the probability of the query given the candidate.

We first find documents which are relevant to the query topic, and then score each candidate by aggregating over all documents associated with that individual. That is,

$$p(q|ca) \propto \sum_{d} p(q|d)p(ca|d). \tag{4}$$

To determine $p(q|d)$, the probability of a query given a document, we use a standard language modeling for IR approach (see Section 2.2). To estimate the strength of the association between document d and candidate ca, $p(ca|d)$, we assume that an association score $a(d, ca)$ has been calculated for all documents and candidates. To turn these associations into probabilities, we put

$$p(ca|d) = \frac{a(d, ca)}{\sum_{ca' \in C} a(d, ca')}, \tag{5}$$

where C is a set of all candidates. The probability $p(ca|d)$ expresses the level of contribution that candidate ca made to document d.

The modeling described here corresponds to the expert finding *Model 2* using *candidate-centric* associations, introduced by Balog et al. [2]. For a more detailed account of the modeling we refer the reader to [2].

3.2 Document-Candidate Associations

Document-candidate associations form an essential part of the model presented in Section 3.1. We need to assign non-negative association scores $a(d, ca)$ to all document-candidate pairs.

The recognition of candidates is a (restricted and) specialized named-entity recognition task, and we approach it in a rule-based manner. We introduce two binary association methods (A_0, A_1) that return 0 or 1 depending on whether the document d is associated with candidate ca.

A_0**: NAME_MATCH** returns 1 if the name of the candidate appears in the document (first and last names are mandatory, middle names are optional)

A_1**: EMAIL_MATCH** returns 1 if the e-mail address of the candidate appears in the document.

Then, we combine the extraction methods from the two groups, and association scores are defined by considering the linear combination of their outcomes:

$$a(d, ca) = 0.55 \cdot A_0(d, ca) + 0.45 \cdot A_1(d, ca). \tag{6}$$

3.3 Supporting Documents

Runs submitted for the Expert Search task required not only a ranked list of experts, but also a ranked list of (up to 20) documents for each returned candidate that support the person's expertise on the given topic. We performed the selection of supporting documents in the following manner. For each topic q_i we ranked documents according to $p(q_i|d)$. For each candidate ca, considered as an expert, the top (up to) 20 documents that are associated with the person $(a(d, ca) > 0)$ were returned as support.

3.4 Query Expansion

We experimented with expanding the original query with noun phrases, which are extracted from the ⟨`desc`⟩ and ⟨`narr`⟩ fields of the topic. We applied the same method described earlier for the discussion search task (see Section 2.3.2).

3.5 Topical Profiles

A *topical profile* of an individual is a record of the types and areas of skills and knowledge of that individual, together with an identification of levels of 'competency' in each [3].

The profile of a candidate is represented as a vector, where each element of the vector corresponds to the person's skills on the given knowledge area. This skill is expressed by a score (not a probability), reflecting the person's (absolute) knowledge on the given topic. In the TREC Enterprise setting we used the query topics $(q_i, i = 1 \ldots n)$ as knowledge

areas. Then, the profile of a candidate becomes:

$$profile(ca) = \langle score(ca, q_1), score(ca, q_2), \ldots, score(ca, q_n) \rangle$$

Balog and de Rijke [3] introduced several methods for calculating the 'competency' scores for candidate and knowledge area pairs. We adopted their best performing setting ("Profiling Method 1") as follows.

For each knowledge area (query topic) q_i a query-biased subset of documents D_{q_i} is obtained by using the top n (= 500) documents retrieved for the query q_i. We iterate over the relevant documents, and sum up the relevance of those that are associated with the given candidate. Formally, the score of an individual ca given the knowledge area q_i is:

$$score(ca, q_i) = \sum_{d \in D_{q_i}, a(d,ca)>0} p(q_i|d) \quad (7)$$

Conceptually, this method is similar to the model we used for expert finding, but associations are not turned into probabilities, thus their strength is not estimated—practically (and realistically) speaking, we simply cannot capture the extent to which the candidate is responsible for a document's content, compared to other individuals that may also be associated with the same document.

To make use of the extracted profiles we introduce a reranking method which adjusts the results of the expert finding algorithm using the individuals' profiles. Specifically, if a knowledge area ranks low on a person's profile, we push the candidate down on the list of experts returned as the system's output. The intuition behind this method is to rank candidates high that have a reasonable knowledge on the topic (compared to others within the organization), moreover their work is focused on the given area. We expect this idea to have a precision enhancing effect, while possibly hurting recall.

We do not make any assumptions about the scores produced by the expert finding (EF) and profiling (PR) methods, which leaves us no other option than to use the ranking of their results. We combine the ranks in a multiplicative way:

$$rank(ca, q_i) = \frac{1}{rank_{EF}(ca, q_i)} \cdot \frac{1}{rank_{PR}(ca, q_i)}. \quad (8)$$

Balog and de Rijke [3] experimented with two ways of combining scores, here we took their best performing setting.

3.6 Runs

We submitted the following 4 runs:

UvAbase Baseline run (expert finding only)

UvAPOS Baseline run + POS query expansion

UvAprofiling Combination of expert finding and profiling (as specified in Equation 8)

UvAprofPOS UvAprofiling + POS query expansion

3.7 Results

Table 4 and 5 give our overall results for the Expert Search task, using the various evaluation measures proposed by the track organizers; our best score per measure is indicated in bold face. Results presented in Table 4 are computed based solely on expert ranking, while Table 5 contains results where candidates without any positive support document retrieved were considered irrelevant.

The scores produced by our baseline method (`UvAbase`) are higher than the highest scores achieved on the TREC 2005 Expert Search topics. We believe that this may be due to the nature of the topics. This year's (manual) queries (and assessments) resulted in a more realistic test set.

The use of topical profiles (`UvAprofiling`) shows a very positive impact on the precision scores, and our reranking method—in spite of being fairly simple—improved significantly on all measures. The results support the view that the expert finding and profiling methods capture evidence that differs in nature, and the combination of these two approaches is beneficial to retrieval performance. Further improvements could be pursued using more sophisticated methods for combining the retrieval results.

The query expansion technique we applied (`UvAPOS`, `UvAprofPOS`) has a negative impact on retrieval performance. At this stage, we do not have a clear explanation for this, but anecdotal evidence suggests that expansion causes serious topic drift because the number of relevant documents associated with any given candidate expert is fairly small.

4 Conclusions

In this paper we described our participation in the TREC 2006 Enterprise track. Following up on the approach we used last year [1], we employed a standard language modeling setting for both tasks.

Our aim for the discussion search task was to experiment with various query expansion techniques. Our first method employs blind relevance feedback, but instead of using the top ranked documents, we also include the contents of the accompanying threads. Our second method enriches the query by adding noun phrases from the description and narrative fields. We also experimented with combining the outcomes of the different approaches. Results indicate that adding terms from the description and narrative fields helps in most cases but not all. Thread-based query expansion did not deliver the desired results, due to topic drift.

As to the expert search task, our baseline method calculates the probability of a candidate being an expert given the query topic. This probability is estimated by iterating over

Run	#rel_ret	MAP	r-prec	bpref	P@5	P@10	P@20	P@30	RR1
UvAbase	**967**	0.3280	0.3777	0.3684	0.3959	0.4082	0.3776	0.3456	0.5065
UvAPOS	936	0.2053	0.2145	0.2585	0.2898	0.2510	0.2306	0.2007	0.4260
UvAprofiling	**967**	**0.4664**	**0.4957**	**0.4707**	**0.6612**	**0.5878**	**0.4959**	**0.4367**	0.8510
UvAprofPOS	936	0.4249	0.4533	0.4358	0.6286	0.5571	0.4653	0.4116	**0.8517**

Table 4: Results for the Expert Search task (computed based solely on expert ranking, supporting documents are not taken into account).

Run	#rel_ret	MAP	r-prec	bpref	P@5	P@10	P@20	P@30	RR1
UvAbase	**686**	0.2049	0.2703	0.3099	0.3143	0.3102	0.2816	0.2578	0.4559
UvAPOS	596	0.1152	0.1273	0.2225	0.1429	0.1327	0.1296	0.1156	0.2587
UvAprofiling	**686**	**0.3016**	**0.3637**	**0.3743**	**0.4980**	**0.4265**	**0.3582**	**0.3238**	**0.7177**
UvAprofPOS	596	0.2588	0.3121	0.3248	0.4367	0.3837	0.3122	0.2735	0.6259

Table 5: Results for the Expert Search task (supporting documents are taken into account).

all documents that are associated with the given person. We experimented with query expansion techniques for the expert search task as well, but these automatic methods were not able to deliver the desired results, and had a negative effect on retrieval performance. Furthermore, we introduced the topical profile of an individual, which reflects the person's competency on a set of knowledge areas. The expert search topics were used as knowledge areas, and the topical profile of each W3C candidate was calculated. A rank-based combination of expert finding and profiling methods resulted in remarkable improvements over the baseline.

Acknowledgments

Krisztian Balog was supported by the Netherlands Organisation for Scientific Research (NWO) under project numbers 220-80-001, 600.065.120 and 612.000.106. Edgar Meij's work was carried out in the context of the Virtual Laboratory for e-Science project (http://www.vl-e.nl). This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ). Maarten de Rijke was supported by NWO under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 600.065.120, 612-13-001, 612.-000.106, 612.066.302, 612.069.006, 640.001.501, 640.002.-501, and by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104.

References

[1] L. Azzopardi, K. Balog, and M. de Rijke. Language modeling approaches for enterprise tasks. In E. M. Voorhees and L. P. Buckland, editors, *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, Gaithersburg, Maryland, November 15-18 2005.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA, 2006. ACM Press.

[3] K. Balog and M. de Rijke. Determining expert profiles (with an application to expert finding). In *IJCAI '07: Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2657–2662, 2007.

[4] N. J. Belkin, C. Cool, J. Head, J. Jeng, D. Kelly, S. jeng Lin, L. Lobash, S. Park, P. A. Savage-Knepshield, and C. Sikora. Relevance feedback *versus* local context analysis as term suggestion devices: Rutgers' trec-8 interactive track experience. In D. Harman and E. Voorhees, editors, *TREC-8, Proceedings of the Eighth Text Retrieval Conference.*, pages 565–574, Washington, D. C., 2000. NIST.

[5] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 390–397, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-369-7.

[6] E. A. Fox and J. A. Shaw. Combination of multiple searches. In D. K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 243–252. National Institute for Standards and Technology. NIST Special Publication 500-215, 1994.

[7] D. He. Cleaned W3C-lists collection (for TRECENT 2005), 2005. http://www.sis.pitt.edu/~daqing/w3c-cleaned.html.

[8] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *SIGIR '93: Proc. of the 16th annual international ACM SIGIR conference on Research*

and development in information retrieval, pages 329–338, 1993.

[9] J. Kamps and M. de Rijke. The effectiveness of combining information retrieval strategies for european languages. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1073–1077, New York, NY, USA, 2004. ACM Press.

[10] J. M. Ponte. Language models for relevance feedback. In W. B. Croft, editor, *Advances in Information Retrieval*, The Kluwer International Series in Information Retrieval, chapter 3, pages 73–95. Kluwer Academic Publishers, Boston, 2000.

[11] R. Sun, C.-H. Ong, and T.-S. Chua. Mining dependency relations for query expansion in passage retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 382–389, New York, NY, USA, 2006. ACM Press. ISBN 1-59593-369-7.

[12] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000. ISSN 1046-8188.

[13] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 49–56, Tampere, Finland, 2001. ACM Press.