

Voting techniques for expert search

Craig Macdonald · Iadh Ounis

Received: 11 November 2006 / Revised: 19 March 2007 / Accepted: 10 June 2007
© Springer-Verlag London Limited 2007

Abstract In an expert search task, the users' 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 users' query. In this paper, we propose a novel approach for predicting and ranking candidate expertise with respect to a query, called the Voting Model for Expert Search. In the Voting Model, we see the problem of ranking experts as a voting problem. We model the voting problem using 12 various voting techniques, which are inspired from the data fusion field. We investigate the effectiveness of the Voting Model and the associated voting techniques across a range of document weighting models, in the context of the TREC 2005 and TREC 2006 Enterprise tracks. The evaluation results show that the voting paradigm is very effective, without using any query or collection-specific heuristics. Moreover, we show that improving the quality of the underlying document representation can significantly improve the retrieval performance of the voting techniques on an expert search task. In particular, we demonstrate that applying field-based weighting models improves the ranking of candidates. Finally, we demonstrate that the relative performance of the voting techniques for the proposed approach is stable on a given task regardless of the used weighting models, suggesting that some of the proposed voting techniques will always perform better than other voting techniques.

Keywords Voting · Expert finding · Expertise modelling · Expert search · Information retrieval · Ranking · Data fusion

Extended version of 'Voting for candidates: adapting data fusion techniques for an expert search task'.
C. Macdonald and I. Ounis. In *Proceedings of ACM CIKM 2006*, Arlington, VA, 2006.
doi: 10.1145/1183614.1183671.

C. Macdonald (✉) · I. Ounis
Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, Scotland, UK
e-mail: craigm@dcs.gla.ac.uk

I. Ounis
e-mail: ounis@dcs.gla.ac.uk

1 Introduction

With the advent of the vast pools of information and documents in large Enterprise organisations, collaborative 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 [14] found that engineers in product-development organisations often intertwine looking for informative documents with looking for 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 [45] identified five scenarios when people may seek an expert as a source of information to complement other sources:

1. *Access to non-documented information*—e.g., in an organisation where not all relevant information is documented.
2. *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.
3. *Leveraging on another's expertise (group efficiency)*—e.g., finding a piece of information that a relevant expert would know/find with less effort than the seeker.
4. *Interpretation need*—e.g., deriving the implications of, or understanding, a piece of information.
5. *Socialisation need*—the user may prefer that the human dimension be involved, as opposed to interacting with documents and computers.

An *expert search*¹ system is an Information Retrieval (IR) system that can aid users with their “expertise need” in the above scenarios. In contrast with classical document retrieval where documents are retrieved, an expert search system supports users in identifying informed people: The user formulates a query to represent their topic of interest to the system; the system then ranks *candidate* persons with respect to their predicted expertise about the query, using available documentary evidence.

Expert search systems make use of evidence of expertise to rank candidates. Predominantly, these systems work by using 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 [9, 11, 20, 42].

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 persons 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 [11]. However, this process is cumbersome, and may not reflect a rich vocabulary of their expertise, nor the changing interests of people. Alternatively, the expert search system can implicitly and automatically generate each profile from a corpus of documents. There are several strategies for associating documents to candidates, to generate a profile of their expertise:

- Documents containing the candidate's name: exact or partial match [9].
- Emails sent or received by the candidate [3, 6, 10].
- The candidate's homepage on the Internet or intranet and their Curriculum Vitae [26].
- Documents written by the candidate [26].

¹ In this work, we use the terms expert search and expert finding interchangeably.

- Team, group or department-level evidence [27].
- Web pages visited by the candidate [44].

In the Voting Model, each candidate's profile is modelled as a set of documents, which is built automatically from a corpus of documents (described in Sect. 4). Then we consider a *ranking of documents* with respect to the expert search query. We see each document retrieved as an implicit vote for the candidate whose profile contains that document. Therefore, expert search can be modelled as a voting problem, where the votes from documents to candidates have to be combined. We propose many techniques, called voting techniques, to aggregate the document votes into a ranking of candidates. The voting techniques are inspired by techniques from the data fusion field.

The retrieval performance of an expert search system is an important issue: If an expert search system suggests incorrect experts, then this could lead the user to contacting these people inappropriately. 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.

Expert search has been a retrieval task in the Enterprise tracks of the Text REtrieval Conferences (TREC) since 2005 [8], aiming to evaluate expert search approaches. The TREC forum provides IR researchers with a means to evaluate their retrieval systems on a shared *test collection*. Generally speaking, a test collection consists of a common corpus of documents, a series of queries (known as topics), and a corresponding set of relevance assessments. For the expert search task, the test collection consisted of a corpus, a list of experts identified in the corpus, and a list of expertise topics with corresponding relevance assessments. The retrieval performances of participating systems are evaluated by using relevance assessments to calculate precision and recall-based measures, such as Precision @ 10 (P@10), and Mean Average Precision (MAP).

To assess the usefulness of the proposed voting techniques from the Voting Model, we use the TREC W3C test collection and the TREC 2005 and 2006 Enterprise track expert search tasks. Note that the voting techniques rely on the weighting model used to generate the underlying ranking of documents. Hence, we experiment with a range of probabilistic weighting models, to determine if the performances of the voting techniques are stable across the range of weighting models experimented with. The obtained results show that the Voting Model for Expert Search is very effective compared with the results of TREC participating systems, while making no use of query or collection-specific heuristics.

The success of the Voting Model depends on the quality of the underlying ranking of documents—in particular, how high and frequently documents from relevant candidates are ranked. In order to improve the underlying ranking of documents, we further refine the representation of documents used by the retrieval system, to take the structure of documents into account. In particular, the structure of each document can be represented as separate fields during indexing and retrieval. We use content, title and anchor text of incoming hyperlinks as separate document fields during retrieval. Each document is represented by these fields. We demonstrate that applying a weighting model that uses these fields improves the performance of the voting techniques.

The structure of this paper is as follows: We review related work in Sect. 2. In Sect. 3, we describe how expert search can be modelled as a voting problem. To aggregate the votes from the documents to candidates and produce a single ranking of candidates, we propose various voting techniques. We describe our experimental setup in Sect. 4, and evaluate the proposed voting approach across a selection of document weighting models in Sect. 5. In Sect. 6, we use field-based weighting models in an expert search context, and show how this

improves the performance of the voting techniques adapted for the approach. In Sect. 7, we demonstrate that the performance of the voting techniques is stable across various weighting models and settings. Finally, we provide concluding remarks and suggestions for future work in Sect. 8.

2 Related work

There is some previous work on expert search models where candidate profiles consist of a set of documents. Having defined profiles of expertise for each candidate, an expert search system needs to accurately rank the candidate profiles in response to a user query. Craswell et al. [9] proposed concatenating the terms of all documents in each profile into “virtual documents” (i.e., one large virtual document for each candidate, containing all the expertise evidence for that candidate), 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 described below.

Liu et al. [20] 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 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 [10,26,42]. Two approaches make use of the HITS algorithm [17] to calculate “repute” and “resourcefulness” scores for each candidate [6,44]. McLean et al. [27] used a graph structure to propagate expertise evidence between members of a project team.

The advent of the expert search task in the recent TREC 2005 and 2006 Enterprise tracks has stimulated research interest in expert search [8,43]. From this, there have been three main approaches for expert search: Balog et al. [4] proposed the use of language models in expert search based on two formal models. Their first model is based on Craswell et al virtual document approach described above. The second has similarities to the proposed Voting Model, in that evidence from distinct documents in the candidate profiles are combined. Similarly, Fang and Zhai [12] applied relevance language models to the expert search task. In contrast, the probabilistic approach proposed by Cao et al. [7] and the hierarchical language models proposed by Petkova and Croft [34] 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 can only be computed using the language modelling approach.

In this work, we consider a different and novel approach to ranking expertise. In particular, we consider expert search as a voting problem. Using the ranked list of retrieved documents for the expert search query (generated using any IR approach or document search engine), 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. The problem is how to aggregate the votes for each candidate so as to produce the final ranking of experts. In Sect. 3, we show how existing data fusion techniques can be appropriately adapted to voting techniques, to combine votes for candidates. Our approach contrasts from the three main related approaches, as we are not restricted to using any one technique for creating the initial

underlying ranking of documents: Any standard retrieval technique, such as probabilistic [39], language modelling [15] or Divergence from Randomness [1] can be used to generate the document ranking.

3 Expert search as a voting problem

Data fusion techniques—also known as metasearch techniques—are used to combine separate rankings of documents into a single ranking, with the aim of improving over the performance of any constituent ranking. 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 [13] defined several data fusion techniques (CombSUM, CombMNZ, etc.), and these have been the object of much research since (for examples, see [18,29,41]).

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 introduced in Sect. 1, we see expert search as a voting problem: In this work, 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 introduce twelve voting techniques 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 $R(Q)$. Consider the simple example in Fig. 1. The ranking of documents with respect to the query has retrieved documents $D_b >_{rank} D_c >_{rank} D_a >_{rank} D_d$ in that order. Using the candidate profiles, candidate C_1 has then accumulated 2 votes, C_2 2 votes, C_3 3 votes and C_4 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 $C_3 >_{rank} C_1 >_{rank} C_2$. By using appropriate vote aggregation techniques (voting techniques), we can have different rankings of candidates. In the remainder of this paper, we introduce twelve voting techniques which are inspired by the data fusion field, and evaluate them to establish how well they model the proposed voting paradigm for expert search.

Fig. 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 ($profile(C)$)

R(Q)			profiles
Rank	Docs	Scores	
1	D_b	5.3	profile(C_1): { D_a, D_d, D_c }
2	D_c	4.2	profile(C_2): { D_b }
3	D_a	3.9	profile(C_3): { D_a, D_c, D_d }
4	D_d	2.0	profile(C_4): { D_f, D_g }

We determine the score of the candidate with respect to the query, denoted $score_cand(C, Q)$, as the aggregation of votes of all documents d that are retrieved, but which also belong to the profile of the candidate (i.e., $d \in R(Q) \cap profile(C)$). We have two central intuitions about expert search: firstly, a candidate that has written prolifically about a topic of interest (ie has 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. We consider three forms of evidence when aggregating the votes to each candidate, based on these intuitions:

- A The number of retrieved documents voting for each candidate
- B The scores of the retrieved documents voting for each candidate
- C The ranks of the retrieved documents voting for each candidate

We examine and evaluate 12 voting techniques based on known data fusion techniques, which aggregate the votes from the single ranking of documents into a single ranking of candidates, using appropriate forms of evidence.

It is of note that the proposed voting techniques are not straightforward uses of existing data fusion techniques, because in the normal application of 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 document-to-candidate associations of the candidate profiles.

3.1 Voting techniques

We now show how some established data fusion techniques can be adapted for expert search. Firstly, we examine the Reciprocal Rank (RR) data fusion technique [47] for expert search. In this data fusion technique, the rank of a document in the combined ranking is determined by the sum of the reciprocal rank received by the document in each of the individual rankings. Adapting the Reciprocal Rank technique to our approach, we define the score of a candidate's expertise as

$$score_cand_{RR}(C, Q) = \sum_{d \in R(Q) \cap profile(C)} \frac{1}{rank(d, Q)} \quad (1)$$

where $rank(d, Q)$ is the rank of document d in the document ranking $R(Q)$. Intuitively, from an expert search perspective, RR will highly rank candidates that have many of their profile documents rank near the top of the document ranking $R(Q)$. RR uses forms of evidence A & C, but with more focus on C.

In CombSUM [13]—a score aggregation technique—the score of a document is the sum of the normalised scores received by the document in each individual ranking. CombSUM can also be used in expert search. In this case, the score of a candidate's expertise is

$$score_cand_{CombSUM}(C, Q) = \sum_{d \in R(Q) \cap profile(C)} score(d, Q) \quad (2)$$

where $score(d, Q)$ is the score of the document d in the document ranking $R(Q)$, as defined by a suitable document weighting model. Intuitively, a candidate's expertise with respect to a query is scored as the sum of the relevance score of all the documents in $R(Q)$ that are voting for that candidate. CombSUM uses forms of evidence A & B. Similarly, CombMNZ [13] can be adapted for expert search:

$$score_cand_{CombMNZ}(C, Q) = \|R(Q) \cap profile(C)\| \cdot \sum_{d \in R(Q) \cap profile(C)} score(d, Q) \quad (3)$$

Table 1 Summary of the twelve voting techniques, inspired from the data fusion field, that are used in this paper

Name	Relevance score of candidate is:
Votes	$\ D(C, Q)\ $
RR	sum of inverse of ranks of docs in $D(C, Q)$
BordaFuse	sum of ($\ R(Q)\ $ - ranks of docs in $D(C, Q)$)
CombMED	median of scores of docs in $D(C, Q)$
CombMIN	minimum of scores of docs in $D(C, Q)$
CombMAX	maximum of scores of docs in $D(C, Q)$
CombSUM	sum of scores of docs in $D(C, Q)$
CombANZ	$\text{CombSUM} \div \ D(C, Q)\ $
CombMNZ	$\ D(C, Q)\ \times \text{CombSUM}$
expCombSUM	sum of exp of scores of docs in $D(C, Q)$
expCombANZ	$\text{expCombSUM} \div \ D(C, Q)\ $
expCombMNZ	$\ D(C, Q)\ \times \text{expCombSUM}$

$D(C, Q)$ is the set of documents $R(Q) \cap \text{profile}(C)$. $\|\cdot\|$ is the size of the described set

where $\|R(Q) \cap \text{profile}(C)\|$ is the number of documents from the profile of candidate C that are in the ranking $R(Q)$. This has a similar intuition to CombSUM (Eq. (2)), except that candidate with a larger number of votes are scored higher. Again, CombMNZ uses forms of evidence A & B, but places more emphasis on A than CombSUM.

Normally, in the CombSUM and CombMNZ data fusion techniques, it is necessary to normalise the scores of documents across all input rankings [29]. However, in Eqs. (2) and (3), no score normalisation is necessary: Indeed, in our case, as stressed above, only one input ranking of documents is involved, and hence the scores are all comparable.

Table 1 summarises all the voting techniques that we use and evaluate in this work. In addition to the three techniques described above, we also use: a technique that we call Votes, which simply counts the number of retrieved documents of each candidate profile; a technique inspired by BordaFuse [5], which works by linearly summing the ranks of the retrieved documents; and several other score aggregation techniques first defined by Fox and Shaw [13]. The final three voting techniques listed in the table, namely expCombSUM, expCombANZ and expCombMNZ, are slight variants of CombSUM, CombANZ and CombMNZ, respectively. In these variants, the score of each document is transformed by applying the exponential function (e^{score}), as suggested by Ogilvie and Callan [31]. Applying the exponential function boosts the scores of highly ranked documents, which, in turn, boosts the retrieval scores of candidates associated to highly ranked documents (by emphasising the most highly scored documents in evidence form B).

Other data fusion techniques could also have been considered in this work for adaptation to the Voting Model, including one based on Condorcet voting-theory [30], a technique that models score distributions [25], and a logical regression model [40]. However, in this work, due to the more complex nature of these techniques and due to the fact that their adaption to the Voting Model would be non-trivial, we focus the evaluation of our proposed voting approach on the techniques in Table 1.

It is of note that the CombSUM voting technique is very similar to the 2nd formal model proposed by Balog et al. [4]. In particular, the Voting Model can emulate Model 2 from [4] if Hiemstra’s language modelling was used to generate the initial ranking $R(Q)$.² However, the Voting Model is more general in two senses: the document ranking $R(Q)$ can be generated

² Indeed, we have conducted experiments using Hiemstra’s language modelling with CombSUM, and found that we could reproduce the results reported in [4].

using any approach including but not restricted to language modelling; and secondly there are more diverse methods of aggregating the votes than just the simple sum of scores. Indeed, in the following experiments, we will show that the voting techniques from the Voting Model can be applied using a selection of weighting models, and that various voting techniques can be successfully applied to the expert search task.

4 Experimental setting

In the following, we aim to demonstrate that voting is an effective approach for expert search and that the voting techniques applied are suitable to implement the proposed approach. As discussed above, in the Voting Model, any technique can be used to generate the underlying document ranking $R(Q)$. In our experiments, we hypothesise that the choice of an effective weighting model will have little effect on the relative ranking of voting techniques. In this case, while the weighting model can have an effect on the magnitude of the retrieval performance, the main parameter in the Voting Model is the choice of voting technique, and how well it covers the three forms of evidence described in Sect. 3. Hence we experiment using three different statistical document weighting models to generate $R(Q)$.

To evaluate our approach, we use the Expert Search tasks of the TREC 2005 and TREC 2006 Enterprise tracks. The TREC Enterprise test collection consists of 331,037 documents collected from the World Wide Web Consortium (W3C) website in 2005 [8]. 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 (www), personal homepages (people), official standards and recommendation documents, email discussion list archives (lists), a wiki (esw), and a source code repository (dev).

The W3C test collection includes a list of 1,092 candidate experts. We assess the retrieval accuracy of our expert search approach using the 50 topics of the TREC 2005 Expert Search task (EX1-EX50), and the 49 topics of the TREC 2006 task (EX52-EX104). The retrieval performance is evaluated using Mean Average Precision (MAP)—to assess the overall quality of the ranking—and Precision @ 10 (P@10), to assess the accuracy of the top-ranked candidates retrieved by the system.

As the expert search task in TREC 2005 was a pilot task, the candidate rankings for each query submitted by participating groups were not assessed as such, but instead were evaluated on their accuracy to a ground truth - in this case the W3C working group memberships. For TREC 2006, the submitted rankings from participating groups were evaluated by manual relevance assessing of each candidate for each query. For this to be achievable, systems submitted supporting documents for each candidate, which assessors used to judge each candidate (For example, systems could provide top-scoring documents from each suggested candidate's profile to justify that candidate's relevance). Therefore, because the relevance assessments for the TREC 2006 task were more 'complete', the retrieval accuracies achievable by expert search systems are far higher than for the TREC 2005 pilot task.

To generate the profile for each candidate, we use the Unix `grep` command to identify documents from the collection which contain an exact match of the candidate's full name. Note that this article improves on the CIKM version of this work by applying a candidate profile set that performs robustly on both TREC tasks [19]. This improved candidate profile set contains on average more than 12 times as many documents per candidate than that

Table 2 Breakdown of documents by subsection of the TREC W3C Collection, and of the candidate profile set used in these experiments

Subsection	Collection		Profiles	
Total	3,31,037	100%	1,37,161	100%
dev	62,509	18.9%	7,762	5.7%
esw	19,605	5.9%	3,480	2.5%
lists	1,98,394	59.9%	1,02,414	74.7%
other	3,538	1.1%	401	0.3%
people	1,016	0.3%	1,001	0.7%
www	45,975	13.9%	22,103	16.1%

applied in the CIKM version. The set of documents identified for each candidate C form their $profile(C)$. Table 2 shows a break down of the TREC W3C collection by subsection of the collection, showing the number of documents in each subsection of the collection, and how well the subsection is represented in the candidate profiles. Overall 41% of the documents in the collection are included in one or more candidate profiles. Moreover, it appears that most candidates are only represented in the corpus by the emails they have sent, because the lists subsection forms the largest part of the expertise evidence in the profiles.

In this work, we index the W3C collection and carry out all experiments using the Terrier platform [32,33]. 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 the proposed Voting Model using the twelve voting techniques listed in Table 1 with three statistically different document weighting models to generate the underlying ranking $R(Q)$. The first of these weighting models is the well-established probabilistic Okapi BM25 [39], where the relevance score of a document d for a query Q is given by

$$score(d, Q) = \sum_{t \in Q} w^{(1)} \frac{(k_1 + 1)tfn}{k + 1 + tfn} \frac{(k_3 + 1)qtf}{k_3 + qtf} \tag{4}$$

where qtf is the frequency of the query term t in the query; k_1 and k_3 are parameters, for which the default setting is $k_1 = 1.2$ and $k_3 = 1000$ [38]; $w^{(1)}$ is the idf factor, which is given by

$$w^{(1)} = \log_2 \frac{N - N_t + 0.5}{N_t + 0.5} \tag{5}$$

N is the number of documents in the whole collection. N_t is the document frequency of term t . The normalised term frequency tfn is given by

$$tfn = \frac{tf}{(1 + b) + b \cdot \frac{l}{avg_l}}, \quad (0 \leq b \leq 1) \tag{6}$$

where tf is the term frequency of the term t in document d . b is the term frequency normalisation hyper-parameter, for which the default setting is $b = 0.75$ [38]. l is the document length in tokens and avg_l is the average document length in the collection.

The remaining two weighting models tested are from the Divergence from Randomness (DFR) framework [1]. The first of these, PL2, is robust and performs particularly well for tasks requiring high early-precision [35]. For the PL2 model, the relevance score of a document d for a query Q is given by

$$score(d, Q) = \sum_{t \in Q} qtw \cdot \frac{1}{tfn + 1} \left(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn) \cdot \log_2 e + 0.5 \cdot \log_2 (2\pi \cdot tfn) \right) \quad (7)$$

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 qtf/qtf_{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:

$$tfn = tf \cdot \log_2 \left(1 + c \cdot \frac{avg_l}{l} \right), \quad (c > 0) \quad (8)$$

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 $c = 1.0$ [1].

The DLH13 document weighting model is a generalisation of the parameter-free hypergeometric DFR model in a binomial case [2, 22]. The hypergeometric model assumes that the document is a sample, and the population is from the collection. For the DLH13 document weighting model, the relevance score of a document d for a query Q is given by:

$$score(d, Q) = \sum_{t \in Q} \frac{qtw}{tf + 0.5} \cdot \left(\log_2 \left(\frac{tf \cdot avg_l}{l} \cdot \frac{N}{F} \right) + 0.5 \log_2 \left(2\pi tf \left(1 - \frac{tf}{l} \right) \right) \right) \quad (9)$$

Note that the DLH13 weighting model has no term frequency normalisation component, as this is assumed to be inherent to the model. Hence, DLH13 has no term frequency normalisation hyper-parameters that require tuning. Indeed, all variables are automatically computed from the collection and query statistics.

The BM25 and PL2 document weighting models include hyper-parameters, which can be tuned using relevance assessments to improve retrieval performance. In our experiments, we assess the performance of the voting techniques, both using the default parameter settings for each weighting model, and when, for each voting technique, the parameters of the weighting model have been empirically set to maximise MAP on the TREC 2005 task. This allows assessment of the maximum potential of the proposed approach for the TREC 2005 task, and how well the approach performs on the TREC 2006 task given realistic training data. Note again that the DLH13 model has no parameters that need to be tuned, and therefore is deployed directly in the expert search task.

5 Experimental results

Table 3 shows the retrieval performance of the proposed voting approach, using the twelve voting techniques, across three weighting models, namely BM25, PL2, and DLH13 for both the TREC 2005 and TREC 2006 expert search tasks. In these experiments, the default setting is used for BM25 and PL2 (see Sect. 4). Table 4 shows the retrieval performance on both the TREC 2005 and TREC 2006 when the term frequency hyper-parameters of the weighting models are empirically set for TREC 2005. This enables us to determine the maximum potential of each technique and weighting model on the TREC 2005 task. Moreover, it

Table 3 Performance of the 12 voting techniques for expert search, on both the TREC 2005 and TREC 2006 Enterprise track Expert Search tasks

Fusion	BM25			PL2			DLH13		
	MAP	ΔMAP	P@10	MAP	ΔMAP	P@10	MAP	ΔMAP	P@10
<i>TREC 2005</i>									
Votes	0.1725	(+23%)	0.2800	0.1616	(+15%)	0.2460	0.1612	(+15%)	0.2620
RR	0.2372>>	(+69%)	0.3560	0.2150>>	(+53%)	0.3400	0.2023>>	(+44%)	0.3140
BordaFuse	0.1875>>	(+34%)	0.2980	0.1699>	(+21%)	0.2660	0.1738>	(+24%)	0.2820
CombANZ	0.0222<<	(-84%)	0.0140	0.0230<<	(-84%)	0.0140	0.0187<<	(-87%)	0.0060
CombMED	0.1141<	(-19%)	0.1600	0.0949<<	(-32%)	0.1320	0.0997<<	(-29%)	0.1580
CombMIN	0.0505<<	(-64%)	0.1120	0.0482<<	(-66%)	0.1020	0.0575<<	(-59%)	0.1120
CombMAX	0.2379>>	(+70%)	0.3200	0.2251>>	(+61%)	0.3020	0.2179>>	(+55%)	0.2920
CombSUM	0.1781>	(+27%)	0.2840	0.1690>	(+21%)	0.2660	0.1671>	(+19%)	0.2700
CombMNZ	0.1756>	(+25%)	0.2780	0.1653>	(+18%)	0.2540	0.1634	(+17%)	0.2640
expCombANZ	0.0199<<	(-86%)	0.0080	0.0227<<	(-84%)	0.0120	0.0197<<	(-86%)	0.0060
expCombSUM	0.2281>>	(+63%)	0.3240	0.2294>>	(+64%)	0.3340	0.2173>>	(+55%)	0.3160
expCombMNZ	0.2055>>	(+47%)	0.3160	0.2120>>	(+51%)	0.3280	0.2040>>	(+46%)	0.3100
<i>TREC 2006</i>									
Votes	0.5164>>	(+51%)	0.6551	0.4885>>	(+43%)	0.5959	0.5049>>	(+48%)	0.6306
RR	0.5663>>	(+66%)	0.6551	0.5373>>	(+57%)	0.6122	0.5643>>	(+65%)	0.6816
BordaFuse	0.5425>>	(+59%)	0.6469	0.5127>>	(+50%)	0.5980	0.5316>>	(+56%)	0.6490
CombANZ	0.0096<<	(-97%)	0.0082	0.0083<<	(-98%)	0.0041	0.0082<<	(-98%)	0.0061
CombMED	0.1866<<	(-45%)	0.2143	0.2050<<	(-40%)	0.2245	0.2282<<	(-33%)	0.2388
CombMIN	0.0600<<	(-82%)	0.1245	0.0852<<	(-75%)	0.1551	0.1001<<	(-71%)	0.1531
CombMAX	0.5065>>	(+48%)	0.6061	0.5031>>	(+47%)	0.5653	0.5190>>	(+52%)	0.6245
CombSUM	0.5294>>	(+55%)	0.6510	0.5066>>	(+48%)	0.5918	0.5188>>	(+52%)	0.6388
CombMNZ	0.5234>>	(+53%)	0.6531	0.5023>>	(+47%)	0.5898	0.5133>>	(+50%)	0.6328
expCombANZ	0.0112<<	(-97%)	0.0041	0.0100<<	(-97%)	0.0041	0.0092<<	(-97%)	0.0041
expCombSUM	0.5545>>	(+63%)	0.6592	0.5399>>	(+58%)	0.6122	0.5469>>	(+60%)	0.6735
expCombMNZ	0.5520>>	(+62%)	0.6571	0.5316>>	(+56%)	0.6224	0.5502>>	(+61%)	0.6837

We use the default settings for the weighting models (see Sect. 4). Relative MAP differences from the median run of the participating groups of the TREC 2005 and TREC 2006 Enterprise tracks are shown (MAP 0.1402 and 0.3412, respectively). 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 voting technique for each weighting model and evaluation measure is highlighted in bold

allows to determine the extent that the setting for TREC 2006 can be determined using the TREC 2005 task.

Firstly, it is noticeable that the retrieval performance of the voting techniques differs between the TREC 2005 and TREC 2006 tasks. This is expected, as the relevance assessments for the TREC 2006 task are more complete, and the easier nature of this task is reflected by the increased median MAP achieved by the participating groups for TREC 2006 (TREC 2005 median MAP 0.1402; TREC 2006 median MAP 0.3412). Tables 3 and 4 also present

Table 4 Performance of the 12 voting techniques for expert search

Fusion	BM25			PL2		
	MAP	Δ MAP	P@10	MAP	Δ MAP	P@10
<i>TREC 2005</i>						
Votes	0.1742 ^{>}	(+24%)	0.2880	0.1649 ^{>}	(+18%)	0.2480
RR	0.2398 ^{>>}	(+71%)	0.3620	0.2258 ^{>>}	(+61%)	0.3560
BordaFuse	0.1908 ^{>>}	(+36%)	0.3120	0.1733 ^{>>}	(+24%)	0.2800
CombANZ	0.0238 ^{>>}	(−83%)	0.0160	0.0248 ^{<<}	(−82%)	0.0180
CombMED	0.1192 ^{<}	(−15%)	0.1600	0.1006 ^{<<}	(−28%)	0.1540
CombMIN	0.0555 ^{<<}	(−60%)	0.1100	0.0593 ^{<<}	(−58%)	0.1120
CombMAX	0.2435 ^{>>}	(+74%)	0.3220	0.2352 ^{>>}	(+68%)	0.3320
CombSUM	0.1788 ^{>>}	(+28%)	0.2920	0.1698 ^{>}	(+21%)	0.2660
CombMNZ	0.1763 ^{>>}	(+26%)	0.2880	0.1662 ^{>}	(+19%)	0.2520
expCombANZ	0.0229	(−84%)	0.0160	0.0257 ^{<<}	(−82%)	0.0180
expCombSUM	0.2288 ^{>>}	(+63%)	0.3340	0.2377 ^{>>}	(+70%)	0.3480
expCombMNZ	0.2078 ^{>>}	(+48%)	0.3300	0.2144 ^{>>}	(+53%)	0.3240
<i>TREC 2006</i>						
Votes	0.5301 ^{>>}	(+56%)	0.6510	0.4920 ^{>>}	(+44%)	0.6184
RR	0.5560 ^{>>}	(+63%)	0.6490	0.5414 ^{>>}	(+59%)	0.6184
BordaFuse	0.5498 ^{>>}	(+61%)	0.6612	0.5146 ^{>>}	(+51%)	0.5959
CombANZ	0.0092 ^{>>}	(−97%)	0.0041	0.0094 ^{<<}	(−97%)	0.0082
CombMED	0.2046 ^{<<}	(−40%)	0.2388	0.1638 ^{<<}	(−52%)	0.2041
CombMIN	0.0755 ^{<<}	(−78%)	0.1347	0.0893 ^{<<}	(−74%)	0.1265
CombMAX	0.5128 ^{>>}	(+50%)	0.6020	0.4957 ^{>>}	(+45%)	0.5694
CombSUM	0.5399 ^{>>}	(+58%)	0.6510	0.5043 ^{>>}	(+48%)	0.5918
CombMNZ	0.5347 ^{>>}	(+57%)	0.6510	0.4996 ^{>>}	(+46%)	0.5837
expCombANZ	0.0108 ^{<<}	(−97%)	0.0041	0.0123	(−96%)	0.0082
expCombSUM	0.5476 ^{>>}	(+60%)	0.6612	0.5325 ^{>>}	(+56%)	0.6061
expCombMNZ	0.5561 ^{>>}	(+63%)	0.6633	0.5335 ^{>>}	(+56%)	0.6184

The parameter of each weighting model (b or c) is empirically tuned to maximise MAP on the TREC 2005 data (see Sect. 4). Relative MAP differences from the median run of the participating groups of the TREC 2005 and TREC 2006 expert search tasks are shown (MAP 0.1402 and 0.3412 respectively): the notations are the same as in Table 3. DLH13 is not included as it does not have any parameters that need tuning

the statistical significance of results, when compared to the median run of all participants of each TREC year, using the Wilcoxon Matched-Pairs Signed-Rank test.³

In their default settings, the relative retrieval performance of the voting techniques is overall consistent across the three weighting models and two tasks. Moreover, as expected, retrieval accuracy is generally enhanced in Table 4 over Table 3. In particular for TREC 2005, all voting techniques benefit when the weighting model is tuned. However, this tuning does not always result in an increase in performance for the TREC 2006 task. In terms of weighting models, BM25 outperforms PL2 and DLH13 on the TREC 2005 task, while

³ We do not have access to the P@10 of the median run for either year.

DLH13 and BM25 are roughly comparable for TREC 2006, even when BM25 has been tuned as per Sect. 4.

Most of the voting techniques lead to a clear increase in performance over the median runs. In particular, applying either CombSUM, CombMAX, CombMNZ or the exponential variants always results in a statistically significant increase in MAP from the baseline (except CombMNZ using DLH13 for TREC 2005). However, by examining Tables 3 and 4 in detail, we can make a number of observations concerning the voting techniques. Firstly, the Votes technique, which simply counts the number of document votes for each candidate, shows good performance. The rank-based techniques, RR and BordaFuse, both perform well across the three weighting models. Note the good performance of RR on P@10 for all weighting models and tasks. RR highly scores candidate profiles that have documents occurring at the very top of the ranking, suggesting that the highly ranked documents contribute more to the expertise of a candidate, and should be considered as stronger votes (evidence form C). In contrast, the BordaFuse technique assigns linearly scaled votes across the document ranking to candidates, without emphasising the strength of votes by top ranked documents, which, as expected slightly hinders the retrieval performance compared to the high-rank focused RR.

On the other hand, the score-based voting techniques have varying effectiveness, depending on the exact combination of evidence applied. CombMAX works extremely well for TREC 2005, but is not as effective as other voting techniques on the TREC 2006 task. CombMAX ranks candidates by their most highest ranked profile documents, without taking into account the number of votes for a candidate profile. Its relatively strong performance demonstrates that the most highly ranked document for each candidate is a good indicator of its expertise, without taking into account any additional votes from $R(Q)$.

The reasonably good effectiveness of CombSUM and CombMNZ mirrors previous studies of their use in classical data fusion [28,29]. In particular, for expert search, both take into account the strength of the document votes, i.e., the magnitude of the score for each retrieved document of the candidate's profile. Moreover, CombMNZ adds a second component, the number of votes for each candidate, explaining its slight overall performance edge over CombSUM.

The high performance on both tasks of the exponential variants of CombSUM and CombMNZ, expCombSUM and expCombMNZ, is expected, and can be explained in that the exponential function increases the scores of the highly-scored documents more than the low-scored documents, increasing the strength of their votes. Hence a candidate with many weak votes will be lower ranked, while a candidate with fewer stronger votes will be higher ranked. In terms of MAP, expCombSUM and expCombMNZ outperform all other techniques across all weighting models for TREC 2006, and are only beaten by CombMAX for BM25 and DLH13 on the TREC 2005 task.

The CombANZ, CombMIN, and expCombANZ techniques do not perform well on either task, because they focus too much on the low scoring documents of each profile, which, intuitively, are not good indicators of expertise. Interestingly, taking the median of the scored documents in a profile (CombMED) outperforms taking the average (CombANZ). This finding is inconsistent with previous experiments using these techniques for classical data fusion [13]. A possible interpretation is that the denominator component of CombANZ impairs the evidence in the distribution of the candidate scores.

From the results, 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, evidence form A), and the relative magnitude of the retrieval scores in the candidate's profile (strong votes, evidence form B). The techniques Votes and CombMAX exemplify

each of these indicators respectively. Moreover, the robustly performing CombMNZ and expCombMNZ techniques combines both these indicators. The rank-based voting technique RR also combines votes with a focus on highly-scored documents (in this case manifested by high-ranks, evidence form C).

Overall, we have shown that the proposed Voting Model, using voting techniques inspired by data fusion techniques, can be effectively applied to expert search. Indeed, the best performing runs in Table 3 would rank as high as the top three participants in TREC 2005 Enterprise track runs, and as high as the top two participants in the TREC 2006 Enterprise track runs, without using any collection-specific heuristics, nor any parameter tuning. The training for Table 4, as expected, enhances the retrieval performance for the TREC 2005 task, however it does not enhance the performance of all techniques on the TREC 2006 task. As discussed in Sect. 4, this is probably due to the fact that the TREC 2005 task (a pilot study) was evaluated in a different manner to the TREC 2006 task, and hence is not a representative training set for TREC 2006.

The proposed voting techniques are low-cost, and are easy to deploy in an operational Enterprise setting. The proposed voting techniques perform robustly using a selection of statistically different document weighting models to generate $R(Q)$. Moreover, the results of the experiments show that the voting techniques are overall consistent across the different weighting models, showing that the main parameter of the Voting Model is not the weighting model used to generate $R(Q)$, but the voting technique applied.

In the next section, we will show that we can significantly improve on the performance of the proposed approach by improving the quality of the underlying document ranking.

6 Use of document structure

The quality of the underlying document ranking returned by the IR system in response to the expert search query is important to the success of the proposed voting approach. If the quality of the document ranking is improved, then we naturally hypothesise that the ranking of candidates will also likely improve.

Experiments in the Web and Enterprise tracks have shown that when the structure of documents (fields) is taken into account by a retrieval system, then the retrieval performance can be improved [21, 46]. For example, a Web document can be represented by three fields: the body, the title, and the anchor text of its incoming hyperlinks. Robertson et al. [37] and Plachouras and Ounis [36] then showed improved retrieval effectiveness in Web tasks when the contribution of each field to the document ranking was controlled by the use of weights. We hypothesise that the retrieval performance of our proposed voting approach for expert search could be improved if the quality of the underlying document ranking is increased. In this section, we use field-based document weighting models that account for the document structure to improve the quality of document ranking, and assess the effect on the proposed voting approach for expert search.

In the remainder of this section, we use variations of the BM25 and PL2 models that take fields into account. Next, we apply these fields-based document weighting models in our voting approach for expert search.

6.1 Field-based document weighting models

The BM25 weighting model can be extended into a field-based document weighting model, BM25F [46], by replacing Eq. (6) with

Table 5 The number of tokens (#tokens) and average document length (avg_l) of each field of the W3C collection

Field	#tokens	avg_l
body	302,447,426	913.64
anchor text	25,048,853	75.67
title	4,037,394	12.20
total	331,533,673	1001.51

$$tf_n = \sum_f w_f \cdot \frac{tf_f}{(1 - b_f) + b_f \cdot \frac{l_f}{avg_l_f}}, \quad (0 \leq b_f \leq 1) \tag{10}$$

where tf_f is the term frequency of term t in field f of document d , l_f is the length of field f in d , and avg_l_f is the average length of documents in f . The normalisation applied to terms from field f can be controlled by the field hyper-parameter, b_f , while the contribution of the field is controlled by the weight w_f .

Similarly, we can extend the PL2 document weighting model to handle fields. The so-called Normalisation 2 (Eq. (8)) is replaced with *Normalisation 2F* [21], so that the normalised term frequency tf_n corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f :

$$tf_n = \sum_f \left(w_f \cdot tf_f \cdot \log_2 \left(1 + c_f \cdot \frac{avg_l_f}{l_f} \right) \right), \quad (c_f > 0) \tag{11}$$

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 . Having defined Normalisation 2F, the PL2 model (Eq. (7)) can be extended to PL2F by using Normalisation 2F.

In the following experiments, we index the body, anchor text and titles of documents as separate fields using Terrier. Table 5 shows the breakdown of the statistics of each field on the W3C collection. As in Sect. 4, we remove stopwords and apply the first two steps of Porter’s stemming algorithm. We again train the parameter settings using the 50 expert search task topics from TREC 2005 Enterprise track, and report the obtained retrieval effectiveness on the TREC 2005 and TREC 2006 Expert Search tasks. Similar to Sect. 5, this allows us to assess the maximum contributions a field-based model can have using the TREC 2005 task, and how well it can work in a realistic train/test setting using the TREC 2006 task.

We follow [46] to optimise the involved hyper-parameter values and the field weights as follows. Firstly, for each voting technique, the hyper-parameter for each field (c_f or b_f) is tuned using a simulated annealing. During this, the w_f of that field is set to 1, and the weights of the other fields are set to 0. Once good hyper-parameter values have been found, a three-dimensional simulated annealing is used to find the optimal w_f values. Contrary to Zaragoza et al. [46], who assumed that the body field should have a weight of 1, we do not assume any constraints on the weights of any fields. Moreover, as the training is done for each voting technique individually, there is a total of 144 parameters. Hence, for reasons of brevity, we have chosen not to provide the settings obtained.

6.2 Experiments and results

Table 6 shows the retrieval performance of the proposed voting approach using the twelve voting techniques, and the BM25F and PL2F field-based weighting models. Retrieval performance is shown on both the TREC 2005 and TREC 2006 Expert Search tasks. From

Table 6 Performance of the 12 voting techniques for expert search, when used with two field-based document weighting models

Fusion	BM25F				PL2F			
	MAP	Δ MAP	P@10	Δ P@10	MAP	Δ MAP	P@10	Δ P@10
<i>TREC 2005</i>								
Votes	0.2007 \gg	(+15%)	0.3180 $>$	(+13%)	0.1826 $>$	(+11%)	0.2840 $>$	(+15%)
RR	0.2862 \gg	(+19%)	0.4240	(+17%)	0.2668 $>$	(+18%)	0.3920	(+10%)
BordaFuse	0.2162 $>$	(+13%)	0.3460	(+11%)	0.1935	(+12%)	0.3060	(+9%)
CombANZ	0.0264	(+11%)	0.0160	(0%)	0.0258	(+4%)	0.0200	(+11%)
CombMED	0.1456 $>$	(+22%)	0.2000 $>$	(+25%)	0.1403 $>$	(+39%)	0.2080 $>$	(+35%)
CombMIN	0.0846 \gg	(+52%)	0.1620 \gg	(+47%)	0.0893 $>$	(+50%)	0.1640	(+46%)
CombMAX	0.2917 $>$	(+20%)	0.4160 \gg	(+29%)	0.2801 \gg	(+19%)	0.4040	(+21%)
CombSUM	0.2128 \gg	(+19%)	0.3440 \gg	(+18%)	0.1913	(+12%)	0.3140 $>$	(+18%)
CombMNZ	0.2075 \gg	(+18%)	0.3420 \gg	(+19%)	0.1803 $>$	(+8%)	0.2820 $>$	(+12%)
expCombANZ	0.0263	(+15%)	0.0200	(+25%)	0.0264	(+3%)	0.0180	(0%)
expCombSUM	0.2892 \gg	(+26%)	0.4240 \gg	(+27%)	0.2891 \gg	(+22%)	0.4180 $>$	(+20%)
expCombMNZ	0.2740 \gg	(+31%)	0.4100 \gg	(+24%)	0.2726 $>$	(+27%)	0.3940 $>$	(+22%)
<i>TREC 2006</i>								
Votes	0.5139	(−3%)	0.6367	(−2%)	0.4871	(−1%)	0.6224	(+6%)
RR	0.5712 $>$	(+3%)	0.6878 \gg	(+6%)	0.5385	(−1%)	0.6469	(+5%)
BordaFuse	0.5277	(−4%)	0.6408	(−3%)	0.5049	(−2%)	0.6163	(+3%)
CombANZ	0.0097	(+6%)	0.0041	(0%)	0.0088	(−6%)	0.0041	(−50%)
CombMED	0.2434 \gg	(+19%)	0.2878 $>$	(+21%)	0.1975 \gg	(+20%)	0.2612 $>$	(+28%)
CombMIN	0.0868	(+15%)	0.1224	(−9%)	0.0960	(+7%)	0.1592	(+26%)
CombMAX	0.5073	(−1%)	0.5796	(−4%)	0.5021	(+1%)	0.5857	(+2%)
CombSUM	0.5184	(−3%)	0.6367	(−2%)	0.4921	(−2%)	0.6204	(+5%)
CombMNZ	0.5275	(−1%)	0.6449	(−1%)	0.5104 $>$	(+2%)	0.6286 $>$	(+7%)
expCombANZ	0.0118	(+10%)	0.0122	(+200%)	0.0121	(−1%)	0.0143	(+75%)
expCombSUM	0.5484	(0%)	0.6531	(−2%)	0.5355	(+1%)	0.6306	(+4%)
expCombMNZ	0.5585	(0%)	0.6551	(−1%)	0.5515	(+3%)	0.6571	(+6%)

Relative improvements over the equivalent entry in Table 4 are shown. Statistically significant improvements at $p \leq 0.05$ are denoted $>$; and significant improvements at $p \leq 0.01$ are denoted \gg

the obtained results, we can see that the use of fields has led to an overall improvement in effectiveness compared to Table 4, for both MAP and P@10. In particular, for a large number of cases on the TREC 2005 task, there is a statistically significant improvement over the corresponding entry in Table 4. For example, expCombMNZ shows a MAP of 0.2740 for BM25F, compared to only 0.2078 for BM25—a statistically significant improvement of 31%. The relative performance of the voting techniques remains mostly consistent with the previous experiments.

By introducing fields into the underlying document ranking technique, we are able to rank highly more documents that are good indicators of expertise for the query. In general, this increased quality of the document ranking leads to an increased performance by the proposed voting approach on all voting techniques. In particular, performance is increased for all

voting techniques on the TREC 2005 task. This infers that when a document ranking can be appropriately fine-tuned to rank higher more documents associated with relevant candidates, then the voting techniques can take this into account, enhancing retrieval effectiveness.

For the TREC 2006 task, as noted in Sect. 5 and expected, training using the TREC 2005 topics did not provide the same increases in retrieval performance. While some of the mediocre or poor techniques can be improved (eg CombANZ or CombMED), it appears that training using the TREC 2005 data does not give the correct evidence about the relative importance of the underlying document structure for the high performing voting techniques. This is perhaps due to the different schemes for relevance assessing between the two tasks, as discussed in Sect. 4. Among the well-performing techniques on the TREC 2006 task, RR shows the most improvement for BM25F (+3% MAP and +6% P@10), while expCombMNZ shows the same margin of improvements for PL2F. This underlines the stability of these techniques, which always perform extremely well across all weighting models and tasks, because they use appropriate mixes of the forms of evidence defined in Sect. 3.

The obtained results are in the top two most effective techniques reported for both the TREC 2005 and TREC 2006 Expert Search tasks [8,43], while not using any collection-dependent means, such as focusing on the pages containing many of the answers. This is very encouraging, as our approach could be extended to include other factors, such as document and candidate priors, degree of association between documents and profiles, and the proximity of query terms to candidate names etc. Moreover, it can be shown that query expansion can be applied, in either a document-centric or a candidate-centric manner, to improve the quality of $R(Q)$ and hence improve the accuracy of generated candidate ranking [23]. Our voting approach is general, and can easily be applied in an Enterprise setting independent of the collection and its structure.

7 Stability of voting techniques

As mentioned in the hypothesis in Sect. 4, our voting approach relies on the voting techniques to provide a suitable aggregation of the votes by documents for candidates. In this section, we test the robustness and stability of the voting techniques used in the proposed approach, across the seven weighting schemes and settings applied in Tables 3, 4 and 6. Using MAP as the evaluation measure, for each weighting model, we order the adapted voting techniques from the best to the worst performing. For example, from the TREC 2005 task in Table 6, the ordering for BM25F has CombMAX as the best and expCombANZ as the worst voting technique.

Figures 2 and 3 show the retrieval performance in terms of MAP of each voting technique across all weighting schemes and settings, for the TREC 2005 and TREC 2006 tasks, respectively. From Fig. 2, we can see that for the TREC 2005 task, there are three distinct groups of fusion techniques—the best ($\text{MAP} \geq 0.20$); the middle range (0.17–0.20 MAP), and the rest (MAP 0.16 and below). For the easier TREC 2006 task (Fig. 3), we see that more of the techniques give roughly comparable performance ($\text{MAP} \geq 0.50$). Moreover, from Fig. 2, we can observe that the lines joining the performance of each fusion technique on the different weighting models are mostly parallel. This suggests that the relative performance of the voting techniques is stable on this task, since their ordering remains unchanged regardless of the used weighting model. In Fig. 3, more techniques have comparable performance, and there are more swaps between their relative performances. However, across the two figures, it is apparent that the choice of weighting model is not as important as the choice of voting technique.

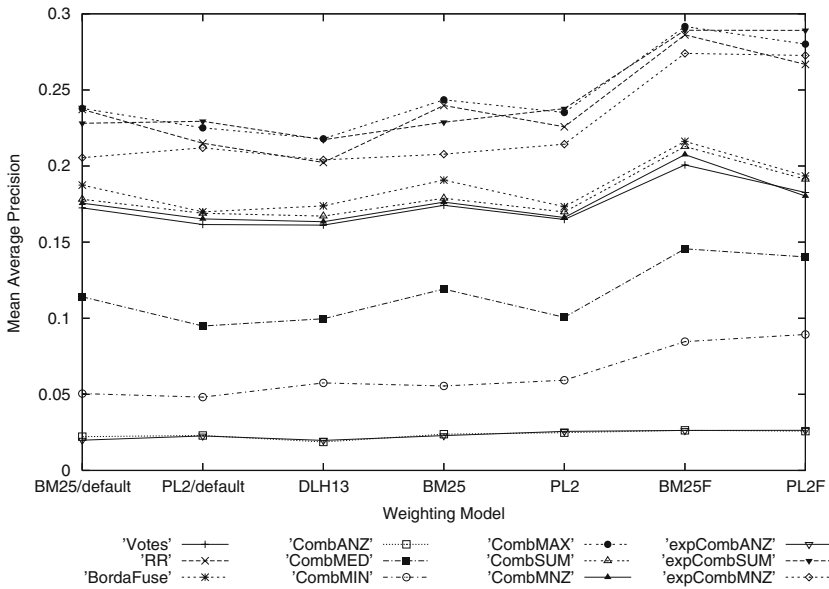


Fig. 2 The performance of voting techniques plotted across various weighting models on the TREC 2005 task. BM25/default, PL2/default and DLH13 are the default settings of those document weighting models (Table 3); BM25 and PL2 are from Table 4; and BM25F and PL2F are from Table 6

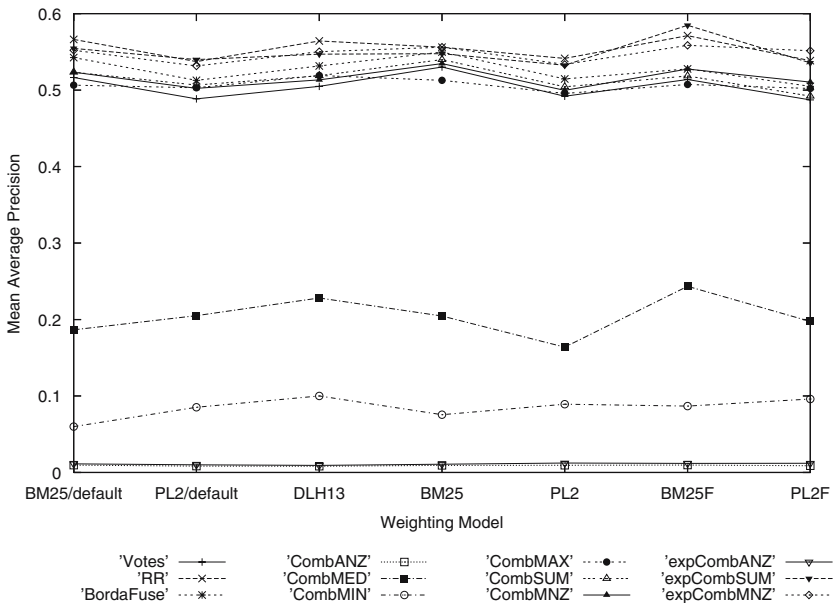


Fig. 3 The performance of voting techniques plotted across various weighting models on the TREC 2006 task. Notation as in Fig. 2

To check that the voting techniques are indeed stable across different document weighting models, we can use a statistical concordance measure. Kendall's [16] W of concordance measures the concordance of n items over a set of m rankings. W is in the range $W \in [0,1]$,

where $W = 1$ are identical rankings, and $W = 0$ are completely disagreeing rankings. We use Kendall's W to measure the concordance of the seven weighting models and the twelve voting techniques for each TREC task. Moreover, using Table 6 in [16], we can calculate the significance of such concordance. For Fig. 2 (TREC 2005), the concordance is $W = 0.82$, while for Fig. 3 (TREC 2006), the concordance is $W = 0.36$ (both values are significant at $p \leq 0.01$).

Hence for the TREC 2005 task, we can see that there is a statistically significant concordance between the rankings of the voting techniques, showing that the relative performance of the various techniques are indeed very stable, regardless of the weighting model used. For the TREC 2006 task, the concordance is weaker (but still significant), due to the similarity of many of the techniques in the $\text{MAP} \geq 0.50$ range, and hence the increased number of swaps compared to the TREC 2005 rankings. We can conclude that although we cannot predict the absolute performance of each voting technique on an arbitrary weighting model, we can conclude that some techniques are always more likely to perform better than others. These are the ones which model the important sources of expertise evidence, A, B & C, introduced in Sect. 3.

8 Conclusions and future work

In this paper, we proposed that expert search can be seen as a voting problem, where documents from an initial ranking $R(Q)$ vote for the candidates with relevant expertise. We call this the Voting Model for Expert Search, and proposed twelve voting techniques for our proposed approach, inspired by the data fusion field. Three statistically different document weighting models were tested, to assess the effectiveness and stability of the data fusion techniques in our approach. The evaluation was conducted in the context of the expert search tasks of the TREC 2005 and TREC 2006 Enterprise tracks.

The results in Sect. 5 show that our proposed approach is effective when using appropriate adapted voting techniques that use the forms of evidence (A, B or C) introduced in Sect. 3, namely the number of votes, the scores of associated documents, and the ranks of associated documents, respectively. While the techniques have varying degrees of performance, some of them consistently outperform others, regardless of the applied document weighting model. The most successful techniques usually integrate the most highly ranked or scored documents of the profile (forms of evidence B or C—strong votes), and the number of retrieved documents from the profile (evidence form A—number of votes). Our experimental results also suggest that the quality of the underlying ranking of documents is important in enhancing the retrieval performance of the expert search system. Indeed, we showed that a recent Web IR technique—i.e., the use of fields to represent the document structure—very often leads to marked performance improvements, which are sometimes significant (see Sect. 6).

We also demonstrate that the relative performance of the voting techniques is stable across the various weighting models and settings applied. Indeed, when the voting techniques are compared across various weighting models, the concordance of their relative performance rankings shows that some of the data fusion techniques are always more likely to outperform others.

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. Moreover, we have successfully deployed an expert search system based on these techniques [22].

Comparing the Voting Model with other existing techniques, we note that in certain circumstances, the Voting Model is similar to the language modelling formal model of Balog

et al. [4]. In particular, the Voting Model can emulate the language modelling approach Model 2 of Balog et al. if CombSUM is used to combine the scores of documents ranked by the language models of Hiemstra [15]. In contrast, the voting approach is more general, as any technique can be used to rank the documents in $R(Q)$ —it is not restricted to language modelling approach. Moreover, as we have shown, other more effective techniques exist for combining the votes of the documents in $R(Q)$ (eg expCombSUM or expCombMNZ).

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 within our approach could increase the retrieval effectiveness of the expert search system. Moreover, we are keen to evaluate our proposed approach on another expert search test collection. Finally, as we work towards our overall objective of having a better understanding of expert search task, we would like to investigate in more detail how the document ranking $R(Q)$ and the final ranking of candidates correlate, and how the ranking $R(Q)$ can be directly evaluated so that we have a better understanding of the connection between the performance of $R(Q)$ and the effectiveness of the expert search system.

References

1. Amati G (2003) Probabilistic models for information retrieval based on divergence from randomness. PhD thesis, University of Glasgow, Glasgow, UK
2. Amati G (2006) Frequentist and Bayesian approach to information retrieval. In: Lalmas M, MacFarlane A, Rügger S et al (eds) Proceedings of ECIR 2006. Lecture Notes in Computer Science, vol 3936. Springer, London, pp 13–24. doi: 10.1007/11735106_3
3. Balog K, de Rijke M (2006) Finding experts and their details in e-mail corpora. In: Carr L, De Roure D, Iyengar A et al (eds) Proceedings of WWW 2006. ACM Press, Edinburgh, pp 1035–1036. doi: 10.1145/1135777.1136002
4. Balog K, Azzopardi L, de Rijke M (2006) Formal models for expert finding in enterprise corpora. In: Efthimiadis E, Dumais S, Hawking D et al (eds) Proceedings of ACM SIGIR 2006. ACM Press, Seattle, pp 43–50. doi: 10.1145/1148170.1148181
5. Aslam JA, Montague M (2001) Models for metasearch. In: oft WB, Harper D, Kraft D et al (eds) Proceedings of ACM SIGIR 2001. ACM Press, New Orleans, pp 276–284. doi: 10.1145/383952.384007
6. Campbell CS, Maglio PP, Cozzi A, et al (2003) Expertise identification using email communications. In Proceedings of ACM CIKM 2003. ACM Press, New Orleans, pp 528–531. doi: 10.1145/956863.956965
7. Cao Y, Li H, Liu J et al (2005) Research on expert search at enterprise track of TREC 2005. In: Proceedings of TREC-2005. NIST, Gaithersburg
8. Craswell N, de Vries AP, Soboroff I (2005) Overview of the TREC-2005 enterprise track. In: Proceedings of TREC-2005. NIST, Gaithersburg
9. aswell N, Hawking D, Vercoustre A-M et al (2001) Panoptic expert: searching for experts not just for documents. In: Ausweb Poster Proceedings, Queensland, Australia
10. Dom B, Eiron I, Cozzi A et al (2003) Graph-based ranking algorithms for e-mail expertise analysis. In: Zaki MJ, Aggarwal C (eds) Proceedings of ACM SIGMOD DMKD Workshop 2003. ACM Press, San Diego, pp 42–48
11. Dumais ST, Nielsen J (1992) Automating the assignment of submitted manuscripts to reviewers. In: Belkin NJ, Ingwersen P, Pejtersen AM (eds) Proceedings of ACM SIGIR 1992, Copenhagen, Denmark, pp 233–244. doi: 10.1145/133160.133205
12. Fang H, Zhai C (2007) Probabilistic models for expert finding. In: Amati G, Carpineto C, Romano G (eds) Proceedings of ECIR 2007. Lecture Notes in Computer Science vol 4425. Springer, Rome, pp 418–430. doi: 10.1007/978-3-540-71496-5_38
13. Fox EA, Shaw JA (1994) Combination of multiple searches. In: Proceedings of TREC-2. NIST, Gaithersburg
14. Hertzum M, Pejtersen AM (2000) The information-seeking practises of engineers: searching for documents as well as for people. Inf Process Manage 36(5):761–778. doi: 10.1016/S0306-4573(00)00011-X
15. Hiemstra D (2001) Using language models for information retrieval. PhD thesis, University of Twente, The Netherlands

16. Kendall MG (1955) Rank correlation methods, 2nd edn. Charles Griffin, London
17. Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. *J ACM* 46(5):604–632. doi: 10.1145/324133.324140
18. Lee JH (1997) Analyses of multiple evidence combination. In: Belkin NJ, Willett P, Narasimhalu AD (eds) *Proceedings of ACM SIGIR 1997*, ACM Press, Philadelphia, pp 267–276. doi: 10.1145/258525.258587
19. Lioma C, Macdonald C, Plachouras V, et al (2007) University of Glasgow at TREC 2006: experiments in terabyte and enterprise tracks with terrier. In: *Proceedings of TREC 2006*. NIST, Gaithersburg
20. Liu X, oft WB, Koll M (2005) Finding experts in community-based question-answering services. In: Schek H-J, Fuhr N, Chowdhury A (eds) *Proceedings of ACM CIKM 2005*, ACM Press, Bremen, pp 315–316. doi: 10.1145/1099554.1099644
21. Macdonald C, He B, Plachouras V, et al (2006) University of Glasgow at TREC 2005: experiments in terabyte and enterprise tracks with terrier. In: *Proceedings of TREC-2005*. NIST, Gaithersburg
22. Macdonald C, Ounis I (2006) Searching for expertise using the terrier platform. In: Efthimiadis E, Dumais S, Hawking D et al (eds) *Proceedings of ACM SIGIR 2006*. ACM Press, Seattle WA, pp 732. doi: 10.1145/1148170.1148345
23. Macdonald C, Ounis I (2007) Using relevance feedback in expert search. In: Amati G, Carpineto C, Romano G (eds) *Proceedings of ECIR 2007*. Lecture Notes in Computer Science, vol 4425. Springer, Rome, pp 418–430. doi: 10.1007/978-3-540-71496-5_39
24. Macdonald C, Plachouras V, He B, Lioma C, Ounis I (2006) University of Glasgow at WebCLEF 2005: experiments in per-field normalisation and language specific stemming. In: Peters C, Gey FC, Gonzalo et al (eds) *Proceedings of CLEF workshop 2005*. Lecture Notes in Computer Science, vol 4022. Springer, Vienna, Austria, pp 898–907. doi: 10.1007/11878773_100
25. Manmatha R, Rath T, Feng F (2001) Modelling score distributions for combining the outputs of search engines. In: oft WB, Harper D, Kraft D et al (eds) *Proceedings of ACM SIGIR 2001*. ACM Press, New Orleans LA, pp 267–275. doi: 10.1145/383952.384005
26. Maybury M, D'Amore R, House D (2001) Expert finding for collaborative virtual environments. *Commun ACM* 44(12):55–56. doi: 10.1145/501338.501343
27. McLean A, Vercoustre A-M, Wu M (2003) Enterprise PeopleFinder: combining evidence from Web pages and corporate data. In: Hawking D, Bruza P, Thom J (eds) *Proceedings of the 8th Australasian Document Computing Symposium (ADCS'03)*
28. Montague M, Aslam JA (2001) Metasearch consistency. In: oft WB, Harper D, Kraft D et al (eds) *Proceedings of ACM SIGIR 2001*. ACM Press, New Orleans, pp 386–387. doi: 10.1145/383952.384030
29. Montague M, Aslam JA (2001) Relevance score normalization for metasearch. In: *Proceedings of ACM CIKM 2001*. ACM Press, Atlanta, pp 427–433. doi: 10.1145/502585.502657
30. Montague M, Aslam JA (2002) Condorcet fusion for improved retrieval. In *Proceedings of ACM CIKM 2002*. ACM Press, McLean, pp 538–548. doi: 10.1145/584792.584881
31. Ogilvie P, Callan J (2003) Combining document representations for known-item search. In: Clarke C, Cormack G, Callan J et al (eds) *Proceedings of ACM SIGIR 2003*. Toronto, Canada, pp 143–150. doi: 10.1145/860435.860463
32. Ounis I, Amati G, Plachouras V et al (2005) Terrier Information Retrieval Platform. In: Losada D, Fernández-Luna JM (eds) *Proceedings of ECIR 2005*. Lecture Notes in Computer Science, vol 3408. Springer, Santiago de Compostela, pp 517–519. doi: 10.1007/b107096
33. Ounis I, Amati G, Plachouras V et al (2006) Terrier: a high performance and scalable information retrieval platform. In: Beigbeder M, Buntine W, Gen Yee W (eds) *Proceedings of the OSIR Workshop 2006*. ACM Press, Seattle, pp 18–25
34. Petkova D, oft WB (2006) Hierarchical language models for expert finding in enterprise corpora. In: Lu CT, Bourbakis NG (eds) *Proceedings of ICTAI 2006*. IEEE, Washington, DC, pp 599–608. doi: 10.1109/ICTAI.2006.63
35. Plachouras V, He B, Ounis I (2004) University of Glasgow at TREC2004: experiments in Web, robust and terabyte tracks with terrier. In: *Proceedings of TREC-2004*. NIST, Gaithersburg
36. Plachouras V, Ounis I (2007) Multinomial randomness models for retrieval with document fields. In: Amati G, Carpineto C, Romano G (eds) *Proceedings of ECIR 2007*. Lecture Notes in Computer Science, vol 4425. Springer, Rome, pp 28–39. doi: 10.1007/978-3-540-71496-5_6
37. Robertson SE, Zaragoza H, Taylor M (2004) Simple BM25 extension to multiple weighted Fields. In: Gravano L, Zhai CX, Herzog O (eds) *Proceedings of ACM CIKM 2004*. ACM Press, Washington, DC, pp 42–49. doi: 10.1145/1031171.1031181
38. Robertson SE, Walker S, Hancock-Beaulieu M, et al (1995) Okapi at TREC-4. In: *Proceedings of TREC-4*. NIST, Gaithersburg
39. Robertson SE, Walker S, Hancock-Beaulieu M, et al (1992) Okapi at TREC. In: *Proceedings of TREC-1*. NIST, Gaithersburg

40. Savoy J, Calvé AL, Vrajitoru D (1997) Report on the TREC-5 experiment: data fusion and collection fusion. In: Proceedings of TREC-5. NIST, Gaithersburg, MD
41. Shaw JA, Fox EA (1994) Combination of multiple searches. In: Proceedings of TREC-3. NIST Gaithersburg
42. Sihn W, Heeren F (2001) Xpertfinder—expert finding within specified subject areas through analysis of E-mail communication. In: Proceedings of Euromedia 2001, Valencia, Spain, pp 279–283
43. Soboroff I, de Vries AP, aswell N (2006) Overview of the TREC-2006 enterprise track. In: Proceedings of TREC-2006. NIST, Gaithersburg
44. Wang J, Chen Z, Tao L, Ma WY, Wenyin L (2002) Ranking user's relevance to a topic through link analysis on web logs. In: Proceedings of WIDM 2002 workshop, McLean, VA, pp 49–54
45. Yimam-Seid D, Kobsa A (2003) Expert finding systems for organizations: problem and domain analysis and the DEMOIR approach. *J Organizat Comput and Elec Commerce* 13(1):1–24
46. Zaragoza H, aswell N, Taylor M, et al (2004) Miosoft Cambridge at TREC-13: Web and HARD tracks. In: Proceedings of TREC-2004. NIST, Gaithersburg
47. Zhang M, Song R, Lin C, et al (2002) Expansion-based technologies in finding relevant and new information: THU TREC2002: Novelty Track experiments. In: Proceedings of TREC-2002. NIST, Gaithersburg

Authors Biography



Craig Macdonald is a PhD research student at the Department of Computing Science at the University of Glasgow. His research interests includes Information Retrieval in Enterprise, Web and Blog settings. He is a co-ordinator of the Blog track at TREC 2006 and 2007, and is a developer of the open source Terrier IR platform. He holds a BSc (Hons) from the University of Glasgow.



Iadh Ounis is a Reader in the Department of Computing Science at the University of Glasgow, which he joined as a Lecturer in 1999. He holds a PhD degree from the University Joseph Fourier, Grenoble. He has been an active researcher in information retrieval since 1994. His current research focuses on parameters-free probabilistic IR Models, Intranet, Enterprise, Web search and large-scale text retrieval systems building. He has recently been coordinating the TREC Blog Track, the purpose of which is to explore information seeking behaviour in the blogosphere. He is the principle investigator of the high-performance and scalable Terrier search engine, the open source version of which has been downloaded thousands of times since becoming available in November 2004.