University of Glasgow at TREC 2007: Experiments in Blog and Enterprise Tracks with Terrier

David Hannah, Craig Macdonald, Jie Peng, Ben He, Iadh Ounis

Department of Computing Science
University of Glasgow
Scotland, UK

{hannahd,craigm,pj,ben,ounis}@dcs.gla.ac.uk

ABSTRACT

In TREC 2007, we participate in four tasks of the Blog and Enterprise tracks. We continue experiments using Terrier[1] [14], our modular and scalable Information Retrieval (IR) platform, and the Divergence From Randomness (DFR) framework. In particular, for the Blog track opinion finding task, we propose a statistical term weighting approach to identify opinionated documents. An alternative approach based on an opinion identification tool is also utilised. Overall, a 15% improvement over a non-opinionated baseline is observed in applying the statistical term weighting approach. In the Expert Search task of the Enterprise track, we investigate the use of proximity between query terms and candidate name occurrences in documents.

1. INTRODUCTION

This year, in our participation in TREC 2007, we participate in the Enterprise and Blog tracks. For both tracks, we continue the research and development of the Terrier platform, and continue developing state-of-the-art weighting models using the Divergence from Randomness (DFR) paradigm.

In the expert search task of the Enterprise track, we continue our research on our voting techniques for expert search on the new CERC test collection. In particular, we investigate the usefulness of candidate and query term proximity and also how query expansion can be successfully applied to the expert search task. For the document search task, we investigate the combination of document priors, and techniques to take feedback documents into account.

In our first participation in the Blog track, we participate in all tasks, namely the opinion finding task (and polarity subtask), and the blog distillation (aka. feed search) task. In the opinion finding task, we deploy two opinion detection techniques. The first is based on a dictionary of weighted terms, which we use to identify opinions in blog documents. The second technique is based on the application of the OpinionFinder tool [19] to detect subjectivity and opinions in documents.

Lastly for the blog distillation task, we view this as a ranking of aggregates, which is similar to the expert search task. For this reason, our participation in the blog distillation task revolves around the adaption of our voting techniques for expert search.

Our paper is structured as follows: We describe the DFR weighting models that we apply in this work in Section 2; and the indexing procedure that we used in Section 3. Both document search tasks are then described, namely the opinion finding task of the Blog track in Section 4, and the document search task of the Enterprise track in Section 5. We describe the expert search task of the Enterprise track in Section 6, followed by the closely-related blog (feed) distillation task of the Blog track in Section 7.

[1]Information on Terrier can be found at: http://ir.dcs.gla.ac.uk/terrier/

2. MODELS

Following from previous years, our research in Terrier centres in extending the Divergence From Randomness framework (DFR) [1]. The remainder of this section is organised as follows. Section 2.1 presents existing field-based DFR weighting models, while Section 2.2 presents our existing DFR model, which captures term dependence and proximity. Section 2.3 presents the Bo1 DFR term weighting model for query expansion.

2.1 Field-based Divergence From Randomness (DFR) Weighting Models

Document structure (or fields), such as the title and the anchor text of incoming hyperlinks, have been shown to be effective in Web IR [4]. Robertson et al. [18] observed that the linear combination of scores, which has been the approach mostly used for the combination of fields, is difficult to interpret due to the non-linear relation between the scores and the term frequencies in each of the fields. In addition, Hawking et al. [6] showed that the length normalisation that should be applied to each field depends on the nature of the field. Zaragoza et al. [20] introduced a field-based version of BM25, called BM25F, which applies length normalisation and weighting of the fields independently. Macdonald et al. [8] also introduced *Normalisation 2F* in the DFR framework for performing independent term frequency normalisation and weighting of fields.

In this work, we use a field-based model from the DFR framework, namely PL2F. Using the PL2F model, the relevance score of a document d for a query Q is given by:

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

where λ is the mean and variance of a Poisson distribution, given by $\lambda = F/N$; F is the frequency of the query term t in the whole 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.

In PL2F, tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f, known as *Normalisation 2F* [8]:

$$tfn = \sum_f \left(w_f \cdot tf_f \cdot \log_2(1 + c_f \cdot \frac{avg\_l_f}{l_f}) \right), (c_f > 0) \quad (2)$$

where tf_f is the frequency of term t in field f of document d; l_f is the length in tokens of field f in document d, and avg_l_f is the average length of the field across all documents; c_f is a hyper-parameter for each field, which controls the term frequency normalisation; the importance of the term occurring in field f is controlled by the weight w_f.

Note that the classical DFR weighting model PL2 can be generated by using *Normalisation 2* instead of Normalisation 2F for tfn in Equation (1) above. Normalisation 2 is given by:

$$tfn = tf \cdot \log_2(1 + c \cdot \frac{avg\_l}{l})(c > 0) \quad (3)$$

where tf is the frequency of term t in the document d; l is the length of the document in tokens, and avg_l is the average length of all documents; c is a hyper-parameter that controls the normalisation applied to the term frequency with respect to l.

2.2 Term Dependence in the Divergence From Randomness (DFR) Framework

We believe that taking into account the dependence and proximity of query terms in documents can increase the retrieval effectiveness. To this end, we extend the DFR framework with models for capturing the dependence of query terms in documents. Following [2], the models are based on the occurrences of pairs of query terms that appear within a given number of terms of each other in the document. 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:

$$score(d, Q) = \sum_{t \in Q} qtw \cdot score(d, t) + \sum_{p \in Q_2} score(d, p) \quad (4)$$

where $score(d, t)$ is the score assigned to a query term t in the document d; p corresponds to a pair of query terms; Q_2 is the set that contains all the possible combinations of two query terms. In Equation (4), the score $\sum_{t \in Q} qtw \cdot score(d, t)$ can be estimated by any DFR weighting model, with or without fields. The weight $score(d, p)$ of a pair of query terms in a document is computed as follows:

$$score(d, p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}) \quad (5)$$

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 randomness model from the DFR framework, such as the Poisson approximation to the Binomial distribution. P_{p2} corresponds to the probability of seeing the query term pair 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 $score(d, t)$ is that the former depends on counts of occurrences of the pair of query terms p, while the latter depends on counts of occurrences of the query term t.

This year, we apply the pBiL2 randomness model [7], which does not consider the collection frequency of pairs of query terms. It is based on the binomial randomness model, and computes the score of a pair of query terms in a document as follows:

$$score(d, p) = \frac{1}{pfn + 1} \cdot \Big(\begin{aligned} & - \log_2 (avg\_w - 1)! + \log_2 pfn! \\ & + \log_2(avg\_w - 1 - pfn)! \\ & - pfn \log_2(p_p) \\ & - (avg\_w - 1 - pfn) \log_2(p'_p) \Big) \end{aligned} \quad (6)$$

where $avg\_w = \frac{T - N(ws-1)}{N}$ is the average number of windows of size ws tokens in each document in the collection, N is the number of documents in the collection, and T is the total number of tokens in the collection. $p_p = \frac{1}{avg\_w - 1}$, $p'_p = 1 - p_p$, and pfn is the normalised frequency of the tuple p, as obtained using Normalisation 2: $pfn = pf \cdot \log_2(1 + c_p \cdot \frac{avg\_w - 1}{l - ws})$. When Normalisation 2 is applied to calculate pfn, pf is the number of windows of size ws in document d in which the tuple p occurs. l is the length of the document in tokens and $c_p > 0$ is a hyper-parameter that controls the normalisation applied to pfn frequency against the number of windows in the document.

2.3 The Bo1 Term Weighting Model for Query Expansion

Terrier implements a list of DFR-based term weighting models for query expansion. The basic idea of these term weighting models is to measure the divergence of a term's distribution in a pseudo-relevance set from its distribution in the whole collection. The higher this divergence is, the more likely the term is related to the query's topic. Among the term weighting models implemented in Terrier, Bo1 is one of the best-performing ones [1].

The Bo1 term weighting model is based on the Bose-Einstein statistics. Using this model, the weight of a term t in the exp_doc top-ranked documents is given by:

$$w(t) = tf_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2(1 + P_n) \quad (7)$$

where exp_doc usually ranges from 3 to 10 [1]. Then, the top exp_term with the largest $w(t)$ from the exp_doc top-ranked documents are selected to be added to the query. exp_term is usually larger than exp_doc [1]. P_n is given by $\frac{F}{N}$. F is the frequency of the term in the collection, and N is the number of documents in the collection. tf_x is the frequency of the query term in the exp_doc top-ranked documents.

Terrier employs a parameter-free function to determine qtw when query expansion has been applied (see Equation (1)). The query term weight of a query term is then given as follows:

$$\begin{aligned} qtw &= \frac{qtf}{qtf_{max}} + \frac{w(t)}{\lim_{F \to tf_x} w(t)} \\ &= F_{max} \log_2 \frac{1 + P_{n,max}}{P_{n,max}} + \log_2(1 + P_{n,max}) \end{aligned} \quad (8)$$

where $\lim_{F \to tf_x} w(t)$ is the upper bound of $w(t)$. $P_{n,max}$ is given by F_{max}/N. F_{max} is the F (frequency in the collection) of the term with the maximum $w(t)$ in the top-ranked documents. If a query term does not appear in the most informative terms from the top-ranked documents, its query term weight remains equal to the original one.

3. INDEXING

This year we participate in both the Blog and Enterprise tracks. The test collection for the Blog track is the TREC Blogs06 test collection [10], which is a crawl of 100k blogs over an 11-week period. During this time, the blog posts (permalinks), feeds (RSS XML

etc.) and homepages of each blog were collected. In our participation in the Blog track, we index only the permalinks component of the collection. There are approximately 3.2 million documents in the permalinks component.

For the Enterprise track, a new collection has been deployed this year, namely the CSIRO Enterprise Research Collection (CERC), which is a crawl of the `csiro.au` domain (370k documents). CSIRO is a real Enterprise-sized organisation, and this collection is a more realistic setting for experimentation in Enterprise IR than the previous enterprise W3C collection.

For indexing purposes, we treat both collections in the same way, using the Terrier IR platform [14]. In particular, to support the field-based weighting models, we index separate fields of the documents, namely the content, the title, and the anchor text of the incoming hyperlinks. Each term is stemmed using Porter's English stemmer, and normal English stopwords are removed.

4. BLOG TRACK: OPINION FINDING TASK

In our participation in the opinion finding task, we aim to test two novel approaches to opinion detection. The first one is a light-weight dictionary-based statistical approach, and the second one applies techniques in Natural Language Processing (NLP) for subjectivity analysis. We conduct experiments to see to which extent these two approaches improve the performance in opinion detection over the baseline. We introduce the two opinion detection approaches in Section 4.1 and discuss our experiments in Section 4.2.

4.1 Opinion Detection Approaches

Firstly, inspired by participants in last year's opinion finding task [15], we propose a dictionary-based statistical approach to opinion detection based on a list of approximately 12,000 English words derived from various linguistic sources. For a set of training queries, we assume that D(Rel) is the document set containing all relevant documents, and D(opRel) is the document set containing all opinionated relevant documents. D(opRel) is a subset of D(Rel). For each term t in the word list, we measure $w_{opn}(t)$, the divergence of the term's distribution in D(opRel) from that in D(Rel). This divergence value measures how a term stands out from the opinionated documents, compared with all relevant, yet not necessarily opinionated, documents. The higher the divergence is, the more opinionated the term is. In our experiments, the opinion weight $w_{opn}(t)$ is assigned using the Bo1 term weighting model in Equation (7). We submit the 100 most weighted terms as a query Q_{opn} to the system, and assign an opinion score $Score(d, Q_{opn})$ to each document according to Q_{opn}, using the PL2 document weighting model (see Equations (1) & (3)) with the default parameter setting $c = 1$.

For each retrieved document for a given new query Q, we combine the relevance score $Score(d, Q)$ produced by a document weighting model (e.g. PL2F in Equations (1) & (2)) with the opinion score $Score(d, Q_{opn})$. Our combination method is as follows:

$$Score_{com}(d, Q) = \frac{-k}{log_2 P(op|d, Q_{opn})} + Score(d, Q) \quad (9)$$

where the final combined relevance score $Score_{com}(d, Q)$ is the sum of the raw relevance score $Score(d, Q)$ with the inverse form of the logarithm function of opinion probability $P(op|d, Q_{opn})$. k is a scaling factor. Based on training on last year's opinion finding task queries, we use $k = 600$ in our submitted runs. The opinion probability $P(op|d, Q_{opn})$ is mapped from the opinion score $Score(d, Q_{opn})$ by the following equation:

$$P(op|d, Q_{opn}) = \frac{Score(d, Q_{opn})}{\sum_{d \in Ret(Q_{opn})} Score(d, Q_{opn})} \quad (10)$$

where $Ret(Q_{opn})$ is the set of documents containing at least one of the 100 most opinionated terms in the dictionary. The final document ranking for a given new query Q is based on the combined relevance score $Score_{com}(d, Q)$. We have experimented with different combination methods such as the linear combination and the rank-based combination on last year's opinion finding task topics. The above combination method seems to be the most effective.

Our second opinion detection approach uses OpinionFinder [19], a freely available toolkit, which identifies subjective sentences in text. For a given document, we adapt OpinionFinder to produce an opinion score for each document, based on the identified opinionated sentences. We define the opinion score $Score(d, OF)$ of a document d produced by OpinionFinder as follows:

$$Score(d, OF) = sumdiff \cdot \frac{\#subj}{\#sent} \quad (11)$$

where $\#subj$ and $\#sent$ are the number of subjective sentences and the number of sentences in the document, respectively. $sumdiff$ is the sum of the $diff$ value of each subjective sentence in the document, showing the confidence level of subjectivity estimated by OpinionFinder.

For a given new query, such an opinion score is then combined with the relevance score $Score(d, Q)$ to produce the final relevance score in the same way as described above for the dictionary-based approach. The only difference is to use $Score(d, Q_{opn})$ instead of $Score(d, OF)$ in Equations (9) & (10). The parameter k in Equation (9) is set to 100 based on training on last year's opinion finding task topics.

Our dictionary-based approach is light-weight because the opinion scoring of the documents are performed offline (i.e. prior to retrieval), and such a scoring process is not computationally expensive. Compared with the dictionary-based approach, our second approach is based on the NLP subjectivity analysis techniques, which is more computationally expensive than the first one - for instance calculating opinion scores from the dictionary takes a few seconds, while running OpinionFinder on a subset of the collection can takes weeks of CPU hours.

4.2 Experiments

All our six submitted runs use the PL2F field-based weighting model in Equations (1) & (2). Our opinion retrieval runs are summarised in Table 1. Firstly, on top of the title-only baseline (uogBOPF), run uogBOPFProx tests the use of the DFR-based pBiL2 term proximity model (Equation (6)) in enhancing retrieval performance. Run uogBOPFProxW differs from uogBOPFProx by the use of our first opinion detection approach. Secondly, compared to the title-description baseline (uogBOPFTD), run uogBOPFTDW uses the first opinion detection approach, while run uogBOPFTDOF applies our NLP-based opinion detection approach using OpinionFinder. Because of the time constraint, we only finished parsing a small portion of the retrieved documents using OpinionFinder for our submitted run. In this paper, we also report the results obtained based on a complete parsing of the retrieved documents using OpinionFinder.

Finally, one polarity run was submitted, where the opinion categorisation is based on the dictionary-based opinion finding approach. For each type of opinion relevance degree (positive, negative or mixed), we measure the divergence of each term's distri-

Run	Techniques
uogBOPF(Base)	T-only queries + PL2F
uogBOPFProx	uogBOPF + proximity
uogBOPFProxW	uogBOPFProx + dictionary
uogBOPFTD(Base)	TD queries + PL2F
uogBOPFTDW	uogBOPFTD + dictionary
uogBOPFTDOF	uogBOPFTDW + OpinionFinder

Table 1: Techniques applied in the submitted runs in the Blog track opinion finding task.

Run	MAP(rel)	P@10(rel)	MAP(op)	P@10(op)
median	0.3340	-	0.2416	-
Title-only runs				
uogBOPF(Base)	0.3532	0.6120	0.2596	0.4200
uogBOPFProx	**0.3812**	**0.6740**	**0.2817**	**0.4540**
uogBOPFProxW	**0.4160**	**0.7200**	**0.3264**	**0.5520**
Title-description runs				
uogBOPFTD(Base)	0.3868	0.7420	0.2971	0.4880
uogBOPFTDW	**0.4033**	0.7600	**0.3182**	**0.5580**
uogBOPFTDOF	0.3872	0.7300	0.2995	0.4920
uogBOPFTDOFa	**0.4064**	0.7560	**0.3251**	**0.5620**
uogBOPFPol	RAccuracy: 0.1460		median: 0.1227	

Table 2: Results of submitted runs in the opinion finding task. uogBOPFTDOFa is an additional run for which the parsing of the retrieved documents using OpinionFinder is completed. uogBOPFPol is our polarity run. All submitted runs are above the median of all participating systems. A value in bold indicates a significant difference ($p \leq 0.05$) from the baseline run according to the Wilcoxon matched-pairs signed-ranks test.

bution in the documents with this type of opinion relevance degree from its distribution in all relevant documents. We submit the top 100 positive, negative or mixed terms as a query to the system to score the polarity orientation of the documents in the collection. Each document is then categorised into the type (i.e. positive, negative or mixed) with the highest score.

Table 2 summarises the retrieval performance of our submitted runs in terms of topic relevance (rel) and opinion finding (op). In this table, a value in bold indicates a significant difference ($p \leq 0.05$) from the baseline run according to the Wilcoxon matched-pairs signed-ranks test. From the three title-only runs, we find that runs uogBOPFProx and uogBOPFProxW provide a statistically significant improvement over the baseline run uogBOPF in both topic relevance and opinion finding. This shows that the use of term proximity and the weighted dictionary is helpful in finding opinionated documents. In particular, the dictionary-based approach markedly improves the baseline (15.8% between uogBOPFProx and uogBOPFProxW in MAP, see Table 2). Moreover, it is interesting to see that the use of the dictionary for opinion finding improves the retrieval performance in both topic relevance and opinion finding. This is probably due to the fact that the blog articles are often opinionated. As a result, an approach improving the opinion finding performance is likely to improve the topic relevance. From the three title-description (TD) runs, we also observe an improvement in both topic relevance and opinion finding brought by the weighted dictionary. In addition, row uogBOPFTDOFa gives the result obtained using OpinionFinder with a complete parsing of the retrieved documents. Compared with the baseline uogBOPFTD, OpinionFinder brings a statistically significant improvement in both topic relevance and opinion finding, if the parsing of the retrieved documents is complete. Finally, the last row shows that our only submitted polarity run gives a RAccuracy (a ranked classification accuracy measure [13]) that is higher than the median of all participants.

Run	MAP(rel)	P@10(rel)	MAP(op)	P@10(op)
Title-only runs				
uogBOPF(Base)	0.3464	0.5960	0.2583	0.4260
uogBOPFProx	**0.3809**	**0.6580**	**0.2847**	**0.4720**
uogBOPFProxW	**0.4076**	**0.7100**	**0.3256**	**0.5540**
Title-description runs				
uogBOPFTD(Base)	0.3797	0.7300	0.2847	0.4820
uogBOPFTDW	0.3892	0.7300	**0.3100**	0.4840
uogBOPFTDOFa	0.3963	0.7480	**0.3133**	**0.5440**

Table 3: Results of submitted runs in the opinion finding task when the document fields feature is disabled. A value in bold indicates a significant difference ($p \leq 0.05$) from the baseline run according to the Wilcoxon matched-pairs signed-ranks test.

Run	MAP(rel)	P@10(rel)	MAP(op)	P@10(op)
Title-only runs				
uogBOPF(Base)	0.3677	0.6180	0.2722	0.4380
uogBOPFProx	**0.4041**	**0.6800**	**0.3007**	**0.4840**
uogBOPFProxW	**0.4114**	**0.7100**	**0.3279**	**0.5540**
Title-description runs				
uogBOPFTD(Base)	0.3967	0.7220	0.2968	0.4980
uogBOPFTDW	0.3897	0.7180	0.3060	**0.5440**
uogBOPFTDOFa	0.3950	0.7280	0.3082	0.5600

Table 4: Results of submitted runs in the opinion finding task when the document fields feature is disabled and language filter is applied. A value in bold indicates a significant difference ($p \leq 0.05$) from the baseline run according to the Wilcoxon matched-pairs signed-ranks test.

In our additional runs, we investigate if the use of document fields helps improve the retrieval performance. Table 3 provides the results obtained for our runs when the document fields feature is disabled. In this case, only the document content is used for retrieval. By comparing the results with (Table 2) and without (Table 3) the use of document fields, we find no statistically significant difference in the retrieval performance in these two tables. The results suggest that the document content index is adequate enough for this task. The use of additional document structure information does not seem to be beneficial.

Finally, we apply a language filter that removes non-English documents from the retrieved set. Table 4 contains the results obtained if the document fields feature is disabled and a language filter is applied. Compared with the results obtained without the use of the language filter (see Table 3), we find that the retrieval performance is markedly improved when the opinion finding feature is disabled.

In the Blog track opinion finding task, we have mainly tested two novel approaches for detecting subjectivity in documents. The light-weight statistical dictionary-based approach provides statistically significant improvement in opinion retrieval over the baseline (15.8% between uogBOPFProx and uogBOPFProxW in MAP, see Table 2); The NLP-based approach using OpinionFinder also achieves similar improvement when the parsing for the retrieval documents is complete. Moreover, from our additional runs, we find that the use of document fields does not seem to be helpful, and the use of the language filter is beneficial.

5. ENTERPRISE TRACK: DOCUMENT SEARCH TASK

In our participation in the Document Search task, we aim to test a list of techniques for using the feedback documents for enhancing the retrieval performance, using different sources of evidence, including the click-distance, inlinks, and a combination of inlinks

with URL-length. The feedback documents are given by the track organisers which are known to be relevant to the topics used in this task. Section 5.1 describes the different sources of evidence of relevance used in our experiments. Section 5.2 presents our experiments in this task.

5.1 Different Sources of Evidence

We apply three different sources of evidence for utilising the feedback documents, namely click-distance, inlinks, and URL-length.

The underlying hypothesis of the click-distance evidence is that the documents, adjacent to a known relevant document in the linking structure, are likely to be relevant. We conduct a breath-first search for the shortest path in the hyperlink graph between each document in the ranking and the feedback documents. If a shortest path $minDist$ is not found within a $maxDist$ links of the feedback document, then the distance is assumed to be $maxDist + 1$. The click-distance evidence is then combined with the relevance score $Score(d, Q)$ by the inverse form of the *sigmoid* function in [5]:

$$Score_{com}(d, Q) = w\frac{(minDist + 0.5)^a + k^a}{(minDist + 0.5)^a} + Score(d, Q) \quad (12)$$

where w, a and k are parameters. We use the parameter values suggested in [5] which are $w = 1.8$, $a = 0.6$ and $k = 1$.

In addition to the click-distance evidence, we considered the following two sources of query-independent evidence, namely inlinks and URL-length:

- Inlinks: Documents in the Web are connected through hyperlinks. A hyper-link is a connection between a source and a target document. A high number of incoming links indicates that many documents consider the given document to be of a high quality.
- URL-length: Simply counts the number of symbols in the URL. For example trec.nist.gov has a character length 13.

When using a query-independent feature for retrieval, the relevance score of a retrieved document d for a query Q is altered in order to take the document prior probability into account as follows:

$$score(d, Q) = score(d, Q) + log(P(E)) \quad (13)$$

where P(E) is the prior probability of the query independent feature E in document d.

From our previous study in [16], we found that it is possible to make a further improvement on the retrieval performance if we integrate more than one source of query-independent evidence. We used the conditional combination method, which takes the possible dependence between query-independent evidence into account [16]. Two query-independent features as combined by:

$$P(E1, E2) = P(E2|E1) \cdot P(E1) \quad (14)$$

where P(E1) is the prior probability of the query independent feature E1; $P(E2|E1)$ is the conditional probability of the query independent feature E2, given E1; $P(E1|E2)$ is the probability that both E1 and E2 occur [16]. Naturally, we can extend this technique to combine more than two sources of query-independent evidence.

When using the combination of query-independent feature described in Equation (14) for retrieval, the score of a retrieved document d for a query Q is altered, in order to take the combined query-independent evidence into account as follows:

$$score(d, Q) = score(d, Q) + log(P(E1, E2)) \quad (15)$$

Run	Techniques
uogEDSF(Base)	PL2F
uogEDSINLPRI	PL2F+inlinks
uogEDSComPri	PL2F+combining inlinks with URL length
uogEDSCLCDIS	PL2F+click-distance

Table 5: Techniques applied in the submitted runs in the Enterprise track document search task.

5.2 Experiments

We submitted four runs, all of which apply the PL2F DFR fields-based weighting model in Equations (1) & (2). Our submitted runs are summarised in Table 5. Our baseline run is uogEDSF, which applies the PL2F field-based weighting model. The parameter values used in PL2F are shown in Table 14On top of the baseline (uogEDSF), run uogEDSINLPRI tests the inlinks query-independent evidence, and run uogEDSComPri tests the combination of inlinks with URL-length. Finally, run uogEDSCLCDIS tests the use of click-distance. In our submitted runs, the training of the query-independent evidence, namely inlinks and URL-length, was done using the given feedback documents. The target evaluation measure of the training process is the Mean Reciprocal Rank (MRR) since there are only very few feedback documents per query.

Table 6 summarises the results of our submitted runs on the final 50 judged queries. The table shows that our baseline (uogEDSF) provides a robust retrieval performance that is higher than the median MAP of all participating systems. Run uogEDSCLCDIS, which uses the click-distance evidence, performs slightly better than our baseline, but with no statistically significant difference according to the Wilcoxon matched-pairs signed-ranks test. In addition, we see that the use of the query-independent evidence does not improve MAP and precision at 10 though it helps in MRR (inlinks especially can make a statistically significant improvement over the baseline) since it is the target evaluation measure for our training. We suggest this is due to the fact that our training process overfitted the small number of the given feedback documents.

Therefore, we re-run the experiments where the training of the query independent evidence is conducted on the .GOV collection (TREC 2003 Web Track mixed task), by optimising MAP. In this case the feedback documents are not used. The results presented in Table 6 show that the use of query-independent evidence does not improve MAP and P@10 over the baseline (the only exception is the URL-length evidence on P@10). However, the MRR measure is improved for all three sources of query-independent evidence.

We also re-run the experiments where the training of the query independent evidence is conducted on the .GOV2 collection (TREC 2006 namedpage finding task), by optimising MRR. The results show that the use of query independent evidence leads to improvements over the baseline at P@10 and MRR measures (the only exception is inlinks on P@10). However, no improvement over the baseline is observed for MAP (see bottom part of Table 6).

Overall, with various different training settings, it was not possible to improve the baseline MAP by using the query independent evidence, suggesting that the training issue needs to be further investigated (e.g. use of more training queries)

To conclude, in this task, we have tested the use of different sources of evidence for utilising the feedback documents. According to our experimental results, the use of click-distance works the best in our submitted runs with a slight positive difference from the baseline; The use of the query-independent feature can improve precision at 10 and MRR over the baseline if the training is appropriately conducted. More training data is possibly required for a better performance on MAP.

Run	-			MAP	P@10	MRR
median	-			0.3072	-	-
	Feature	Training Data	Training Measure			
Official Runs						
uogEDSF(Base)	-	Feedback documents	MRR	0.3393	0.4840	0.8092
uogEDSINLPRI	inlinks	Feedback documents	MRR	0.2694	0.4600	**0.8680**
uogEDSComPri	inlinks + URL length	Feedback documents	MRR	0.2190	0.4820	0.8505
uogEDSCLCDIS	click distance	Feedback documents	MRR	0.3442	0.4940	0.8236
Unofficial Runs						
-	URL length	Feedback documents	MRR	0.3002	0.4840	0.8381
-	inlinks	TREC 2003 Web Track mixed task	MAP	0.3162	0.4740	0.8531
-	inlinks + URL length	TREC 2003 Web Track mixed task	MAP	0.2322	0.4720	0.8511
-	URL length	TREC 2003 Web Track mixed task	MAP	0.3281	0.4880	0.8110
-	inlinks	TREC 2006 namedpage finding task	MRR	0.3000	0.4680	0.8548
-	inlinks + URL length	TREC 2006 namedpage finding task	MRR	0.2382	0.4940	0.8566
-	URL length	TREC 2006 namedpage finding task	MRR	0.3249	0.5140	0.8183

Table 6: The results of our official and unofficial runs in the Enterprise track Document Search task. The second row contains the median MAP of all participating systems. Value in bold indicates a significant difference ($p \leq 0.05$) from the baseline run according to the Wilcoxon matched-pairs signed-ranks test.

6. ENTERPRISE TRACK: EXPERT SEARCH TASK

We participated in the expert search task of the TREC 2007 Enterprise track, with the aim of continuing to test and develop our Voting Model for expert search [9]. In the expert search task, systems are asked to rank candidate experts with respect to their predicted expertise about a query, using documentary evidence of expertise found in the collection.

Our participation to the expert search task of TREC 2007 strengthens the Voting Model for expert search by testing it on a new test collection. We also test two forms of proximity and two forms of query expansion. In particular, we investigate how the proximity of candidate name occurrences to query terms can be applied within an expert search system. Indeed, a document may contain occurrences of several candidates' names. The closer a candidate name occurs to the terms of the query, the more likely that the document is a higher quality indicator of expertise. In this technique, we strengthen votes from expertise evidence where the candidate's name occurs in close proximity to the terms of the query.

Moreover, we compare two techniques for query expansion (QE) when applied to the expert search task. In the first of these, document-centric QE [11], QE is performed on the underlying ranking of documents. In the second, known as candidate topic-centric QE [12], where the pseudo-relevant set is taken as the top-ranked profile documents associated to the top-ranked candidates.

The remainder of this section is structured as follows: In Section 6.1, we give an overview of the Voting Model for expert search; Section 6.2 details how we take candidate and query term proximity into account in the Voting Model; and Section 6.3 provides details on both forms of query expansion; We detail the experimental setup in Section 6.4; and provide results and conclusions in Section 6.5.

6.1 Voting Model

In our voting model for expert search, instead of directly ranking candidates, we consider the *ranking of documents*, with respect to the query Q, which we denote $R(Q)$. We propose that the ranking of candidates can be modelled as a voting process, from the retrieved documents in $R(Q)$ to the profiles of candidates: every time a document is retrieved and is associated with a candidate, then this is a vote for that candidate to have relevant expertise to Q. The votes for each candidate are then appropriately aggregated to form a ranking of candidates, taking into account the number of voting documents for that candidate, and the relevance score of the voting documents. Our voting model is extensible and general, and is not collection or topics dependent.

In [9], we defined twelve voting techniques for aggregating votes for candidates, adapted from existing data fusion techniques. In this work, we apply only the robust and effective expCombMNZ voting technique for ranking candidates. expCombMNZ ranks candidates by considering the sum of the exponential of the relevance scores of the documents associated with each candidate's profile. Moreover, it includes a component which takes into account the number of documents in $R(Q)$ associated to each candidate, hence explicitly modelling the number of votes made by the documents for each candidate. Hence, in expCombMNZ, the score of a candidate C's expertise to a query Q is given by:

$$score\_cand_{expCombMNZ}(C,Q) = \|R(Q) \cap\ profile(C)\| \cdot \sum_{d\ \in\ R(Q) \cap\ profile(C)} exp(score(d,Q)) \quad (16)$$

where $\|R(Q) \cap\ profile(C)\|$ is the number of documents from the profile of candidate C that are in the ranking $R(Q)$.

6.2 Candidate - Query Term Proximity

Some types of documents can have many topic areas and many occurrences of candidate names (for instance, the minutes of a meeting). In such documents, the closer a candidate's name occurrence is to the query terms, the more likely that the document is a high quality indicator of expertise for that candidate [3, 17].

We define the proximity of candidate and query terms in terms of the DFR term proximity document weighting models defined in Section 2.2. The term proximity model is designed to measure the informativeness in a document of a pair of query terms occurring in close proximity. We adapt this to the expert search task and into the expCombMNZ voting technique (Equation (16)), by measuring the informativeness of a query term occurring in close proximity to a candidate's name, as follows:

$$score\_cand_{expCombMNZProx}(C,Q) = \|R(Q) \cap\ profile(C)\| \cdot \sum_{d\ \in\ R(Q) \cap\ profile(C)} exp(score(d,Q) + \sum_{p=name(C)\times t \in Q} score(d,p)) \quad (17)$$

Here p is a tuple consisting of a term t from the query and the full name of candidate C. $score(d,p)$ can be calculated using any DFR weighting model [7], however, for efficiency reasons, we use

the pBiL2 model (Equation (6)) because it does not consider the frequency of tuple p in the collection but only in the document.

Hence, in this way, we are able to use the same weighting model to count and weight candidate occurrences in close proximity to query terms as we proposed in [7] to weight the informativeness of query terms occurring in close proximity. Note that the approach proposed here does not remove evidence of expertise for a candidate where the candidate's name does not occur near a query term, as this may result in a relevant candidate not being retrieved for a difficult query (i.e. the relevant candidate had only sparse evidence of expertise). Instead, candidate with names occurring in close proximity to query terms are given stronger votes in the Voting Model, and hence should be ranked higher in the final ranking of candidates.

6.3 Query Expansion in Expert Search

Query Expansion (QE) has previously been shown to be useful in adhoc document retrieval tasks. We have been investigating how QE can be applied in the expert search task. In particular, we have proposed two forms of QE. Firstly, using the underlying ranking of documents $R(Q)$ applied in the voting model, it is clear that query expansion can be applied on $R(Q)$, to improve the quality of the ranking of documents, and hence the accuracy of the ranking of candidates [11]. In this scenario, the pseudo-relevant set is the top-ranked documents in the document ranking $R(Q)$.

However, it would be better to apply QE in expert search where the items of the pseudo-relevant set are in fact the top-ranked candidates. While we proposed candidate centric QE in [11], this method did not perform well, due to the occurrence of topic drift within candidate profiles. Topic drift is when a candidate has many interests represented in their profile, and using all documents in the profile can cause the QE to fail, by selecting expansion terms unrelated to the original query [12]. We hence proposed a new form of QE for expert search, known as candidate topic centric QE, in which the pseudo-relevant set contains only the top-ranked documents for the top-ranked candidate profiles [12].

6.4 Experimental Setup

In contrast to previous years of the expert search task at TRECs 2005 and 2006, for TREC 2007, there is no 'master list' of candidates deployed as a central part of the test collection. Indeed, participants are more realistically expected to identify and rank candidates themselves.

To identify candidates, we looked to identify email addresses in the CERC collection. As many email addresses are obfuscated to avoid detection by spam robots, we attempted to identify as many alternatives for the @ symbol as possible (for example `_AT_`, `{at}` etc.). Once the email addresses were extracted, we removed email addresses not matching the format `firstname.lastname@csiro.au`. Lastly, because there are many example email addresses in the corpus, we removed the email addresses containing the sequence of characters `name` (e.g. the actual address `firstname.lastname@csiro.au` is removed).

The second stage is associating documents with each candidate. The full name of each candidate is derived from their email address. We look for documents that match the full name (and hence the email address) of each candidate C, and associate them with that candidate.

For all our submitted runs, we apply the same index used in the document search task. Standard stopwords are removed, and Porter's English stemmer is applied. Moreover, to generate the underlying ranking of documents, we apply the PL2F document weighting model (Equation (1)), with three fields, namely content, title and anchor text. For query expansion, we apply the Bo1 term weighting model from the DFR framework (Equation (7)). For training, we used the same setting for the PL2F document weighting as that applied in the document search task of the Enterprise task - see Table 14. For training expert search specific features of runs (e.g. query expansion, candidate-query term proximity), we trained on TREC 2005 and 2006 expert search tasks.

Run Name	Salient Features
uogEXFeMNZ	PL2F & expCombMNZ voting technique
uogEXFeMNZP	+ query term proximity
uogEXFeMNZcP	+ candidate query term proximity
uogEXFeMNZdQ	+ document centric QE
uogEXFeMNZQE	+ candidate topic centric QE

Table 7: Salient features of our Enterprise track expert search task submitted runs.

Unfortunately, two small bugs affected our identifying of candidates and documents. Firstly, we did not correctly describe the HTML entity for the @ symbol, by mistakenly writing `&64;` instead of `&#64;`. This had the effect of not identifying 302 candidates (of which 16 relevant candidates were omitted). Secondly, our custom written Perl scripts only identified at most one expert per line of the HTML documents in the CERC collection. This had the effect of omitting a total of 346 candidate-document associations.

6.5 Experiments and Results

We submitted four runs to the expert search task of the Enterprise track. Along with the unsubmitted baseline, these were:

- uogEXFeMNZ: is our baseline run (unsubmitted). It applies the PL2F DFR document weighting model (Equations (1) & (2)) to generate the underlying ranking of documents, combined with the expCombMNZ voting technique to rank experts.
- uogEXFeMNZP: improves upon the baseline run by applying query term proximity, pBiL2, (Equation (6)) to the underlying ranking of documents.
- uogEXFeMNZcP: applies the candidate - query term proximity technique described in Section 6.2 above. Baseline is uogEXFeMNZ.
- uogEXFeMNZdQ: applies document-centric QE [11] to the baseline.
- uogEXFeMNZQE: applies candidate-topic-centric QE [12] to the baseline.

The salient features of the runs are described in Table 7. Moreover, Table 8 details the retrieval performance of the submitted runs and our unsubmitted baseline run (uogEXFeMNZ). Retrieval performance is measured in terms of mean average precision (MAP) and mean reciprocal rank (MRR). Moreover, we also provide corrected retrieval performances using the improved candidate profile sets. From the results, it is firstly noticeable that using the improved candidate profile sets markedly improves the retrieval performance of all runs. The largest of these improvements in the run applying candidate - query term proximity (uogEXFeMNZcP), which jumps from 0.3138 to 0.4419 MAP (41% improvement). Over all the corrected runs, this run performs the best, followed by the normal query term proximity approach (uogEXFeMNZPP). Candidate topic centric QE shows a very small improvement over the

Run Name	Submitted MAP	Submitted MRR	Corrected MAP	Corrected MRR
Best	0.70010	0.9345	0.7010	0.9345
Median	0.2468	0.5011	0.2468	0.5011
uogEXFeMNZcP	**0.3138**	0.4475	**0.4419**	**0.5802**
uogEXFeMNZdQ	0.3122	**0.4597**	0.3748	0.5266
uogEXFeMNZP	0.3042	0.4239	0.3811	0.5024
uogEXFeMNZQE	0.2686	0.3670	0.3783	0.5149
uogEXFeMNZ	-	-	0.3782	0.5057

Table 8: The mean average precision (MAP) and mean reciprocal rank (MRR) of our Enterprise track expert search task submitted runs, as well as the median performance achieved by all participating systems. Submitted is using the profile set we use for our submitted runs, while corrected depicts the retrieval performance when the improved profile sets are applied. All runs use title only topics. uogEXFeMNZ is an additional, unsubmitted baseline run.

Method	MAP	MRR
VotesProx	0.2171	0.2754
CombMAXProx	0.4332	**0.6034**
CombSUMProx	0.3190	0.4139
CombMNZProx	0.2569	0.3223
expCombSUMProx	0.4370	0.5905
expCombMNZprox	**0.4419**	0.5802
MRRProx	0.3148	0.4081

Table 9: Comparison of the retrieval performance of various voting techniques in the Enterprise track expert search task.

corrected baseline, while document centric QE shows a small decrease. Compared to the median, the corrected runs are well above the median performance for MAP, and above median MRR also.

Table 9 details the performance of a selection of voting techniques from the Voting Model for expert search [9]. All voting techniques use the PL2F document weighting model as well as the improved candidate profile sets and candidate query term proximity. From this we can see that the CombMAX voting technique is best for the MRR evaluation measure, and the MRR voting technique performs best for the MAP evaluation measure. The CombMAX, expCombSUM and expCombMNZ techniques markedly improve over the Votes, CombSUM, CombMNZ and MRR voting techniques, for both evaluation measures.

6.6 Expert Search Task Conclusions

Overall, we demonstrated that the voting techniques from the Voting Model can be successfully applied on the new and more realistic CERC expert search test collection. Our results show the candidate - query term proximity method we proposed can be effectively applied to the expert search task, and will result in a marked increase in both MAP and MRR retrieval performances. Finally, (for the second year running), we have been affected by a bug in the generation of candidate profiles, and that the accuracy of candidate expertise evidence is a highly important factor in the retrieval performance of the expert search system.

In terms of experimental conclusions, the proposed candidate query term proximity technique shows a marked improvement (17%) over the baseline, followed by the more traditional proximity applied on the document ranking. For the query expansion, the document centric QE did not work while the candidate topic centric QE showed little positive difference from the baseline, and not as much as the increases exhibited by the proximity runs.

7. BLOG TRACK: FEED DISTILLATION TASK

In TREC 2007, we also participated in the blog distillation (feed distillation) task of the Blog track, where we aim to test the applicability of our novel voting model for Expert Search [9] to this task. Firstly, in the blog distillation task, the aim of each system is to identify the blogs (feeds[2]) that have a principle recurring interest in the query topic [13]. We believe that the blog distillation task can be seen as a voting process: A blogger with an interest in a topic will blog regularly about the topic, and these blog posts will be retrieved in response to a query topic. Each time a blog post is retrieved about a query topic, that can be seen as a vote for that blog to have an interest in the topic area. Indeed, this task is then very similar to the expert search task, in that both tasks aggregate the documents that are ranked in response to a query. Hence, our main investigation in our TREC 2007 participation is to determine if our Voting Model for expert search (which we also applied for the expert search task in Section 6) can be successfully applied to this task also.

In this task, we have three central research hypotheses: Firstly, is the voting paradigm depicted by the Voting Model for expert search an accurate depiction of the blog distillation task, and hence can the voting techniques be successfully applied in this task; Secondly, can we improve the effectiveness of the blog distillation system by giving less consideration to feeds that do not have a cohesive set of associated documents; and lastly, can the anchor text from homepages to homepages be of benefit to the retrieval performance of the search engine.

7.1 Cohesiveness

In [12], we defined three measures of cohesiveness for expert search, within the context of query expansion for expert search. A measure of cohesiveness examines all the documents associated with an aggregate, and measures on average, how different each document is from all the documents associated to the aggregate. In particular, we proposed three measures of cohesiveness, based on Cosine distance, Kullback-Leibler divergence, and profile size. In TREC 2007, we aim to test whether the cohesiveness measures can successfully identify feeds that blog about a very diverse set of topics, and hence are less likely to be principally devoted to the area of the query topic.

In TREC 2007, we only apply the cohesiveness measure based on the Cosine distance, as the profile size-based measure is not intuitive in the blogosphere context, while the multiple logarithm function calls in the Kullback-Leibler divergence based cohesiveness measure make it slower to apply at the scale of the TREC Blogs06 collection. The cohesiveness of a blog B can be measured using the Cosine measure from the vector-space framework as follows:

$$Cohesiveness_{Cos}(B) = \frac{1}{\|posts(B)\|} \cdot \sum_{d \in posts(B)} \frac{\sum_{t \in posts(B)} tf_d \cdot tf_B}{\sqrt{\sum_{t \in d} (tf_d)^2} \sqrt{\sum_{t \in posts(B)} (tf_B)^2}} \quad (18)$$

where $posts(B)$ denotes the set of blog posts (i.e. documents) associated with blog B. Moreover, tf_d is the term frequency of term t in document d, and tf_B is the total term frequency of term t in all documents associated with blog B (denoted $t \in posts(B)$). $Cohesiveness_{Cos}$ measures the mean divergence between every document in the blog and the blog itself. Note that $Cohesiveness_{Cos}$

[2]In this task, we will use the term feed and blog interchangeably, as each blog in the collection has one corresponding feed.

is bounded between 0 and 1, where 1 means that the documents represent the entire blog completely.

We integrate the cohesiveness score with the $score\_cand(B,Q)$ of a blog to a query as follows:

$$score\_cand(B,Q) = score\_cand(B,Q) + log(1 + Cohesiveness_{Cos}(B)) \quad (19)$$

7.2 Additional Homepage Anchor Text

Blogs often link to other blogs with similar interests, particularly in the 'blogroll' at the side of a blog's homepage. Because of this linkage, we choose to use some additional anchor text information from the blogrolls from blogs in the collection. In doing so, we hoped that this would bring additional textual evidence about the blogger's interests, and also would identify the more authoritative blogs (i.e. those most linked to by other blogs in the general topic area).

Recall from Section 3 that we have already indexed the permalinks (documents) part of the Blogs06 collection. In doing so, we extracted all anchor text coming *from* documents *to* documents. However, such anchor text does not include the blogroll normally found on the homepage of each blog.

Therefore in addition, we extracted anchor text linking from the homepage component of the collection to other homepages in the collection. This obtained, on average, an additional 49 tokens of anchor text per blog, in addition to the mean 21 tokens of anchor text already associated with each document in the collection.

We combined the additional anchor text information with the blog scores as follows:

$$score\_cand(B,Q) = score\_cand(B,Q) + score\_cand_{Anch}(B,Q) \quad (20)$$

where $score\_cand_{Anch}(B,Q)$ is the score of additional homepage anchor text calculated using the PL2 weighting model (Equations (1) & (3)). This has the effect of boosting blogs that have anchor text linking to their homepages containing query terms.

7.3 Experimental Setup

Similar to our participation in the opinion finding task of the Blog track, and as described in Section 3, we index the permalinks component of Blogs06 collection using the Terrier IR platform [14], by removing standard stopwords and applying Porter's stemming for English. Note that we do not index the feeds component of the collection.

For the underlying ranking of blog posts, we apply the PL2F field-based document weighting model (Equation (1)), using the content, title and anchor text of incoming hyperlinks as the fields. The parameter values for PL2F were exactly the same as those applied in the opinion finding task - i.e. they were trained on the TREC 2006 opinion finding task (Actual values are reported in Table 13).

For the associations between blog posts (documents) and blogs, we use the mappings provided by the collection, so that every document is associated to its corresponding blog. Hence, when a blog document is retrieved for a query topic, this can be seen as a vote for the corresponding blog to have an interest in the topic area.

7.4 Experiments and Results

We submitted 4 runs to the Blog Distillation task of the TREC-2007 Blog Track, which test our hypotheses for this task. The first run is a baseline run.

- *uogBDFeMNZ* is our baseline run. It uses the PL2F weighting model together with expCombMNZ voting technique to score the predicted relevance of feeds to the query topic.
- *uogBDFeMNZP* improves on the baseline run, by boosting the rank of documents in the document ranking where the query terms occur in close proximity. We use the PBiL2 DFR term dependence model (Equation (6)) to model the proximity of query terms in the documents.
- *uogBDFeMNZpC* investigates our cohesiveness hypothesis in this task. Feeds with a cohesive set of blog posts that all discuss similar topic(s) will be ranked higher than feeds with a highly diverse set of associated blog posts.
- *uogBDFeMNZhA* investigates how the application of additional anchor text can be used to improve the performance of the blog retrieval system.

Run Name	Salient Features
uogBDFeMNZ	PL2F & expCombMNZ voting technique
uogBDFeMNZhA	+ homepage anchor text
uogBDFeMNZpC	+ cohesiveness
uogBDFeMNZP	+ query term proximity

Table 10: Salient features of our Blog track feed distillation task submitted runs.

Run Name	MAP	MRR	P@10
Median	0.2035	-	-
uogBDFeMNZ	0.2909	0.7686	0.5222
uogBDFeMNZhA	0.2340	0.7610	0.4667
uogBDFeMNZpC	0.2685	0.7620	0.5111
uogBDFeMNZP	**0.2923**	**0.7834**	**0.5311**

Table 11: The mean average precision (MAP), Reciprocal Rank (MRR), and precision at 10 (P@10) of our submitted Blog track feed distillation task runs, as well as the median performance achieved by all participating systems. MRR and P@10 medians are not available.

Table 10 summarises the salient features of our submitted runs. Moreover, Table 11 presents the results of the submitted runs. The evaluation measures in this task are Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Precision at 10 (P@10). From the results, we observe that all submitted runs are perform markedly higher than the median of all participating system. Moreover, the run applying proximity (uogBDFeMNZP) improves slightly over the baseline (uogBDFeMNZ) for all three evaluation measures. Runs applying additional anchor text (uogBDFeMNZhA) and cohesiveness (uogBDFeMNZpC) do not improve over the baseline.

In Table 12, we compare the retrieval effectiveness of various voting techniques we proposed in [9] - note that we do not apply proximity since it does not bring marked improvements over the voting techniques alone. From the results, it is noticeable that the expCombMNZ technique performs best for MAP and P@10 evaluation measures, while expCombSUM performs best for the MRR measure. The Votes, CombSUM, CombMNZ and MRR techniques all perform similarly on the MAP evaluation measure, while the MRR voting technique is better for the MRR evaluation measure, and not as good as others for P@10.

7.5 Blog Distillation Task Conclusions

Our participation to the blog distillation task at the TREC 2007 Blog track was successful as it demonstrated that the voting techniques that we proposed in [9] can be successfully applied to this task. While these techniques have been previously tested in smaller Enterprise settings with thousands of experts, this task has a much

Method	MAP	MRR	P@10
Votes	0.2574	0.6108	0.4867
CombMAX	0.2074	0.7034	0.3756
CombSUM	0.2669	0.6399	0.4867
CombMNZ	0.2631	0.6249	0.4844
expCombSUM	0.2663	**0.7726**	0.5000
expCombMNZ	**0.2909**	0.7686	**0.5222**
MRR	0.2666	0.7711	0.4511

Table 12: Comparison of the retrieval performance of various voting techniques in the Blog track feed distillation task.

larger setting of potential experts (i.e. 100,000 blogs), and given the promising retrieval performance demonstrated here by our runs, that it performs well in this larger setting.

8. CONCLUSIONS

In TREC 2007, we participate in two tracks, namely the Blog track and the Enterprise track. For the Blog track, our main research conclusion in the opinion finding task is that a purely statistical and lightweight approach based on a weighted dictionary is very effective in detecting opinions, in particular, leading to 15.8% improvement over the baseline. In addition, the OpinionFinder tool can be as effective as the dictionary-based approach if applied on all of the retrieved documents. For the blog distillation task, we showed a connection to the expert search task, by successfully adapting voting techniques that we have previously developed for expert search.

In the Enterprise track, our main finding for the document search task is that the click distance is the most effective feature for identifying related documents based on the feedback evidence. The use of priors and their combination did not work as well as expected, mainly due to a lack of adequate training. Finally, in the expert search task, the use of candidate and query term proximity benefited retrieval performance markedly, and that for the query expansion techniques, the more realistic candidate-centric form helped most, improving slightly on the baseline.

Acknowledgements

We would like to thank Erik Graf for his assistance in running the OpinionFinder tool, and Christina Lioma for identifying various linguistical sources of opinionated words. Moreover, we would like to thank the three friendly assessors who assisted us in our TREC assessment workload this year.

9. REFERENCES

[1] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow, 2003.

[2] G. Amati, C. Carpineto and G. Romano. Italian Monolingual Information Retrieval with Prosit. In *CLEF 2001*, 257–264, 2002.

[3] Y. Cao, H. Li, J. Liu, S. Bao. Research on expert search at enterprise track of TREC 2005. In *Proceedings of TREC 2005* Gaithersburg, USA, 2006.

[4] N. Craswell and D. Hawking. Overview of TREC 2004 Web track. In *Proceedings of TREC 2004*, Gaithersburg, USA, 2004.

[5] N. Craswell, S.E. Robertson, H. Zaragoza and M. Taylor. Relevance weighting for query independent evidence. In *Proceesings of SIGIR 2005*, Salvador, Brazil, 2005.

[6] D. Hawking, T. Upstill and N. Craswell. Towards better Weighting of Anchors. In *Proceedings of SIGIR 2004*, 512–513, Sheffield, UK, 2004.

[7] C. Lioma, C. Macdonald, V. Plachouras, J. Peng, B. He and I. Ounis. University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of TREC 2006*, Gaithersburg, USA, 2007.

[8] C. Macdonald, V. Plachouras, B. He, C. Lioma and I. Ounis. University of Glasgow at WebCLEF 2005: Experiments in Per-Field Normalisation and Language Specific Stemming. *CLEF 2005*, 898–907, 2006.

[9] C. Macdonald and I. Ounis. Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In *Proceedings of CIKM 2006* , Arlington, USA, 2006.

[10] C. Macdonald and I. Ounis. The TREC Blogs06 Collection : Creating and Analysing a Blog Test Collection *DCS Technical Report TR-2006-224*. Department of Computing Science, University of Glasgow. 2006. `http://www.dcs.gla.ac.uk/~craigm/publications/macdonald06creating.pdf`

[11] C. Macdonald and I. Ounis. Using Relevance Feedback in Expert Search. In *Proceedings of ECIR 2007*, Rome, Italy, 2007.

[12] C. Macdonald and I. Ounis. Expertise drift and query expansion in expert search. In *Proceedings of CIKM 2007*, Lisbon, Portugal, 2007.

[13] C. Macdonald, I. Ounis and I. Soboroff. Overview of the TREC 2007 Blog Track. In *Proceedings of TREC 2007*, Gaithersburg, USA, 2007.

[14] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of OSIR 2006 Workshop*, Seattle, USA, 2006

[15] I. Ounis, M. de Rijke, C. Macdonald, G. Mishne and I. Soboroff. Overview of the TREC 2006 Blog Track. In *Proceedings of TREC 2006*, Gaithersburg, USA, 2006.

[16] J. Peng and I. Ounis. Combination of Document Priors in Web Information Retrieval. In *Proceedings of ECIR 2007*, Rome, Italy, 2007.

[17] D. Petkova, W.B. Croft. Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of ICTAI 2006*, 599–608, Washington D.C., USA.

[18] S. Robertson, H. Zaragoza and M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of the CIKM 2004*, 42–49, Washington DC, USA, 2004.

Parameter	Equations(s)	Value(T)	Value(TD)
PL2F w_{body}	(1) & (2)	1.19	0.97
PL2F w_{title}	(1) & (2)	3.88	1.11
PL2F w_{anchor}	(1) & (2)	0.11	0.77
PL2F c_{body}	(1) & (2)	9.34	2.73
PL2F c_{title}	(1) & (2)	22.28	98.70
PL2F c_{anchor}	(1) & (2)	8.16	0.84
pBiL2 c_p	(6) & (3)	40.00	-
pBiL2 $dist$	(6)	2	-

Table 13: The parameter values used in our TREC 2007 Blog track opinion finding task runs for title-only (T) and title-description (TD) queries. The title-only settings are also used in the blog distillation task.

Parameter	Equation(s)	Value
PL2F w_{body}	(1) & (2)	1.197
PL2F w_{title}	(1) & (2)	33.20
PL2F w_{anchor}	(1) & (2)	43.23
PL2F c_{body}	(1) & (2)	12.25
PL2F c_{title}	(1) & (2)	66.83
PL2F c_{anchor}	(1) & (2)	45.79

Table 14: The parameter values used in our submitted runs to the TREC 2007 Enterprise Track.

[19] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff and S. Patwardhan. OpinionFinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 on Interactive Demonstrations*. Vancouver, British Columbia, Canada.

[20] H. Zaragoza, N. Craswell, M. Taylor, S. Saria and S. Robertson. Microsoft Cambridge at TREC 13: Web and Hard Tracks. In *Proceedings of the TREC 2004*, Gaithersburg, USA, 2004.