Macdonald, C. and Ounis, I. (2007) Using relevance feedback in expert search. *Lecture Notes in Computer Science* 4425:pp. 431-443.

http://eprints.gla.ac.uk/3764/

Deposited on: 19 October 2007

Using Relevance Feedback in Expert Search

Craig Macdonald and Iadh Ounis

Department of Computing Science,
University of Glasgow, G12 8QQ, UK
{craigm,ounis}@dcs.gla.ac.uk

Abstract. In Enterprise settings, expert search is considered an important task. In this search task, the user has a need for expertise - for instance, they require assistance from someone about a topic of interest. An expert search system assists users with their "expertise need" by suggesting people with relevant expertise to the topic of interest. In this work, we apply an expert search approach that does not explicitly rank candidates in response to a query, but instead implicitly ranks candidates by taking into account a ranking of document with respect to the query topic. Pseudo-relevance feedback, aka query expansion, has been shown to improve retrieval performance in adhoc search tasks. In this work, we investigate to which extent query expansion can be applied in an expert search task to improve the accuracy of the generated ranking of candidates. We define two approaches for query expansion, one based on the initial of ranking of documents for the query topic. The second approach is based on the final ranking of candidates. The aims of this paper are two-fold. Firstly, to determine if query expansion can be successfully applied in the expert search task, and secondly, to ascertain if either of the two forms of query expansion can provide robust, improved retrieval performance. We perform a thorough evaluation contrasting the two query expansion approaches in the context of the TREC 2005 and 2006 Enterprise tracks.

1   Introduction

In large Enterprise settings with vast amounts of digitised information, it is often important that a user is not only be able to identify documents that are relevant to a topic of interest, but also to find people that have relevant expertise to the topic. People are a critical source of information because they can explain and provide arguments about why specific decisions were made [5]. Hence, in addition to classical document Information Retrieval (IR) systems, there is a growing interest in the research community to build accurate expert search systems. An *expert search* system aids a user in their "expertise need" by identifying people with relevant expertise to the topic of interest.

The retrieval performance of an expert search system is very important. If an expert search system suggests incorrect experts, then this could lead the user to contacting these people inappropriately. Similar to document IR systems, the accuracy of an expert search system can be measured using the traditional

IR evaluation measures: *precision*, the accuracy of suggested candidate experts; and *recall*, the number of candidate experts with relevant expertise retrieved. Expert search was a retrieval task in the Enterprise tracks of the Text REtrieval Conferences (TREC) since 2005 [4], aiming to evaluate expert search approaches.

Pseudo-relevance feedback (PRF) [10] has been used in adhoc search tasks to improve the performance of document IR systems. PRF describes the process of automatically examining top-ranked documents in an IR system ranking, and using information from these documents to improve the ranking of documents. This is done by assuming that the top-ranked documents are relevant, and using information from this 'pseudo-relevant set' to improve the accuracy of the ranking by expanding on the initial query and re-weighting the query terms[1].

In this paper, we explore how query expansion can be applied in an expert search task. To this end, we experiment with an expert search system that is based on the voting model for expert search [8]. In this model, documents are firstly associated with candidates to represent the candidates expertise. Then the voting model considers the ranking of documents with respect to the query, in order to generate an accurate ranking of candidates. The voting model for expert search is interesting for these experiments, as we can apply query expansion using the underlying ranking of documents as the pseudo-relevant set. Moreover, we investigate how query expansion can be applied if the ranking of candidates is used as the pseudo-relevant set. We call these approaches document-centric, and candidate-centric query expansion respectively.

In this work, our objectives are two fold: firstly, to determine if query expansion can be successfully applied in expert search; and secondly, to analyse both forms of query expansion, allowing conclusions to be drawn concerning the applicability and effectiveness of both approaches. In order to fully understand the applicability of query expansion, we experiment using two statistically different models from the Divergence from Randomness (DFR) framework for extracting informative terms from the pseudo-relevant set - one model based on the Bose-Einstein statistics and is similar to Rocchio [10], and one based on Kullback-Leibler divergence [1]. Furthermore, we experiment using two different voting techniques for ranking candidates, using the topics and relevance assessments for the W3C collection from the TREC 2005 and 2006 Expert Search tasks. Conclusions are drawn across the two voting techniques applied.

Section 2 provides further detail on the model for expert search that we employ in this work, and demonstrates the baselines achieved using this approach. Section 3 defines how query expansion can be applied to expert search. We experimentally investigate both approaches for query expansion in Section 4. Section 5 investigates the effect of varying the parameters of query expansion, to assess the maximum potential and stability of each approach. Section 6 provides concluding remarks and suggestions for future work.

---

[1] In this work, we use the terms pseudo-relevance feedback and query expansion interchangeably.

2   Expert Search

Modern expert search systems for Enterprise settings work by using documents to form the profile of textual evidence for each candidate. The candidate's profile represent the expertise of the candidate expert to the expert search system. This documentary evidence can take many forms, such as intranet documents, documents or emails authored by the candidates, or even emails sent by the candidate or web pages visited by the candidate (see [8] for an overview). In this work, the profile of a candidate is considered to be the set of document associated with the candidate. These candidate profiles can then be used to rank candidates automatically in response to a query.

This work uses the voting approach for expert search proposed by Macdonald & Ounis in [8], which considers the problem of expert search as a voting process. Instead of directly ranking candidates, it considers the *ranking of documents*, with respect to the query Q, denoted by $R(Q)$. The ranking of candidates can then be modelled as a voting process, from the retrieved documents in $R(Q)$ to the profiles of candidates: every time a document is retrieved and is associated with a candidate, then this is a vote for that candidate to have relevant expertise to Q. Each document retrieved by the IR system, that is associated with the profile of a candidate, can be seen as a vote for that candidate to have relevant expertise to the query topic. The ranking of the candidate profiles can then be determined by aggregating the votes of the documents. Eleven voting techniques for ranking experts were defined in [8], which each employ various sources of evidence that can be derived from the ranking of documents with respect to the query topic. In this work, we only use the CombSUM and exp-CombMNZ voting techniques, because they provide robust results on the W3C collection. The CombSUM technique ranks candidates by considering the sum of the relevance scores of the documents associated with each candidate's profile. Hence the relevance score of a candidate expert C with respect to a query Q, $score\_cand(C, Q)$, is:

$$score\_cand(C, Q) = \sum_{d \in R(Q) \cap profile(C)} score(d, Q) \tag{1}$$

where $profile(C)$ is the set of documents associated with candidate C, and $score(d, Q)$ is the relevance score of the document in the document ranking $R(Q)$. For expCombMNZ, the relevance score of a candidate C's expertise to a query Q is given by:

$$score\_cand_{expCombMNZ}(C, Q) = \|R(Q) \cap profile(C)\|$$
$$\cdot \sum_{d \in R(Q) \cap profile(C)} exp(score(d, Q)) \tag{2}$$

where $\|R(Q) \cap profile(C)\|$ is the number of documents from the profile of candidate C that are in the ranking $R(Q)$, and $exp()$ is the exponential function. expCombMNZ is similar to CombSUM, but also includes a second component

which takes into account the number of documents in $R(Q)$ associated to each candidate, hence explicitly modelling the number of votes made by the documents for each candidate. The exponential function boosts candidates that are associated to highly scored documents (strong votes).

In the remainder of this section, we define the strong baselines that we deploy for our experiments. Secondly, we provide details on two statistically different query expansion (QE) techniques based on the Divergence from Randomness (DFR) framework. We employ two QE techniques in our experiments to ensure our drawn conclusions are general.

2.1   Baselines

In this section, we define our experimental setup, and the baselines we use in this work. Our experiments are carried out in the setting of the Expert Search tasks of the TREC Enterprise track, 2005 and 2006. The TREC W3C collection is indexed using Terrier [9], removing standard stopwords and applying the first two steps of Porters stemming algorithm. Initial experimental results have shown that applying only this weaker form of stemming results in increased high precision without degradation in mean average precision (MAP) for this task.

Next, we generate the profiles of documentary evidence of expertise for the candidates: for each candidate, documents which contain an exact match of the candidates full name are used as the profile of the candidate.

From the two TREC expert search tasks, we have a total of 99 topics with relevance assessments. Documents are ranked using the DLH13 document weighting model from the DFR framework. The DLH13 document weighting model is a generalisation of the parameter-free hypergeometric DFR model in a binomial case [2,7]. The hypergeometric model assumes that the document is a sample, and the population is from the collection. For the DLH13 document weighting model, the relevance score of a document d for a query Q is given by:

$$score(d, Q) = \sum_{t \in Q} \frac{qtw}{tf + 0.5} \cdot \left(\log_2(\frac{tf \cdot avg\_l}{l} \cdot \frac{N}{F}) \right.$$
$$\left. + 0.5 \log_2 \left(2\pi tf(1 - \frac{tf}{l}) \right) \right) \tag{3}$$

where tf is the term frequency of the term t in document d, F is the frequency of the query term in the collection and N is the number of documents in the collection. l is the length of the document d in tokens, and avg_l is the average document length in the whole collection. The query term weight qtw is given by qtf/qtf_{max}. qtf is the query term frequency. qtf_{max} is the maximum query term frequency among the query terms.

We chose to experiment using DLH13 because it has no term frequency normalisation parameter that requires tuning, as this is assumed to be inherent to the model. Moreover, DLH13 performs robustly on many collections and tasks without any need for parameter tuning [7]. By applying DLH13, we remove the presence of any term frequency normalisation parameter in our experiments.

In this work, we could also experiment with other weighting models. However, it was shown that the relative performance rankings of the voting techniques were concordant across a selection of weighting models, on the same W3C collection [8]. This infers that conclusions drawn using one document weighting model should be applicable to any other state-of-the art model.

Table 1 shows the retrieval performances achieved by the baseline expert search approach we employ in this paper. We report MAP and P@10 evaluation measures. The retrieval performance is reported on the TREC 2005 and 2006 topics. In addition, we also report the median run of MAP for each year (the median P@10 runs are not available). As apparent from Table 1, the voting

Table 1. Baseline performances of CombSUM and expCombMNZ, using the DLH weighting model, on the 2005 and 2006 TREC Enterprise track, expert search tasks. For TREC 2005, topics only had one fields, while for TREC 2006, we use title-only (short) queries. Mean average precision (MAP) and Precision at 10 (P@10) measure are reported. The MAP median runs of all participants from the respective year of TREC are given. Moreover, the best result for each measure are emphasised.

	TREC 2005		TREC 2006	
	MAP	P@10	MAP	P@10
Median	0.1402	-	0.3412	-
CombSUM	**0.2037**	**0.3240**	0.5188	0.6388
expCombMNZ	**0.2037**	0.3100	**0.5502**	**0.6837**

techniques are clearly performing well above the median run for both years. Moreover, these results for the TREC 2005 Enterprise task are similar to those of the 3rd top group participating that year. For TREC 2006, the ranking results are not yet publicly available, but with such a large margin over the median run, these results appear strong. The voting approach is robust and general, as it is not dependent on heuristics based on the used enterprise collection.

In [3], Balog et al. defined a language modelling approach for expert search. However, in contrast to the voting approach by Macdonald & Ounis, their approach can only be applied in a language modelling setting. The voting model approach is more flexible, because any approach (including language modelling) can be used to generate the document ranking $R(Q)$. However, there are some similarities between the two approaches. In particular, for the voting approach, if Hiemstra's language modelling approach [6] was used to generate $R(Q)$, and CombSUM applied to combine the scores for candidates, then this would be identical to the candidate ranking formula of Equation (4) in [3]. For this reason, we do not experiment using the Balog et al. language modelling approach, as its characteristics are encapsulated in the CombSUM voting technique[2].

---

[2] In fact, experimental evaluation of Balog et al's approach on the same profile set, provides similar results to Hiemstra's language model combined with CombSUM voting technique.

Because the voting approach allows any IR technique to be used to generate the ranking of documents $R(Q)$, we wish to determine the extent to which the performance of the approach can be improved by increasing the quality of the document ranking. An obvious way to apply QE is to use the top-ranked documents in $R(Q)$ as the pseudo-relevant set. However, we also propose an alternative approach for applying query expansion, namely using the top-ranked candidates as the pseudo-relevant set.

2.2   Query Expansion Models

In our investigation into query expansion (QE) in expert search, we need to determine if the QE model employed has any effect on the conclusions concerning our two approaches for query expansion. Hence, we employ two statistically different QE models from the DFR framework [1], known as term weighting models, for extracting informative terms from the pseudo-relevant set of top-ranked documents. DFR term weighting models measure the informativeness of a term by considering the divergence of the term occurrence in the pseudo-relevant set from a random distribution.

Terrier provides various DFR-based term weighting models for query expansion. We experiment with two term weighting models to understand the importance of the choice of model. One term weighting model, known as Bo1, is based on Bose-Einstein statistics and is similar to Rocchio [1]. The other is based on the Kullback Leibler (KL) divergence between the pseudo-relevant set sample and the collection. In Bo1, the informativeness $w(t)$ of a term t is given by:

$$w(t) = tf_x \cdot log_2 \frac{1 + P_n}{P_n} + log_2(1 + P_n) \tag{4}$$

where tf_x is the frequency of the term in the pseudo-relevant set, and P_n is given by $\frac{F}{N}$. F is the term frequency of the query term in the whole collection and N is the number of documents in the collection.

Alternatively, $w(t)$ can be calculated using the Kullback Leibler divergence term weighting model [1]:

$$w(t) = P_x \cdot log_2 \frac{P_x}{P_c} \tag{5}$$

where $P_x = \frac{tf_x}{l_x}$ and $P_c = \frac{F}{token_c}$. l_x is the size in tokens of the pseudo-relevant set, and $token_c$ is the total number of tokens in the collection. Note that unlike Bo1, KL uses the size of the pseudo-relevant set while measuring divergence.

Using either Bo1 or KL to define $w(t)$, the top exp_term informative terms are identified from the top exp_doc ranked documents, and these are added to the query ($exp\_term \geq 1$, $exp\_doc \geq 2$). The default setting for these parameters is $exp\_doc = 3$ and $exp\_term = 10$, suggested by Amati in [1] after extensive experiments. Finally, for both the Bo1 and KL term weighting models, the query term frequency qtw of an expanded query term is given by [1]:

$$qtw = qtw + \frac{w(t)}{w_{max}(t)} \tag{6}$$

where $w_{max}(t)$ is the maximum $w(t)$ of the expanded query terms. qtw is initially 0 if the query term was not in the original query.

3   Applying QE in Expert Search Task

Our work concerns the applicability of QE to expert search. The application of QE in adhoc search tasks is known to improve retrieval performance. Using the voting model described in Section 2, it can be seen that the quality of the generated ranking of candidates is dependent on how well $R(Q)$ ranks documents associated with relevant candidates. Then any improvement in the quality of the document ranking should improve the accuracy of the retrieved candidate ranking, because the document ranking votes will be more accurate, and hence the aggregated ranking of candidates will be more accurate.

We call *document-centric query expansion*, the approach that considers the top-ranked documents of the document ranking $R(Q)$ as the pseudo-relevant set. We hypothesise that the candidate ranking generated by applying a voting technique to the refined document ranking will have increased retrieval performance, when compared to applying the voting technique to the initial $R(Q)$.

Moreover, we propose a second approach called *candidate-centric query expansion* where the pseudo-relevant set is taken from the final ranking of candidates generated by a query. If the top-ranked candidates are defined to be the pseudo-relevance set, then we can extract informative terms from the corresponding candidate profiles, and use these to generate a refined ranking of documents. In using this expanded query, we hypothesise that the document ranking will become nearer to the expertise area of the initially top-ranked candidates, and hence the generated candidate ranking will likely include more candidates with relevant expertise.

In the following section, we assess the usefulness of both forms of QE compared to the baseline approaches defined in Section 2. It is of note that typically, each candidate profiles will many associated documents. Hence, applying candidate-centric QE will consider far more tokens of text in the top-ranked candidates, than applying document-centric QE. In particular, Table 2 details the statistics of the documents of the W3C collection, and the document candidate associations we use in this work. Of particular note is the size in tokens of profiles compared to documents - the average profile size is 76 times larger than the average document, while the largest candidate profile is a massive 444 times larger than the largest document in the collection. Due to the large difference between candidate profiles and documents, it is possible that the default settings of $exp\_doc = 3$ and $exp\_term = 10$ may not be suitable for candidate-centric query expansion. In the remainder of the paper, we assess whether the default settings are in fact suitable for document-centric and, in particular, candidate-centric query expansion.

Table 2. Statistics of the W3C collection, and of the candidate-document associations used in this work

W3C Collection	
Number of Documents	331,037
Size of Collection (tokens)	310,720,411
Average size of a Documents (tokens)	9,385
Largest Document (tokens)	50,001
Number of Candidates	1,092
Size of all Candidate Profiles (tokens)	779,840,190
Average size of a Candidate Profile (documents)	913
Average size of a Candidate Profile (tokens)	714,139
Largest Candidate Profile (documents)	88,080
Largest Candidate Profile (tokens)	22,182,816

4   Experimental Results

Table 3 shows the results of document-centric and candidate-centric forms of QE, using both the Bo1 and KL term weighting models. For both Bo1 and KL, the default setting of extracting the top $exp\_term = 10$ most informative terms from the top $exp\_doc = 3$ ranked documents or candidates [1] is applied. Statistically significant improvements from the baselines are shown using the Wilcoxon signed rank test. At first inspection, it appears that query expansion can be successfully applied in an Expert Search task to increase retrieval performance.

Table 3. Results for query expansion using the Bo1 and KL term weighting models. Results are shown for the baseline runs, with document-centric query expansion (DocQE) and candidate-centric query expansion (CandQE). The best results for each measure, term weigthing model and voting technique combination are emphasised. Statistically significant improvements ($p \leq 0.05$) over the corresponding baseline are marked by *, while significant improvements ($p \leq 0.01$) are denoted **.

		TREC 2005		TREC 2006	
		MAP	P@10	MAP	P@10
Baselines	CombSUM	**0.2037**	**0.3240**	0.5188	0.6388
	expCombMNZ	0.2037	0.3100	0.5502	0.6837
Bo1					
DocQE	CombSUM	0.1742	0.2860	**0.5216**	**0.6510**
	expCombMNZ	**0.2185**	**0.3340***	**0.5606**	**0.6959**
CandQE	CombSUM	0.1473	0.2240	0.4203	0.5388
	expCombMNZ	0.1760	0.2500	0.4554	0.5939
KL					
DocQE	CombSUM	0.1805	0.2880	**0.5296**	**0.6490**
	expCombMNZ	**0.2231***	**0.3400****	**0.5689***	**0.7020**
CandQE	CombSUM	0.1627	0.2560	0.5195	0.6265
	expCombMNZ	0.2031	0.3100	0.5600	0.6592

Moreover, the document-centric QE outperforms the candidate-centric QE on both MAP and P@10, in all settings. It is possible that the default setting of exp_doc and exp_term used is not suitable for candidate-centric query expansion, because of the size of the candidate profiles being considered in the pseudo-relevant set. In particular, it can be seen that applying document-centric QE results in an increase over the baseline for the TREC 2006 topics, and when using expCombMNZ for the TREC 2005 topics - some of these improvements are statistically significant ($p <= 0.05$). Compared to the respective baselines, applying candidate-centric QE results in a degradation in performance for most settings using the TREC 2005 topics. Document-centric QE provides an increase in MAP and P@10 over the baselines, except when using CombSUM for the TREC 2005 topics.

Overall, the KL term weighting model performs better in terms of MAP and P@10 when compared to the baselines, than Bo1 achieves. This is interesting as previous thorough experiments on various test collections shows that Bo1 performs consistently better than KL on adhoc search tasks [1]. Note also, that applying document-centric QE to expCombMNZ will always results in an increase in performance if it increased the performance of the CombSUM baseline. This can be explained by the fact that the generated refined document ranking by applying QE is identical. It appears then that expCombMNZ is better than CombSUM at converting the refined document ranking into a ranking of candidates, in line with the same results for unrefined document rankings. Moreover, this follows the persistent high performance of expCombMNZ observed by Macdonald & Ounis in [8]. QE using documents has been well tested in classical IR systems, so it is no surprise that it performs well here in increasing the quality of the document ranking. However, as discussed in Section 3, candidate profiles are many times larger than standard documents, so it is possible that the default setting of $exp\_term = 10, exp\_doc = 3$ is not as suitable for candidate-centric QE. In the next section, we assess the extent to which the setting of the QE parameters can affect the retrieval performance of either forms of QE.

5   Effect of Query Expansion Parameters

In this section, we investigate the extent to which the parameters for QE have an effect on the retrieval performance of the expert search task. The parameters of query expansion are exp_doc, the number of top-ranked documents or candidates to be considered as the pseudo-relevance set, and exp_term, the number of informative terms to be added to the query. To fully investigate their effect, we perform a large-scale evaluation of many parameter combinations. We aim to conclude if one of document-centric or candidate-centric QE is more stable with respect to various parameter settings, and to have a better comparison of the two forms of QE, as well as the term weighting model employed.

For these experiments, we use the expCombMNZ voting technique, using only the TREC 2006 topics, as this is the best performing setting (see Section 4). To assess the stability of the approaches with respect to exp_term and exp_doc, we
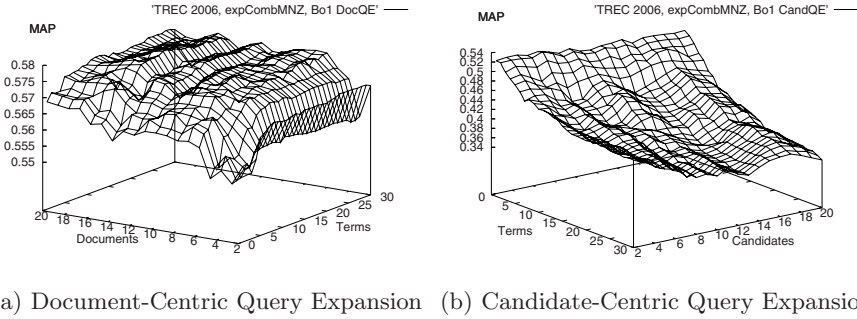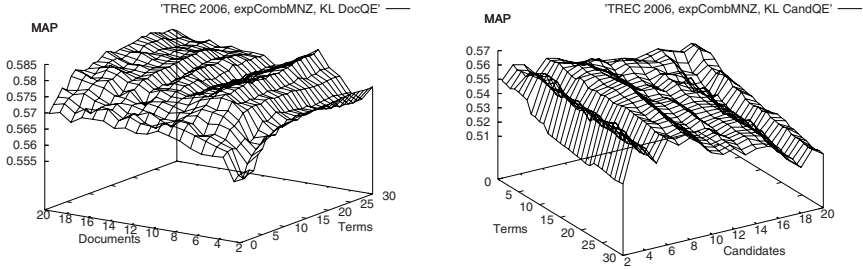
(a) Document-Centric Query Expansion   (b) Candidate-Centric Query Expansion

Fig. 1. Surface plots of MAP for expCombMNZ, using the Bo1 term weighting model, when the exp_doc documents or candidates and exp_term terms query expansion parameters are varied

vary them and record the MAP of the generated run. In particular, we vary $2 \leq exp\_doc \leq 20$ and $1 \leq exp\_term \leq 30$. This generates a matrix of 570 points per setting. Figures 1 & 2 present surface plots of the Bo1 and KL QE settings, using the expCombMNZ voting techniques. In each figure, (a) uses document-centric query expansion, and (b) uses candidate-centric query expansion[3]. From the Figure 1 (a), shows that the number of document used as the pseudo-relevant set in document-centric QE has some effect on the retrieval performance of the generated ranking of candidates. In particular, it appears that using the 3 top-ranked documents is not a good setting, as can be seen from the crevice running across the surface plot on $exp\_doc = 3$; $exp\_doc = 2$ and $exp\_doc \geq 4$ are better settings. With respect to terms considered in the document-centric QE, using less than 10 terms means a drop-off in MAP, while for $exp\_term \geq 10$, the retrieval performance is stable. Indeed, the best performance achieved in Figure 1 (a) is MAP 0.5799, for $exp\_term = 16$ and $exp\_doc = 15$, compared to 0.5606 from Table 3, with default setting ($exp\_term = 10$, $exp\_doc = 3$).

Figure 1 (b) shows that as more terms are considered in candidate-centric QE, the MAP degrades. In particular, expanding the query by only 1 term ($m = 1$), still does not achieve the baseline MAP of 0.5502 from Table 3. In this case, varying the number of candidate profiles considered by the QE mechanism has little affect for a low number of terms. As the number of terms increases to 30, considering less profiles is favoured. The best setting on this figure is $exp\_term = 2$ and $exp\_doc = 6$, which gives a markedly better MAP of 0.5306, compared to the default setting of 0.4554.

For Figure 2, the patterns are similar for the Bo1 term weighting model as in Figure 1 (a). Again, the crevice for $exp\_term = 3$ is apparent. In addition, there is also a slight crevice in MAP at $exp\_term = 11$ for $exp\_doc > 10$. For candidate-centric query expansion (Figure 2 (b)), as the number of terms considered increases, there is again a decrease in MAP, but not as noticeable as in Figure 1 (b). Moreover, MAP is not as stable as exp_term increases.

---

[3] Note that some figures have different orientation to allow easier viewing.

(a) Document-Centric Query Expansion (b) Candidate-Centric Query Expansion

Fig. 2. Surface plots of MAP for expCombMNZ, using the KL term weighting model, when the exp_doc documents or candidates and exp_term terms query expansion parameters are varied

Best settings for Figure 2 are (a) $exp\_term = 24$, $exp\_doc = 20$ (MAP 0.5827), and (b) $exp\_term = 6$, $exp\_doc = 3$ (MAP 0.5627), compared to the default settings of 0.5689 and 0.5600 respectively.

Overall, our large-scale experiments has allowed us to draw some conclusions concerning the applicability and stability of both forms of query expansion. Document-centric QE performs robustly, although exp_doc and exp_term should not be too small - in particular a fairly flat MAP surface is exhibited for $exp\_term \geq 6$ and $exp\_doc \geq 10$. For candidate-centric query expansion, more profound influencing of MAP is apparent as exp_doc and exp_term are varied. From our experiments, $3 \geq exp\_doc \geq 8$ and $exp_term \leq 5$, exhibit the most stable MAP surfaces for this form of query expansion. In particular, the quality of terms decreases rapidly, which is possibly due to the large and varied size of candidate profiles. In summary, overall it appears that document-centric QE is the more stable and effective of the two approaches.

6 Conclusions

We have investigated pseudo-relevance feedback QE in an Enterprise expert search setting. It was shown how query expansion can be applied in two different manners in the context of the voting approach for expert search, namely document-centric and candidate-centric QE. Experiments were carried out using two different voting techniques, and two different query expansion term weighting models. Topics from the TREC 2005 and 2006 Enterprise track Expert Search tasks were used. The results showed that firstly, QE can be successfully applied in expert search and secondly, using the default setting for query expansion, document-centric QE outperforms candidate-centric QE.

By performing a large-scale evaluation of the effect of the QE parameter settings, we observed that document-centric QE is stable with $exp\_term \geq 6$ and $exp\_doc \geq 10$. In contrast, candidate-centric QE was observed to be stable with

respect to the number of candidate profiles considered (*exp_doc*), but increasing the number of expansion terms caused degradations in retrieval performance. Overall, the document-centric QE was more stable and consistently outperformed candidate-centric QE. The major difference when performing candidate-centric QE is that candidate profiles can be extremely large when compared to the documents considered in document-centric QE. We hypothesise that modern QE techniques struggle to identify informative terms when presented with such a large sample. In particular, the more terms identified by candidate-centric QE, the worse the retrieval performance. This also explains the better performance of the KL term weighting model for candidate-centric QE, as KL accounts for the size of the pseudo-relevant set when measuring the informativeness of terms.

Another possible explanation for the less stable performance of candidate-centric QE is due to 'topic drift'. A candidate profile contains many documents that represent the various interests of a candidate. When candidate-centric QE is performed, the expanded query terms may describe other common, not relevant interests of the candidates in the pseudo-relevant set, causing more candidates with these incorrect interests to be retrieved erroneously. Topic drift is less likely to occur with document-centric QE as documents are smaller and more likely to be about a single topic.

In the future, we would like to develop advanced forms of QE suitable for use for candidates. This would combine the best properties of document-centric and candidate-centric QE by only considering the top-ranked documents from the top-ranked candidates profiles as the pseudo-relevant set. An alternative possible approach for extracting informative terms from top-ranked candidate profiles might involve clustering the profile documents in each profile, to identify important interest areas of the candidates.

References

1. G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness.* PhD thesis, Department of Computing Science, University of Glasgow, 2003.
2. G. Amati. Frequentist and bayesian approach to information retrieval. In *Advances in Information Retrieval, 28th European Conference on IR Research, ECIR 2006*, pages 13–24. Springer, April 2006.
3. K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th ACM SIGIR 2006*, pages 43–50, Seattle, WA. August 2006.
4. N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC-2005 Enterprise Track. In *Proceedings of the 14th Text REtrieval Conference (TREC-2005)*, 2005.
5. M. Hertzum and A. M. Pejtersen. The information-seeking practices of engineers: searching for documents as well as for people. *Inf. Process. Manage.*, 36(5):761–778, 2000.
6. D. Hiemstra. *Using language models for information retrieval.* PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.
7. 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 14th Text REtrieval Conference (TREC 2005)*, 2005.

8. C. Macdonald and I. Ounis. Voting for candidates: Adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM CIKM 2006*. Arlington, VA. November 2006.
9. I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the OSIR Workshop 2006*, pages 18–25, August 2006.
10. J. Rocchio. *Relevance feedback in information retrieval*. Prentice-Hall, Englewood Cliff. NJ.