The Computer Journal Advance Access published March 6, 2008

doi:10.1093/comjnl/bxm112

Searching for Expertise: Experiments with the Voting Model

CRAIG MACDONALD* AND IADH OUNIS

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK
**Corresponding author: craigm@dcs.gla.ac.uk*

In an expert search task, the user's need is to identify people who have relevant expertise to a topic of interest. An expert search system predicts and ranks the expertise of a set of candidate persons with respect to the user's query. In this work, we propose a novel approach for estimating and ranking candidate expertise with respect to a query. We see the problem of ranking experts as a voting problem, which we model using adaptations of data fusion techniques. We extensively investigate the effectiveness of the voting approach and the associated data fusion techniques across a range of document weighting models, in the context of the TREC 2005 and TREC 2006 Enterprise track settings. The evaluation results show that the voting paradigm is very effective, without using any collection-specific heuristics. Additionally, we further analyse two main features of the voting model, namely the manner in which document votes are combined and the effect of the underlying document ranking. First, for the combination of document votes, we hypothesise that candidate with large profiles can introduce bias in the generated ranking of candidates. We propose and integrate into the model a candidate length normalisation technique that removes bias towards prolific candidate experts. Secondly, we investigate the relative effects of applying various retrieval enhancing techniques to improve the quality of the underlying document ranking, to investigate how each technique improves the retrieval effectiveness of the generated ranking of candidates. At each stage, we experiment extensively and draw conclusions. Our results show that the voting techniques proposed are indeed effective, across several different document weighting models and settings. Secondly, we see that candidate profile length normalisation can help improve retrieval accuracy when applied to the candidate profile sets. Lastly, we show that increasing the quality of the underlying ranking of candidates can enhance the retrieval accuracy of the generated ranking of candidates.

Keywords: Expert search; expert finding; voting model; information retrieval; enterprise information retrieval

Received 1 February 2007; revised 7 September 2007

1. INTRODUCTION

With the advent of the vast pools of information and documents in large enterprise organisations, users regularly have the need to find not only documents, but also people with whom they share common interests, or who have specific knowledge in a required area.

Hertzum and Pejtersen [1] found that engineers in product development organisations often intertwine looking for informative documents with informed people. People are a critical source of information because they can explain and provide arguments about why specific decisions were made.

Yimam-Seid and Kobsa [2] identified five scenarios when people may seek an expert as a source of information to complement other sources:

(i) *Access to non-documented information*—For example, in an organisation where not all relevant information is documented.
(ii) *Specification need*—the user is unable to formulate a plan to solve a problem, and resorts to seeking experts to assist them in formulating the plan.
(iii) *Leveraging on another's expertise* (*group efficiency*)—for example, finding a piece of information that a relevant expert would know/find with less effort than the seeker.
(iv) *Interpretation need*—for example, deriving the implications of, or understanding, a piece of information.
(v) *Socialisation need*—the user may prefer that the human dimension be involved, as opposed to interacting with documents and computers.

An *expert search* system is an information retrieval (IR) system that can aid users with their 'expertise need' in the above scenarios. In contrast with classical IR systems where users search for documents, an expert search system supports users in identifying informed people: users formulate a query to represent their topic of interest to the system; the system then ranks *candidate* persons with respect to their estimated expertise about the query, using available documentary evidence.

The retrieval performance of an expert search system—the accuracy of the suggested candidates—is an important issue. An expert search system should aim to rank candidate experts while maximising the traditional evaluation measures in IR: *precision*, the accuracy of suggested candidates expertise; and *recall*, the number of candidates with relevant expertise retrieved.

The running of the expert search task in the recent TREC 2005 and TREC 2006 Enterprise tracks [3, 4] has increased interest in the expert search area. The TREC forum provides researchers with shared interests in IR research areas to evaluate their retrieval systems on a shared test collection. For the expert search task, this test collection consisted of a common corpus, a list of experts identified in the corpus and a list of expertise queries, known as topics. The retrieval performances of participating systems are evaluated by deriving a ground truth, known as relevance assessments, and then evaluating each system using precision and recall-based measures, such as Precision @ 10 (P@10), and mean average precision (MAP). For the Enterprise track of 2005 and 2006 the test collection consisted of 331 037 documents collected from the World Wide Web Consortium (W3C) website in 2005 [3]. For research purposes, the W3C is a useful example of an enterprise organisation, as it operates almost entirely over the Internet. Moreover, its documents are freely available online. This allows research on an enterprise-level corpus, without the intellectual property issues normally associated with obtaining such a corpus. The corpus is also wide ranging, containing the main W3C Web presence, personal homepages, official standards and recommendation documents, email discussion list archives, a wiki and a source code repository.

The corpus is also wide-ranging, containing the main W3C Web presence, personal homepages, standards documents, email discussion list archives, a wiki, and a source code repository.

Expert search systems typically use a *profile* of evidence for each candidate that indicates their expertise. These profiles can be generated manually by the candidate, or automatically by the system using documentary evidence. In this work, we will only consider automatically generated profiles of evidence, which are represented as sets of documents associated with each candidate expert.

In [5], we introduced the Voting Model for expert search. This work extends the research into the voting model with further hypotheses and experiments, to better understand both the model and the expert search task.

The voting model considers a *ranking of documents* with respect to the expert search query. This document ranking contains implicit evidence of expertise. For example, a user with an expertise need might examine a ranking of related documents, looking for an author who has written a highly relevant document, or looking for an author who has written prolifically about the general topic area. For these reasons, the ranking of documents is fundamental to the voting model for expert search. We see each document retrieved as an implicit vote for the candidate whose profile contains that document. We propose several ways to aggregate document votes into a ranking of candidates, using our intuitions about the manual search process a user might exhibit. These intuitions are manifested by appropriate adaptations of data fusion techniques, which we call voting techniques.

Normal IR systems use document weighting models, that rank documents with respect to their estimated relevance to the query, thus creating a ranking of documents. The voting techniques proposed above are dependent on the document weighting model used to generate the underlying ranking of documents. Therefore, we evaluate the voting techniques using a selection of state-of-the-art probabilistic document weighting models. Evaluation is performed using the TREC 2005 and 2006 Enterprise expert search tasks described above. The obtained evaluation results show that applying the voting model to expert search is very effective compared with the other systems participating in the TREC tracks. Moreover, the voting model is general, making no use of collection-specific heuristics, and has no dependence on the document weighting models used to generate the underlying ranking of documents.

This paper poses two research hypotheses in the context of the voting model for expert search. First, we investigate the effect of candidate profile length on the retrieval performance of the voting model: because each candidate can have a varying amount of expertise evidence in their profile, we suggest that candidates with longer profiles can be unfairly favoured in the voting model, because these longer profiles are more likely to gain a vote at random from the document ranking. We hypothesise that taking into account the variable length of candidate profiles is required, and call this candidate length normalisation. Secondly, since the voting model depends on the input ranking of documents with respect to the query, the ranking of documents can have a profound effect on the accuracy of the generated ranking of candidates. We hypothesise that increasing the quality of this ranking of documents will increase the accuracy of the generated ranking of candidates. In this article, we provide detailed experiments and analysis of these hypotheses in terms of the voting model.

1.1. Related work

There is some previous work on expert search models where candidate profiles consist of a set of documents. Craswell

et al. [6] proposed concatenating the terms of all documents in each profile into virtual documents, and ranking these using a traditional IR system. However, this approach lacks granularity, as the contribution of each document in a candidate's profile is not measured individually, making this approach less effective than other approaches.

Liu *et al*. [7] addressed the expert search problem in the context of a community-based question–answering service. They applied three different language models, and experimented with varying the size of the candidate profiles. They concluded that retrieval performance can be enhanced by including more evidence of expertise in the profiles (in this case questions or answers written by the candidate).

Social network analysis also features in some related work to expert search. Graph-based techniques are used to infer connections between candidates, and are particularly useful on corpora of email communications [8–10]. For instance, Dom *et al*. [8] proposed that a series of email communications could be seen as a directed graph and that the act of sending an email to another person (represented as an edge from a node to another) implied the expertise on the sender. Moreover, two approaches make use of the HITS algorithm [11] to calculate 'repute' and 'resourcefulness' scores for each candidate [12,13]. In [14], McLean *et al*. use a graph structure to propagate expertise evidence between members of a project team. Finally, various expert search approaches were proposed by participants in the TREC 2005 Enterprise track [3], and techniques such as document structure and clustering were applied. This includes that of Balog *et al*. [15], who proposed the use of language models in expert search. They proposed several models for expert search; however the approach is limited to the use of language modelling to provide the estimates for the relevance of a document to the query. In Section 3, we show that the language modelling approach is a special case of the voting model. Similarly to Balog *et al*., Fang and Zhai [16] applied relevance language models to the expert search task. In contrast, the probabilistic approach proposed by Cao *et al*. [17] and the hierarchical language models proposed by Petkova and Croft [18] do not consider expertise evidence on a document level, but instead work on a more fine-grained approach using windowing. In all these approaches, the relevance computation of documents to the queries can only be computed using the language modelling approach. In contrast, our proposed method does not rely on a particular approach to rank document to the query. Moreover, our approach is based on three sources of evidence based on intuitions about the expert search task, and not on the marginalisation of conditional probabilistic distribution.

1.2. Contributions

The main contributions of this article are as follows: We propose a novel, general model for modelling expert search that takes as input a ranking of documents and generates a ranking of candidates. The model defines many different techniques for combining the votes of documents, based on intuitions concerning the best way to derive experts from a ranking of documents. Our extensive evaluation of these various voting techniques has shown that the model can generate accurate rankings of candidates.

The proposed model is not dependent on any features of the collection or on the queries used by the system. Any standard retrieval technique, such as Divergence from Randomness (DFR) [19], probabilistic or language modelling [20] can be used to generate the underlying document ranking. It can be easily deployed in an enterprise setting without the need of any specific facilities provided by the enterprise search engine, other than a ranking of documents with respect to the query.

Moreover, by applying and experimenting with the various voting techniques for expert search, we are able to better understand the expert search task. In particular, we are able to determine the important sources of evidence in expert search: First, the prolific aspect of a candidate expert—whether they have written many documents about the topic of interest (we call this number of votes). Secondly, how on-topic the documents in their profile are? If they have document(s), which are exactly about the topic, then it is likely that the candidate has relevant expertise than a candidate who has only written about related topics (we call this strength of votes).

Next, we show that the performance of an expert search system is linked to quality of the expertise evidence—that is, how the profiles of documents are associated with the candidate experts. We experiment using four different candidate profile sets, determined using different techniques, and find that the most exact method of matching candidates to document performs the most robustly. Next, we investigate the need for candidate length normalisation, using extensive experimentation. We find that the usefulness of normalisation depends on the task and the voting technique being applied. Finally, we investigate the effect of increasing the quality of the document ranking, using a selection of IR techniques to increase precision and recall. The experimental results show that applying such techniques to increase the quality of the underlying document ranking can enhance the retrieval accuracy of the generated ranking of candidates.

1.3. Outline

This article is organised as follows:

- Section 2 introduces how documentary evidence is used to represent the expertise of candidates, known as candidate profiles. We describe various methods for building profiles of evidence for candidates, and

define the four different expertise profile sets we use in this work.
- Section 3 details the motivations for the voting model for expert search. We define several voting techniques, which we use to aggregate the votes for candidates from the document ranking, and evaluate these across the expertise profiles defined in Section 2, and three statistically different document weighting models.
- Section 4 investigates the effect of profile length in the voting model. In particular, it describes our hypothesis that candidates who have long profiles (for instance, prolific authors) can have an unfair bias in the candidate ranking, and that a normalisation component is required to be integrated into the model. We describe two ways of normalising the estimation of candidate expertise relative to profile size, and conclude with experiments across all four expert profile sets.
- Section 5 describes the second hypothesis in this article, which poses that the accuracy of the generated ranking of candidates can be improved by increasing the quality of the underlying document ranking. We then investigate the extent to which improvements in the document ranking affect the generated candidate ranking, by applying a variety of renown and state-of-the-art retrieval techniques to the document ranking.
- Section 6 provides concluding remarks and directions for future research.

2. EVIDENCE OF EXPERTISE FOR EXPERT SEARCH SYSTEMS

Expert search systems make use of textual evidence of expertise to rank candidates. Predominantly, these systems work by generating a *profile* of textual evidence for each candidate. The profiles represent the system's knowledge of the expertise of each candidate, and they are ranked in response to a user query [6, 7, 10, 21].

There are two requirements for an expert search system: a list of candidate persons that can be retrieved by the system and some textual evidence of the expertise of each candidate to include in their profile. In most enterprise settings, a staff list is available and this list defines the candidate profiles that can be retrieved by the system. Candidate profiles can be created either explicitly or implicitly: candidates may explicitly update their profile with an abstract or list of their skills and expertise [21]; or alternatively, the expert search system can implicitly and automatically generate each profile from a corpus of documents. There are several strategies for automatically associating documents to candidates, to generate a profile of their expertise:

- Documents containing the candidate's name: exact or partial match [6].
- Emails sent or received by the candidate [8, 12, 22].
- The candidate's homepage on the Web or intranet and their CV [9].
- Documents written by the candidate [9].
- Team, group or department-level evidence [14].
- Web pages visited by the candidate [13].

In this work, we focus on implicit evidence of expertise and, in particular, we assess the performance and stability of our model across a selection of different methods for generating the candidate profiles. In the TREC W3C collection, which we use for evaluation in this work, the authorship of all documents is not readily available, so we identify expertise evidence using documents that contain variations of the candidate names. We apply four techniques for generating candidate profiles, based on occurrences of the candidates' names in the documents of the W3C collection, namely:

- Last Name: documents containing the last name of the candidates.
- Full Name: documents containing the exact full name of the candidates.
- Full Name + Aliases: documents containing the full name of the candidates and variations of their names.
- Email Address: documents matching exactly the email addresses of the candidates.

These techniques cover a spectrum of accuracy of the profiles: profiles should contain as much evidence as possible for a given candidate (i.e. minimising false-negatives), without incorrectly associating too much evidence with a candidate (i.e. minimising false-positives).

Table 1 details the statistics of the four different profile sets. The Full Name and Email Address sets are the most exact, in that they should only match documents that contain the name or email address of the candidate, respectively. In contrast, the Last Name profile set can mismatch evidence for candidates. For instance, consider the scenario that two candidates have identical surnames—both candidates will be associated with

TABLE 1. Statistics of the candidate profile sets employed in this work.

	Last Name	Full Name	Full Name + Aliases	Email Address
Candidates with evidence (of 1092)	923	720	810	470
Mean candidate profile size	881.82	286.23	381.78	191.59
Largest candidate profile size	50 767	18 674	44 330	25 571
Percentage of collection documents in profile set	66.8	41.4	52.2	20.7

documents that should have only been associated to either candidate. This is reflected in the statistics of the table, as the Last Name profile set has the largest average profile size.

3. VOTING MODEL FOR EXPERT SEARCH

In this work, we consider a different and novel approach to ranking expertise. In particular, we consider that expert search is a voting process. Using the ranked list of retrieved documents for the expert search query, we propose that the ranking of candidates can be modelled as a voting process using the retrieved document ranking and the set of documents in each candidate profile. This is manifested from two intuitions: first, a candidate who has written prolifically about a topic of interest (i.e. many on-topic documents in their profile) is likely to have relevant expertise; and secondly, the more about a topic the documents in their profile are, the stronger is the likelihood of relevant expertise. The problem is how to aggregate the votes for each candidate so as to produce an accurate final ranking of experts.

We design various voting techniques, which aggregate these votes from the single ranking of documents into a single ranking of candidates, using evidence based on intuitions described above. In particular, we are inspired by the aggregation of document rankings in the meta search field. Data fusion techniques—also known as meta search techniques—are used to combine separate rankings of documents into a single ranking, with the aim of improving over the performance of the constituent rankings. Each time a document is retrieved by a ranking, an implicit vote has been made for that document to be included higher in the combined ranking. Fox and Shaw [23] defined several data fusion techniques (CombSUM, CombMNZ, etc.), and these have been the object of much research since (e.g. see [24–26]). Two main classes of data fusion techniques exist: those that combine rankings using the ranks of the retrieved documents and those that combine rankings using the scores of the retrieved documents.

As proposed in [5] and introduced in Section 1, we see expert search as a voting problem. In particular, the profile of each candidate is a set of documents associated to them to represent their expertise. We then consider a *ranking of documents* by an IR system with respect to the query. Each document retrieved by the IR system that is associated with the profile of a candidate can be seen as an implicit vote for that candidate to have relevant expertise to the query. The ranking of the candidate profiles can then be determined from the votes. In this work, we adapt well-known data fusion techniques in IR, such as CombSUM, to aggregate the votes for candidates by the retrieved documents.

Let $R(Q)$ be the set of documents retrieved for query Q, and the set of documents belonging to the profile of candidate C be denoted profile(C). In expert search, we need to find a ranking of candidates, given as $R(Q)$. Consider the simple example in Fig. 1. The ranking of documents with respect to the query has retrieved documents $\{D_b >_{\text{rank}} D_c >_{\text{rank}} D_a >_{\text{rank}} D_d\}$. Using the candidate profiles, candidate C_1 has then accumulated two votes, C_2 has one vote, C_3 has three votes and C_4 has no votes. Hence, if all votes are counted as equal, and each document in a candidate's profile is equally weighted, a possible ranking of candidates to this query could be $\{C_3 >_{\text{rank}} C_1 >_{\text{rank}} C_2 >_{\text{rank}} C_4\}$ if all votes were considered equally. While counting the number of votes as evidence of expertise may be sufficient to produce a ranking of candidates, doing so would not take into account the additional fine-grained evidence that is readily available, for instance, the scores and ranks of the documents in $R(Q)$. By using appropriate vote aggregation techniques to combine the score and/or ranks of the documents, we can have different rankings of candidates.

We determine the score of the candidate with respect to the query, score_cand(C, Q), as the aggregation of votes of all retrieved documents d, which also belong to the profile of the candidate [i.e. $d \in R(Q) \cap \text{profile}(C)$]. From our two intuitions on expert search, we consider three forms of evidence when aggregating the votes to each candidate:

(i) the number of retrieved documents voting for each candidate;
(ii) the scores of the retrieved documents voting for each candidate;
(iii) the ranks of the retrieved documents voting for each candidate.

We design various voting techniques, which aggregate these votes from the single ranking of documents into a single

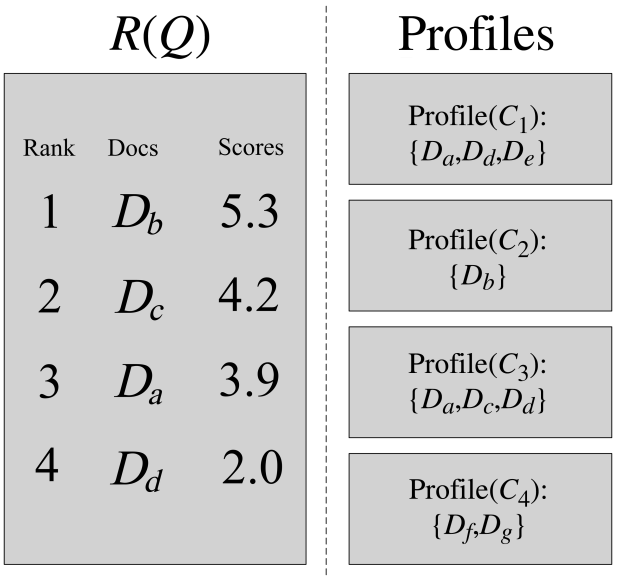

FIGURE 1. A simple example from expert search. The ranking $R(Q)$ of documents (each with a rank and a score) must be transformed into a ranking of candidates using the documentary evidence in the profile of each candidate (profile(C)).

ranking of candidates, using evidence based on intuitions described above. In particular, we are inspired by the aggregation of document rankings in the meta search field. We adapt data fusion techniques to aggregate the votes from the single ranking of documents $R(Q)$ into a ranking of candidates, using the appropriate forms of evidence.

Our application of data fusion techniques differs from conventional applications as follows. Typically, when applying data fusion techniques, several rankings of documents are combined into a single ranking of documents. In contrast, our novel approach aggregates votes from a single ranking of documents into a single ranking of candidates, using the votes between $R(Q)$ and the candidate profiles.

We now show how some established data fusion techniques can be adapted for expert search. In [5], we defined and experimented with 11 voting techniques suitable for ranking candidates. These include a selection of score- and rank-based fusion techniques. In this work, we only consider a selection of score-based voting techniques, as these exhibit a higher degree of accuracy than rank-based voting techniques, due to the fact that the score information is more fine-grained than the rank information.

First, we adapt the CombSUM [23] (a score aggregation technique) for expert search. In this data fusion technique, the score of a document is the sum of the normalised scores received by the document in each individual ranking. Adapting the CombSUM technique to our approach, we score a candidate's expertise with respect to a query, as the sum of the relevance score of all the documents in $R(Q)$ that are voting for that candidate. Formally, score_cand (C, Q) is calculated as

$$\text{score\_cand}_{\text{CombSUM}}(C, Q) = \sum_{d \in R(Q) \cap \text{profile}(C)} \text{score}(d, Q), \quad (1)$$

where score(d, Q) is the score of the document d in the document ranking $R(Q)$ as given by a search engine for query Q. Similarly, CombMNZ [23] can be adapted for expert search. This has a similar intuition to CombSUM [Equation (1)], except that candidates with a larger number of votes are scored higher. In CombMNZ, candidates are scored with respect to a query as

$$\text{score\_cand}_{\text{CombMNZ}}(C, Q) = \|R(Q) \cap \text{profile}(C)\| \sum_{d \in R(Q) \cap \text{profile}(C)} \text{score}(d, Q), \quad (2)$$

where $\|R(Q) \cap \text{profile}(C)\|$ is the number of documents from the profile of candidate C that are in the ranking $R(Q)$ that is, the number of votes for this candidate. CombMNZ explicitly models both the (i) number of votes and (ii) strength of votes sources of expertise evidence.

Finally, the CombMAX technique is based on the intuition that the extent to which a candidate is prolific is not important, but instead examines the extent to which they have written a document about the topic. In particular, in CombMAX, each candidate only gets at most one vote—this being from the most highly scored document in their profile. This document represents the most on-topic part of their profile: candidates with only a single 'vaguely' on-topic document should be ranked lower than candidates with a definitely on-topic document. In the CombMAX technique [23], the score of a candidate with respect to a query is

$$\text{score\_cand}_{\text{CombMAX}}(C, Q) = \text{Max}(\text{score}(d, Q) : d \in R(Q) \cap \text{profile}(C)), \quad (3)$$

where Max$(\cdot)$ is the maximum of the described set. CombMAX concentrates on the most strong vote, i.e. evidence (ii).

Normally, in the CombSUM, CombMNZ and CombMAX data fusion techniques, it is necessary to normalise the scores of documents across all the rankings [26]. However, in Equations (1)–(3), no score normalisation is necessary. Indeed, in our case, as stressed above, only one ranking of documents is involved, and hence the scores are all comparable.

In the following sections, we test each of these voting techniques with extensive experimentation.

3.1. Experimental setup

In the following, we aim to demonstrate that voting is an effective approach for expert search and that the data fusion techniques adapted are suitable to implement the proposed approach. We use three statistically different document weighting models to assess the extent to which the performance of the voting techniques is affected by the choice of weighting model. In particular, we apply language modelling and two Divergence from Randomness document weighting models to represent the state-of-the-art. We could test other state-of-the-art document weighting models (e.g. BM25). However, the conclusions would likely be similar—for example BM25 has been shown to perform similarly to the PL2 Divergence from Randomness weighting models in both Expert Search and Web settings [5, 27].

To evaluate our approach, we use the expert search tasks of the TREC 2005 and TREC 2006 Enterprise tracks. The W3C corpus used for these tasks includes a list of 1 092 candidate experts. We use a set of 50 topics from the TREC 2005 expert search task and the title field a further set of 49 topics from the TREC 2006 task. Retrieval performance is evaluated using Mean Average Precision (MAP)—to assess the overall accuracy of the generated ranking of candidates—and precision at 10 (P@10)—to assess the accuracy of the top-ranked candidates retrieved by the system.

We index the W3C collection using Terrier [28, 29]. During indexing, each document is represented by its textual content

and the anchor text of its incoming hyperlinks. Stopwords are removed, and as we would like to favour high precision, we use a weak stemming algorithm, which only applies the first two steps of Porter's stemming algorithm.

We test our proposed voting model using the three voting techniques described above, and using three statistically different document weighting models to generate the document ranking $R(Q)$. First, it is of note that for the voting model, if Hiemstra's language modelling approach [20] 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 [15]. For this reason, so as to include a state-of-the art baseline in our experiments, the first document weighting model we apply is based on language modelling. In language modelling, the probability $P(d|Q)$ of a document d being generated by a query Q is estimated as follows [20]:

$$\text{score}(d, Q) \propto P(d|Q) = \frac{P(d) \cdot P(Q|d)}{P(Q)}, \tag{4}$$

where $P(d)$ is the document prior probability, for which we use a uniform prior. $P(Q)$ can be ignored since it does not depend on the documents and, therefore, does not affect the ranking of documents. score(d, Q) is given by [20]

$$\text{score}(d, Q) = \sum_{t \in Q} \log\left(1 + \frac{\lambda_{\text{LM}} \cdot \text{tf} \cdot \text{Tok}}{(1 - \lambda_{\text{LM}}) \cdot F \cdot l}\right), \tag{5}$$

where λ_{LM} is the Jelinek–Mercer hyper-parameter between 0 and 1, tf is the term frequency of query term t in a document d, l is the length of document d, that is the number of tokens in the document, F is the term frequency of query term t in the collection and Tok is the total number of tokens in the collection. Following [20], we set $\lambda_{\text{LM}} = 0.15$. We denote this document weighting model as LM.

The remaining two weighting models tested are from the Divergence from randomness (DFR) framework [19]. The first of these, PL2, is robust and performs particularly well for tasks requiring high early-precision [30]. For the PL2 model, the relevance score of a document d for a query Q is given by

$$\text{score}(d, Q) = \sum_{t \in Q} \text{qtw} \cdot \frac{1}{\text{tfn} + 1}\Big(\text{tfn} \log_2 \frac{\text{tfn}}{\lambda} + (\lambda - \text{tfn}) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot \text{tfn})\Big), \tag{6}$$

where λ is the mean and variance of a Poisson distribution. It is given by $\lambda = F/N$. F is the frequency of the query term in the collection and N is the number of documents in the whole collection. The query term weight qtw is given by $\text{qtf}/\text{qtf}_{\text{max}}$. qtf is the query term frequency. qtf_{max} is the maximum query term frequency among the query terms.

The normalised term frequency tfn is given by the so-called Normalisation 2 from the DFR framework:

$$\text{tfn} = \text{tf} \cdot \log_2\left(1 + c \cdot \frac{\text{avg\_l}}{l}\right) \quad (c > 0), \tag{7}$$

where tf is the term frequency of the term t in document d and l is the length of the document. avg_l is the average document length in the whole collection. c is the hyper-parameter that controls the normalisation applied to the term frequency with respect to the document length. The default value is $c = 1.0$ (see [19]).

The DLH13 document weighting model is a generalisation of the parameter-free hypergeometric DFR model in a binomial case [31, 32]. 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 (variables are as above)

$$\text{score}(d, Q) = \sum_{t \in Q} \frac{\text{qtw}}{\text{tf} + 0.5} \cdot \left(\log_2\left(\frac{\text{tf} \cdot \text{avg\_l}}{l} \cdot \frac{N}{F}\right) + 0.5 \log_2\left(2\pi\text{tf}\left(1 - \frac{\text{tf}}{l}\right)\right)\right). \tag{8}$$

In the following section, we evaluate the voting techniques across the three document weighting models detailed above, for the expert search tasks of the TREC 2005 and TREC 2006 Enterprise tracks. Note that the language modelling and PL2 document weighting models have parameter to tune (λ_{LM} and c, respectively). However, DLH13 has no parameters that require tuning. Indeed, all variables are automatically computed from the collection and query statistics. For λ_{LM} and c, we use the default settings, but note that retrieval effectiveness could be improved if these were empirically tuned—for instance, using suitable training data.

3.2. Evaluation

In order to assess the effectiveness of the voting model as an approach for expert search, we experiment with the CombSUM, CombMAX and CombMNZ voting techniques, across the three document weighting models defined above, and the four candidate profile sets described in Section 2. Tables 2 and 3 detail the retrieval performance for the TREC 2005 and TREC 2006 Expert search tasks, respectively. On analysing these results, we note that performance on TREC 2006 is notably higher than for TREC 2005. This corresponds to the difficulty of the task, as experienced by all participants in the corresponding TREC tracks. Indeed, the median MAP

TABLE 2. Evaluation using the expert search task of TREC 2005 Enterprise track, of three voting techniques, using three different document weighting models and four candidate profile sets.

	Last Name		Full Name		Name and Aliases		Email Address	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
CombSUM								
LM	0.0898≪	0.1540	**0.1589**>	**0.2680**	0.1367	0.2320	0.1135 <	0.2000
PL2	0.0835≪	0.1520	**0.1688**>	**0.2660**	0.1336	0.2100	0.1172	0.2200
DLH13	0.0877≪	0.1580	**0.1670**>	**0.2700**	0.1342	0.2340	0.1212	0.2060
CombMAX								
LM	0.1288	*0.1800*	**0.1980**≫	**0.2620**	0.1836>	0.2400	0.1110 <	0.1980
PL2	*0.1370*	0.1660	***0.2247***≫	***0.3020***	0.1950>	*0.2640*	0.1315	*0.2560*
DLH13	0.1326	0.1680	**0.2176**≫	**0.2920**	*0.1971*≫	0.2520	0.1288	0.2360
CombMNZ								
LM	0.0872≪	0.1288	**0.1565**	**0.2620**	0.1346	0.2400	0.1116 <	0.1980
PL2	0.0812≪	0.1460	**0.1652**>	**0.2540**	0.1311	0.2120	0.1169	0.2160
DLH13	0.0861≪	0.1580	**0.1633**	**0.2640**	0.1323	0.2280	0.1187	0.2080

Retrieval performance is measured using MAP and P@10. Statistical significance, using the Wilcoxon signed rank test, is computed with respect to the median run of the participating groups of the TREC 2005 Enterprise track are shown (MAP 0.1402): statistically significant improvements at $P \leq 0.05$ are denoted >; significant improvements at $P \leq 0.01$ are denoted ≫. Similarly, statistically significant degradations in MAP are denoted < and ≪, respectively. The best performance in each row is in bold, and the best in each column is Italicised.

TABLE 3. Evaluation using the expert search task of TREC 2006 Enterprise track, of three voting techniques, using three different document weighting models and four candidate profile sets.

	Last Name		Full Name		Name and Aliases		Email Address	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
CombSUM								
LM	0.2630≪	0.3286	**0.5084**≫	**0.6184**	0.3924>	0.5224	0.3471	0.5184
PL2	0.2384≪	0.3204	**0.5066**≫	**0.5918**	0.3727	0.4939	0.3544	0.5102
DLH13	0.2505≪	*0.3367*	**0.5188**≫	***0.6388***	0.3933>	0.5184	*0.3709*	0.5367
CombMAX								
LM	*0.2792*≪	0.3347	**0.4936**≫	**0.5959**	0.4233>	0.5306	0.3416	0.5204
PL2	0.2299≪	0.2816	**0.5031**≫	**0.5653**	0.3902	0.4490	0.3617	0.5429
DLH13	0.2727≪	*0.3367*	***0.5190***≫	**0.6245**	*0.4446*>	*0.5408*	0.3696	*0.5633*
CombMNZ								
LM	0.2573≪	0.3163	**0.4998**≫	**0.6122**	0.3874	0.5143	0.3437	0.5102
PL2	0.2320≪	0.3143	**0.5023**≫	**0.5898**	0.3686	0.4816	0.3505	0.5041
DLH13	0.2509≪	0.3327	**0.5133**≫	**0.6327**	0.3886>	0.5102	0.3683	0.5306

Notation as in Table 2. Median run had MAP 0.3412.

for the TREC 2005 task was 0.1402 [3], while for TREC 2006 the median MAP was 0.3412 [4].[1] Additionally, using the Wilcoxon signed rank test, both tables denote statistically significant differences in MAP compared to these median runs.

Examining Tables 2 and 3 in detail, we can make the following observations. First with respect to the document ranking $R(Q)$, across all voting techniques and candidate profile sets, the choice of document weighting model used to generate the document ranking overall seems to have little effect on the quality of the results, as the results in terms of MAP and P@10 are fairly consistent. This implies that all of the modern weighting models applied are able to rank documents accurately to the extent that the voting techniques can infer expertise from the ranking. Secondly, the voting

[1]The median runs for P@10 were not provided.

techniques perform well as all three can outperform the median MAP of both TREC 2005 and TREC 2006. Moreover, the choice of voting technique applied can have a small impact on the quality of results. In particular, the CombSUM and CombMNZ techniques appear to have equivalent performance. Noticeable is the good performance of the CombMAX voting technique, particularly for P@10, even though it does not take into account the number of votes for a candidate profile. This shows that the most highly ranked document for each candidate is a very good indicator of candidates' expertise.

Finally, the choice of candidate profile set can have a major impact on the retrieval effectiveness of the generated candidate rankings. In particular, we note that the Full Name set provides the best performance, in terms of MAP and P@10 for both TREC 2005 and TREC 2006. In most cases, the improvements in MAP from the median run of that year's TREC participants are statistically significant. As expected, the Last Name set performs on the whole much worse than the other sets—we hypothesise that the Last Name set is too noisy, and contains too much misassociated evidence (false-positives). Moreover, the retrieval performance of CombMAX is less affected by this candidate profile set than the other voting techniques, especially for the TREC 2005 topics. This can be explained, as CombMAX only takes into account one item from each candidate's profile, and hence this noise affects it to a lesser extent than the other voting techniques. For the Email Address set, which is the smallest candidate profile set because it only contains is smaller, and only matches documents containing candidate email addresses—it appears that this set misses vital evidence of the candidate's expertise that is expressed in documents which only contain the candidate names. Hence, as expected, this set does not exhibit any significant improvements from the median runs, and does, on a few occasions, exhibit some significant degradations. On the other hand, we hypothesise that the Name and Aliases set must contain too many false-positives, caused by the variations of candidate names matching documents incorrectly. Only some settings show significant improvements from the median runs for this profile set. Overall, the Full Name candidate profile set performs best across both tasks, weighting models and fusion techniques.

As mentioned above, the large difference between the retrieval performance between the TREC 2005 and TREC 2006 tasks is expected: for the TREC 2005 expert search task, which was seen as a pilot task [3], the ground truth was taken from the W3C working groups. In contrast, for TREC 2006 each suggested candidate by a system was assessed for having relevant expertise, by looking for documents which supported those conclusions. Hence, the TREC 2006 contained a fuller assessment of all the candidates provided by systems. This explains the higher retrieval accuracy figures seen for the TREC 2006 task by all participating systems.

3.3. Conclusions

From the results in Tables 2 and 3, we can surmise that good indicators of expertise of a candidate seem to be the number of documents in the candidate's profile retrieved for a query (number of votes), and the relative magnitude of the retrieval scores in the candidate's profile (strength of votes). Indeed, the strongly performing CombMNZ combines both these indicators, while the CombMAX focuses on the most strongly voting documents.

In terms of weighting models, all weighting models performed similarly, without the need to tune any of their parameters, particularly on the best–performing Full Name candidate profile set. For the remainder of this article, we concentrate only on the PL2 weighting model, because the use of PL2 allows flexibilities, such as the application of candidate length normalisation, the use of document structure and of term proximity information, as will be shown in Sections 4 and 5.

Overall, we have shown that the proposed approach using the voting techniques can be effectively applied to expert search. Indeed, the retrieval performances exhibited in Table 2 would have been placed as a third group at TREC 2005, while for TREC 2006 the results would have been placed at the level of the second group.[2] Note, however, that these techniques do not take any collection- or topic-specific heuristics into account. Moreover, no parameters have been trained to maximise accuracy.

4. NORMALISING CANDIDATES VOTES

While the voting approach produces effective expert search retrieval, we hypothesise that it can be biased towards candidates with many associated documents: the more documents that are associated with a candidate (a large profile), the more likely that the candidate is to receive a vote from the document ranking by chance. This is because a large candidate profile is more likely to have misassociated documents, that cause the candidate to be incorrectly retrieved.

Similarly, to the introduction of document length normalisation in document retrieval models, we normalise candidates that have long profiles, in order to prevent them from gaining too many votes from the document ranking by chance.

In this work, we adapt a classical document length normalisation technique, *Normalisation 2* from the DFR framework [19] [see Equation (7)], and integrate it into the voting model for expert search.

Important to Normalisation 2 is the definition of length. In this work, we experiment with two methods of measuring the size of candidate profiles: first, by the number of tokens in the candidate profile (total term occurrences); and, secondly, by measuring the profile size as the number of

[2]To provide a comparable basis, the ranking of submitted runs is performed for automatic runs using only the title field of the topics.

associated documents with the candidate. The first of these is a more accurate measure of the length of the profile, as documents within a profile can have varying lengths. However, all the weighting models applied in this task take into account document length; therefore the generated document ranking should have no bias towards documents of short or long length. Hence, we also experiment with measuring the length of profiles in terms of documents.

To apply candidate-length normalisation to the voting model, we alter the score of a candidate score_cand(C, Q), as follows:

$$\text{score\_cand}_{\text{Norm2}}(C, Q) = \text{score\_cand}(C, Q) \cdot \log_2 \left(1 + c_{\text{pro}} \cdot \frac{\text{avg\_l\_pro}}{\text{l\_pro}}\right), \quad (9)$$

where avg_l_pro is the average length of all candidate profiles, and l_pro is the length of the profile of candidate C. Both avg_l_pro and l_pro can be counted either in terms of tokens, or in terms of documents. c_{pro} is a hyper-parameter controlling the amount of candidate profile length normalisation applied ($c_{\text{pro}} > 0$). The higher the value of c_{pro}, the less normalisation is applied to score_cand(C, Q). As with Amati [19], we suggest that $c_{\text{pro}} = 1$ is a good initial setting. Note that a smaller c_{pro} value will penalise larger profiles more.

In the following sections, we evaluate the candidate length normalisation by applying it to all our voting techniques, and across the four candidate profile sets we created in Section 2. Moreover, we experiment with varying the value assigned to c_{pro}, to assess what effect this has on retrieval performance.

4.1. Evaluation

If the baseline uses the voting technique is called M, say, then MNorm2T denotes the use of the voting technique M in conjunction with candidate length normalisation applied using Equation (9) calculated using the total number of tokens in the documents associated to the profile as the measure of the profile size. Similarly, MNorm2D denotes when Normalisation 2 is applied using the number of documents as the measure of profile size. Tables 4 and 5 present the experiments made by applying Normalisation 2 as candidate length normalisation, to both the TREC 2005 and TREC 2006 expert search tasks. Our baselines (using the PL2 document weighting model) are the equivalent runs from Tables 2 and 3 without any normalisation applied.

On analysing Tables 4 and 5, the following can be observed. First, the benefit of candidate length normalisation is different across the voting techniques applied. In particular, for the TREC 2005 setting, CombSUM overall performs best with the application of candidate length normalisation using the number of documents in a profile (Norm2D), with the exception of Last Name. For the TREC 2006 data, Norm2T is better than Norm2D than Norm2D for CombSUM, though it is better not to apply any normalisation for the Full Name and Email Address candidate profile sets.

Similarly to CombSUM, CombMNZ performs better using Norm2D for TREC 2005 (with the exception of P@10 for Last Name), while Norm2T is better than Norm2D on the TREC 2006 topics, although again it is better not to apply any normalisation for the Full Name and Email Address candidate profile sets.

In contrast to CombSUM and CombMNZ, the CombMAX voting technique almost always works best with no normalisation applied (the only exception here is P@10 for Last Name on the TREC 2005 topics; however, the improvement in applying normalisation is not significant). Note that this is expected, as CombMAX can only receive at most one vote from the document ranking, and hence, the application of candidate length normalisation for this technique is unnecessary, because large candidate profiles have less chance to overinfluence the ranking of candidates. In particular, the results for CombSUM and CombMNZ with normalisation applied are comparable to those of CombMAX without normalisation.

Examining the effect of normalisation across the different candidate profile sets, we again see a variation between Tables 4 and 5. While disregarding CombMAX, for the TREC 2005 topics, all profile sets perform better with candidate length normalisation applied, while for the TREC 2006 topics, only Last Name and Name and Aliases profile sets show improvement in applying normalisation. It is of interest that the noisiest profile sets are the ones which improve with normalisation, and that normalisation is required most for the voting techniques that are susceptible to profile noise (CombSUM and CombMNZ).

In contrasting Tables 4 and 5, we note that candidate profile length normalisation is more important for the TREC 2005 topics than for the TREC 2006 topics.

In particular, for the highest performing candidate profile set, Full Name, no normalisation is necessary for TREC 2006, and applying any normalisation can result in a serious impact to the retrieval performance, particularly for the CombMAX voting technique. These degradations of performance are mostly statistically significant.

Finally, the performance of Norm2D and Norm2T components is overall extremely similar— compared to the base line, in no case does one form of normalisation benefit retrieval performance while another hinders. In the next section, we vary the candidate profile length normalisation parameter, c_{pro}, to see the effect that this has on the accuracy of the generated candidate ranking, and will investigate further the apparent similarity between Norm2D and Norm2T.

4.2. Effect of varying candidate length normalisation

In this section, we observe the effect of the candidate profile normalisation component, by measuring MAP as the c_{pro}

TABLE 4. Evaluation on the expert search task of the TREC 2005 Enterprise track, of the use of candidate profile length normalisation in expert search.

	Last Name		Full Name		Name and Aliases		Email Address	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
CombSUM	0.0835	0.1520	0.1688	0.2660	0.1336	0.2100	0.1172	0.2200
CombSUMNorm2T	**0.1981**$\gg$	**0.3340**$\gg$	0.2108	0.3300	0.2113	0.3320	0.1333$>$	0.2700
CombSUMNorm2D	0.1911$\gg$	0.3280$\gg$	**0.2221**$>$	**0.3500**$\gg$	**0.2233**$\gg$	**0.3400**	**0.1369**$>$	**0.2900**$>$
CombMAX	**0.1370**	0.1660	**0.2247**	**0.3020**	**0.1950**	**0.2640**	**0.1315**	**0.2560**
CombMAXNorm2T	0.0929$<$	**0.1780**	0.1182$\ll$	0.1980	0.1140$\ll$	0.1960$<$	0.0851$\ll$	0.2120
CombMAXNorm2D	0.0863$\ll$	0.1660	0.1277$\ll$	0.2080$\ll$	0.1178$\ll$	0.1960$<$	0.0860$<$	0.2200
CombMNZ	0.0812	0.1460	0.1652	0.2540	0.1311	0.2120	0.1169	0.2160
CombMNZNorm2T	0.1574$\gg$	**0.2520**$\gg$	0.1987$>$	0.3000$>$	0.1908$\gg$	0.3060$\gg$	0.1350$\gg$	0.2400
CombMNZNorm2D	**0.1600**$\gg$	0.2460$\gg$	**0.2200**$\gg$	**0.3320**$\gg$	**0.2052**$\gg$	**0.3260**$\gg$	**0.1362**$\gg$	**0.2700**$\gg$

We test two forms of normalisation, namely normalisation in terms of tokens (Norm2T), and in terms of documents (Norm2D), with three voting techniques, across four different candidate profile sets. For each setting, the baseline (without normalisation) from Table 2 is given. Retrieval performance is measured using MAP and P@10. Statistically significant improvements from the baseline at $P \leq 0.05$ are denoted $>$; significant improvements at $p \leq 0.01$ are denoted $\gg$. Similarly, statistically significant degradations from the baseline are denoted $<$ and $\ll$, respectively. The best performance for each setting is in bold, and the best in each column is italicised.

TABLE 5. Evaluation on the expert search task of the TREC 2006 Enterprise track, of the use of candidate profile length normalisation in expert search.

	Last Name		Full Name		Name and Aliases		Email Address	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
CombSUM	0.2384	0.3204	**0.5066**	**0.5918**	0.3727	**0.4939**	**0.3544**	**0.5102**
CombSUMNorm2T	**0.3340**$\gg$	**0.4571**$\gg$	0.3641$\ll$	0.4592$\ll$	**0.3904**	0.4898	0.3011$\ll$	0.4490$<$
CombSUMNorm2D	0.2939$>$	0.4082$>$	0.3389$\ll$	0.4204$\ll$	0.3430$<$	0.4224$<$	0.2775$\ll$	0.4184$\ll$
CombMAX	**0.2299**	**0.2816**	**0.5031**	**0.5653**	**0.3902**	**0.4490**	**0.3617**	**0.5429**
CombMAXNorm2T	0.0699$\ll$	0.1286$\ll$	0.1159$\ll$	0.1429$\ll$	0.1058$\ll$	0.1510$\ll$	0.2083$\ll$	0.2898$\ll$
CombMAXNorm2D	0.0626$\ll$	0.1204$\ll$	0.1237$\ll$	0.1306$\ll$	0.1095$\ll$	0.1347$\ll$	0.2001$\ll$	0.2776$\ll$
CombMNZ	0.2320	0.3143	**0.5023**	**0.5898**	0.3686	0.4816	**0.3505**	**0.5041**
CombMNZNorm2T	**0.3505**$\gg$	**0.4592**$\gg$	0.4731	0.5694	**0.4590**$\gg$	**0.5531**$\gg$	0.3416	0.4939
CombMNZNorm2D	0.3250$\gg$	0.4388$\gg$	0.4498$<$	0.5388	0.4271$\gg$	0.5224	0.3249	0.4898

We test two forms of normalisation, namely normalisation in terms of tokens (Norm2T), and in terms of documents (Norm2D), with three voting techniques, across four different candidate profile sets. For each setting, the baseline (without normalisation) from Table 3 is given. Notation is as in Table 4.

value is varied. Figures 2–4 show the MAP for the CombSUM, CombMAX and CombMNZ voting techniques, when either Norm2D or Norm2T is applied. All four candidate profile sets are experimented with. Note that for Normalisation 2, the c parameters have effect on exponential scale; therefore, to cover the parameter space with the minimum number of settings, the x-axis of each figure is in log scale.

The figures allow us to draw the following observations: length normalisation is desirable for the CombSUM voting techniques on the TREC 2005 data (Fig. 2a), as a definite peak in performance is reached for most profile sets in this figure. In Fig. 4a, when CombMNZ is applied on the TREC 2005 data, the smallest c_{pro} is most desirable, showing that candidate length normalisation is important for this voting technique in this setting. In particular, this infers that the number of votes evidence is not desirable for the TREC 2005 task, and it would be better to apply the CombSUM voting technique in this setting, because the small c_{pro} values are balancing out the effect of the $\|R(Q) \cap \text{profile}(C)\|$ component in CombMNZ. In contrast, for the other four settings, i.e., Figs 2b, 3 and 4b, MAP increases as the c_{pro} value is increased, that is, as less candidate profile length

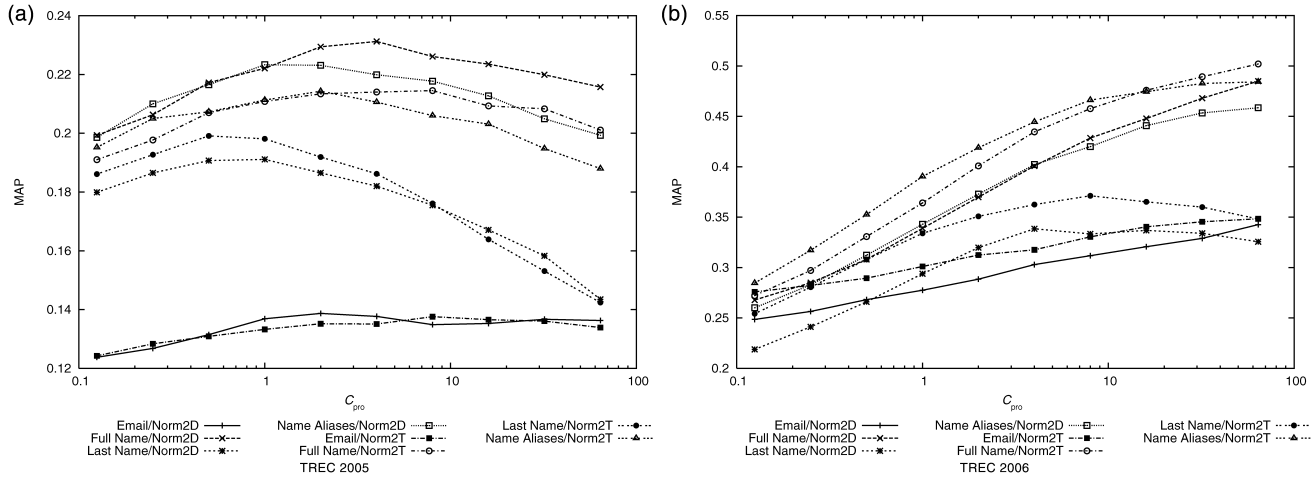

FIGURE 2. The MAP retrieval accuracy of CombSUM voting technique when varying amounts of candidate length normalisation components (c_{pro}) are applied. Both forms of candidate length normalisation are tested (Norm2D and Norm2T) over the four candidate profile sets.

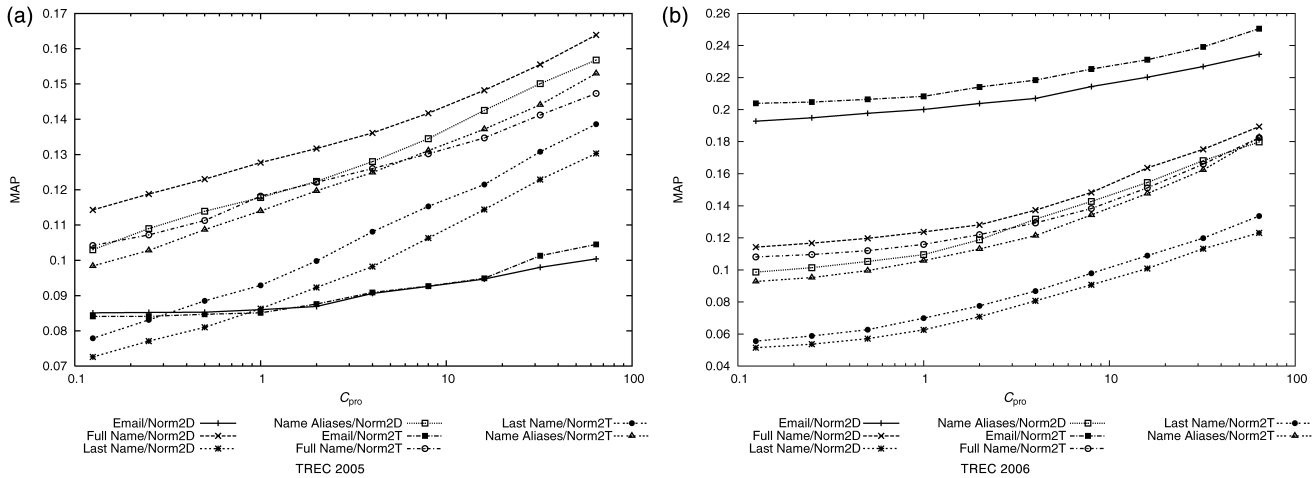

FIGURE 3. The MAP retrieval accuracy of CombMAX voting technique when varying amounts of candidate length normalisation components (c_{pro}) are applied. Both forms of candidate length normalisation are tested (Norm2D and Norm2T) over the four candidate profile sets.

normalisation is applied. The reason that candidate length normalisation appears incompatible with CombMAX is explained above, while for CombSUM and CombMNZ on the TREC 2006 data, we can infer that number of votes evidence is useful, and hence it is better not to apply candidate length normalisation.

The final observation from the figures is that the plot lines for different forms of normalisation are paired and parallel, that is, a line representing Norm2D in a given setting is usually very similar to the line representing Norm2T. From this observation, we can conclude that both forms of candidate length normalisation are roughly equivalent, and any differences in retrieval performance between the two can be eliminated by a slight variance of the c_{pro} parameter. This suggests that both ways of measuring candidate length are correlated, and hence, for this collection, there is no variance in the average document lengths of the documents in each candidate's profile, thus making candidate length normalisation unnecessary. Indeed Spearman's ρ on the Full Name candidate profile set shows 0.97 correlation between the number of documents in a profile and the number of tokens.

4.3. Conclusions

In conclusion, we have seen that candidate length normalisation is necessary in some settings to improve the retrieval performance of some voting techniques, under certain noisy conditions. In particular, the evaluation showed that normalisation is more useful on the more difficult TREC 2005 topics than on the TREC 2006 ones. Moreover, TREC 2005 topics perform best with length normalisation in terms of documents, while for TREC 2006, length normalisation in terms of tokens is more effective, but only for noisy candidate profile sets and when CombMAX is not applied. We conclude that it is

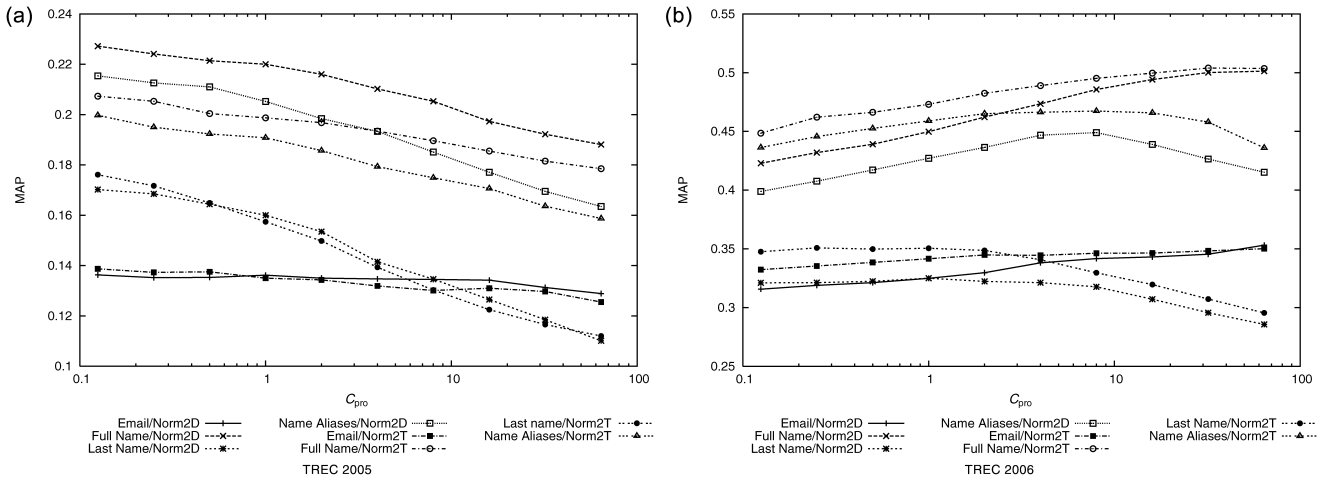

FIGURE 4. The MAP retrieval accuracy of CombMNZ voting technique when varying amounts of candidate length normalisation components (c_{pro}) are applied. Both forms of candidate length normalisation are tested (Norm2D and Norm2T) over the four candidate profile sets.

important to take length normalisation into account in the voting model, as it can massively improve the performance of some voting techniques, particularly when inaccurate or noisy candidate profile sets are applied.

In the remaining experiments in this paper, we use the Full Name profile set, as this is the best performing setting. For this set, no normalisation is usually needed, particularly on TREC 2006. Moreover, this profile set gives the best results across all voting techniques, good document weighting models and tasks, and hence is a good baseline for use in the rest of this article.

5. IMPROVING THE DOCUMENT RANKING

In the proposed voting model for expert search, the accuracy of the retrieved list of candidates is dependent on two features: the manner in which the votes from the documents in the document ranking $R(Q)$ are combined, and the document ranking $R(Q)$. In Sections 3 and 4, we investigated different ways in which document votes could be combined, into a ranking of candidates. In contrast, this section investigates the relative benefit of applying to an expert search system three document retrieval techniques that normally improve the effectiveness of a document search engine.

In terms of the voting techniques described above, the accuracy of the generated ranking of candidates is dependent on how well the document ranking $R(Q)$ ranks documents associated with relevant candidates—we call this the quality of the document ranking. Relevant candidates should have a mix of highly ranked documents that are about the topic (strong votes) or written prolifically around the topic (number of votes). We have no way of measuring the 'quality' of the document ranking directly, so instead we vary the document ranking and evaluate the accuracy of the generated ranking of candidates to draw conclusions about the type of retrieval techniques that should be deployed. We naturally hypothesise that applying retrieval techniques that typically increase the precision and/or recall of a normal document IR system will increase the quality of the document ranking in the expert search system, and hence will increase the performance of the generated candidate ranking.

In Section 3, we saw that the choice of document weighting model applied to generate the document ranking $R(Q)$ has little effect on the overall candidate ranking. In this section, we further test our document ranking hypothesis by applying three techniques, which we believe will increase the quality of the document ranking.

First, the structure of HTML documents in Web and Enterprise settings can bring additional information to an IR system—for instance, does the term occur in the title or content of the document, in an emphasised tag (such as ⟨H1⟩), or does it occur in the anchor text of the incoming hyperlinks to the documents. We know that taking into account the structure of documents can allow increased precision for document retrieval [27], particularly on the W3C collection [33]. Hence, we apply a field-based weighting model from the DFR framework, to take the structure of each document into account when ranking the documents. For instance, this model allows the higher scoring of documents where query terms occur in the title or anchor text of the incoming hyperlinks of the documents, than when they occur in the content of the document alone. By taking the structure of the document into account, we expect to see a higher document ranking, particularly with more on-topic documents at the top of the document ranking.

Secondly, we use a novel information-theoretic model, based on the DFR framework, for incorporating the dependence and proximity of query terms in the documents. We believe that query terms will occur close to each other in

on-topic documents, and by modelling this co-occurrence and proximity of the query terms, we can increase the quality of the document ranking, by ranking these on-topic documents higher in the document ranking $R(Q)$. A high-quality document ranking will be aggregated into an accurate ranking of candidates.

Thirdly, we investigate the use of query expansion in the expert search setting. Query expansion describes the process of implicit relevance feedback, where a ranking is improved by assuming that some of the top-ranked items are relevant, known as the pseudo-relevant set. Additional query terms are then extracted from the top-ranked documents and added to the initial query. By rerunning the initial query, recall is improved, as relevant documents not containing the initial query terms are retrieved. Meanwhile, precision is also increased, as the query terms are re-weighted to bring the document ranking closer to the general topic area of the query. In applying query expansion to the document ranking $R(Q)$, we hope to bring more on-topic documents into and higher ranked in the document ranking, therefore increasing its quality for the expert search techniques.

In the following sections, we detail the retrieval-enhancing techniques deployed, explain the experiments carried out and present experimental results for each test of the hypothesis.

5.1. Field-based document weighting model

A field-based weighting model takes into account separately the influence of a term in a field of a document (e.g., in the title, content, H1 tag or even in the anchor text of the incoming hyperlinks). Such a model was suggested by Robertson *et al*. [34], where the weighted term frequencies from each field were combined before being used in ranking. Robertson found this to be superior to the postretrieval combination of scores of weighting models applied on different fields. However, as found by Zaragoza *et al*. [35], the distribution of term occurrences varies across different fields. They found that a combination of the frequencies of a term in the various fields is best performed after the document length normalisation component of the weighting model is applied. Similarly, we use a field-based weighting model called PL2F, which is a derivative of the document weighting model PL2 [Equation (6) in Section 3]. In the PL2F model, the document length normalisation step is altered to take a more fine-grained account of the distribution of query term occurrences in the different fields. The so-called Normalisation 2 [Equation (7)] is replaced with *Normalisation 2F* [32, 36], so that the normalised term frequency tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f:

$$\mathrm{tfn} = \sum_f \left(w_f \cdot \mathrm{tf}_f \cdot \log_2 \left(1 + c_f \cdot \frac{\mathrm{avg\_l}_f}{l_f} \right) \right) \quad (c_f > 0), \quad (10)$$

where c_f is a hyper-parameter for each field controlling the term frequency normalisation, and the contribution of the field is controlled by the weight w_f. Together, c_f and w_f control how much impact term occurrences in a field have on the final ranking of documents. tf_f is the term frequency of term t in field f of document d, and l_f is the number of tokens in field f of the document. $\mathrm{avg\_l}_f$ is the average length of the field in the collection. Having defined Normalisation 2F, the PL2 model [Equation (6)], can be extended to PL2F by using Normalisation 2F [Equation (10)] to calculate tfn.

In the following, we compare the retrieval performance of the generated ranking of candidates, when PL2 and PL2F are used to rank documents in the document ranking $R(Q)$. The field we apply are content, title and anchor text of incoming hyperlinks. However, the additional parameters in the PL2F weighting model, compared to PL2 (i.e. c_f and w_f for each field), mean that the weighting models required training to set them to appropriate values. Due to the lack of availability of representative training data for TREC 2005, we use the parameter values for c_f and w_f trained to maximise MAP using the TREC 2005 topics (as in [5]), and then test using the TREC 2006 topics. Moreover, to provide a comparable baseline, we also train the c parameter of PL2. Table 6 shows the accuracy of the ranking of candidate generated when using PL2 or PL2F, compared across three different voting techniques, all using the Full Name candidate profile set. In this table, PL2/default denotes the default $c = 1$ setting of PL2, PL2/trained denotes when the c parameter has been trained using the TREC 2005 topics, and PL2F/trained denotes when the *cf* and *wf* parameters have been trained using the TREC 2005 topics.

From the results in Table 6, we can see that the retrieval performance of PL2F is better for both MAP and P@10 measures, on both the TREC 2005 and TREC 2006 tasks. However, the gain for TREC 2005 is larger than the gain for TREC 2006, and there is no statistically significant improvement for TREC 2006. First, this is because the model has been overfitted to maximise MAP on the TREC 2005 topics, so its good performance here is expected. Moreover, we suspect that the training from [5] (on the TREC 2005 task) is not sufficiently representative for the easier TREC 2006 task. It is noticeable that the when the c parameter of PL2 is trained on the 2005 topics, this can result in a degradation compared to the default $c = 1$ in the 2006 topics. However, the relative weighting of the importance of the fields seems to be consistent between the topic sets, as the fields setting trained on the 2005 topics always increases retrieval performance on the TREC 2006 topics. Moreover, if more representative training existed, we would expect a larger, statistically significant margin of improvement on the TREC 2006 task.

5.2. Term dependence

When more than one query term occurs in a document, it is more likely to be relevant to a query than if a single query

TABLE 6. Applying the field-based model PL2F, to enhance the quality of the document ranking $R(Q)$.

	CombSUM		CombMAX		CombMNZ	
	MAP	P@10	MAP	P@10	MAP	P@10
TREC 2005						
PL2/default	0.1688	0.2660	0.2247	0.3020	0.1652	0.2540
PL2/trained	0.1698	0.2660	0.2352	0.3320	0.1669	0.2540
PL2F/trained	**0.1805**$>$	**0.2920**	**0.2760**$\gg$	**0.3980**$\gg$	**0.1673**	**0.2560**
TREC 2006						
PL2/default	0.5066	0.5918	0.5031	0.5653	0.5023	0.5898
PL2/trained	0.5042	0.5918	0.4922	0.5612	0.4997	0.5878
PL2F/trained	**0.5085**	**0.6082**	**0.5048**	**0.5918**	**0.5048**	**0.6000**

Results are shown for the TREC 2005 and TREC 2006 expert search tasks, for MAP and P@10. PL2/default denotes the default $c = 1$ setting of PL2, PL2/trained denotes when the c parameter has been trained using the TREC 2005 topics, and PL2F/trailed denotes when the c_f and w_f parameters have been trained using the TREC 2005 topics. Statistically significant improvements from the baseline at $P \leq 0.05$ are denoted $>$; significant improvements at $P \leq 0.01$ are denoted $\gg$. The best retrieval performance in each setting is emphasised.

term appears. Moreover, it has been shown that when query terms occur near to each other in a document—in proximity—it can be a further indicator of relevance [37]. Such term dependence and proximities can also be modelled using the DFR framework, by using document weighting models that capture the probability of the occurrence of pairs of query terms in the document and the collection [38]. Following [39], the term dependence weighting models are based on the probability that two terms should occur within a given proximity. The introduced weighting models assign scores to pairs of query terms, in addition to the single query terms.

The score of a document d for a query Q is given as follows:

$$\text{score}_p(d, Q) = \text{score}(d, Q) + \sum_{p \in Q_2} \text{score}(d, p), \qquad (11)$$

where score(d, t) is the score assigned to a query term t in the document d with respect to query Q, score (d, p) is the score assigned to a query term pair P from Q_2, which is the set that contains all the possible combinations of two query terms. In Equation (11), the score, score(d, Q) is the existing score of the document, calculated using single query terms, such as by PL2 [Equation (6)]. The score(d, p) of a query term pair in a document is computed as follows [38]:

$$\text{score}(d, p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}), \qquad (12)$$

where P_{p1} corresponds to the probability that there is a document in which a pair of query terms p occurs a given number of times. P_{p1} can be computed with any DFR model, such as the Poisson approximation to the binomial distribution. P_{p2} corresponds to the probability of seeing the query term pair p once more, after having seen it a given number of times. P_{p2} can be computed using any of the after-effect models in the DFR framework. The difference between score(d, p) and a classical document weighting model is that the former employs counts of occurrences of the query term pairs in a document, while the latter depends only on the counts of occurrences of each query term.

For example, applying term dependence and proximity with the weighting model PL2 [see Equations (6), (7)] results in a new version of PL2, denoted pPL2, where the prefix p stands for proximity. pPL2 estimates score(d, p) as follows:

$$\text{score}(d, p) = \frac{1}{\text{pfn} + 1}\left(\text{pfn} \cdot \log_2 \frac{\text{tfn}}{\lambda_p} + (\lambda - \text{pfn}) \cdot \log_2 \text{e} + 0.5 \cdot \log_2(2\pi \cdot \text{tfn})\right), \qquad (13)$$

where $\lambda_p = F_p/N$. F_p corresponds to the number of documents in which the pair of query terms p appears within dist terms of each other, and N is the number of documents in the whole collection. pfn is the normalised frequency of a query term pair occurring in document d, calculated using Normalisation 2 [Equation (7)]. The occurrence of a query term pair p in a document d is counted as the number of windows of size dist that contain the pair of query terms p. As suggested by Lioma *et al*. in [38], we use dist = 5.

We apply the term dependence model to the document ranking, using the PL2 weighting model with default setting $c = 1$ as the baseline. Again, we test across three voting techniques and using the Full Name candidate profile set. Table 7 presents the results for the TREC 2005 and TREC 2006 expert search tasks. On analysing the results, we can see that the retrieval performance, in terms of MAP and P@10, of the baselines is improved when the term dependence model is applied, for both the TREC 2005 and 2006 expert search tasks, regardless of the voting technique applied. In particular, there are statistically significant gains in P@10 in five out of

TABLE 7. Applying the term dependence model pPL2 to increase the quality of the document ranking $R(Q)$.

	CombSUM		CombMAX		CombMNZ	
	MAP	P@10	MAP	P@10	MAP	P@10
TREC 2005						
PL2/default (baseline)	0.1688	0.2660	0.2247	0.3020	0.1652	0.2540
+ term dependence pPL2	**0.1934**	**0.3040**$>$	**0.2441**	**0.3360**$\gg$	**0.1883**	**0.2920**$>$
TREC 2006						
PL2/default (baseline)	0.5066	0.5918	0.5031	0.5653	0.5023	0.5898
+ term dependence pPL2	**0.5382**$\gg$	**0.6429**$\gg$	**0.5226**	**0.6224**	**0.5257**	**0.6286**$>$

The baseline, denoted PL2/default, uses the PL2 weighting model with default setting $c = 1$. Results are shown for the TREC 2005 and TREC 2006 expert search tasks, for MAP and P@10. Statistically significant improvements from the baseline at $P \leq 0.05$ are denoted $>$; significant improvements at $P \leq 0.01$ are denoted $\gg$. The best retrieval performance in each setting is emphasised.

six settings ($P \leq 0.05$), and there are significant improvements in MAP and P@10 for CombSUM on the TREC 2006 topics ($P \leq 0.01$). Although the increases in performance shown by CombMAX are substantial (only significant in one case), these increases are expected, as the influence of term dependence is likely to affect the highest part of the ranking, which is where CombMAX is most affected. Overall, we conclude that the application of query term proximity evidence to the document ranking can enhance the retrieval accuracy of the generated ranking of candidates.

In summary, it appears that the use of the term dependence model to improve the quality of the document ranking can improve the accuracy of the generated candidate ranking, and significantly improve the high precision of candidate ranking. In particular, it appears that applying term dependence brings new evidence, and is more likely to improve the accuracy of an expert search system than the inclusion of document structure evidence such as a fields-based weighting model.

Like Web IR, we believe expert search to be a high precision task—a user is unlikely to contact all experts retrieved for a query to ask for assistance, and instead will concentrate on the most highly ranked experts. It should be noted that for the best setting (CombSUM + Term dependence on TREC 2006, MAP 0.5382), the average reciprocal rank of the first relevant expert is 0.9167, and a relevant expert is ranked first for 87% of queries (Success at rank 1).

Indeed this level of performance would have ranked between the first and second groups at the TREC 2006 expert search task.

5.3. Document-centric query expansion

In this section, we examine whether the application of query expansion can be used to increase the quality of the document ranking. Query expansion describes the process where a few top items in a ranking are assumed to be relevant to the query (known as the pseudo-relevant set). The IR system can then make use of the term occurrence information in the pseudo-relevant set to expand the query with new query terms. By rerunning using the improved query, a new ranking is created, typically with higher precision and recall.

In applying query expansion in expert search, we assume that the top-ranked documents in the document ranking $R(Q)$ contain relevant information to the topic, and hence are good indicators of expertise [40]. We can then apply automatic relevance feedback, in the form of query expansion, to generate an expanded query. We hypothesise that by regenerating the document ranking $R(Q)$ using the expanded query, we should have more highly ranked documents that are about the topic of interest. Then, aggregating the votes for candidates from this improved ranking of documents should result in a more accurate ranking of candidates. We call this form of query expansion as document-centric query expansion, because it acts purely on the document ranking.

Terrier [28, 29] deploys several DFR models for weighting query terms for the purpose of query expansion. We use the Bo1 term weighting model, which is based on Bose–Einstein statistics and is similar to Rocchio [19]. In Bo1, the informativeness $w(t)$ of a term t is given by

$$w(t) = tf_x \cdot \log_2\left(\frac{1 + P_n}{P_n}\right) + \log_2(1 + P_n), \qquad (14)$$

where tf_x is the frequency of the term in the pseudo-relevant set, and P_n is given by 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.

The top exp_term informative terms are identified from the top exp_doc ranked documents, and these are added to the query (exp_term ≥ 1, exp_doc ≥ 2). We use the default setting for these parameters, that is, exp_doc $= 3$ and exp_term $= 10$, as suggested by Amati [19] after extensive experiments. Finally, the query term frequency qtw of an

TABLE 8. Applying document-centric query expansion to increase the quality of the document ranking $R(Q)$.

	CombSUM		CombMAX		CombMNZ	
	MAP	P@10	MAP	P@10	MAP	P@10
TREC 2005						
PL2 baseline	0.1688	0.2660	0.2247	0.3020	0.1652	0.2540
+ QE	**0.1818**>	**0.2860**	**0.2465**	**0.3360**>	**0.1760**	**0.2820**
TREC 2006						
PL2 baseline	0.5066	0.5918	0.5031	0.5653	0.5023	0.5898
+ QE	**0.5074**	**0.6020**	**0.4904**	**0.5694**	**0.5008**	**0.6061**

The baseline, denoted PL2/default, uses the PL2 weighting model with default setting $c = 1$. Results are shown for the TREC 2005 and TREC 2006 expert search tasks, for mean average precision (MAP) and precision @ 10 (P@10). Statistically significant improvements from the baseline at $P \leq 0.05$ are denoted >.

expanded query term is given by

$$\text{qtw} = \text{qtw} + \frac{w(t)}{w_{\max}(t)}, \tag{15}$$

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.

We experiment with applying document-centric query expansion on both the TREC 2005 and 2006 expert search tasks. Our baseline document ranking is created using the PL2 document weighting model, using the default setting $c = 1$, as in Tables 2 and 3. Similarly to before, we experiment with query expansion across three voting techniques and using the Full Name candidate profile set. Table 8 details the results of the experiments using document-centric query expansion. According to the results, we can see that applying document-centric query expansion affects retrieval performance on both tasks. For TREC 2005, the retrieval performance is enhanced for all voting techniques, for both MAP and P@10 measures—in some cases these improvements are significant. For TREC 2006, P@10 is always enhanced, but there is a slight drop in MAP for the CombMAX and CombMNZ voting techniques.

We conclude that document-centric query expansion can be applied for improving the quality of the document ranking. However, the disappointing improvements in retrieval performance lead us to conclude that perhaps the default query expansion parameters of exp_doc = 3 and exp_term = 10 as suggested by Amati are not ideal in the expert search setting, to provide a high quality document ranking. In [40], we examined in detail the effect on retrieval performance of varying the exp doc and exp term parameters. Indeed, for the experimental setting applied in that paper (DLH13 DFR weighting model combined with the expCombMNZ voting technique), the best setting achieved on the TREC 2006 topics was MAP 0.5799, a marked increase over the default setting of query expansion in that paper. Again, this level of performance would have ranked between the first and second groups at the TREC 2006 expert search task. To conclude, these experiments suggest that query expansion can be used in some way to improve the accuracy of an expert search system. It is of note that we have since investigated the query expansion problem in expert search in more detail, and proposed several hybrid query expansion techniques that improve expert search system accuracy in the expert search task [41].

5.4. Conclusions

In conclusion, we have examined three techniques for improving the retrieval performance of the underlying ranking of documents used by the expert search model account. These were the use of a field-based weighting model to take into account a more refined account of the distribution of query terms in the structured documents; the use of a term dependence model that takes into account the co-occurrence and proximity of query terms in the documents; and lastly, the application of query expansion on the document ranking.

All of these techniques demonstrated potential to increase the accuracy of the expert search system, in terms of MAP and/or P@10. If representative training data were available for both tasks, it is likely that the retrieval accuracy would increase further. In particular, from the experiments conducted, it seems that the use of term dependence brings the largest increase in retrieval accuracy, however, given a proper setting, the application of query expansion may also yield marked improvements in retrieval performance. Finally, applying all three techniques at oncethat is, fields-based weighting model, query expansion and term dependencefurther enhances retrieval performance: on the TREC 2006 task, using the voting technique CombSUM, an MAP of 0.5417 was achieved (P@10 0.6612). This is better than the second ranked group s runs at TREC 2006, and hence is comparable to the state-of-the-art. Comparing to the best group s runs at TREC 2006, it is apparent that this final boost in performance (MAP 0.5947 [4]) was achieved by using appropriately sized windowing of query terms and candidate occurrences, thus considering expertise evidence on a less granular level than the individual documents of each candidate's profile [17]. This increases the quality of the candidate profile, which as demonstrated in this article, is an important factor in the performance of an expert search system. We plan to adapt our profile set to take into account this fine-grained evidence, the use of which does not change the proposed voting model.

We conclude that state-of-the-art techniques can be successfully applied to increase the quality of the document ranking, and hence improve the accuracy of the generated ranking of candidates. Moreover, all the techniques applied to increase

the quality of the document ranking had the effect of increasing the high precision, such as P@10, of the generated ranking of candidates. This is important, as we believe that expert search is a high precision task: user satisfaction is likely to be correlated with a high precision measure such as P@10, as they will select a candidate in the top 10 results, say, rather than contacting each suggested expert in a list of 100.

6. CONCLUSIONS

Expert search is an important task in enterprise environments. This article describes the voting model for expert search, to accurately rank candidates with respect to their estimated expertise to a query.

In the voting model, the ranking of documents with respect to the query (denoted $R(Q)$) is considered to contain implicit information about the expertise of candidates. We see this as implicit votes by documents to their associated candidates. We model this information using a selection of voting techniques to combine the votes of documents into an accurate ranking of candidates.

The voting model is flexible, as it can take as input, the output of any normal document search engine that gives a ranking of documents in response to a query. In this work, we selected three state-of-the-art document weighting models to generate the underlying document ranking. Moreover, many techniques can be applied to generate the candidate document associations (candidate profile sets). Furthermore, numerous voting techniques can be applied to aggregate the votes by the documents for the candidates. Additionally, normalisation can be optionally applied to prevent any bias in the candidate ranking to candidates with large profiles.

We thoroughly evaluated several voting techniques in the context of the expert search tasks of the TREC 2005 and 2006 Enterprise tracks. To assess the effectiveness and robustness of the techniques, we experimented over four candidate profile sets and three state-of-the-art statistically different document weighting models.

The results show that the proposed voting model is effective when using appropriate voting techniques and appropriate candidate document associations. The most successful techniques integrate one or more of the following features: the most highly ranked documents, or even just the single highest ranked document (strong vote(s)), and the number of retrieved documents from the profile (number of votes). Our experiments show that the quality of the candidate document associations is important for good retrieval accuracy. This is exemplified by the fact that the Full Name candidate profile set performed best overall throughout our experiments. Moreover, as discussed above, the use of windowing would increase the quality of the candidate profile sets further, by discarding expertise evidence for a candidate that does not appear near terms of the query.

Furthermore, we examined the effect of candidate profile size in the voting model. Our experimental results suggest that for more difficult topics, and more noisy candidate profile sets, candidate length normalisation can be useful.

Finally, we investigated the effect of the document ranking on the accuracy of the generated ranking of candidates. We showed that using techniques to increase the quality of the document ranking could increase the retrieval performance of the generated candidate ranking.

The approach proposed in this paper is general in the sense that it is not dependent on heuristics from the used enterprise collection, and can be easily operationally deployed with little computational overhead, even on an existing intranet search engine. In particular, the voting model is not dependent on the techniques used to generate the underlying document ranking, or the method used to generate the profiles of the experts. For example, the introduction of windowing would not force a change in the proposed voting model.

Looking forward, our future work directions have several prongs. Firstly, we will investigate the selective use of query expansion. Indeed, if the performance of the query is predicted to be poor, applying query expansion can lead to a further degradation of retrieval performance [42]. The development of query performance predictors for expert search will allow the effective, selective, application of query expansion in the expert search task.

Secondly, it is apparent that integrating evidence about the proximity of query terms to candidate name occurrences in documents can increase the performance of an expert search system, by removing false expertise evidence from candidates.

Moreover, this work can be naturally extended to integrate prior knowledge. For example, we believe that not all documents are likely to be good indicators of expertise, and furthermore that not all candidates are likely to be experts. Designing and integrating document and candidate priors with our approach could increase the retrieval effectiveness of the expert search system.

In particular, this work focuses solely on the use of the ranking of documents to predict expertise. Our future research directions will investigate the human aspect of expert search. For instance, people do not work in isolation within an organisation: they work together with other people; author documents together; and communicate with others e.g. by emails; moreover, some experts are also authorities of their field. Therefore, there is a large potential to increase the effectiveness of an expert search engine using such social network analysis, citation analysis and other related techniques. It is hoped that the new test collection—crawled from the website of an Australian government research organisation—created for the TREC 2007 Enterprise track will allow the human aspects of expert search to be tested.

REFERENCES

[1] Hertzum, M. and Pejtersen, A.M. (2000) The information-seeking practices of engineers: Searching for documents as well as for people. *Inf. Process. Manage.*, **36**, 761–778.

[2] Yimam-Seid, D. and Kobsa, A. (2003) Expert finding systems for organisations: Problem and domain analysis and the DEMOIR approach. *J. Organ. Comput. Electron. Commerce.*, **13**, 1–24.

[3] Craswell, N., de Vries, A.P. and Soboroff, I. (2006) Overview of the TREC-2005 Enterprise Track. *Proc. TREC-2005*, Gaithersburg, MD, USA, 17–24. NIST Press, Gaithersburg, MD, USA.

[4] Soboroff, I., de Vries, A.P. and Craswell, N. (2007) Overview of the TREC-2006 Enterprise Track. *Proc. TREC-2006*, Gaithersburg, MD, USA, 28–42. NIST Press, Gaithersburg, MD, USA.

[5] Macdonald, C. and Ounis, I. (2006) Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. *Proc. 2006 ACM CIKM Int. Conf. Information and Knowledge Management*, Arlington, VA, USA, November 6–11. ACM Press, NY, USA.

[6] Craswell, N., Hawking, D., Vercoustre, A.-M. and Wilkins, P. (2001) Panoptic expert: Searching for Experts not Just for Documents. *Proc. 9th Australasian World Wide Web Conf. (AusWeb03)*, Gold Coast, Australia, July 5–19. Norsearch Conference Services, Gold Coast, Australia.

[7] Liu, X., Croft, W.B. and Koll, M. (2005) Finding Experts in Community-based Question-Answering Services. *Proc. 2005 ACM CIKM Int. Conf. Information and Knowledge Management*, Bremen, Germany, October 31 to November 5. pp. 315–316. ACM Press, NY, USA.

[8] Dom, B., Eiron, I., Cozzi, A. and Zhang, Y. (2003) Graph-Based Ranking Algorithms for E-mail Expertise Analysis. *Proc. ACM SIGMOD DMKD Workshop 2003*, San Diego, CA, USA, June 13, pp. 42–48. ACM Press, NY, USA.

[9] Maybury, M., D'Amore, R. and House, D. (2001) Expert finding for collaborative virtual environments. *Commun. ACM*, **44**, 55–56.

[10] Sihn, W. and Heeren, F. (2001) Xpertfinder – Expert Finding within Specified Subject Areas through Analysis of E-mail Communication. *Proc. EuroMedia 2001: Sixth Annual Scientific Conf. Web Technology, New Media, Communications and Telematics Theory, Methods, Tools and Applications. Valencia, Spain*, April 18–20, pp. 279–283. European Technology Institute, Ostend, Belgium.

[11] Kleinberg, J.M. (1999) Authoritative sources in a hyperlinked environment. *J. ACM*, **46**, 604–632.

[12] Campbell, C.S., Maglio, P.P., Cozzi, A. and Dom, B. (2003). Expertise Identification using Email Communications. *Proc. 2003 ACM CIKM Int. Conf. Information and Knowledge Management*, New Orleans, LO, USA, November 2–8, pp. 528–531. ACM Press, NY, USA.

[13] Wang, J., Chen, Z., Tao, L., Ma, W.-Y. and Wenyin, L. (2001) Ranking User's Relevance to a Topic through Link Analysis on Web Logs. *4th ACM CIKM Int. Workshop on Web Information and Data Management (WIDM'02)*, McLean, VA, USA, pp. 49–54. ACM Press, NY, USA.

[14] McLean, A., Vercoustre, A.-M. and Wu, M. (2003) Enterprise PeopleFinder: Combining Evidence from Web Pages and Corporate Data. *Proc. 8th Australasian Document Computing Symp. (ADCS'03)*, Canberra, Australia, December 15, pp. 59–62. Commonwealth Scientific and Industrial Research Organisation, Canberra, Australia.

[15] Balog, K., Azzopardi, L. and de Rijke, M. (2006) Formal Models for Expert Finding in Enterprise Corpora. *Proc. 29th Ann. Int. ACM SIGIR Conf. Research and Development on Information Retrieval*, Seattle, WA, USA, August 6–11, pp. 43–50. ACM Press, NY, USA.

[16] Fang, H. and Zhai, C. (2007) Probabilistic Models for Expert Finding. *Proc. 29th European Conf. IR Research (ECIR 2007)*, Rome, Italy, *Lecture Notes in Computer Science*, **Vol. 4425**, pp. 418–430. Springer, Berlin, Germany.

[17] Cao, Y., Li, H., Liu, J. and Bao, S. (2005) Research on Expert Search at Enterprise Track of TREC 2005. *Proc. TREC-2005*, Gaithersburg, MD, USA. NIST Press, Gaithersburg, MD, USA.

[18] Petkova, D. and Croft, W.B. (2006) Hierarchical Language Models for Expert Finding in Enterprise corpora. *Proc. ICTAI 2006*, Washington DC, USA, pp. 599–608. IEEE Press, NY, USA.

[19] Amati, G. (2003) Probabilistic Models for Information Retrieval based on Divergence from Randomness. PhD Thesis, Department of Computing Science, University of Glasgow, UK.

[20] Hiemstra, D. (2001) Using Language Models for Information Retrieval. PhD Thesis, Centre for Telematics and Information Technology, University of Twente, The Netherlands.

[21] Dumais, S.T. and Nielsen, J. (1992) Automating the Assignment of Submitted Manuscripts to Reviewers. *Proc. 15th Ann. Int. ACM SIGIR Conf. Research and Development in Information Retrieval. Copenhagen, Denmark*, June 21–24, pp. 233–244. ACM Press, NY, USA.

[22] Balog, K. and de Rijke, M. (2006) Finding Experts and their Details in E-mail Corpora. *Proc. 15th Int. World Wide Web Conf. (WWW2006)*, Edinburgh, UK, May 23–26, pp. 1035–1036. ACM Press, NY, USA.

[23] Fox, E.A. and Shaw, J.A. (1994) Combination of Multiple Searches. *Proc. TREC-2*, Gaithersburg, MD, USA. pp. 243–252. NIST Press, Gaithersburg, MD, USA.

[24] Lee, J.H. (1997) Analyses of Multiple Evidence Combination. *Proc. 20th Ann. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Philadelphia, PA, USA. July 27–31, pp. 267–276. ACM Press, NY, USA.

[25] Shaw, J.A. and Fox, E.A. (1994) Combination of Multiple Searches. *Proc. TREC-3*, Gaithersburg, MD, USA. NIST Press, Gaithersburg, MD, USA.

[26] Montague, M. and Aslam, J.A. (2001) Relevance Score Normalisation for Metasearch. *Proc. 2001 ACM CIKM Int. Conf. Information and Knowledge Management*, Atlanta, GA, USA. November 5–10, pp. 427–433. ACM Press, NY, USA.

[27] Plachouras, V. (2006) Selective Web Information Retrieval. PhD Thesis, Department of Computing Science, University of Glasgow, UK.

[28] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. and Johnson, D. (2005) Terrier Information Retrieval Platform. *Proc. 27th European Conf. IR Research (ECIR 2005)*, *Lecture Notes in Computer Science*, **Vol. 3408**, Springer, Berlin, Germany, pp. 517–519.

[29] Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C. and Lioma, C. (2006) Terrier: A High Performance and Scalable Information Retrieval Platform. *Proc. OSIR Workshop 2006*, Seattle, WA, USA, August 12, pp. 18–25. ACM Press, NY, USA.

[30] Plachouras, V., He, B. and Ounis, I. (2005) University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. *Proc. TREC-2004*, Gaithersburg, MD, USA. NIST Press, Gaithersburg, MD, USA.

[31] Amati, G. (2006) Frequentist and Bayesian Approaches to Information Retrieval. *Advances in Information Retrieval, 28th European Conf. IR Research, ECIR 2006*, London, UK, April 10–12, *Lecture Notes in Computing Science*, **Vol. 3936**, Springer, Berlin, pp. 13–24.

[32] Macdonald, C., He, B., Plachouras, V. and Ounis, I. (2006) University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise tracks with Terrier. *Proc. TREC-2005*, Gaithersburg, MD, USA. NIST Press, Gaithersburg, MD, USA.

[33] Macdonald, C. and Ounis, I. (2006) Combining Fields in Known-item Email Search. *Proc. 29th Ann. Int. ACM SIGIR Conf. Research and Development on Information Retrieval*, Seattle, WA, USA, August 6–11, pp. 675–676. ACM Press, NY, USA.

[34] Robertson, S., Zaragoza, H. and Taylor, M. (2004) Simple BM25 Extension to Multiple Weighted Fields. *Proc. 2004 ACM CIKM Int. Conf. Information and Knowledge Management*, Washington, DC, USA, November 8–13, pp. 42–49. ACM Press, NY, USA.

[35] Zaragoza, H., Craswell, N., Taylor, M., Saria, S. and Robertson, S. (2005) Microsoft Cambridge at TREC 13: Web and Hard Tracks. *Proc. TREC-2004*, Gaithersburg, MD, USA. NIST Press, Gaithersburg, MD, USA.

[36] Macdonald, C., Plachouras, V., He, B., Lioma, C. and Ounis, I. (2006) University of Glasgow at WebCLEF 2005: Experiments in Per-field Normalisation and Language Specific Stemming. *Accessing Multilingual Information Repositories: 6th Workshop of the CrossLanguage Evaluation Forum (CLEF 2005)*, *Lecture Notes in Computer Science*, **Vol. 4022.** Springer, Berlin, Germany, pp. 898–907.

[37] Hearst, M.A. (1996) Improving Full-text Precision on Short Queries using Simple Constraints. *Symp. Document Analysis and Information Retrieval (SDAIR)*, Las Vegas, NV, USA. April 15–17.

[38] Lioma, C., Macdonald, C., Plachouras, V., Peng, J., He, B. and Ounis, I. (2007) University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise tracks with Terrier. *Proc. TREC-2006*, Gaithersburg, MD, USA. NIST Press, Gaithersburg, MD, USA.

[39] Amati, G., Carpineto, C. and Romano, G. (2002) Italian Monolingual Information Retrieval with PROSIT. Advances in Cross-Language Information Retrieval, *3rd Workshop of the Cross-Language Evaluation Forum, CLEF 2002*, Rome, Italy, September 19–20, *Lecture Notes in Computing Science*, **Vol. 2785**. Springer, Berlin, Germany, pp. 257–264.

[40] Macdonald, C. and Ounis, I. (2007) Using Relevance Feedback in Expert Search. *Proc. 29th European Conf. IR Research (ECIR 2007)*, Rome, Italy, *Lecture Notes in Computer Science*, **Vol. 4425**, Springer, Berlin, Germany, pp. 431–443.

[41] Macdonald, C. and Ounis, I. (2007) Expertise Drift and Query Expansion in Expert Search. *Proc. 2007 ACM CIKM Int. Conf. Information and Knowledge Management*, Lisbon, Portugal, November 6–8. ACM Press, NY, USA.

[42] Yom-Tov, E., Fine, S., Carmel, D. and Darlow, A. (2005) Learning to Estimate Query Difficulty: including Applications to Missing Content Detection and Distributed Information Retrieval. *Proc. 28th Ann. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Salvador, Brazil, August 15–19, pp. 512–519. ACM Press, NY, USA.