DERI at TREC 2008 Enterprise Search Track

Ronan Cummins and Colm O'Riordan

Digital Enterprise Research Institute, National University of Ireland, Galway
firstname.lastname@deri.org

Abstract. This paper describes the work carried out by DERI for the Enterprise Search track at TREC 2008. We participated in both the expert search task and document search task of the track. For both tasks we made use of novel learned term-weighting schemes. For the expert search task, we used two different approaches (namely a profiling approach and a two-stage document centric approach). We found that the document centric approach outperforms the profiling approach on previous years TREC data. For the document search task we adopted a standard retrieval framework and made use of the learned term-weighting schemes previously developed for the ad hoc retrieval task.

1 Introduction

Traditional Information Retrieval deals with determining the relevance of a document given a user need. However, in large modern organisations, employees have often accumulated the unique expertise in a specific topic area themselves. Automatically identifying experts in a certain area given a specific topic is therefore a useful goal in attempting to satisfy someone's specific information needs on a specific topic. Expert search is the problem of finding and ranking experts in a large corpus of semi-structured or unstructured documents given a user need.

The expert search task of the enterprise track of TREC [2, 8] has been run since 2005 and has provided a corpus, topics and associated relevant experts to enable researchers to develop techniques in advancing the area of expert search. This is the first participation of DERI[1] in TREC. The evaluation metrics used are similar to those used in the standard IR document retrieval task. The document search task of the enterprise track assumes a user request (e.g. an email communication) for information about an organisation or activity in which they may be engaged. The retrieval task is to return a set of key pages (e.g. homepages or project overview pages) for a specific query. There is high critera on relevance for this task.

This paper presents our experiments concerning both tasks in the Enterprise Search track (i.e. both the expert search task and the document search task). We outline two of the main approaches used in expert search systems. We study the performance of various term-weighting schemes applied to both approaches. We also attempt to learn term-weighting features useful for expert search for one of

[1] Digital Enterprise Research Institute, Galway

the approaches. Furthermore, we study the best method of aggregating document scores in the two-stage approach to expert search for different term-weighting schemes. These two main approaches are profiling (identifying candidates and then creating a collection of terms from the corpus for each candidate) and a two-stage approach (initially ranking documents with respect to a topic and then aggregating the document scores for documents associated with candidates in order to rank the candidates). The approach adopted by us for the document search task is based on a standard retrieval framework. However, instead of using a standard term-weighting scheme (like $BM25$) we use learned term-weighting schemes and compare them to more standard schemes. Our approach is purely content based and does not use link analysis features as yet.

The remainder of the paper is as follows: Section 2 outlines the two most common approaches to ranking candidate experts based on their associated documents. Section 3 outlines the experiments and results for the expert search task, while section 4 outlines the experiments and results for the document search task. Our conclusions are presentd in section 5.

2 Expert Search

There have been two main approaches to the problem of expert search adopted by most researchers. This section outlines both of these approaches and some new term-weighting schemes that we use with both of these models for expert search.

2.1 Profiling approach

The profiling approach to expert search consists of firstly identifying candidates in the corpus and then extracting keywords from the corpus which are associated with each candidate. Typically, terms occurring near the appearance of a candidate are extracted and added to the candidate profile. In most approaches, the size of this window is at the document level. Therefore, terms that co-occur in the documents which contain the candidate identifiers are added to the profile. In essence, the profile of a candidate is created by concatenating documents in which the candidate occurs. Once all the profiles have been created, there exist N profiles corresponding to the number of potential experts within the corpus. These 'bag of word' profiles can be matched against a specific topic using a standard term-weighting scheme (e.g. BM25). This approach substitutes profiles for documents in the retrieval model. It is a very simple model but is efficient, as once the collection has been indexed and the profiles created (which can be done during indexing) only the profiles have to be ranked at run-time.

2.2 Two-Stage Approach

The two-stage approach to expert search first ranks the documents in the collection to the topic using a standard term-weighting scheme (e.g. BM25). Then it

aggregates the *score* of the documents which are associated with a candidate to produce a final ranking of candidates. Recent research [9, 10] has modelled this approach as a voting problem and researched various strategies of aggregating the strengths of votes of documents for specific candidates. Many fusion techniques have been experimented with to deal with the aggregation of document scores.

For the two-stage approach, we only have to deal with combining scores from a single ranked list of documents. The following fusion (or aggregation) techniques combine the scores of documents (which are associated to a candidate) when matched against a specific topic:

$$combSUM(Q, C_i) = \sum_{d \in R(Q) \cap D(C_i)} (S(Q, d)) \tag{1}$$

where C_i is candidate i, d is a document, Q is a query (topic), $D(C_i)$ is the set of documents associated with C_i, $R(Q)$ is the ranking of document when given query Q and $S(Q, d)$ is the score of document d given query Q. Thus, $combSUM$ is a summation of the documents scores associated with the candidate C_i. A related ad hoc fusion approach $combNSUM$ simply sums up the top N document scores associated with C_i.

2.3 Term-Weighting

Standard term-weighting approaches can be utilised for both of the aforementioned approaches to expert search. For the profiling approach, each profile can be treated as a document and a term-weighting scheme such as $BM25$ [12] or the pivoted document normalisation scheme [13] can be used to rank the profiles. It is ultimately the term-weighting scheme that is applied to each profile that ultimately determines the performance of the approach.

The performance of the two-stage approach to expert search is determined by the method used to initially rank the documents (i.e. the term-weighting scheme) and the aggregation method use to combine the scores of the top N document associated with the candidate. The default $BM25$ scheme is used in this paper as a benchmark along with the following learned term-weighting schemes:

$$ES(D, Q) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D}{tf_t^D + 0.45 \cdot \sqrt{\frac{dl}{dl_{avg}}}} \cdot \sqrt{\frac{cf_t^3 \cdot N}{df_t^4}} \cdot tf_t^Q \right) \tag{2}$$

where D is a document (or possibly profile depending on the model adopted), Q is a query, tf_t^D is the frequency of a term t in D and tf_t^Q is the frequency of the term in the query Q. dl and dl_{avg} are the length and average length of the documents respectively measured in non-unique terms. N is the number of documents in the collection, df_t is the number of documents in which term t appears and cf_t is the frequency of the term in the entire collection. This function which was learned using genetic programming for the ad-hoc retrieval task and has no tuning parameters. The following scheme is a partially learned

weighting scheme [3] as the normalisation part of the scheme is taken from the $BM25$ scheme:

$$ES7(D,Q) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D \cdot tf_t^Q}{tf_t^D + 0.2 \cdot (0.25 + 0.75 \cdot \frac{dl}{dl_{avg}})} \cdot log\left(\frac{cf_t + \frac{1}{2 \cdot \sqrt{\sqrt{cf_t}}}}{df_t}\right) \cdot \sqrt{\frac{N}{df_t} \cdot \left(\frac{1}{df_t} + 1\right)}\right) \tag{3}$$

3 Experiments in Expert Search

3.1 Preprocessing and candidate identification

For the CSIRO collection (TREC 2008) we removed standard stop-words and stemmed the remaining terms using Porter's stemming algorithm [11]. Candidates were identified using their email addresses. For TREC 2005 and 2006 a list of candidates was explicitly given with the corpus. For TREC 2007, candidates had to be identified by extracting email addresses. The method used by us was to extract email addresses and use them as potential candidates. We used the strings "@csiro.au" and a few common variations (e.g. "at_csiro_dot_au") that people may used to limit spam. This approach led us to identifying 2,910 experts in the collection.

For associating documents to candidates for both approaches, we considered the email address and the first name and surname in the email address. For example, if "joe.bloggs@csiro.au" was the candidates email address, we considered documents which contained either "joe.bloggs@csiro.au" or "joe bloggs" to be associated to that specific candidate. In our preliminary experiments, this approach of associating documents with candidates showed improved performance over using only the email address of the candidates. Indeed, it has been indicated in previous studies that one of the best method of associating the topics of interest for a specific candidate is to use the candidates full name and aliases [10].

3.2 Profiling Approach

For the profiling approach, we use GP to find ranking functions. We follow previous research [4] by dividing the search for useful functions into two stages. We develop term-weighting for ranking these profiles incrementally. We develop global schemes which aim to discover the usefulness of the search term based on measures in the documents, profiles and collection as a whole. When a suitable global scheme has been discovered, measures from the individual profile can be utilised to develop a profile specific measure of usefulness for a term. Table 1 shows the measures (terminals) used in determining a global term-weighting scheme for this approach. We also used the functions outlined in Table 3 as inputs to our GP.

We used a GP population of size 500 run for 40 generations on the TREC 2007 data using both short (query fields) and long queries (query and narrative

fields) for all our experiments. We ran our GP four times and present the results of the top two runs on our training data and used MAP as the fitness function. The training data is sizeable and are solutions are limited in size to a certain length in order to discover general solutions. None of evolved schemes outperform a simple binary weighting for this global term-weighting problem. Even *idf* did not outperform a simple binary weighting on the terms occurring in the profile. Thus, in a global sense the best scheme treats all terms equally when appearing in a profile. From this preliminary experiment, we have identified that using a binary weighting for the global weighting is sufficient when adopting a profiling approach to expert finding. Considering that fact that the number of profiles is small and the fact that each profile contains a large number of terms (because the profiles are made up of multiple documents), it is not surprising that most of the profiles contain at least one occurrence of each of the the query terms making an *idf* type function redundant. Hence, it is the local (or within-profile) part of the scheme that will be more useful for effective retrieval.

Table 1. Global Measures

Measure	Description
df	No. of documents in which a term occurs
cf	Total occurrences of a term in the corpus
pf	No. of profiles in which a term occurs
pcf	Total occurrences of a term in all profiles
V	No. of unique terms in corpus
C	Total no. of terms in corpus
E	No. of experts (profiles)
N	No. of documents in corpus
10	a constant
1	a constant
0.5	a constant

From a profile specific perspective, we can use the set of documents which make up a specific profile to gather features about a specific profile. These features are listed in Table 2. Although all of documents associated with a candidate are concatenated to form a profile, we can extract certain extra information (e.g. the number of documents that make up a profile) during preprocessing with little or no extra cost. Two of the best functions evolved are $EP1$ and $EP2$.

$$EP1(Q,D) = 20 + log(0.5 \cdot \frac{\sqrt{tf}}{tl}) + log(\frac{pdf}{df_e} \cdot \frac{pdf^2}{df_e}) \quad (4)$$

where tf is the frequency of a term in the profile, tl is the length of the profile in words, pdf is the number of document in the profile in which a term occurs (i.e. a term which occurs in all of the documents that makeup a profile would be likely to be more important) and df_e is the number of documents in the profile.

Table 2. Profile Specific Measures

Measure	Description
tf	No. of occurrences of a term in the profile
pdf	No. of documents that make up the profile in which the term occurs
l	No. of unique terms in the profile (vector length)
tl	Total number of terms in the profile (length)
l_{avg}	Average length of all profiles (measured by vector length)
tl_{avg}	Average length of all profiles (measured by total length)
cf_e	Total no. of occurrences of candidate identifier (i.e. frequency of candidate in profile)
df_e	No. of documents that makes up the profile (i.e. document frequency of candidate)
10	a constant
1	a constant
0.5	a constant

Table 3. Functions

Function	Description
$\times \div + -$	standard arithmetic functions
log	natural log
$\sqrt{\ }$	the square-root
sq	square
exp	exponential

$$EP2(Q,D) = (\frac{pdf}{df_e}) \cdot tl_{avg} \cdot log(log(df_e^2)) + (\frac{pdf}{df_e}) \cdot (tl_{avg} - 1) \cdot log(log(df_e)) + 2 \cdot l_{avg} + tl_{avg} \quad (5)$$

where tl_{avg} and l_{avg} are the average length of the profiles and average length of the profile vectors respectively. $BM25$ seems to be quite a robust retrieval model as it performs well using this approach. Normalisation is a very important part of a term-weighting scheme when dealing with large profiles which vary considerably in size for the profile model [1]. For example the average profile vector length is 2,792 terms while the average document vector length is less than 500. The profiles also vary considerably in length as a few long profiles contain many documents (over 50 documents) while many smaller profiles only contain one or two documents.

Table 4. Details of Expert Search Runs

Run	Model Adopted	Topic Fields	Weighting	Stemmed	Stopword Rem.
DERIrun1	Profile	Query and Narrative	EP1(Q,D)	Yes	Yes
DERIrun2	Profile	Query Only	EP2(Q,D)	Yes	Yes
DERIrun3	Document Centric	Query Only	ES7(Q,D)	Yes	Yes
DERIrun4	Document Centric	Query and Narrative	ES(Q,D)	Yes	Yes

Table 4 describes the runs submitted to this years expert search task, while Table 5 presents results for the same approach on last years data. The astericks indicate that the formula evolved was trained on that data.

Table 5. MAP for Profiling approach (TREC 2007 data)

Run	Scheme	Topic Fields	MAP
baseline	$BM25$	Query and Narrative	0.2549
DERIrun1	$EP1$	Query and Narrative	0.3082*
baseline	$BM25$	Query Only	0.2377
DERIrun2	$EP2$	Query Only	0.2979*

We can see that $EP1$ and $EP2$ outperform $BM25$ on the TREC 2007 data. However, this may well be because this is the training data on which $EP1$ and $EP2$ were learned. It will be interesting to see how $EP1$ and $EP2$ perform on this years test data (TREC 2008).

3.3 Two-Stage Document Centric Approach

The performance of this approach is directly dependent on the performance of the document ranking function. The ranking of documents is done a priori and then the scores of the top N documents which are associated to the candidate are aggregated in some way. This final score is then used to rank the candidates. As learned functions have already been developed for the ad hoc document retrieval task [6, 14, 4] , we can use some of these (e.g. ES and $ES7$) as they were learned to optimise MAP. However, for this approach the aggregation of the scores for the top N documents associated to the candidate is an important aspect. It has been suggested in previous research that the best fusion approach is to choose the best associated document score as a measure of the relevance of a specific candidate [10]. This fusion method is called *combMAX*. We evaluate four ranking functions (pivoted document length normalisation, BM25, ES and ES7) using the $combNSUM$ fusion technique.

We used the $combNSUM$ method for aggregating score on all of the previous expert search TREC collections (2005, 2006 and 2007) for a number of different values of N. In Figure 1 we can see that for three of the four term-weighting functions for the *combNSUM*, the performance tends to decrease after the top five documents which are associated with the candidate are aggregated. All the values are averaged results from the three previous years data.

Table 6 shows the results of the runs when used on last years data. $ES7$ performs comparably to $BM25$ on short queries. The performance of the ES scheme for long queries on last years data is surprisingly poor. We are interested in the performance of this scheme on this years data.

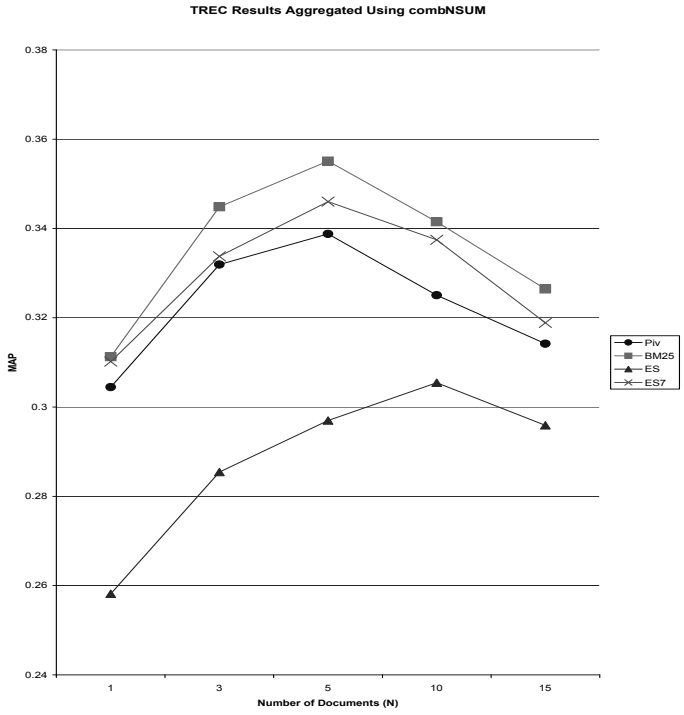

Fig. 1. Performance (MAP) for varying N for combNSUM

4 Experiments in Document Search

For the document search task, we stemmed terms using Porter's algorithm [11] and removed standard stopwords [2]. We submitted 4 runs for the document search task. Details of the runs submitted are outlined in Table 7.

Table 8 shows the results of the document search task on previous TREC data (TREC 2007). It shows that the BM25 scheme outperforms our evolved term-weighting schemes on this data. This is surprising as our results show that on most ad hoc TREC data ES and $ES7$ outperform $BM25$. Furthermore, we expected that ES and $ES7$ would actually perform very well on longer queries (using both query and narrative Fields) as our previous studies have indicated

[2] http://www.lextek.com/manuals/onix/stopwords1.html

Table 6. MAP for document-centric approach (TREC 2007 data) using $comb5SUM$

Run	Scheme	Topic Fields	MAP
baseline	$BM25$	Query Only	0.3140
DERIrun3	$ES7$	Query Only	0.3038
baseline	$BM25$	Query and Narrative	0.3770
DERIrun4	ES	Query and Narrative	0.2314

Table 7. Details of Document Search Runs

Run	Topic Fields	Weighting	Stemmed	Stopword Rem.
DERIrun5	Query Only	ES(Q,D)	Yes	Yes
DERIrun6	Query Only	ES7(Q,D)	Yes	Yes
DERIrun7	Query and Narrative	ES(Q,D)	Yes	Yes
DERIrun8	Query and Narrative	ES7(Q,D)	Yes	Yes

this. This could be due a bias in this collection as most groups tend to submit runs which are created by systems which use $BM25$. It could also be because the task for document search in the enterprise track is a different task to that of ad hoc retrieval. The task of document search in enterprise search is to return key or authoritative pages such as homepages and documents dedicated to the topic, rather than pages that only briefly mention the topic. As metioned in the task description there is a somewhat high critera on relevance. It will be interesting to see the performance of these term-weighting schemes on this years TREC data.

Table 8. Results of Document Search Runs on TREC 2007

Run	Weighting Scheme	Topic Fields	MAP
Baseline	BM25	Query Only	0.4414
Baseline	BM25	Query and Narrative	0.4590
DERIrun5	ES(Q,D)	Query Only	0.3927
DERIrun6	ES7(Q,D)	Query Only	0.4307
DERIrun7	ES(Q,D)	Query and Narrative	0.2473
DERIrun8	ES7(Q,D)	Query and Narrative	0.3537

5 Conclusion

In this paper, we outlined the approaches used by DERI in this years Enterprise Search track. We experimented with a number of different weighting schemes.

For the profiling approach, we search, using evolutionary computation, the available sources of evidence and combinations thereof to identify which features are useful in achieving good performance (measured using MAP). For the second approach, the two-stage expert search approach, we examine the problem of aggregating scores from the ranked list of documents. We find that for the profiling

approach, contrary to our initial expectations, that a simple binary weighting scheme of the terms occurring in the profiles performs well and in fact outperforms more complex weighting approaches such as *idf* and our evolved schemes. With respect to the two stage approach, we compare different fusion techniques for a range of underlying weighting schemes. In our results *comb5SUM* was found to be optimal over several data sets.

For the document search task we used previously evolved term-weighting schemes. We failed to see any improvements over a standard benchmark on last years TREC data. We suggest two possible reasons for this due to the fact that these term-weighting schemes perform very well for the ad hoc task of TREC.

References

1. Krisztian Balog and Maarten de Rijke. Associating people and documents. pages 296–308. 2008.
2. Nick Craswell and Arjen P. De Vries. Overview of the trec-2005 enterprise track. In *In The Fourteenth Text REtrieval Conf. Proc. (TREC*, 2005.
3. Ronan Cummins and Colm O'Riordan. An evaluation of evolved term-weighting schemes in information retrieval. In *CIKM*, pages 305–306, 2005.
4. Ronan Cummins and Colm O'Riordan. Evolving local and global weighting schemes in information retrieval. *Information Retrieval*, 9(3):311–330, 2006.
5. Ronan Cummins and Colm O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artificial Intelligence Review*, 2008.
6. Weiguo Fan, Ming Luo, Li Wang, Wensi Xi, and Edward A. Fox. Tuning before feedback: combining ranking function discovery and blind feedback for robust retrieval. In *the Proceedings of the 27th Annual International ACM SIGIR Conference*, U.K., 2004. ACM.
7. M. Gordon. Probabilistic and genetic algorithms in document retrieval. *Commun. ACM*, 31(10):1208–1218, 1988.
8. Arjen P. De Vries Ian Soboroff and Nick Craswell. Overview of the trec-2006 enterprise track. In *In The Fifthteenth Text REtrieval Conf. Proc. (TREC*, 2006.
9. Craig Macdonald and Iadh Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 387–396, New York, NY, USA, 2006. ACM.
10. Craig Macdonald and Iadh Ounis. Searching for expertise: Experiments with the voting model. *The Computer Journal*, 2008.
11. M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
12. Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. Okapi at TREC-3. In *In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST*, 1995.
13. Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM Press, 1996.
14. Andrew Trotman. Learning to rank. *Inf. Retr.*, 8(3):359–381, 2005.