University of Glasgow at TREC 2008: Experiments in Blog, Enterprise, and Relevance Feedback Tracks with Terrier

Ben He, Craig Macdonald, Iadh Ounis, Jie Peng, Rodrygo L.T. Santos

Department of Computing Science
University of Glasgow
Scotland, UK

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

1. INTRODUCTION

In TREC 2008, we participate in the Blog, Enterprise, and Relevance Feedback tracks. In all tracks, we continue the research and development of the Terrier platform[1] centred around extending state-of-the-art weighting models based on the Divergence From Randomness (DFR) framework [26]. In particular, we investigate two main themes, namely, proximity-based models, and collection and profile enrichment techniques based on several resources.

In the Blog track, we aim to improve our opinion detection techniques and to integrate various new blog-specific features into our Voting Model [18]. For the baseline ad-hoc task, we aim to build strongly performing baselines by applying two different techniques. The first one boosts documents in which query terms co-occur in a given window size, and the second one applies query expansion using collection enrichment. Non-English documents are also removed from the retrieved results.

In the opinion-finding task, we experiment with two main opinion detection approaches. The first one improves our TREC 2007 dictionary-based approach by automatically building an internal opinion dictionary from the collection itself. We measure the opinionated discriminability of each term using an information-theoretic divergence measure based on the relevance assessments of previous years. The second approach is based on the OpinionFinder tool, which identifies subjective sentences in text. In particular, we introduce a novel method to measure the informativeness of query terms occurring in close proximity to subjective sentences.

In the blog distillation task, we have two research themes. Firstly, we aim to extend our Voting Model with a component to focus on a balanced and neutral retrieval that does not favour prolific bloggers. The Voting Model is based on the intuition that a relevant blogger will post repeatedly around a topic area. By treating each relevant post as a vote for that blog to be relevant, we can infer a ranking of blogs. This approach is based on voting techniques inspired by electoral social choice theory and data fusion [18]. Neutrality is an important concept in an election – each candidate should have an equal chance of getting elected. Similarly, bloggers should have an equal chance of getting retrieved for a query, regardless of how many posts they have made. With this in mind, we investigate the application of normalisation techniques in this task.

In our participation in the first Relevance Feedback track, we aim to develop new techniques on top of our DFR query expansion models. First, we expand the query by measuring the divergence of a term's distribution in a relevance set to its distribution in the whole collection using the Kullback-Leibler (KL) divergence measure. This relevance set can be either pseudo, which consists of the top returned documents, or explicit, which consists of the judged relevant documents as in the provided feedback sets B to E.

Second, we expand the query from surrogates of the documents, instead of all their text content. These document surrogates are created by a low-cost, syntactically-based information processing model, which uses surface-syntactic evidence to automatically identify informative content and to reduce the noise from any textual input. We use the surface-syntactic approach to prune the feedback documents before selecting the query expansion terms, allowing (noisy) terms carried in unusual syntactic structures to be ignored.

In the Enterprise track, we participate in both the document and the expert search tasks. In keeping with one of the central themes for our TREC 2008 participation, we investigate the application of suitable external evidence in both tasks.

In the document search task, we investigate how external resources can be used to enhance the retrieval performance through a collection enrichment approach. Our external resources are obtained from the top-ranked results produced by different commercial search engines. Furthermore, we test how the selective application of collection enrichment can provide further improvements.

In the expert search task, we have two aims. Firstly, to extend and further test our novel Voting Model and, secondly, to use external evidence of expertise. The Voting Model takes into account various sources of evidence of candidate's expertise, by examining the ranking of documents with respect to the query, and inferring votes for candidates to be relevant. The model is expanded to take into account several input rankings of documents. In particular, we use document rankings produced by Web search engines as an external evidence of the expertise of the candidates.

The remainder of this paper is structured as follows. Section 2 describes the DFR models we use in TREC 2008. Section 3 details our indexing specifications. Section 4 describes our approaches in each of the Blog track post retrieval tasks along with their corresponding evaluation. Section 5 details our participation in the Enterprise document search task, while Section 6 discusses the results of our first participation in the Relevance Feedback track. Sections 7 and 8 cover our participation in the Enterprise track expert search task and on the Blog track blog distillation task, respectively. Finally, Section 9 presents our final remarks.

[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 on extending the Divergence From Randomness framework (DFR) [1]. In Section 2.1, we present existing DFR weighting models we experimented with in TREC 2008, while, in Section 2.2, we present our existing DFR model that captures terms dependence and proximity. Section 2.3 presents the Bo1 and KL DFR term weighting models for query expansion.

2.1 Divergence From Randomness Weighting Models

Document structure (i.e. fields, such as titles) has been shown to be useful when ranking documents. A field-based weighting model is one that considers the occurrences of query terms in different fields. Robertson et al. [31] 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. [10] showed that the length normalisation that should be applied to each field depends on the nature of the field. Zaragoza et al. [36] introduced a field-based version of BM25, called BM25F, which applies length normalisation and weighting of the fields independently. Macdonald et al. [25] 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, and qtf_{max} is the maximum query term frequency among all query terms. In PL2F, tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f, a technique known as *Normalisation 2F* [25]:

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

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:

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

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 term frequency normalisation with respect to the document length.

Another weighting model used in our participation in TREC is the InLB model, which is applied in the Blog track opinion-finding task. This model applies the Inverse Document Frequency and Laplace succession for document weighting [1], as well as BM25's term-frequency normalisation function [32]. In InLB, for a given document d and query Q, the relevance score is given by:

$$score(d,Q) = \sum_{t \in Q} w(d,t) = \sum_{t \in Q} \frac{qtw \cdot tfn}{tfn+1} \log_2 \frac{N+1}{df+0.5} \quad (4)$$

where the query term weight qtw is given by qtf/qtf_{max}, qtf is the query term frequency, and qtf_{max} is the maximum query term frequency among all query terms. N is the number of documents in the collection, and df is the number of documents containing the query term t. The normalised term frequency tfn is given by BM25's normalisation function [32] as follows:

$$tfn = \frac{tf}{(1-b) + b \cdot \frac{l}{avg\_l}} \quad (5)$$

where tf is the within-document term frequency, l is the document length, and avg_l is the average document length in the whole collection. b is a free parameter. In this paper, we set b to 0.2337 after training on the 100 topics from the TREC 2006 and 2007 Blog track opinion-finding tasks, numbered 851 to 950.

The last weighting model used in this work is the DPH model, also derived from the DFR framework. Using DPH, the relevance score of a document *d* for a query *Q* is given by [2]:

$$score(d,Q) = \sum_{t \in Q} \frac{qtw(1-F)^2}{tf+1} \cdot \big(tf \cdot \log_2(tf \cdot \frac{avg\_l}{l}\frac{N}{TF})\big) + 0.5 \cdot \log_2(2\pi \cdot tf \cdot (1-F)) \quad (6)$$

where *F* is given by tf/l, *tf* is the within-document frequency, and *l* is the document length in tokens. avg_l is the average document length in the collection, *N* is the number of documents in the collection, and *TF* is the term frequency in the collection. Note that DPH is a parameter-free model. All variables in its formula can be directly obtained from the collection statistics. No parameter tuning is required to optimise DPH, and we can rather focus on studying query expansion. *qtw* is the query term weight and is given by qtf/qtf_{max}, where *qtf* is the query term frequency and qtf_{max} is the maximum query term frequency among all query terms.

2.2 Terms Dependence in the Divergence From Randomness 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 [3], 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:

$$score(d,Q) = \sum_{t \in Q} qtw \cdot score(d,t) + \sum_{p \in Q_2} score(d,p) \quad (7)$$

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, and Q_2 is the set that contains all possible combinations of two query terms. In Equation (7), $\sum_{t \in Q} qtw \cdot score(d,t)$ can be estimated by any DFR weighting model, with or without fields. The $score(d,p)$ of a pair of query terms in a document is computed as follows:

$$score(d,p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}) \quad (8)$$

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 applied the pBiL2 randomness model [17], 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:

$$score(d,p) = \frac{1}{pfn+1} \cdot \Big(\begin{aligned} & - \log_2 (avg\_w - 1)! + \log_2 pfn! \\ & + \log_2 (avg\_w - 1 - pfn)! \\ & - pfn \log_2 (p_p) \\ & - (avg\_w - 1 - pfn) \log_2 (p'_p) \Big) \end{aligned} \quad (9)$$

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 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 given by Normalisation 2: $pfn = pf \cdot \log_2(1 + c_p \cdot \frac{avg\_w - 1}{l - ws})$. In Normalisation 2, 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 the pf frequency with respect to the number of windows in the document.

2.3 Term Weighting Models 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 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:

$$w(t) = tf_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2 (1 + P_n) \quad (10)$$

where exp_doc usually ranges from 3 to 10 [1]. Another parameter involved in the query expansion mechanism is exp_term, the number of terms extracted from the exp_doc top-ranked documents. exp_term is usually larger than exp_doc [1]. P_n is given by $\frac{F}{N}$, F is the frequency of the term t in the collection, and N is the number of documents in the collection. tf_x is the frequency of the query term t in the exp_doc top-ranked documents.

Terrier employs a parameter-free function to determine the query term weight qtw (see Equation (1)), which is given as follows:

$$\begin{aligned} qtw &= \frac{qtf}{qtf_{max}} + \frac{w(t)}{\lim_{F \to tf_x} w(t)} \quad (11) \\ &= F_{max} \log_2 \frac{1 + P_{n,max}}{P_{n,max}} + \log_2 (1 + P_{n,max}) \end{aligned}$$

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 frequency F of the term with the maximum $w(t)$ in the top-ranked documents. If a query term does not appear among the most informative terms from the top-ranked documents, its query term weight remains equal to the original one.

Another term weighting model employed by Terrier is based on the KL divergence measure. Using the KL model, the weight of a term t in the feedback document set D is given by [1]:

$$w(t) = p(t|D) \cdot \log_2 \frac{p(t|D)}{p(t|Coll)} \quad (12)$$

where $p(t|D) = tf_x / c(D)$ is the probability of observing the term t in the feedback document set D, tf_x is the frequency of the term t in the set D and $c(D)$ is the number of tokens in this set. $p(t|Coll) = TF/c(Coll)$ is the probability of observing the term t in the whole collection, TF is the frequency of t in the collection, and $c(Coll)$ is the number of tokens in the collection. In our experiments, the feedback document set contains the exp_doc top-ranked documents, from which the exp_term most weighted terms by KL are then extracted.

Using KL, the query term weight qtw is also determined by Equation (11), while the upper bound of $w(t)$ is given by:

$$\lim_{F \to tf_x} w(t) = \frac{F_{max} \cdot \log_2 \frac{c(Coll)}{l_x}}{l_x} \quad (13)$$

where F_{max} is the collection frequency F of the term with the maximum $w(t)$ in the top-ranked documents, l_x is the length of the feedback documents, and *c(Coll)* is the number of tokens in the whole collection. Note that the DFR query expansion framework is similar to Rocchio's relevance feedback method [33]. The difference is that the former considers the whole feedback document set as a bag of words, while the latter averages term weights over single feedback documents.

3. INDEXING

In TREC 2008, we participate in the Blog, Enterprise, and Relevance feedback tracks. The test collection for the Blog track is the TREC Blogs06 collection [23], 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 Relevance feedback track, the test collection is the large-scale .GOV2 collection, which has an uncompressed size of 426G. For indexing purposes, we treat the above two collections in the same way. Using the Terrier IR platform [26], we create content-based indices, including the document content and the titles.

For the Enterprise track, we use the CSIRO Enterprise Research Collection (CERC), which is a crawl of the `csiro.au` domain (370k documents). CSIRO is a real Enterprise-sized organisation. 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.

For all the three collections, each term is stemmed using Porter's English stemmer, and normal English stopwords are removed.

4. BLOG TRACK: BLOG POST RETRIEVAL TASKS

Following the TREC guidelines, in the Blog post retrieval tasks, namely, baseline ad-hoc, opinion-finding, and polarity tasks, we

submit runs based on all 150 topics developed so far. In this section, however, unless otherwise stated, we report our results on the new topics only, i.e., topics 1001-1050. Additionally, all our runs use only the title of the topics.

4.1 Baseline Ad-hoc Retrieval Task

Following the Blog track guidelines, in the baseline ad-hoc retrieval task, we submit two runs solely aimed at retrieving topic-relevant documents, i.e., with no opinion feature enabled. Our first baseline, uogBLProx, applies the InLB document weighting model and the pBiL2 term proximity model, as described in Equations (4) and (9), respectively. In addition, we remove non-English blog posts from the returned results as language filtering was shown to be beneficial in previous Blog tracks [24, 27]. On top of uogBLProx, our second baseline, uogBLProxCE, applies the Bo1 term weighting model for query expansion on an external collection, namely, the Aquaint2 collection, a timely news resource.

Table 1 summarises the retrieval performance of our two submitted baseline runs in terms of both topic-relevance (rel) and opinion-finding (op). The median of the participating groups in this task for the 2008 topics, 1001-1050, is also shown. From the table, we can see that our baselines markedly outperform the TREC median performance, both in terms of topic-relevance and opinion-finding.

Run	MAP$_{rel}$	P@10$_{rel}$	MAP$_{op}$	P@10$_{op}$
TREC median	0.3529	0.6960	0.2890	0.5700
uogBLProx	0.4141	0.6840	0.3464	0.5820
uogBLProxCE	0.4219	0.7060	0.3531	0.6100

Table 1: Results of submitted runs in the baseline ad-hoc retrieval task for topics 1001-1050.

4.2 Opinion-Finding Task

In the opinion-finding task, we experiment with two main approaches for detecting opinionated documents. The first approach improves our TREC 2007 dictionary-based approach by automatically building an internal opinion dictionary from the collection itself. The second approach is based on the OpinionFinder tool, which identifies subjective sentences in text. In particular, we introduce a novel method to measure the informativeness of query terms occurring in a close proximity to subjective sentences.

In our first opinion detection approach [12], a dictionary of subjective terms is automatically derived from the target collection without requiring any manual effort. In particular, from the list of all terms in the collection ranked by their within-collection frequency in descending order, a skewed query model is applied to filter out those that are too frequent or too rare [5]. This aims to remove terms with too little or too specific information and which thus cannot be interpreted as general opinion indicators for different queries. Using the Bo1 term weighting model (Equation (10)) and a training set comprising the 100 topics for the TREC 2006 and 2007 Blog track opinion-finding tasks, 851-950, the remaining terms from the list are weighted based on the divergence of their distribution in the set $D(opRel)$ of relevant and opinionated documents retrieved for these topics against that in the set $D(rel)$ of relevant documents retrieved for the same set of topics.

To compare with the dictionary derived from the collection itself, we also manually generate a dictionary compiled from various linguistic resources such as OpinionFinder [35]. This dictionary contains around 12,000 English words, mostly adjectives, adverbs and nouns, which are supposed to be subjective. In this paper, we denote the manually edited dictionary by the *external dictionary*, and the automatically derived one by the *internal dictionary*.

We submit the 100 highly weighted terms from either dictionary as a query Q_{opn} and assign an opinion score to the retrieved documents using the InLB DFR weighting model (Equation (4)). For each retrieved document d for a given new query Q, we combine its topic-relevance score – given by a retrieval baseline, which is independent of any expressed opinion in the document – with its opinion score to produce the final document ranking. We have experimented with two combination methods. The first method applies the following linear combination:

$$score_{com}(d,Q) = (1-\alpha)\cdot score(d,Q_{opn}) + \alpha \cdot score(d,Q) \quad (14)$$

where $score(d,Q_{opn})$ and $score(d,Q)$ are scaled by dividing them by the maximum $score(d,Q_{opn})$ and $score(d,Q)$, respectively. α is the free parameter of the linear combination, set after training on the 100 topics from 2006 and 2007.

Our second combination method maps each opinion score to the maximum likelihood of the probability $P(opn|d,Q_{opn})$ of being opinionated as follows:

$$P(opn|d,Q_{opn}) = \frac{score(d,Q_{opn})}{\sum_{d\in Coll} score(d,Q_{opn})} \quad (15)$$

where *Coll* is the entire document collection. Since a high probability is supposed to indicate a high degree of opinion expressed in the document, we would like to have a combined score that is an increasing function of $P(opn|d,Q_{opn})$. Therefore, such a probability $P(opn|d,Q_{opn})$ is combined with the initial relevance score using a logarithmic function as follows:

$$score_{com}(d,Q) = \frac{-k}{\log_2 P(opn|d,Q_{opn})} + score(d,Q) \quad (16)$$

where k is a free parameter, also set by training.

Both score combination methods use the stored opinion scores of all documents, computed during indexing. Therefore, there is only a negligible additional overhead during retrieval.

Our second approach in this task [34] uses OpinionFinder [35], a Natural Language Processing-based subjectivity analysis system, to classify the subjectiveness of every sentence in the Blogs06 collection. After the whole collection is parsed, we index it by considering the sentence tags generated by OpinionFinder as special position markers, so that we can record the positions of every index term with respect to the sentences in which it occurs within a given document. We then boost the scores of the retrieved documents, as given by the InLB weighting model (Equation (4)), based on the proximity between the query terms and the subjective sentences identified in each of these documents according to the equation:

$$score_{com}(d,Q) = (1-\beta)\sum_{p\in Q\times S} score(d,p) + \beta\cdot score(d,Q) \quad (17)$$

where the pair $p = \langle t, s\rangle$ comprises a query term t from the query Q and a subjective sentence s from the set S of all subjective sentences identified in document d. In order to compute the proximity score $score(d,p)$, we apply the pBiL randomness model [17], as given by Equation (9), except that the proximity windows are measured in terms of number of sentences instead of number of tokens and Normalisation 2 is not applied, since it did not show significant improvements in our experiments. The linear combination parameter β is trained on the 2006 and 2007 topics.

4.2.1 Experiments

In 2008, the TREC Blog track organisers provided five strongly performing, yet statistically different baselines. Each of these comprises a list of retrieved documents produced by a "black box"

search engine that retrieves as many topic-relevant documents as possible without applying any specific opinion-finding feature. On top of each of our two baselines, namely, uogBLProx and uogBLProxCE, and the 5 standard baselines provided by the Blog track organisers for TREC 2008, we submitted four runs as follows. The first run applies our dictionary-based approach using either the internal or the external opinion dictionary. The second run applies our TREC 2007 OpinionFinder-based approach [11], while the third run integrates the new proximity of query terms to subjective sentences approach. The last run combines the proximity to subjective sentences feature with our dictionary-based approach. Table 2 describes the nomenclature used by our submitted runs.

Code	Techniques
OPb1	Run based on baseline uogBLProx
OPb2	Run based on baseline uogBLProxCE
OP1-5	Runs based on standard baselines 1-5
ext	Dictionary-based approach (external dictionary)
int	Dictionary-based approach (internal dictionary)
of	OpinionFinder-based approach
Pr	Proximity to OpinionFinder's classified subjective sentences
L	Logarithmic combination
l	Linear combination

Table 2: Techniques applied in the submitted runs in the Blog track opinion-finding task.

Table 3 summarises the retrieval performance of our submitted runs in terms of topic-relevance (rel) and opinion-finding (op) over the 7 baselines. In this table, an asterisk (*) indicates a significant difference ($p \leq 0.05$) from the corresponding baseline run according to the Wilcoxon matched-pairs signed-ranks test. We find that all our 4 approaches provide statistically significant improvement over 5 out of the 7 baselines. As for the remaining baselines, our proximity-based approaches significantly improve over baseline 2, while the approaches that do not employ proximity significantly improve over baseline 4. As a whole, these results show that both of our approaches are effective in finding opinionated documents.

In order to further investigate the robustness of our approaches, Table 4 shows the opinion MAP performances of each of our runs over each of the 5 standard baselines as well as the average performance of these runs across the baselines. For the sake of legibility, the 4 approaches deployed over each baseline are generically referred to as Dict, OpinionFinder, ProxSent, and Dict+ProxSent. For each baseline, we also show the median performance of the 21 TREC runs that were deployed over all the standard baselines. The median of the average improvements of these runs is also shown. From the table, we can see that, besides improving over the baselines, our approaches outperform the TREC medians in most settings. Moreover, on average, all of our techniques provide improvements across the 5 baselines, what further attests their robustness.

Additionally, Table 5 shows the performance results of our best opinion-finding runs for each of the 7 baselines in terms of topic-relevance and opinion-finding MAP across the topics for each of the 3 years the opinion-finding task was run, as well as for the combined topics for the 3 years. From Table 5, we can observe that, for baselines 1, 3, and 4, our opinion-finding performance increases over the three years. Interestingly, in terms of topic-relevance performance, the best results are observed for the TREC 2007 topics across all baselines, which suggests that this set of topics was relatively easier when compared to the 2006 and 2008 topics.

4.3 Polarity Task

In the polarity task, we apply our dictionary-based approach once more, with the exception that the dictionaries used for retrieving positive and negative blog posts are respectively extracted from the strong positive and negative words in OpinionFinder's dictionary. The negative dictionary comprises a total of 2,547 words while the positive one comprises 1,153 words in total.

Run	MAP_{rel}	$P@10_{rel}$	MAP_{op}	$P@10_{op}$
uogBLProx	0.4141	0.6840	0.3464	0.5820
uogOPb1intL	0.4218	0.7260*	0.3607*	0.6220*
uogOPb1ofL	0.4281*	0.7240*	0.3665*	0.6360*
uogOPb1Pr	0.4149	0.7040	0.3629*	0.6300
uogOPb1PrinL	0.4142	0.7040	0.3636*	0.6300
uogBLProxCE	0.4219	0.7060	0.3531	0.6100
uogOPb2intl	0.4237	0.7240	0.3617*	0.6340
uogOPb2ofL	0.4342*	0.7340*	0.3709*	0.6380*
uogOPb2Pr	0.4116	0.7040	0.3597*	0.6200
uogOPb2PrintL	0.4115	0.7100	0.3604*	0.6260
baseline 1	0.4032	0.7320	0.3239	0.5800
uogOP1intL	0.4174*	0.7440	0.3512*	0.6380*
uogOP1ofL	0.4073*	0.7460	0.3526*	0.6460*
uogOP1Pr	0.4115*	0.6880	0.3529*	0.6040
uogOP1PrintL	0.4120*	0.6880	0.3564*	0.5980
baseline 2	0.3107	0.6480	0.2639	0.5500
uogOP2intL	0.3029	0.6400	0.2621	0.5660
uogOP2ofL	0.3123	0.6360	0.2712	0.5540
uogOP2Pr	0.3048	0.6080	0.2692*	0.5420
uogOP2PrintL	0.3045	0.6020	0.2692*	0.5380
baseline 3	0.4343	0.6440	0.3564	0.5540
uogOP3intL	0.4391*	0.7280*	0.3669*	0.6340*
uogOP3ofL	0.4419*	0.6980	0.3728*	0.6060*
uogOP3Pr	0.4315	0.7000	0.3685*	0.6200
uogOP3PrintL	0.4302	0.7020	0.3704*	0.6180
baseline 4	0.4724	0.7440	0.3822	0.6160
uogOP4intL	0.4750	0.7520	0.3964*	0.6400
uogOP4ofL	0.4710	0.7640	0.3963*	0.6600*
uogOP4Pr	0.4431	0.6940	0.3752	0.5980
uogOP4PrintL	0.4397	0.6920	0.3753	0.6040
baseline 5	0.3745	0.7040	0.2988	0.5300
uogOP5extL	0.3713*	0.6900	0.3033*	0.5640*
uogOP5ofL	0.3777*	0.7020	0.3098*	0.5660*
uogOP5Pr	0.3894*	0.7040	0.3312*	0.6160*
uogOP5PrintL	0.3915*	0.7140	0.3345*	0.6240*

Table 3: Results of submitted runs in the opinion-finding task over 7 different baselines for topics 1001-1050. An asterisk (*) indicates a significant difference ($p \leq 0.05$) from the corresponding baseline run according to the Wilcoxon matched-pairs signed-ranks test.

Table 6 shows the performance of our polarity runs in terms of both topic-relevance and opinion-finding across all baselines. Overall, none of our approaches significantly differs from their respective baselines according to the Wilcoxon matched-pairs signed-ranks test with $p \leq 0.05$. In fact, we can observe minor improvements for retrieving negative documents and a slight degradation for retrieving positive documents.

Analogously to the analysis conducted in the previous section, Table 7 shows the negative and positive MAP performances of our deployed approach for the polarity task (DictOF, in the table) over each of the 5 standard baselines as well as the average performances of these runs across the baselines. For each baseline, the median performance of all 10 TREC runs that were deployed over all 5 baselines is also shown. From Table 7, we can see that, although having decreased the baseline performances on average, our approach stands well above the median. These very low median values, in turn, attest the difficulty of this task.

Overall, our participation in the TREC 2008 Blog track was very successful. Our two submitted baseline runs performed well above the median performance of all participants. In the opinion-finding task, most of our approaches provided improvements over all the

	baseline1	baseline2	baseline3	baseline4	baseline5	improvement	
Baseline	0.3239	0.2639	0.3564	0.3822	0.2988	mean	stdev
+Dict	0.3512*	0.2621	0.3669*	0.3964*	0.3033*	+3.18%	3.38%
+OpinionFinder	0.3526*	0.2712	0.3728*	0.3963*	0.3098*	+4.72%	2.40%
+ProxSent	0.3529*	0.2692*	0.3685*	0.3752	0.3312*	+4.67%	5.18%
+Dict+ProxSent	0.3564*	0.2692*	0.3704*	0.3753	0.3345*	+5.22%	5.70%
TREC median	0.3493	0.2705	0.3705	0.3846	0.3010	+0.76%	0.73%

Table 4: Opinion MAP over 5 standard baselines and average improvement for topics 1001-1050. An asterisk (*) indicates a significant difference ($p \leq 0.05$) from the corresponding baseline run according to the Wilcoxon matched-pairs signed-ranks test.

	2006 (851-900)		2007 (901-950)		2008 (1001-1050)		All 150 topics	
Run	MAP_{rel}	MAP_{op}	MAP_{rel}	MAP_{op}	MAP_{rel}	MAP_{op}	MAP_{rel}	MAP_{op}
uogBLProx	0.3366	0.2224	0.4373	0.3265	0.4141	0.3464	0.3960	0.2984
uogOPb1ofL	0.3307	0.2341*	0.4375	0.3520*	0.4281*	0.3665*	0.3988	0.3175*
uogBLProxCE	0.3459	0.2351	0.4507	0.3393	0.4219	0.3531	0.4062	0.3091
uogOPb2ofL	0.3449	0.2428*	0.4529	0.3584*	0.4342*	0.3709*	0.4106*	0.3240*
baseline 1	0.3004	0.1905	0.4043	0.2758	0.4032	0.3239	0.3693	0.2634
uogOP1PrintL	0.3326*	0.2595*	0.4609*	0.3513*	0.4120*	0.3564*	0.4019*	0.3224*
baseline 2	0.3156	0.2296	0.3881	0.3034	0.3107	0.2639	0.3381	0.2656
uogOP2PrintL	0.3082	0.2410*	0.4069*	0.3415*	0.3045	0.2692*	0.3399*	0.2839*
baseline 3	0.3768	0.2545	0.4619	0.3489	0.4343	0.3564	0.4244	0.3199
uogOP3ofL	0.3769	0.2705*	0.4657	0.3665*	0.4419*	0.3728*	0.4282*	0.3366*
baseline 4	0.4300	0.3022	0.5303	0.3784	0.4724	0.3822	0.4776	0.3543
uogOP4intL	0.4240	0.3134*	0.5428*	0.3959*	0.4750	0.3964*	0.4806	0.3686*
baseline 5	0.4046	0.2632	0.5465	0.3805	0.3745	0.2988	0.4419	0.3141
uogOP5PrintL	0.4138	0.3012*	0.5510	0.4104*	0.3915*	0.3345*	0.4521*	0.3487*

Table 5: Results of our best submitted runs for each baseline on the 2008 topics across 4 sets of topics. An asterisk (*) indicates a significant difference ($p \leq 0.05$) from the corresponding baseline run according to the Wilcoxon matched-pairs signed-ranks test.

Run	MAP_{neg}	$P@10_{neg}$	MAP_{pos}	$P@10_{pos}$
uogBLProx	0.1218	0.1320	0.1376	0.1800
uogPLb11	0.1225	0.1440	0.0866	0.1260
uogBLProxCE	0.1176	0.1400	0.1388	0.1760
uogPLb21	0.1176	0.1420	0.1372	0.1700
baseline1	0.1175	0.1700	0.1364	0.1860
uogPL11	0.1076	0.1400	0.1272	0.1820
baseline2	0.0865	0.1420	0.0951	0.1400
uogPL21	0.0867	0.1420	0.0942	0.1340
baseline3	0.1266	0.1520	0.1376	0.1680
uogPL31	0.1203	0.1440	0.1345	0.1800
baseline4	0.1288	0.1600	0.1532	0.1980
uogPL41	0.1301	0.1580	0.1394	0.1700
baseline5	0.1085	0.1680	0.1229	0.1780
uogPL51	0.1067	0.1700	0.1179	0.1440

Table 6: Results of submitted runs in the polarity task for topics 1001-1050 over 7 different baselines.

baselines and the TREC median performance. Moreover, all of them showed to be robust, as demonstrated by their average improvement across the standard baselines. Finally, although not providing a significant improvement over the baselines, our polarity approach performed fairly above the median performance of the participants in this task.

5. ENTERPRISE TRACK: DOCUMENT SEARCH TASK

In our participation in the Enterprise track document search task, we aim to investigate how external resources, such as the Google and Yahoo! Web search engines, can be used to enhance the retrieval performance through a collection enrichment approach. Furthermore, we test how the selective application of collection enrichment can further improve retrieval effectiveness. Section 5.1 describes the selective application of collection enrichment. Section 5.2 presents our experiments.

5.1 Selective Application of Collection Enrichment

For Enterprise document search, query expansion may fail as the Enterprise intranets are often created by a small number of individuals, which lead to the Enterprise collection having limited use of alternative lexical representations. In particular, this could lead to pseudo-relevance sets of poor quality. In this case, it can be beneficial to use collection enrichment, which expands the initial query by taking into account a pseudo-relevance set based on larger and higher-quality external resources [14].

We experiment using five different external resources, namely Wikipedia, Yahoo! Web, Google Web, Google Scholar, and Google Books. The Wikipedia website provides the procedures about how to download the Wikipedia data[2]. For the external resources of Yahoo! Web, Google Web, Google Scholar, and Google Books, we submit queries to each of these search engines and then download their returned results. In particular, we discriminate the Yahoo! Web resource into *Yahoo! Web ANY* and *Yahoo! Web PDF* according to the restriction on the type of the retrieved documents for a given query. For example, Yahoo! Web ANY means that we do not apply any restriction on the retrieved documents when we submit a query to the Yahoo! Web search engine, while Yahoo! Web PDF means that we restrict the retrieved documents to be PDF files only. We make the same kind of discrimination for Google Web and Google Scholar as well.

We hypothesise that not all queries benefit equally from the application of collection enrichment. Therefore, we use query performance predictors to selectively apply collection enrichment on a per-query basis. Various query performance predictors have been studied in [13] and shown to be useful and low-cost. In our experiment, we use two of these predictors, namely, the $\gamma 2$ and the Average Inverse Collection Term Frequency (AvICTF) predictors.

[2] `http://en.wikipedia.org/wiki/Wikipedia:Database_download`

negative	baseline1	baseline2	baseline3	baseline4	baseline5	improvement	
Baseline	0.1175	0.0865	0.1266	0.1288	0.1085	mean	stdev
+DictOF	0.1076	0.0867	0.1203	0.1301	0.1067	-2.76%	3.92%
TREC median	0.0597	0.0457	0.0743	0.0677	0.0453	-48.49%	2.66%
positive	baseline1	baseline2	baseline3	baseline4	baseline5	improvement	
Baseline	0.1364	0.0951	0.1376	0.1532	0.1229	mean	stdev
+DictOF	0.1272	0.0942	0.1345	0.1394	0.1179	-4.60%	3.29%
TREC median	0.0953	0.0547	0.0955	0.0973	0.0708	-36.79%	17.48%

Table 7: Negative and positive MAP over 5 standard baselines and average improvement for topics 1001-1050.

The definition of $\gamma 2$ is given as follows:

$$\gamma 2 = \frac{idf_{max}}{idf_{min}} \tag{18}$$

where idf_{max} and idf_{min} are the maximum and minimum idf among the query terms in the query Q, respectively. The idf of each query term t is computed as follows:

$$idf(t) = \frac{\log_2 (N + 0.5)/N_t}{\log_2 (N + 1)} \tag{19}$$

where N_t is the number of documents in which the query term t appears and N is the number of documents in the whole collection.

The definition of $AvICTF$ is given as follows:

$$AvICTF = \frac{\log_2 \prod_Q \frac{token_{coll}}{tf_{coll}}}{ql} \tag{20}$$

where ql is the query length, tf_{coll} is the number of occurrences of a query term in the whole collection and $token_{coll}$ is the number of tokens in the whole collection.

Our decision mechanism is given as follows:

1. Expand the initial query on the local resource if and only if the prediction score obtained from the local resource is higher than a threshold score and the prediction score obtained from the external resource.

2. Expand the initial query on the external resource if and only if the prediction score obtained from the external resource is higher than a threshold score and the prediction score obtained from the internal resource.

3. Disable the expansion on the initial query if and only if the prediction scores obtained from the external and internal resources are all lower than a threshold score.

In addition, our decision mechanism is summarised in Table 8.

5.2 Experiments

We submitted four runs, all of which apply the PL2F DFR field-based weighting model. More details about this weighting model can be found in Section 2.1. Three document fields, namely, body, title, and anchor text of incoming hyperlinks are used. The details of our submitted runs are given below, while Table 9 summarises their salient features.

- Run uogTrEDbl tests how effective the application of query terms proximity in the DFR framework is by using the pBiL2 randomness model. More details about query terms proximity in the DFR framework can be found in Section 2.2.

- Run uogTrEDQE tests how effective the uniform application of query expansion to all queries is by using the Bo1 term weighting model. More details about the Bo1 term weighting model can be found in Section 2.3.

- As we hypothesise that not all queries benefit equally from the application of collection enrichment, we propose to use a query performance predictor to selectively apply collection enrichment on a per-query basis. Run uogTrEDSelW investigates how effective the selective application of collection enrichment is. The external resource in this case is Wikipedia, and we use $\gamma 2$ as our predictor, as it performed better than the $AvICTF$ predictor during our training process.

- Run uogTrEDSE2 also investigates how effective the selective application of collection enrichment is. In particular, we combine the results from several external resources. After training, we chose to combine the Yahoo! Web ANY and Yahoo! Web PDF external resources and the $AvICTF$ predictor.

Run	Techniques
uogTrEDbl	PL2F + proximity
uogTrEDQE	PL2F + query expansion
uogTrEDSelW	PL2F + selective CE on a single resource
uogTrEDSE2	PL2F + selective CE on combined resources

Table 9: Techniques applied in the submitted runs in the Enterprise track document search task.

Table 10 summarises the results of our official and unofficial runs on the final 63 judged queries. The table shows that the query terms proximity technique makes a marginal improvement on the retrieval performance over the PL2F baseline (ID0 vs. ID1). In addition, we observe that the runs with the application of query expansion or collection enrichment outperform the PL2F baseline (ID0 vs. ID2/ID5). Moreover, the selective application of collection enrichment makes a marked improvement on retrieval performance (ID3 vs. ID2/ID5). In particular, the improvement is significant ($p < 0.05$) in terms of MAP according to the Wilcoxon matched-pairs signed-ranks test. We also find that it is very important to choose an appropriate external resource (ID6 vs. ID7/ID8) and an appropriate predictor (ID3 vs. ID6) before selectively applying collection enrichment. Finally, we notice that the combination of external resources makes a slight improvement over any single external resource (ID4 vs. ID7/ID8).

Overall, in the Enterprise track document search task, we have shown that using query performance predictors to selectively apply collection enrichment on a per-query basis can enhance the retrieval performance.

6. RELEVANCE FEEDBACK TRACK

In the first Relevance Feedback track, we aim to test the effectiveness of the DFR query expansion framework in various relevance feedback settings. In addition, we apply a novel technique that expands queries on surrogates of the feedback documents, instead of the raw documents themselves.

score_L > T	score_E > T	score_L > score_E	Decision
True	True or False	True	local
True or False	True	False	external
False	False	True or False	disabled

Table 8: The decision mechanism of the selective application of collection enrichment. $score\_L$ and $score\_E$ denote the prediction scores on the local and external resources, respectively. T is a threshold score, which needs an appropriate setting. We used $T = 0$ in the submitted runs. $local$, $external$ and $disabled$ in the column $Decision$ indicate expanding the initial query on the $local$ resource, $external$ resource and disabling the expansion, respectively.

ID	Run	Technique	Predictor	MAP	NDCG
0	-	PL2F	-	0.3590	0.5454
1	uogTrEDbl	+ Proximity	-	0.3623	0.5489
2	uogTrEDQE	+Query Expansion	-	0.3718	0.5568
3	uogTrEDSelW	+ Selective CE (wiki)	$\gamma 1$	**0.3891**	**0.5660**
4	uogTrEDSE2	+ Selective CE (YWA + YWP)	AvICTF	0.3658	0.5587
0	-	PL2F	-	0.3590	0.5454
5	-	+ CE (wiki)	-	0.3648	0.5638
6	-	+ Selective CE (wiki)	AvICTF	0.3677	0.5579
7	-	+ Selective CE (YWA)	AvICTF	0.3576	0.5418
8	-	+ Selective CE (YWP)	AvICTF	0.3650	0.5534

Table 10: The results of our official and unofficial runs in the Enterprise track document search task. The highest value in each column is highlighted in bold.

6.1 Document Surrogates Creation

We expand the query from document surrogates, instead of all text content in the documents. The document surrogates are created by a low-cost, syntactically-based information processing model, which uses surface-syntactic evidence in order to automatically identify informative content and to reduce the noise from any textual input. We use the surface-syntactic approach to prune the feedback documents before selecting the query expansion terms, allowing (noisy) terms carried in unusual syntactic structures to be ignored.

We use part-of-speech (POS) n-grams [4, 16] to detect noise in the indexed documents. POS n-grams are n-grams of parts-of-speech, which are extracted from a POS-tagged sentence in a recurrent and overlapping way. For example, for a sentence ABCDEFG, where parts-of-speech are denoted by the single letters A, B, C, D, E, F, G, and where POS n-grams have length $n = 4$, the POS n-grams extracted are ABCD, BCDE, CDEF, and DEFG. The order in which the POS n-grams occur in the sentence is ignored. For each sentence, all possible POS n-grams are extracted.

Our technique is based on the fact that high-frequency POS n-grams correspond mostly to sequences of words that include relatively little noise, whereas low-frequency POS n-grams correspond mostly to sequences of words that include relatively more noise [16]. The only resources needed are a POS tagger and a collection of documents. This can be any collection of documents of a reasonable size, not necessarily the target collection, i.e., the collection from which we retrieve documents [15].

Our methodology is as follows. We extract POS n-grams from a collection of documents and count their frequency. We refer to these POS n-grams as *global POS n-grams*. We rank these global POS n-grams according to their frequency in the collection (in decreasing order). We refer to this ranked list as *global list*. We empirically set a cutoff threshold θ of POS n-gram rank in the global list and we assume that everything below this threshold corresponds to estimated noise (Figure 1). We then extract POS n-grams from the text we wish to process. For each POS n-gram drawn from the text, we determine its position in the global list. Whenever this rank is below the threshold, we remove the POS n-gram and its corresponding sequence of words from the document, regardless of any other POS n-grams that overlap it.

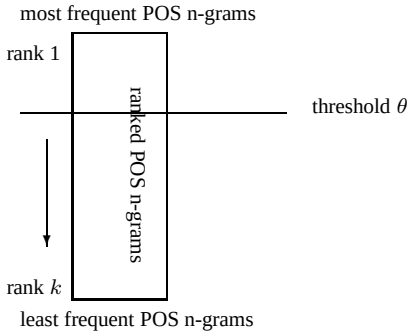

Figure 1: POS n-grams ranked by frequency.

We remove POS n-grams in a uniform way, i.e., by setting θ to the same value for all documents. We use the POS n-grams extracted from WT10G to reduce noise from the index of .GOV2. In particular, we use TreeTagger[3] for the POS tagging of WT10G. The POS n-grams extracted from WT10G provide us with a global list of POS n-grams. Overall, we extract 25,070 POS n-grams from WT10G with $\theta = 17,070$.

6.2 Experiments

In our participation in the Relevance Feedback track, we apply the DPH model (see Equation (6)) for document ranking, and the KL model (Equations (12) & (13)) for query expansion. Moreover, we take into account the proximity between the original query terms by applying the pBiL randomness model [17] as given by Equation (9), except that Normalisation 2 is not applied. We use these three parameter-free models for experimentation so that we focus on studying the query expansion aspect.

As the applied weighting models and the query expansion model are parameter-free, the only parameters that we need to fix are the number of expanded terms (exp_term) extracted from the exp_doc top-ranked documents. For training purposes, we use the 65 odd-numbered Terabyte track ad-hoc topics for which there are at least 4 relevant documents in the top 50 documents returned by DPH. We

[3]Details on the parameters and tagset used can be found in [16].

scan a wide range of possible values of exp_doc and exp_term, namely every exp_doc value within $2 \leq exp\_doc \leq 10$, and every exp_term value within $10 \leq exp\_term \leq 100$ with an interval of 5, in order to maximise MAP. We obtain $exp\_doc = 3$ and $exp\_term = 35$, which are used in our submitted runs.

We submitted two lists of runs for each set of feedback documents as follows. The first list, namely runs uogRF08.{A-E}1, applies query expansion on surrogates of the positive documents for the feedback sets B-E, and on surrogates of the top-3 returned documents for the feedback set A. The second list, namely runs uogRF08.{A-E}2, applies query expansion on positive documents for sets B-E, and on the top-3 returned documents for set A.

Table 11 provides the retrieval performance of our submitted runs as given by three different evaluation measures: Top10 AP, MCT AP, and Stat AP. From Table 11, we conclude the following:

1. It is beneficial to use the positive feedback documents for query expansion. The retrieval performance for sets B-E is markedly higher than that of set A.

2. Query expansion on document surrogates has a better retrieval performance in terms of Top10 AP than query expansion on the raw documents.

Run	Top10 AP	MCT AP	Stat AP
Positive QE on Surrogates			
uogRF08.A1	0.1971	0.0560	0.2919
uogRF08.B1	0.2202	0.0641	0.3075
uogRF08.C1	0.2274	0.0673	0.3205
uogRF08.D1	0.2323	0.0673	0.3379
uogRF08.E1	0.2272	0.0661	0.3420
Positive QE			
uogRF08.A2	0.1921	0.0563	0.2843
uogRF08.B2	0.2088	0.0658	0.3092
uogRF08.C2	0.2118	0.0697	0.3222
uogRF08.D2	0.2219	0.0695	0.3393
uogRF08.E2	0.2234	0.0682	0.3373
TREC median	0.1427	0.0564	0.1946

Table 11: Results of submitted runs in the Relevance Feedback track.

We have also conducted additional experiments to test the effectiveness of the techniques applied in our participation using the 31 topics for which the top-10 returned documents were judged by assessors. We obtain the following results:

- First, we evaluate the usefulness of pseudo-query expansion compared with the first-pass retrieval. We obtain MAP 0.1368 for first-pass retrieval, and MAP 0.1982 for pseudo-query expansion using the top-3 returned documents for relevance feedback. Pseudo-query expansion provides a 44.88% statistically significant improvement over the first-pass retrieval, which attests the effectiveness of this technique.

- Second, we test if the use of positive feedback documents (Pos QE) provides a better retrieval performance than pseudo-query expansion (Pseudo-QE). From Table 12, we can see that positive feedback documents do bring useful information, particularly for the settings which have a relatively larger number of feedback documents (see results for sets D and E).

- Third, we evaluate if query expansion on document surrogate improves the retrieval performance. Table 13 provides the related results. In general, we find no obvious difference between query expansion on the whole documents and query expansion on the document surrogates on the 31 topics used. No statistically significant difference is found between their corresponding MAP values.

Set	Pseudo-QE	Pos QE	*diff.*
B	0.1982	0.2153	8.63
C	0.1982	0.2240	13.02
D	0.1982	0.2330	17.56*
E	0.1982	0.2343	18.21*

Table 12: The MAP values obtained by (pseudo) query expansion (QE), and by positive query expansion (Pos QE). A statistically significant difference between the two MAP values at the 0.05 confidence level is marked with a star.

Set	QE/Pos QE	Sur QE	*diff.*
A	0.1982	0.2001	0.96
B	0.2153	0.2187	1.58
C	0.2240	0.2275	1.56
D	0.2330	0.2376	1.97
E	0.2343	0.2272	-3.03

Table 13: The MAP values obtained by pseudo/positive query expansion (QE/Pos QE), and by query expansion on document surrogates (Sur QE).

Overall, we have shown that expanding queries on positive documents is markedly better than pseudo-relevance feedback. We have also shown that it is beneficial to expand the queries over a refined representation of the feedback documents, namely, the document surrogates. In addition, we have investigated the usefulness of negative feedback documents for query expansion based on an adaptation of Rocchio's relevance feedback algorithm [33] to the DFR query expansion framework. We have not found the negative documents to be useful for relevance feedback, in line with the findings of other participants.

7. ENTERPRISE TRACK: EXPERT SEARCH TASK

We participate in the expert search task of the TREC 2008 Enterprise track with the aim of continuing to test and develop our novel Voting Model [22]. 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.

In our participation in the expert search task this year, we follow our central themes of proximity and enrichment. In particular, we use an advanced proximity extension to the Voting Model, which uses an information-theoretic DFR model to calculate the informativeness of a candidate's name occurring in close proximity to the terms of the query. Moreover, we enrich the profiles of the candidate experts to obtain better evidence of their expertise.

7.1 Voting Model

In expert search, the expertise areas of the candidates are represented to the system by documentary evidence of expertise, known as candidate profiles. In our Voting Model for expert search, instead of directly ranking candidates using these profiles, 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 [22], 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. In expCombMNZ, the score of a candidate C's expertise to a query Q is given by:

$$score(C,Q) = |R(Q) \cap profile(C)| \cdot \sum_{d \in R(Q) \cap profile(C)} exp(score(d,Q)) \quad (21)$$

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

Some types of documents can have many topic areas and many occurrences of candidate names (e.g., 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 [6, 29]. To this end, we define a voting technique, based on expCombMNZ, which takes into account the proximity of each candidate's name occurrence to the query terms in the documents [19]. We measure this proximity using the DFR term proximity model defined in Section 2.2. This model is designed to measure the informativeness of a pair of query terms occurring in close proximity in a document. We adapt this to the expert search task and into the expCombMNZ voting technique (Equation (21)), by measuring the informativeness of a query term occurring in close proximity to a candidate's name. The adapted voting technique, expCombMNZProx, is given as follows:

$$score(C,Q) = |R(Q) \cap profile(C)| \cdot \sum_{d \in R(Q) \cap profile(C)} exp\Big(score(d,Q) + \sum_{p=name(C) \times t \in Q} score(d,p) \Big) \quad (22)$$

Here, $p = \langle t, C \rangle$ is a tuple 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 [17]; for efficiency reasons, we use the pBiL2 model (Equation (9)), as it does not consider the frequency of the 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 that used in other TREC tracks (Blog, Relevance Feedback) to weight query term occurrences in close proximity. Note that this approach 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.

Some voting techniques in the Voting Model can suffer from being biased towards candidates with large candidate profiles (many associated documents). To neutralise this effect, we apply a normalisation function that is called Norm2D [21]:

$$score(C,Q) = score(C,Q) \cdot \log\left(1 + c_{pro} \cdot \frac{avg\_pro}{|profile(C)|}\right) \quad (23)$$

where c_{pro} is a free parameter ($c_{pro} > 0$), $|profile(C)|$ is the size of the profile of candidate C, measured as the number of documents, and avg_pro is the average size of the profile of all candidates. This is inspired by the Normalisation 2 from the DFR framework (Equation (3)).

7.2 Enriching Candidate Profiles

In keeping with our TREC theme this year, we investigate how enterprise data can be enriched by an external source of evidence. In [30], Serdyukov & Hiemstra proposed the use of external evidence in expert search, in particular, by using queries submitted to commercial Web search engines. In this work, we follow their suggestion for identifying useful external evidence. However, we develop more refined methods for ranking the experts. In particular, we actually download and rank all of the expertise evidence derived from a given source.

In order to identify expertise evidence for the candidates on the Web, we build new queries, which we call "evidence identification queries". These evidence identification queries involve both the actual expert search query (from the TREC 2007 Expert search task), and the name of a candidate. We then submit these evidence identification queries to the APIs of a major Web search engine, which will allow Web documents specific to the query and to the candidate to be retrieved. In particular, each query contains:

- the quoted full name of the person: e.g., *"craig macdonald"*,
- the name of the organisation: e.g., *csiro*,
- query terms without any quotations: e.g., *genetic modification*,
- a directive prohibiting any results from the actual organisation Web site: *-site:csiro.au*.

The use of the name of the organisation helps in name disambiguation, to prevent the matching of any content not related to the candidate expert in question. However, this will also prevent the matching of evidence for a candidate from a previous employer. The prohibitive *-site* directive, in turn, ensures that the acquired expertise evidence does not overlap with the intranet collection.

For each of the 50 topics in the TREC 2008 expert search task, we submit the evidence identification queries to the Yahoo! Web search engine, for the top 100 candidates suggested by our baseline expert search engine. From the search listing results, we extract a list of URLs associated to each candidate. For each expert identification query, a maximum of 24 results are extracted, and the corresponding Web pages downloaded. These pages form the profiles of the candidates. Note that these new profiles are *query-biased*, as only documents which are related to query topic(s) are associated to each candidate.

To create the runs, we rank all downloaded documents which have been returned by the Yahoo! Web search engine using the PL2 document weighting model (Equations (1) & (3)). Then, a voting technique is applied to convert this ranking of documents into a ranking of candidates using the query-biased profiles.

7.3 Experiments Setup, Runs, and Results

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

- uogTrEXFeMNZ is our baseline run (unsubmitted). It applies the PL2F DFR document weighting model (see Equations (1) & (2)) to generate the underlying ranking of documents from the CERC collection, combined with the expCombMNZ (Equation (21)) voting technique to rank experts.
- uogTrEXFeMNZP is a baseline run (unsubmitted), which improves upon the baseline run by applying query-term proximity (Equation (9)) before expCombmMNZ.
- uogTrEXfeNP improves upon uogTrEXFeMNZP by applying candidate size normalisation on top of expCombMNZ.
- uogTrEXfePC improves upon the baseline run by applying candidate-query term proximity, expCombMNZProx (Equation (22)), instead of expCombMNZ.
- uogTrEXfeNPC combines the previous two runs, by applying expCombMNZProx, and candidate size normalisation.
- uogTrEXeY is a baseline run (unsubmitted), which applies the PL2 DFR document weighting model on the externally obtained Yahoo! document index.
- uogTrEXmix combines uogTrEXfeNPC and uogTrEXeY by a linear mixture combination.

The salient features of the runs are described in Table 14. Note that, for all runs using the CERC collection, we use the candidate profiles identified during our TREC 2007 participation [9].

Table 15 presents the retrieval performance of the runs described above, as well as the per-topic median and best runs from all TREC participants. From the results, we draw the following observations: all of our submitted runs were above median; our best performing run was uogTrEXfeNPC, which applied expCombMNZProx and normalisation; applying query term proximity did not appear to benefit retrieval performance for MAP, but did improve MRR (uogEXFeMNZ vs. uogEXFeMNZP); candidate normalisation improved retrieval performance (uogEXFeMNZP vs. uogTrEXfeNP and uogTrEXfePC vs. uogTrEXfeNPC); candidate-query term proximity was more effective than query-term proximity, or the baseline (uogTrEXfePC vs. uogEXFeMNZ and uogEXFeMNZP), while applying normalisation on top was more beneficial (uogTrEXfePC vs. uogTrEXfeNPC); lastly, the usefulness of Yahoo! for expertise evidence mining was disappointing (uogTrEXeY), and hindered retrieval performance when combined with uogTrEXfeNPC.

Run Name	Submitted	MAP	MRR	P@10
TREC best	-	0.6844	0.9909	-
TREC median	-	0.3491	0.7829	-
uogEXFeMNZ	✘	0.3484	0.6550	0.3130
uogEXFeMNZP	✘	0.3444	0.7072	0.3148
uogTrEXfeNP	✔	0.3535	0.7079	0.3218
uogTrEXfePC	✔	0.3969	0.7259	0.3636
uogTrEXfeNPC	✔	**0.4126**	**0.7611**	**0.3727**
uogTrEXeY	✘	0.2428	0.5868	0.2436
uogTrEXmix	✔	0.3748	0.7600	0.3473

Table 15: Retrieval performances of the our Enterprise track expert search task runs, and also the TREC per-topic best and median runs.

Overall, we conclude that we successfully participated in the TREC 2008 expert search task of the Enterprise track. All of our submitted runs were above median, and our normalisation and candidate query-term proximity features were successful at increasing baseline retrieval performance. The low performance of the Web-enriched candidate profiles requires further investigation.

8. BLOG TRACK: BLOG DISTILLATION TASK

In TREC 2008, we also participate in the blog distillation task of the Blog track, where we aim to test the applicability of our Voting Model [22] to this task. Firstly, in the blog distillation task, the aim of each system is to identify the blogs that have a principle recurring interest in the query topic [24]. We believe that this 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. In [9] & [20], we showed that the task can be successfully modelled using the Voting Model. With this in mind, many of the techniques we apply in this task are described in Section 7 above: for each candidate expert C, read blog; for each document d, read blog post. The set of posts of each blog forms the blog's "candidate profile".

We also investigate the use of a feature which ascertains if the retrieved posts in a given blog for a topic are spread across the time span of the collection. If a blogger has an interest in a topic area, it is likely that he or she will continue to blog about the topic area repeatedly and frequently. Indeed, the definition for a relevant blog in the blog distillation task gives a clue that the timing of on-topic posts by a blog may have an impact on the overall relevance of the blog. In particular, we believe that a relevant blog will continue to post relevant posts throughout the timescale of the collection.

With this in mind, we break the 11-week period of the Blogs06 collection into a series of DI equal intervals (where DI is a parameter). Then, for each blog, we measure the proportion of its posts in each time interval that were retrieved in response to a query. We call this evidence *recurring interests* (Dates), and define a $Qscore_{Dates}(C,Q)$ for each blog C as follows [20]:

$$Qscore_{Dates}(C,Q) = \sum_{i=1}^{DI} \frac{1+|R(Q) \cap dateInterval_i(posts(C))|}{1+|dateInterval_i(posts(C))|} \quad (24)$$

where $dateInterval_i(posts(C))$ is the number of posts of blog C in the ith date interval. Note that we smooth this probability distribution using Laplace smoothing to combat sparsity problems (e.g., when a blog had no posts in a date interval). We integrate the $Qscore_{Dates}(B,Q)$ evidence as:

$$score(C,Q) = score(C,Q) \times Qscore_{Dates}(C,Q)^{\omega} \quad (25)$$

where $\omega > 0$ is a free parameter. We use $DI = 3$, which approximates the month where the post was made (the corpus time span is 11 weeks), and $\omega = 0.48$. Initial experiments found that using higher values for DI does not change the results, due to the time span of the corpus. Finally, note that, as this evidence requires knowledge of the ranking of posts for a query, it has to be calculated during the retrieval phase, but without adding high overheads.

We submitted 4 runs to the blog distillation task of the TREC 2008 Blog Track, which test our hypotheses for this task. Below, we describe our submitted and unsubmitted runs:

- uogTrBDfe is our baseline run. It uses the PL2F weighting model together with the expCombMNZ voting technique to score the predicted relevance of blogs to the query topic.

Run Name	Submitted	Source	Salient Features
uogEXFeMNZ	✘	CERC	PL2F + expCombMNZ
uogEXFeMNZP	✘	CERC	PL2F + Proximity + expCombMNZ
uogTrEXfeNP	✔	CERC	+ Norm2D
uogTrEXfePC	✔	CERC	PL2F + expCombMNZProx
uogTrEXfeNPC	✔	CERC	+Norm2D
uogTrEXeY	✘	Yahoo!	PL2 + expCombMNZ
uogTrEXmix	✔	(both)	mixture of uogTrEXeY & uogTrEXfeNPC

Table 14: Salient features of our Enterprise track expert search task runs.

- uogTrBDfeN is an unsubmitted baseline run. In addition to PL2F and expCombMNZ, it applies Norm2D to remove any bias towards prolific bloggers.
- uogTrBDfeNP improves on the baseline run, in two ways: Firstly, by boosting the rank of documents in the document ranking where the query terms occur in close proximity using the pBiL2 DFR terms dependence model (Equation (9)); secondly, we use the Norm2D normalisation technique (Equation (23)), which has been shown to be useful on this task [20].
- uogTrBDfeNPD investigates the use of the recurring interests quality evidence, compared to uogTrBDfeNP.
- uogTrBDfeNWD investigates the use of Wikipedia for collection enrichment, compared to uogTrBDfeNP, similarly to our collection enrichment approach in the document search task (Section 5). In particular, to enrich the topics, we apply the Bo1 term weighting model (Equation (10)) on the full content of a Wikipedia index from 2008, and using a pseudo-relevance set of size $exp\_doc = 30$ documents, we expand each query with $exp\_term = 10$ additional terms.

Table 16 summarises the salient features of the submitted and unsubmitted runs. Table 17 presents the results of our submitted runs in the blog distillation task of the Blog track. The evaluation measures in this task are mean average precision (MAP), mean reciprocal rank (MRR), and precision at rank 10 (P@10).

Run Name	Submitted	Salient Features
uogTrBDfe	✔	PL2F & expCombMNZ
uogTrBDfeN	✘	+ Norm2D
uogTrBDfeNP	✔	+ pBiL2 + Norm2D
uogTrBDfeNPD	✔	+ pBiL2 + Norm2D + Dates
uogTrBDfeNWD	✔	+ Wikipedia + Norm2D + Dates

Table 16: Salient features of our Blog track blog distillation task submitted and unsubmitted runs.

Run Name	MAP	MRR	P@10
TREC median	0.2224	-	-
uogTrBDfe	0.2028	0.6595	0.3560
uogTrBDfeN	0.2337	0.6948	0.3720
uogTrBDfeNP	0.2395	0.7267	0.3800
uogTrBDfeNPD	0.2437	0.7310	0.3740
uogTrBDfeNWD	0.2521	0.7425	0.4040

Table 17: The mean average precision (MAP), Reciprocal Rank (MRR), and precision at 10 (P@10) of our Blog track blog distillation task runs, as well as that achieved by all participants. MRR and P@10 achieved by all participants is not available.

From the results, we note that applying any of Norm2D, pBiL2 (proximity), Dates or Wikipedia-based collection enrichment results in an increase in retrieval effectiveness in comparison to the baseline run. Moreover, the incremental combination of these techniques brings further improvements, suggesting that each of them may address a different dimension of the blog distillation problem.

9. CONCLUSIONS

In TREC 2008, we have participated in three tracks, namely the Blog track, the Enterprise track, and the Relevance Feedback track. In particular, we have investigated the effectiveness of our proximity-based models in different tasks as well as the usefullness of external resources for collection and profile enrichment.

In the Blog track opinion-finding task, our main conclusion is that our proposed approaches have significantly improved over all but one of the 7 baselines (our proximity-based approaches do not improve over baseline 4, while the approaches that do not use proximity do not improve over baseline 2). Moreover, the opinion-finding performance of our best-performing runs for all but two baselines has increased across the topics for the three years of the Blog opinion-finding task, with our best topic-relevance performances observed for the 2007 topics over all baselines.

In the Blog track blog distillation task, we have shown that all of our approaches individually improve over the baseline and also over the median of the participating groups. Additionally, the combination of these individual approaches improves even further.

In our participation in the first Relevance Feedback track, we have shown that query expansion on the set of positive feedback documents markedly improves over the first-pass retrieval baseline. Furthermore, the application of query expansion on the document surrogates rather than the raw documents has shown improved performance in terms of Top10 AP.

For the Enterprise track, we have investigated the application of suitable external resources. In the document search task, we have shown that using external resources through collection enrichment to enhance the retrieval performance is very effective. In particular, the selective application of collection enrichment according to the query performance makes a significant improvement in MAP. Moreover, it is very important to choose an appropriate external resource and an appropriate query performance predictor before applying the collection enrichment approach.

In the Enterprise track expert search task, we have successfully applied our Voting Model to rank candidate experts. Moreover, we have investigated the application of candidate size normalisation, candidate query-term proximity, and the enrichment of candidate profiles using a commercial Web search engine. Both candidate size normalisation and candidate-query term proximity have been shown to improve the retrieval performance.

Acknowledgements

We thank Alasdair Gray and Richard McCreadie for assisting us in our TREC assessment workload this year. The work within the Relevance Feedback track is funded by the SIMAP: Simulation modelling of the MAP kinase pathway, EC project 2006-2009.

10. REFERENCES

[1] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence From Randomness*. PhD thesis, Dept.of Computing Science, Univ. of Glasgow, 2003.

[2] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog Track. In *Proceedings of TREC 2007*.

[3] G. Amati, C. Carpineto, and G. Romano. Italian Monolingual Information Retrieval with Prosit. In *Proceedings of CLEF 2001*.

[4] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. Class-Based n-Gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479, 1992.

[5] F. Cacheda, V. Plachouras, and I. Ounis. A Case Study of Distributed Information Retrieval Architectures to Index One Terabyte of Text. *IP&M*, 41(5), 2005.

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

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

[8] N. Craswell and D. Hawking. Overview of TREC-2004 Web track. In *Proceedings of TREC 2004*.

[9] D. Hannah, C. Macdonald, J. Peng, B. He and I. Ounis. University of Glasgow at TREC2007: Experiments in Blog and Enterprise tracks with Terrier. In *Proceedings of TREC 2007*.

[10] D. Hawking, T. Upstill, and N. Craswell. Towards better Weighting of Anchors. In *Proceedings of SIGIR 2004*.

[11] B. He, C. Macdonald, and I. Ounis. Ranking opinionated blog posts using OpinionFinder. In *Proceedings of SIGIR 2008*.

[12] B. He, C. Macdonald, J. He, and I. Ounis. An effective statistical approach to blog post opinion retrieval. In *Proceedings of CIKM 2008*.

[13] B. He, I. Ounis. Query performance prediction. In *Proceedings of SPIRE 2004*.

[14] K. L. Kwok, L. Grunfeld, M. Chan, N. Dinstl, C. Cool. TREC-7 Ad-Hoc, High Precision and Filtering Experiments using PIRCS. In *Proceedings of TREC 1998*.

[15] C. Lioma and I. Ounis. A Syntactically-Based Query Reformulation Technique for Information Retrieval. *IP&M*, 44(1), 2008.

[16] C. Lioma and I. Ounis. Examining the Content Load of Part of Speech Blocks for Information Retrieval. In *Proceedings of ACL 2006*.

[17] 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*.

[18] C. Macdonald. *The Voting Model for People Search*. PhD thesis, Dept. of Computing Science, Univ. of Glasgow, 2009.

[19] C. Macdonald and I. Ounis. High quality evidence of expertise. In *Proceedings of ECIR 2008*.

[20] C. Macdonald and I. Ounis. Key blog distillation: ranking aggregates. In *Proceedings of CIKM 2008*.

[21] C. Macdonald and I. Ounis. Searching for Expertise: Experiments with the Voting Model. *Computer Journal*, in press. 2008.

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

[23] C. Macdonald and I. Ounis. The TREC Blog06 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`

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

[25] 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. In *Proceedings of CLEF 2005*.

[26] 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 Workshop at SIGIR 2006*.

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

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

[29] D. Petkova, W.B. Croft. Hierarchical language models for expert finding in enterprise corpora. In *Proceedings of ICTAI 2006*.

[30] P. Serdyukov and D. Hiemstra. Being Omnipresent To Be Almighty: The Importance of Global Web Evidence for Organizational Expert Finding. In *Proceedings of fCHER Workshop at SIGIR 2008*.

[31] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of CIKM 2004*.

[32] S. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In *Proceedings of TREC 4*.

[33] J. Rocchio. Relevance Feedback in Information Retrieval. In *The Smart Retrieval system – Experiments in Automatic Document Processing.* Salton, G., Ed., 313-323, Prentice-Hall Englewood Cliffs, N.J., USA, 1971.

[34] R. L. T. Santos, B. He, C. Macdonald, and I. Ounis. Integrating proximity to subjective sentences for blog opinion retrieval. In *Proceedings of ECIR 2009*.

[35] 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*.

[36] 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*,