University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise Tracks with Terrier

Christina Lioma, Craig Macdonald, Vassilis Plachouras, Jie Peng, Ben He, Iadh Ounis

Department of Computing Science
University of Glasgow
Scotland, UK

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

ABSTRACT

In TREC 2006, we participate in three tasks of the Terabyte and Enterprise tracks. We continue experiments using Terrier[1], our modular and scalable Information Retrieval (IR) platform. Furthering our research into the Divergence From Randomness (DFR) framework of weighting models, we introduce two new effective and low-cost models, which combine evidence from document structure and capture term dependence and proximity, respectively. Additionally, in the Terabyte track, we improve on our query expansion mechanism on fields, presented in TREC 2005, with a new and more refined technique, which combines evidence in a linear, rather than uniform, way. We also introduce a novel, low-cost syntactically-based noise reduction technique, which we flexibly apply to both the queries and the index. Furthermore, in the Named Page Finding task, we present a new technique for combining query-independent evidence, in the form of prior probabilities. In the Enterprise track, we test our new voting model for expert search. Our experiments focus on the need for candidate length normalisation, and on how retrieval performance can be enhanced by applying retrieval techniques to the underlying ranking of documents.

1. INTRODUCTION

The research scope underlying our participation in TREC 2006 has been to extend our current robust weighting models and retrieval performance enhancing techniques, in novel ways that are theoretically-sound, modular, low-cost, and most importantly, effective. In terms of weighting models, we present two new Divergence From Randomness (DFR) models. The first model aims at combining evidence from document structure, and we test it in the Named Page Finding task of the Terabyte track. The second model aims at modelling term dependence and proximity, and we test it in the Named Page Finding task of the Terabyte track and the Expert Search task of the Enterprise track. In terms of retrieval performance enhancing techniques, we present (i) a refined query expansion mechanism on fields, which combines document field evidence in a linear way, and (ii) a novel noise reduction mechanism for long queries and the index, which uses syntactically-based evidence (parts of speech). We test these two techniques in the Adhoc task of the Terabyte track. We also present a new technique for combining query-independent evidence, in the form of prior probabilities. We test this technique in the Named Page Finding task of the Terabyte track.

In the Enterprise track, we test our novel voting model for expert search. Firstly, we experiment on how candidate length normalisation can be used in the voting model to prevent candidates with too much expertise evidence from gaining an unfair advantage in the voting model. Secondly, we examine how a selection of state-of-the-art retrieval techniques, such as a field-based weighting model, query expansion and term dependence and proximity, can be used to enhance the retrieval performance of the expert search system, by enhancing the quality of an underlying ranking of documents. Conclusions are drawn across two ways of associating documents with candidates to represent their expertise.

The remainder of this paper is organised as follows. Section 2 presents the weighting models used in the Terabyte and the Enterprise tracks. Section 3 presents the hypotheses tested and techniques applied in the Adhoc and Named Page Finding tasks of the Terabyte track, with a discussion of the results. Section 4 presents the hypotheses tested and techniques applied in the Enterprise track, with a discussion of the results. Section 5 summarises our overall participation in TREC 2006.

2. MODELS

Following from previous years, our research in Terrier centres in extending the Divergence From Randomness framework (DFR) [1]. In TREC 2006, we have devised novel, information-theoretic ways of combining evidence from document structure (or fields, such as the title and anchor text), and in modelling term dependence and proximity. Both proposed models are based on the DFR framework, and they are applied very effectively and with little computational overhead.

The remainder of this section is organised as follows. Section 2.1 presents existing field-based DFR weighting models. Section 2.2 introduces our new field-based DFR weighting model, while Section 2.3 presents our new DFR model, which captures term dependence and proximity.

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. [23] 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

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

of the fields. In addition, Hawking et al. [5] showed that the length normalisation that should be applied to each field depends on the nature of the field. Zaragoza et al. [25] introduced a field-based version of BM25, called BM25F, which applies length normalisation and weighting of the fields independently. Macdonald et al. [11] also introduced *Normalisation 2F* in the DFR framework for performing independent term frequency normalisation and weighting of fields.

In this work, we use two field-based models from the DFR framework, namely PL2F and InL2F. 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.

For InL2F, the relevance score of a document d for a query Q is given by:

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

where n_t is the number of documents term t occurs in.

In both PL2F and InL2F, tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f, known as *Normalisation 2F* [11]:

$$tfn = \sum_f \left(w_f \cdot tf_f \cdot \log_2(1 + c_f \cdot \frac{avg\_l_f}{l_f}) \right), (c_f > 0) \quad (3)$$

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 models PL2 and InL2 can be generated by using *Normalisation 2* instead of Normalisation 2F for tfn in Equations (1) & (2) above. Normalisation 2 is given by:

$$tfn = tf \cdot \log_2(1 + c \cdot \frac{avg\_l}{l}) (c > 0) \quad (4)$$

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 the document length.

Note that, following [23], we have also devised a simplified variant of Normalisation 2F, which normalises the sum of the weighted term frequencies from different fields, instead of normalising the term frequencies on a per-field basis. Indeed, this simplified variant of Normalisation 2F allows us to reduce training time, because it has less hyper-parameters to train. The simplified Normalisation 2F, which we denote as Normalisation 2FS, is given as follows:

$$tfn = \sum_f \big(w_f \cdot tf_f \cdot \log_2(1 + c \cdot \frac{avg\_l}{l})\big), (c > 0) \quad (5)$$

Normalisation 2F in Equation (3) has a hyper-parameter c_f for each indexed document field. Unlike Normalisation 2F, Normalisation 2FS has only a single hyper-parameter c for all the indexed document fields. Therefore, we can benefit from having less hyper-parameters to train. In our previous experiments for Adhoc retrieval, we found no significant difference between the retrieval performance obtained using PL2F and PL2FS. For example, for the TREC-9 Web Adhoc task, using title-only queries, the optimised mean average precision (MAP) of PL2F and PL2FS is 0.2071 and 0.2062, respectively. The p-value is 0.07858 using the Wilcoxon matched-pairs signed-ranks test, which indicates an insignificant difference at 5% confidence level.

2.2 Multinomial Divergence From Randomness (DFR) Weighting Model

In TREC 2006, we re-investigate the use of document structure (or fields) in the DFR framework. In both BM25F and the DFR models that employ Normalisation 2F (e.g. PL2F, InL2F), it is assumed that the occurrences of terms in the fields follow the same distribution, because the combination of fields takes place in the document length normalisation component, and not in the probabilistic model [18].

In TREC 2006, we take a different approach by considering that the term occurrences in the fields of documents follow a multinomial distribution. In this way, the combination of the term occurrences from the different fields is modelled in a probabilistic way, and is not part of the document length normalisation.

We introduce a new DFR weighting model, which employs a multinomial randomness model, as follows. The weight of a term in a document ($score(d,t)$) is equal to the product of the information content of two probabilities. Therefore, the relevance score of a document d for a query Q is computed as follows:

$$score(d,Q) = \sum_{t \in Q} qtw \cdot score(d,t) = \sum_{t \in Q} \big(qtw(-\log_2(P_1)) \cdot (1 - P_2)\big) \quad (6)$$

P_1 corresponds to the probability that there is a given number of term occurrences in the fields of a document. P_2 corresponds to the probability of having one more occurrence of a term in a document, after having seen it a given number of times. The probability P_1 is computed using a multinomial randomness model:

$$P_1 = \binom{F}{tfn_1 \quad tfn_2 \quad \dots \quad tfn_k \quad tfn'} \cdot p_1^{tfn_1} \cdot p_2^{tfn_2} \cdots p_k^{tfn_k} \cdot p'^{tfn'} \quad (7)$$

The probability P_2 is computed using the Laplace after-effect model.

$$P_2 = \frac{\sum_f tfn_f}{1 + \sum_f tfn_f} \quad (8)$$

In the above equations, k is the number of fields, tfn_f is the normalised frequency of a term in the field f, which is given by applying Normalisation 2 from Equation (4) to that field. F and N are as defined in Section 2.1. $tfn' = F - \sum_f tfn_f$; p_f is the prior probability of having a term occurrence in the field f of a document, and it is equal to $p_f = \frac{1}{k \cdot N}$; $p' = 1 - \sum_f p_f = \frac{N-1}{N}$.

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

$$score(d,Q) = \sum_{t \in Q} \Big(\frac{qtw}{1+\sum_f tfn_f} \cdot \Big(-\log_2(F!) + \sum_f \big(\log_2(tfn_f!) - tfn_f \log_2(p_f) \big) + \log_2(tfn'!) - tfn' \log_2(p') \Big) \Big) \quad (9)$$

We refer to the multinomial DFR model described in Equation (9) as ML2. In the above equation, the logarithm of the factorial is computed using the Lanzcos approximation of the Γ function [21, p. 213]. The Lanzcos approximation is preferred over the Stirling approximation because it results in lower error [18].

2.3 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 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:

$$score(d,Q) = \sum_{t \in Q} qtw \cdot score(d,t) + \sum_{p \in Q_2} score(d,p) \quad (10)$$

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 (10), 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:

$$score(d,p) = -\log_2(P_{p1}) \cdot (1 - P_{p2}) \quad (11)$$

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.

For example, applying term dependence and proximity with the weighting model InL2 (see Equations (2) and (4)), results in a new version of InL2, which we denote by pInL2, where the prefix p stands for proximity. pInL2 estimates $score(d,p)$ as follows:

$$score(d,p) = \frac{1}{tfn_p + 1} \big(tfn_p \cdot \log_2 \frac{N+1}{n_p + 0.5} \big) \quad (12)$$

where n_p corresponds to the number of documents in which the pair of query terms p appear within $dist$ terms of each other. tfn_p is the normalised frequency of a query term pair p in document d, which can be obtained from applying Normalisation 2 from Equation (4).

A different randomness model, which does not consider the collection frequency of pairs of query terms, is based on the binomial randomness model, and computes the score of a pair of query terms in a document as follows:

$$score(d,p) = \frac{1}{tfn_p + 1} \cdot \Big(- \log_2 (l-1)! + \log_2 tfn_p! + \log_2 (l - 1 - tfn_p)! - tf_p \log_2(p_p) - (l - 1 - tfn_p) \log_2(p'_p) \Big) \quad (13)$$

where $p_p = \frac{1}{l-1}$ and $p'_p = 1 - p_p$. We refer to this binomial DFR model described in Equation (13) as pBiL2.

3. TERABYTE TRACK

In the TREC 2006 Terabyte Track, we participate in the Adhoc and Named Page Finding tasks.

We index the .GOV2[2] collection using Terrier [16], in seven parts (each part having an average size of 3.6 million documents). To support our investigation into the use of document field evidence in retrieval, each of the seven parts consists of three inverted files, one for each of the following document fields: body, title, and anchor text. Standard stopwords are removed from each index. We apply Porter's full stemming for our Adhoc experiments, and Porter's weak stemming for our Named Page Finding experiments. Our choice of stemming is justified by the observation that weak stemming, being less aggressive than full stemming, is better suited for high-precision tasks, such as Named Page Finding.

Following our experiments in the TREC 2005 Terabyte track [10], we use a distributed version of Terrier to reduce up retrieval time. In TREC 2006, we use one broker, and seven query servers, each serving one index part. Moreover, a global lexicon is created in order to speed up the retrieval process, particularly for query expansion.

In the Adhoc task, we adopt a dual approach that generally aims to boost query informativeness on one hand, and reduce noise on the other hand. We boost query informativeness by incorporating different combinations of document field evidence into our query expansion mechanism. We reduce noise using part-of-speech evidence. Specifically, we investigate the following hypotheses:

- H1 For the query expansion mechanism on fields, the linear combination of fields can provide a better retrieval performance, than the uniform combination of fields (Section 3.1.1).
- H2 In a collection of documents, low frequency part-of-speech n-grams correspond to noisy sequences of words, which if removed, can enhance retrieval performance (Section 3.1.2).

In the Named Page Finding task, we investigate a new way of modelling term occurrence in document fields, and a novel theoretically-founded approach for combining multiple sources of query independent evidence. More specifically, we test the following hypotheses:

- H3 Modelling the distribution of term occurrences in document fields as a multinomial distribution is a theoretically-sound and robust approach, which performs at least comparably to other field-based weighting models (Section 2.2).
- H4 Modelling the dependence and proximity of query terms in documents can enhance retrieval effectiveness (Section 2.3).

[2]Information on .GOV2 can be found from http://ir.dcs.gla.ac.uk/test_collections/gov2-summary.htm

H5 Using the conditional combination of multiple sources of query independent evidence, in the form of prior probabilities, can improve retrieval performance over using either source of evidence alone (Section 3.2.1).

The remainder of Section 3 is organised as follows. Section 3.1 presents the linear combination of fields for query expansion (Section 3.1.1), and syntactically-based noise reduction (Section 3.1.2). Section 3.1.3 presents our Adhoc experiments. Section 3.2 presents our participation in the TREC 2006 Named Page Finding task, with an introduction of the techniques tested in Section 3.2.1, and a discussion of the experiments in Section 3.2.2. Section 3.3 summarises our participation in the TREC 2006 Terabyte Track, with conclusions and lessons learnt.

3.1 Adhoc Task

In TREC 2006 we extend our Terrier retrieval platform and implement two retrieval performance enhancing techniques, namely (i) query expansion, which combines document fields in a linear way, and (ii) syntactically-based noise reduction, which is applied to long queries and the index. We experiment with short (Title), and long (Title + Description + Narrative) queries, and report on our results.

3.1.1 Query Expansion on Document Fields

Continuing our experimentation in the TREC 2005 Terabyte Adhoc task, we aim to further improve our query expansion mechanism on document fields, by appropriately combining field evidence available in corpora (hypothesis H1, Section 3). This fine-grained query expansion mechanism uses statistics from various document fields, such as the title, the anchor text of the incoming links, and the body of documents. In TREC 2005, we applied a uniform combination of evidence from different document fields (QEFU) [10]. In TREC 2006, we replace this uniform combination with a more refined linear combination of evidence from different weighted fields (QEFL).

Our query expansion mechanism on document fields is built on top of the Bo1 term weighting model [1], which 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:

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

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 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.

We extend the above Bo1 term weighting model to deal with document fields by a linear combination of the frequencies of the query term in different fields:

$$tf_x = \sum_f wq_f \cdot tf_{xf} \qquad (15)$$

We call the term weighting model in Equation (14), where tf_x is given by Equation (15), the Bo1F term weighting model. In Equation (15), wq_f is the weight of a field f in the exp_doc top-ranked documents, which reflects the relative importance of the associated field in the top-ranked documents. tf_{xf} is the frequency of the query term in field f of the exp_doc top-ranked documents.

Terrier employs a parameter-free function to determine qtw, the query term weight of a query term, which is given as follows:

$$\begin{aligned} qtw &= \frac{qtf}{qtf_{max}} + \frac{w(t)}{\lim_{F \to tf_x} w(t)} \qquad (16) \\ &= F_{max} \log_2 \frac{1+P_{n,max}}{P_{n,max}} + \log_2(1+P_{n,max}) \end{aligned}$$

where qtf is the query term frequency of term t, and qtf_{max} is the maximum qtf among all the query terms in the expanded query; $\lim_{F \to tf_x} w(t)$ is the upper bound of $w(t)$; $P_{n,max}$ is given by $Fmax/N$, where $Fmax$ is the F value of the term with the maximum $w(t)$ in the exp_doc 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. The above formula is parameter-free in the sense that the parameter in Rocchio's query expansion function [24] has been omitted.

Using a field-based weighting model, e.g. PL2F (Equations (1) and (3)), together with Bo1F, there are six field weights involved, namely the weights (w_f) of the three document fields in the weighting model, and the weights (wq_f) of the three document fields in Bo1F. Since it would be very time-consuming to optimise all of these six field weights, we make the following assumptions to reduce the number of field weights to two:

1. For a given field f, we assume that $w_f = wq_f$. This is reasonable because the weight of a field reflects the contribution of the field to the document ranking, which should be consistent in both retrieval and query expansion.

2. Following [10] and [23] , we set the weight of the body field to 1.

By making the above two assumptions, we reduce the number of field weights from six to two, namely the weights of the anchor text and title fields. In addition, we apply the simplified Normalisation 2FS in Equation (5), instead of Normalisation 2F in Equation (3), so that we have only one c hyper-parameter.

In order to train the hyper-parameter c, the field weights, and the parameters exp_doc and exp_term, we adopt two different training strategies. The first training strategy (T1) optimises the parameters over all the 100 old topics used in the TREC 2004 and 2005 Terabyte Adhoc tasks. The parameter values that give the best MAP are used. The second training strategy (T2) splits these 100 old topics into two parts. Each part consists of the 50 topics used in the TREC 2004 or 2005 Terabyte Adhoc task. T2 optimises the parameters over each of the two parts of the old topics. The average of the optimised parameter values for the two parts of the old topics is used. We expect T2 to result in a better retrieval performance than T1 because T2 prevents the training process from being biased towards the set of topics that performs better. Indeed, on the TREC 2005 topics it is easier to achieve high retrieval performance than for the TREC 2004 topics.

3.1.2 Syntactically-based Noise Reduction

This section describes our technique for reducing estimated noise from long queries and documents. We use part-of-speech (POS) n-grams [3, 7] to detect noise in text.

POS n-grams are n-grams (or blocks) 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-gram length l = 4, the POS n-grams extracted are ABCD, BCDE, CDEF, and DEFG. The order in which the POS

Noise Reduction	θ	POS n-grams extracted from	Reduction
NR_q uniform	50	WT10G	47.22%†
	10	.GOV2	63.13%†
NR_q: query length ≤ 40	50	.GOV2	63.69%†
NR_q: query length $50 - 100$	10	.GOV2	
NR_q: query length > 100	5	.GOV2	
NR_iindex	17,070	WT10G	9.39%‡

Table 1: Syntactically-based Noise Reduction Settings. θ displays the value of the threshold in the POS n-gram ranking used. † and ‡ denote reduction in query length (in tokens) and in document pointers in the postings list, respectively.

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 [7]. To test the hypothesis that reducing noise from text using POS n-grams can enhance retrieval performance (H2, Section 3), firstly we reduce estimated noise from long queries in order to enhance retrieval performance by providing more informative queries [9]. We refer to this as NR_q. Secondly, we reduce estimated noise from the collection before it is indexed, in order to improve retrieval precision, at no detrimental cost to retrieval recall [8]. We refer to this as NR_i. The only resources needed are a POS tagger and a collection of documents. This can be any collection of documents of a reasonable size [9], not necessarily the collection from which we retrieve relevant documents.

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 extract POS n-grams from the text we wish to process, i.e. a long query (for NR_q), or a document from the collection to be indexed (for NR_i). 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 query or the document, regardless of any other POS n-grams that overlap it.

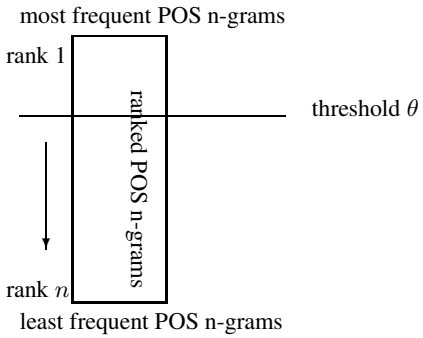

Figure 1: POS n-grams ranked by frequency.

In NR_q, we reduce estimated noise from long queries in two ways: firstly, uniformly for all queries (NR_qU); and secondly, individually on a per query basis (NR_qL). For NR_qU, we use the same threshold θ for all queries. For NR_qL, we use different values of θ according to query length. The intuition behind varying noise reduction according to query length is that the shorter the query, the less noise it is likely to contain. The values of θ according to different query lengths used are displayed in Table 1.

In NR_i we reduce estimated noise from the index from which relevant documents are retrieved. Again, we set the threshold θ, so that everything below θ is considered noisy and removed. We remove POS n-grams in a uniform way, i.e. by setting θ to the same value for all documents (Table 1).

After noise has been reduced using either of the noise reduction techniques described above, we treat the query or index as we would normally treat them. We use the TreeTagger[3] for the POS tagging of WT10G and .GOV2. The POS n-grams extracted from these collections provide us with two separate global lists of POS n-grams. Overall we extract 25,070 POS n-grams from WT10G and 47,018 POS n-grams from .GOV2. We use the POS n-grams extracted from WT10G or .GOV2 to reduce noise from the queries, and the POS n-grams extracted from WT10G to reduce noise from the index of .GOV2. We note that there is not much difference in the POS n-gram ranking between the two collections.

3.1.3 Experiments and Results

We submitted five runs to the Adhoc task. The first two submitted runs test the query expansion mechanism on fields with two different training strategies, respectively (as described in Section 3.1.1). The third submitted run tests the query expansion mechanism on fields with the first training strategy (T1), as well as noise reduction from long queries. The last two submitted runs test the query expansion mechanism on the body of documents only, with noise reduction from long queries and the index. Our collective submitted runs, and their salient features, are summarised in Table 2. The parameter values used in our submitted runs are given in Table 9. A full description of the submitted runs follows.

- *uogTB06QET1* uses the PL2FS weighting model with the simplified Normalisation 2FS; applies query expansion on fields (QEFL) using the Bo1F term weighting model, with training method T1, on short queries.
- *uogTB06QET2* uses the PL2FS weighting model with the simplified Normalisation 2FS; applies query expansion on fields (QEFL) using the Bo1F term weighting model, with training method T2, on short queries.
- *uogTB06S50L* uses the PL2FS weighting model with the simplified Normalisation 2FS; applies query expansion on fields (QEFL) using the Bo1F term weighting model, with training method T1, on long queries; applies uniform noise reduction from the queries (NR_qU), with POS n-grams drawn from WT10G, and $\theta = 50$.
- *uogTB06SS10L* uses the PL2 weighting model with Normalisation 2; applies query expansion on the documents (QE) using the Bo1 term weighting model, on long queries; applies uniform noise reduction from the queries (NR_qU), with POS n-grams drawn from .GOV2, and $\theta = 10$; applies noise reduction in the index (NR_i), with POS n-grams drawn from WT10G, and $\theta = 17,070$.
- *uogTB06SSQL* uses the PL2 weighting model with Normalisation 2; applies query expansion on the documents (QE)

[3]Details on the tagger parameters and tagset used can be found in [7]

Run	Weighting Model	Retrieval Features	Settings	Topic Fields
uogTB06QET1	PL2FS (Eq. 1 & 5)	Bo1F (Eq. 14 & 15)	QEFL: Training T1	T
uogTB06QET2	PL2FS (Eq. 1 & 5)	Bo1F (Eq. 14 & 15)	QEFL: Training T2	T
uogTB06S50L	PL2FS (Eq. 1 & 5)	Bo1F (Eq. 14 & 15), query noise reduction	QEFL: Training T1, NR_qU	TDN
uogTB06SS10L	PL2 (Eq. 1 & 4)	Bo1 (Eq. 14), query & index noise reduction	NR_qU, NR_iU	TDN
uogTB06SSQL	PL2 (Eq. 1 & 4)	Bo1 (Eq. 14), query & index noise reduction	NR_qL, NR_iU	TDN

Table 2: Salient features of submitted Adhoc runs.

using the Bo1 term weighting model, on long queries; applies noise reduction per query length (NR_qL), with POS blocks drawn from .GOV2. For queries of less than 40 words $\theta = 50$; for queries of 41 - 100 words, $\theta = 10$; for queries of more than 100 words, $\theta = 5$ (see Table 1). Applies noise reduction in the index (NR_i), with POS n-grams drawn from WT10G, and $\theta = 17,070$.

For our query expansion mechanism on fields, Table 3 compares the use of the linear combination of fields (QEFL) with the use of the uniform combination of fields (QEFU). The related additional runs corresponding to training method T1 (resp. T2) use the same parameter values applied in run uogTB06QET1 (resp. uogTB06QET2) (see Table 9). In Table 3, we see that the two different training methods have little impact on retrieval performance. Both training methods result in similar MAP performances for both QEFL and QEFU. Moreover, we observe that QEFL outperforms QEFU for both the 50 new topics, and all the 150 topics used. This indicates that our newly proposed linear combination of fields achieves a better retrieval performance than the uniform combination of fields.

In Table 4 we see that reducing estimated noise from the queries improves retrieval performance, compared to using no noise reduction, without query expansion (top part of Table 4, first row). The parameter values of the related additional runs are the same as those used in run uogTB06S50L (see Table 9). With query expansion on the body of documents only, query noise reduction results in slightly worse retrieval performance, compared to using query expansion without noise removal (second part of Table 4, first row). This could be due to the fact that we have trained our query expansion mechanism on long queries before noise reduction, but not on long queries after noise reduction. Query noise reduction reduces query length (from 47.22% to 63.69%, Table 1, column *Reduction*, marked †). Retraining the query expansion mechanism on the reduced queries could provide fairer grounds for comparing the effect of query noise reduction with query expansion. Additionally, in Table 4, we see no marked difference between using query noise reduction with query expansion on the body of the documents only, and using query noise reduction with query expansion on more document fields. Finally, we observe that removing noise from the index slightly damages MAP. However, it appears to benefit high-precision retrieval, as it provides the 2nd highest P@10 score of all official runs of all groups, namely P@10 = 0.6720.

Training	QEFU	QEFL	*diff.* (%)	p-value
50 New topics				
T1	0.3220	**0.3459**	+7.42	0.3396
T2	0.3248	**0.3456**	+6.40	0.1549
All 150 topics				
T1	0.3335	0.3558	+6.69	3.756e-04
T2	0.3338	0.3594	+7.67	9.001e-03

Table 3: MAP of the linear combination of fields vs. the uniform combination of fields. Title-only queries. The weighting model used is PL2F, with Bo1F for query expansion on fields. *QEFL+T1 (resp. QEFL+T2) corresponds to our official run uogTB06QET1 (resp. uogTB06QET2).* Submitted runs are in boldface. QEFU+T1 and QEFU+T2 are baselines for comparison with QEFL. p-values are computed by the Wilcoxon matched-pairs signed-ranks test.

3.2 Named Page Finding Task

The objective of the Named Page Finding task is to find a particular page, given a topic that describes it. A high precision task such as this can benefit from deploying a field-based weighting model that takes into account document structure. For TREC 2006, we test modelling the distribution of term occurrences in document fields as a multinomial distribution (hypothesis H3), using our new multinomial field-based DFR weighting model, ML2 (Section 2.2, Equation (9)). Furthermore, we model term dependence and proximity (hypothesis H4) using the pBiL2 binomial model (Section 2.3, Equation (13)). Lastly, we investigate a novel approach for com-

NR_q	NR_i	Features	MAP	P@10	bPref
none	none		0.3355	0.6240	0.3772
U-WT10G	none		0.3613	0.6320	0.4023
U-GOV2	none		0.3409	0.6240	0.3813
L-GOV2	none		0.3485	0.6400	0.3891
none	none	QE	**0.3966**	0.6680	**0.4446**
U-WT10G	none	QE	0.3853	0.6640	0.4399
U-GOV2	none	QE	0.3806	0.6460	0.4325
L-GOV2	none	QE	0.3898	0.6540	0.4423
U-GOV2	WT10G	QE$\oplus$	0.3686	0.6540	0.4290
L-GOV2	WT10G	QE$\otimes$	0.3728	**0.6720**	0.4404
none	none	QEFL	0.3878	0.6560	0.4398
U-WT10G	none	QEFL$\oslash$	0.3893	0.6580	0.4411
U-WT10G	none	QEFL	0.3770	0.6380	0.4177
L-WT10G	none	QEFL	0.3804	0.6460	0.4257

Table 4: MAP, P@10, and bPref of Adhoc runs with Title + Description + Narrative queries. The weighting model is PL2 (PL2F with fields) and Bo1 for query expansion (Bo1F with fields). $\oplus$ is our official submitted run uogTB06S50L. $\otimes$ is our official submitted run uogTB06SS10L. $\oslash$ is our official submitted run uogTB06SSQL. NR_q and NR_i denote noise reduction in the query and the index, respectively. U and L denote uniform noise reduction and reduction per query length, respectively. QE is query expansion on body only ($QEFL$ is query expansion on fields). Submitted runs are shaded and best scores are in bold.

bining sources of query independent evidence, in the form of prior probabilities (hypothesis H5), which is described in Section 3.2.1. We describe and discuss our experimental runs in Section 3.2.2.

3.2.1 Query-Independent Prior Probabilities

Various sources of query-independent evidence, in the form of prior probabilities, have been shown to be important for Web IR [6]. In this paper, we consider the following three sources of query independent evidence: (i) the information-to-noise ratio of a document [26], (ii) the static absorbing model [20], which is a way of providing authority to documents on the basis of their incoming links, and (iii) the number of incoming links to each document (inlinks). When using query independent evidence 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:

$$score(d, Q) = score(d, Q) + log(P(E)) \quad (17)$$

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

However, it is not clear how several document priors should be combined in a principled way. In particular, some previous work considered the priors to be independent [6], while other hand-tuned linear combinations of priors [15]. Moreover, the independence assumption does not always hold: For example, consider the absorbing model and inlinks priors - while both of these priors increase retrieval accuracy, they are likely to be correlated, because a document with a high number of inlinks is likely to have a high absorbing model score. Therefore, to combine several prior probabilities in a principled manner, we propose a novel combination of prior probabilities. The combination of prior probabilities is given by:

$$P(E_1, E_2) = P(E_2|E_1) \cdot P(E_1) \quad (18)$$

where $P(E_1)$ is the prior probability of the query independent source of evidence E_1; $P(E_2|E_1)$ is the conditional probability of the query independent source of evidence E_2, given E_1; $P(E_1, E_2)$ is the probability that both E_1 and E_2 occur [17]. Naturally, we can can extend this technique for more than two priors.

When using the combination of prior probabilities described in Equation (18) for retrieval, the score of a retrieved document d for a query Q is altered, in order to take the combined prior probabilities into account as follows:

$$score(d, Q) = score(d, Q) + log(P(E_1, E_2)) \quad (19)$$

3.2.2 Experiments and Results

We submitted three runs to the TREC 2006 Named Page Finding task. The first run tests the effectiveness of the new ML2 field-based DFR weighting model, described in Section 2.2. The second run tests the effectiveness of the pBiL2 term dependence and proximity model, described in Section 2.3. The third run tests the combination of prior probabilities using the second run as baseline. A full description of the submitted runs follows:

- *uogTB06M* uses the multinomial DFR weighting model ML2.
- *uogTB06MP* also uses the multinomial DFR weighting model ML2, and adds the term dependence and proximity model pBiL2.
- *uogTB06MPIA* uses the multinomial DFR weighting model ML2 and the term dependence and proximity model pBiL2, while also combining information-to-noise ratio and static absorbing model prior probabilities.

After submitting the above official runs, we discovered that when we approximated the ML2 field-based weighting model, we used the natural logarithm, instead of the correct log_2 in the Lanzcos approximation of the Γ function. We retrained and repeated the submitted runs with the correct logarithm. Table 10 gives the parameter settings applied in this task. Moreover, Table 5 displays the Mean Reciprocal Rank (MRR) of the official submitted runs, and their replacement runs with the corrected logarithm. In addition to the runs submitted, we also experimented with using a different field-based weighting model, namely PL2F, as well as applying each of the three sources of query independent evidence alone, (using Equation (17)), instead of combined as per Equations (18) & (19).

The conclusions we draw from Table 5 are as follows. Firstly, regarding our hypothesis H3, concerning modelling the distribution of term occurrences in document fields as a multinomial distribution, we observe that ML2 (uogTB06M) performs comparably to PL2F (uogTB06PL). This means that ML2 is not only an elegant and theoretically-sound model, but also a readily deployable model, on a par with existing state-of-the-art field-based weighting models, such as PL2F, despite ML2 employing less parameters than PL2F.

Secondly, modelling term proximity appears to assist the retrieval process. In particular, applying proximity to our baselines of uogTB06M and uogTB06PL increases MRR (see uogTB06MP with MRR 0.466 and uogTB06PLP with MRR 0.478 respectively). This validates our hypothesis H4 on the usefulness of term dependence and proximity in the Named Page Finding task.

Thirdly, regarding the application of prior evidence, we see that all three priors applied alone - namely information-to-noise, absorbing model and inlinks - decrease performance compared to the baseline (comparing uogTB06MI, uogTB06MA and uogTB06ML to uogTB06M respectively). However, regarding hypothesis H5 on the combination of query-independent evidence, we observe that retrieval performance can be improved if we choose appropriate document priors (uogTB06MIL). In particular, MRR is improved over the use of no priors (uogTB06M), as well as over the use of any single prior alone (uogTB06MI or uogTB06ML).

Lastly, using both term proximity and the appropriate document priors, we see that retrieval performance is again enhanced compared to the baseline and the combination of priors. In particular, the unofficial run uogTB06MPIL achieves a 5% increase in MRR over our best submitted run (uogTB06MP).

3.3 Terabyte Track Conclusions

In the 2006 Terabyte Track, we participated in the Adhoc and Named Page Finding tasks. We extended our modular Terrier retrieval platform, and tested the following hypotheses. For the Adhoc task, we hypothesised that, for query expansion on document fields, the linear combination of fields can provide better retrieval performance, than the uniform combination of fields. We tested this hypothesis with short queries (Section 3.1.3, Table 3), and found it to be valid. For the same task, we hypothesised that low frequency part-of-speech n-grams found in text, correspond mostly to noise, which if removed, can enhance retrieval performance. We tested this hypothesis on long queries and on the test collection to be indexed, and found it to be valid when query expansion is not applied (Section 3.1.3, Table 4). Query expansion combined with noise reduction lead to a small deterioration in retrieval performance, which could be due to the effect of noise reduction on query length (for noise reduction from the queries). For the Named Page Finding task, we tested the hypotheses that: (i) modelling in a refined way the distribution of term occurrences in document fields, namely as a multinomial distribution, is a theoretically-sound and

Run Name	Submitted	Corrected log
Weighting model only		
uogTB06M	0.448	0.449
uogTB06PL	0.454	
Proximity		
uogTB06MP	0.466	0.467
uogTB06PLP	0.478	
Single Priors		
uogTB06MI	0.440	
uogTB06MA	0.431	
uogTB06ML	0.422	
Combined Priors		
uogTB06MIA	0.413	
uogTB06MIL	0.465	
Proximity + Priors		
uogTB06MPIA	0.463	0.454
uogTB06MPIL	**0.489**	
best	0.7779	
median	0.3706	

Table 5: MRR of the Named Page runs. *Submitted* are the official submitted runs. *Corrected* log are the same runs, using the correct logarithm function. The field-based weighting models used are ML2 (denoted by M), and PL2F (denoted by PL). Term dependence and proximity is denoted by P. I, A and L denote the priors of information-to-noise ratio, static absorbing model and inlinks, respectively. *best* and *median* are the best and median runs submitted among all participants, respectively. Submitted runs are shaded. Our best run is in boldface.

robust approach, which performs comparably to other field-based weighting models; (ii) modelling the dependence and proximity of query terms in documents can enhance retrieval performance; (iii) using a conditional combination of multiple sources of query independent evidence, in the form of prior probabilities can improve retrieval performance, over using a single source of such evidence. We found hypotheses (i) and (ii) to be valid (Section 3.2.2, Table 5), while further work is needed to establish the best combination of priors.

4. ENTERPRISE TRACK

In TREC 2006, we participate in the Expert Search task of the Enterprise track, where we aim to develop and experiment using our novel voting model for Expert Search [14]. Firstly, a set of documents is associated with each candidate to represent the candidate's expertise to the system. Then our voting model considers the ranking of documents with respect to the query, in order to generate an accurate ranking of candidates. For TREC 2006, we experiment to validate the following hypotheses:

1. Candidate Length Normalisation: the profiles of candidates can be of various lengths. We hypothesise that our voting model requires to account for candidate profiles of varying lengths.

2. Document Ranking: in our voting model, we hypothesise that the accuracy of the candidate ranking model depends on the extent to which documents retrieved by the underlying document ranking represents the topic.

To validate our two hypotheses, our research is directed in two areas: firstly, we propose and integrate into the voting model a new theoretically-driven way of combining document votes for candidates, that accounts for the length of each candidate's profile; secondly, to test our document ranking hypothesis, we employ three techniques, namely (i) the use of a field-based weighting model; (ii) query expansion; and (iii) the term dependence and proximity model. These techniques should increase the quality of the document ranking, and we hypothesise that the accuracy of the generated candidate ranking will also be increased.

The remainder of this section is as follows: Section 4.1 describes our voting approach for Expert Search; Section 4.2 discusses the need for candidate profile length normalisation in Expert Search; Section 4.3 describes the effect of the document ranking in the voting approach, and defines techniques which can be applied to increase the quality of the document ranking. In Section 4.4, we present the experimental setup for our runs. We discuss the submitted runs and their results in Section 4.5. We present additional runs in Section 4.6, and give some closing comments in Section 4.7.

4.1 Voting Approaches for Expert Search

Our newly-proposed approach models Expert Search as a voting process [14]. In our model, a candidate's expertise is represented by a profile, which is a set of documents associated with each candidate, to represent that candidate's expertise.

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 [14], we defined eleven voting techniques for aggregating votes for candidates, adapted from existing data fusion techniques. For TREC 2006, we experiment using two voting techniques, namely CombSUM and expCombMNZ. For CombSUM, the score of a candidate C's expertise to a query Q is given by:

$$score\_cand_{CombSUM}(C,Q) = \sum_{d \in R(Q) \cap profile(C)} score(d,Q) \quad (20)$$

where $score(d,Q)$ is the score of document d in the initial ranking of documents $R(Q)$, as given by a suitable document weighting model. In all our runs, we use the DFR InL2 document weighting model, or its field-based variant InL2F to generate $score(d,Q)$ - see Equations (2), (3) & (4).

Secondly, we apply the expCombMNZ voting technique. For expCombMNZ, the score of a candidate C's expertise to a query Q is given by:

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

where $\|R(Q) \cap profile(C)\|$ is the number of documents from the profile of candidate C that are in the ranking $R(Q)$. In the next section, we introduce our candidate length normalisation technique which can be applied to either the voting techniques.

4.2 Candidate Length Normalisation for Expert Search Voting approach

Document length normalisation has been studied in IR for some time, in order to fairly retrieve documents of all lengths. State-of-the-art document weighting models, such as BM25 [22] or those from the DFR framework (for instance PL2 or InL2) [1], all include document length normalisation components. This normalisation component prevents long documents from gaining an unfair advantage in the document ranking. However, our voting model may be susceptible to favouring candidates which have a large profile: consider a candidate with many associated documents in its profile - this candidate has a higher chance of achieving a vote at random from the document ranking, than another candidate that has a smaller profile with fewer associated documents. Hence we hypothesise that we should account for candidate length in our model, so that candidates of all lengths are retrieved fairly.

For TREC 2006, we extend our model to introduce a new technique that explicitly accounts for candidate profile length while ranking candidates. We supplement a voting technique (denoted M), by adding a candidate length normalisation. This normalisation is an adaption of Normalisation 2 from the DFR framework - see Equation (4). Normalisation 2 is used to control any bias towards candidates with longer profile lengths. The combination of a technique M with candidate length normalisation is denoted MNorm2, and is calculated as follows:

$$score\_cand_{MNorm2}(C,Q) = score\_cand_{M}(C,Q) \cdot \log_2(1 + c_{pro} \cdot \frac{avg\_len\_pro}{l_C}), (c_{pro} > 0) \quad (22)$$

where l_C is the number in tokens in all the documents belonging to the profile of candidate C, and avg_len_pro is the average length of all candidate profiles, in tokens. c_{pro} is a hyper-parameter, used to control the influence of normalisation. For TREC2006, we test the use of candidate length normalisation with both CombSUM and expCombMNZ, denoted by CombSUMNorm2 and expCombMNZNorm2 respectively.

4.3 Effect of the Document Ranking

In our voting approach, the quality of the document ranking $R(Q)$ directly affects how well the approach performs. We hypothesise that if we are able to produce a document ranking with many on-topic documents at the top of the document ranking, then we are able to accurately convert the ranking of documents into an effective ranking of candidates. In TREC 2006, we test this hypothesis, using three retrieval techniques to increase the quality of the document ranking.

Firstly, we know that taking into account the structure of documents can allow increased precision for document retrieval, particularly on the W3C collection [12]. Hence, we apply a field-based weighting model from the DFR framework, to take a more refined account of each document field into account when ranking the documents. Namely, we experiment with applying the InL2F field-based weighting model (see Equations (2) & (3)). Using this should increase the number of on-topic documents at the top of the document ranking, compared to using the InL2 model (Equations (2) & (4)).

Secondly, we use the novel information theoretic model, based on the DFR framework, for incorporating the dependence and proximity of the query terms in the documents, as described in Section 2.3. We apply the term dependence and proximity model to improve the number of on-topic documents at the top of the document ranking, as we believe that on-topic documents will have term-dependencies between query terms, and by modelling these, we can bring these to the top of the document ranking. In particular, we use the pInL2 term dependence and proximity model - see Equation (12).

Thirdly, we investigate the use of query expansion (QE) in the expert search setting. We assume that the top-ranked documents in the document ranking are on-topic to the expertise query. By performing query expansion using these top-ranked documents, we aim to bring more on-topic documents into the document ranking [13].

Query expansion is applied using Bo1 (Equation (14)) to weight terms from the top exp_doc ranked documents in $R(Q)$. For Bo1, tf_x is the term frequency of term t in the top exp_doc ranked documents. The exp_term top-ranked terms are then added to query Q and the document ranking $R(Q)$ regenerated. We use the default settings of $exp\_term = 10$ and $exp\_doc = 3$ [1].

4.4 Experimental Setup

We index the W3C collection using the Terrier IR platform [16], by removing standard stopwords and applying Porter's weak stemming. Only documents which were associated with at least one candidate were indexed, which leaves only 52,129 documents in the index. We also index the anchor text of incoming hyperlinks from the entire W3C collection and add these to the documents.

We used two techniques to identify documents from the W3C collection to associate with candidates to represent each candidate's expertise. As described for the *Occurrences* profile sets of our TREC 2005 participation [10], we generate queries which were used to identify documents that mentioned each candidate, based on the occurrences of variations of the candidate's name and email address in the collection. These documents form the *OccurrencesA* profile set of each candidate. All our official runs use this profile set.

Secondly, we use the Unix `grep` command to identify documents from the collection which contain an exact match of the candidate's full name. Each matching document is added to the candidate's profile, to form their *OccurrencesB* profile set. On average, it appeared that the OccurrencesB profile set finds more documents for each candidate than OccurrencesA. We note that this is counter-intuitive, as OccurrencesB should be a subset of OccurrencesA, so we theorise that a bug affected the creation of OccurrencesA for TREC 2005. OccurrencesB is created using a simpler approach, than OccurrencesA.

All our experiments were performed using Terrier. We trained using the 50 TREC 2005 Enterprise track queries. Our optimisation system uses simulated annealing processes to find settings for c and c_{pro} that maximise mean average precision (MAP). Table 11 details the parameter values used for the Expert Search task in TREC 2006.

4.5 Experiments and Results

We submitted 4 runs to the Expert Search task of the Enterprise track, which test our two hypotheses for this task. All official runs used the OccurrencesA profile sets to represent the candidate expertise, and only the title field of the topics. The first three runs test our candidate length normalisation technique. Moreover, they each test a different way of increasing the topicality of the document ranking. The fourth run is a baseline run. More specifically, we submitted the following runs:

- *uogX06csnP* generates a document ranking using the InL2 document weighting model, and applies our CombSUMNorm2 expert search technique described above. Moreover, the pInL2 term dependence and proximity model is applied to increase the topicality of the document ranking. This run tests the candidate length normalisation technique, and uses term de-

Run Name	MAP	bPref	P@10
Best	0.7507	0.7542	-
Median	0.3412	0.3602	-
uogX06csnP	0.2881	0.3120	0.4510
uogX06csnQE	**0.3024**	**0.3292**	0.4429
uogX06csnQEF	0.3011	0.3208	**0.4551**
uogX06ecm	0.2685	0.2991	0.4143
uogX06csn	0.2784	0.3222	0.4224
uogX06csnF	0.2830	0.3195	0.4306

Table 7: The mean average precision (MAP), binary preference (bPref), and precision at 10 (P@10) of our submitted runs, as well as that achieved by all participants, and two additional runs. P@10 achieved by all participants is not available. All runs use the OccurrencesA profile sets, and title only topics.

pendence to test the document ranking hypothesis.

- *uogX06csnQE* also applies InL2 and CombSUMNorm2, but applies query expansion using Bo1 to increase the topicality of the document ranking. This run also tests the candidate length normalisation technique, but uses QE to test the document ranking hypothesis.
- *uogX06csnQEF* is similar to uogX06csnQE, but instead the document ranking takes document structure into account, by using the field-based InL2F weighting model. This run tests the candidate length normalisation technique, and also applies fields and QE to test the document ranking hypothesis.
- *uogX06ecm* uses the expCombMNZ expert search technique, which applies no candidate length normalisation.

Table 6 summaries the salient features of each submitted run, and some additional runs that we will describe in Section 4.6. Table 7 shows the results of the submitted runs, in terms of Mean Average Precision (MAP), binary Preference (bPref) and Precision at 10 (P@10). We also show the overall best and median runs achieved across all participants, as well as two additional baseline runs, namely uogX06csn and uogX06csnF. uogX06csn is the baseline run using InL2 and CombSUMNorm2; uogX06csnF uses InL2F and CombSUMNorm2.

Adding term dependence to the baseline run (uogX06csn vs uogX06csnP) increases retrieval performance, as do fields (uogX06csnF). In particular, adding QE (uogX06csn vs uogX06csnQE) provides the best submitted run. Note that using QE and fields (uogX06csnQEF) does not increase MAP or bPref when compared to QE alone (uogX06csnQE), though Precision at 10 is improved.

4.6 Additional Runs

As explained in Section 4.4, it appears that our OccurrencesA candidate profile sets was affected by a bug, and did not contain as much expertise evidence for each candidate as OccurrencesB - normally, OccurrencesB would be expected to be a subset of OccurrencesA.

For our additional runs, we use only the OccurrencesB candidate profile sets, and perform a selection of runs using this, to allow us to draw firm conclusions, especially concerning the usefulness of candidate length normalisation. We also experiment across all three topic lengths. The salient features of the additional runs are also shown in Table 6.

The results in terms of MAP are shown in Table 8[4]. From the shown results, we can see that our MAP is markedly improved

[4]Note that all runs using OccurrencesB were made using a full index of all 331,037 documents in the W3C collection. This should have little effect on the results, as unassociated documents are not considered by the voting techniques for ranking the experts.

Run Name	MAP		
	T	TD	TDN
uogX06cs	0.5319	0.5409	0.5491
uogX06csQE	0.5458	0.5435	0.5637
uogX06csF	0.5508	0.5394	0.5155
uogX06csQEF	0.5512	0.5564	0.5420
uogX06csn	0.4647	0.4747	0.4805
uogX06csnQE	0.4813	0.4842	0.4983
uogX06csnF	0.4994	0.5302	0.5115
uogX06csnQEF	0.5357	0.5405	0.5366
uogX06ecm	0.5430	0.5567	**0.5746**
uogX06ecmQE	0.5611	0.5511	0.5733
uogX06ecmF	**0.5663**	0.5628	0.5552
uogX06ecmQEF	0.5595	**0.5634**	0.5663
uogX06ecmn	0.5157	0.5264	0.5446
uogX06ecmnQE	0.5395	0.5337	0.5489
uogX06ecmnF	0.5469	0.5442	0.5285
uogX06ecmnQEF	0.5524	0.5595	0.5510
(Averages)	0.5341	0.5380	0.5400

Table 8: The mean average precision (MAP) of a selection of additional runs using the *OccurrencesB* candidate profiles set, across all three topic lengths.

by using the *OccurrencesB* candidates profiles set, compared to the submitted runs in Table 7. In particular, the performance of uogX06csn jumps to MAP 0.4647 using short topics, and uogX06ecm to 0.5430. Applying either QE or fields to either baseline results in an improvement in terms of MAP. For example, comparing uogX06csn with uogX06csnQE; uogX06csn with uogX06csnF; and uogX06ecm with uogX06ecmQE. In each case, applying a technique resulted in an increase in MAP, which validates our document ranking hypothesis. Moreover, in most cases, applying two techniques in runs uogX06csQEF, uogX06csnQEF, and uogX06ecmnQEF improves over applying either QE or fields alone (the exceptions here are uogX06ecmQEF on short and long queries, and uogX06csQEF on long queries). This appears to validate our document ranking hypothesis. Further improvements are obtainable if the exp_doc and exp_term parameters are varied [13].

Next, we examine the usefulness of candidate length normalisation. Comparing uogX06cs with uogX06csn, and uogX06ecm with uogX06ecmn, shows a decrease in MAP across all three topic types. This is mirrored across other runs - for instance, comparing uogX06csQEF with uogX06csnQEF. Note however that decreases in MAP are less marked when applying QE and fields.

Comparing CombSUM and expCombMNZ, we can see that expCombMNZ is at least as good as, and usually better than CombSUM. This mirrors our evaluation using TREC 2005 data [14].

Finally, we examine the effect of topic length on MAP. On average, using title description and narrative topic fields (TDN) is better than title and description (TD), which is better than title only (T). However, the margins between topic types are very narrow, so no solid conclusions can be drawn.

4.7 Expert Search Task Conclusions

Overall, we demonstrated that our expert search model performs in a stable manner.

With regard to our first hypothesis, we require further research to establish the usefulness of candidate length normalisation in expert search. Candidate length normalisation did not appear to be useful

Run Name	Weighting Model	Other Retrieval Techniques	Voting Approach	Topics Fields
Submitted				
uogX06csnP	InL2 (Eqs. (2)&(4))	Term Dependence pInL2 (Eqs. (12)&(4))	CombSUMNorm2	T
uogX06csnQE	InL2	Query Expansion	CombSUMNorm2	T
uogX06csnQEF	InL2F (Eqs. (2)&(3))	Query Expansion	CombSUMNorm2	T
uogX06ecm	InL2	-	expCombMNZ	T
Additional				
uogX06cs	InL2	-	CombSUM	-
uogX06csQE	InL2	Query Expansion	CombSUM	-
uogX06csQEF	InL2F	Query Expansion	CombSUM	-
uogX06csF	InL2F	-	CombSUM	-
uogX06csnF	InL2F	-	CombSUMNorm2	-
uogX06ecmQE	InL2	Query Expansion	expCombMNZ	-
uogX06ecmF	InL2F	-	expCombMNZ	-
uogX06ecmQEF	InL2F	Query Expansion	expCombMNZ	-
uogX06ecmn	InL2	-	expCombMNZNorm2	-
uogX06ecmnQE	InL2	Query Expansion	expCombMNZNorm2	-
uogX06ecmnF	InL2F	-	expCombMNZNorm2	-
uogX06ecmnQEF	InL2F	Query Expansion	expCombMNZNorm2	-

Table 6: Salient features of submitted and additional runs of the expert search task of the Enterprise track.

on the OccurrencesB set. Further experimentation using additional candidate profile sets would provide solid conclusions.

With regard to our document ranking hypothesis, this seems to be validated, because applying known techniques for increasing the quality of the document ranking were shown to increase the retrieval performance of the ranking of candidates. Moreover, on the OccurrencesB candidate profile sets, applying more than one technique (fields and query expansion) resulted in a improvement over either technique alone in most cases.

Our results show that the exact technique applied to associate documents to candidate to represent their expertise has a marked effect on the retrieval performance of the system. Choosing the correct candidate profile set results in a marked increase in performance of our expert search system compared to our submitted run, and the median run of all participants.

5. CONCLUSIONS

In TREC 2006, we participated in the Adhoc and Named Page Finding tasks of the Terabyte track, and the Expert Search task of the Enterprise track. Having such a variety of retrieval tasks to address, ranging from classical adhoc retrieval, to enterprise-oriented expert seach, we focussed on devising new, theoretically-driven, and effective weighting models and retrieval boosting techniques, which would be generic enough, so as to be easily and effectively applied in as many retrieval tasks as possible. Specifically, we extended our Terrier Information Retrieval platform to accommodate two new Divergence From Randomness (DFR) weighting models, which combine evidence on document structure and capture term dependence and proximity, respectively. We used these models in the Terabyte and the Enterprise tracks, and found them to be effective. Additionally, we presented a new query expansion mechanism on fields, which successfully combines evidence in a linear, rather than uniform way and a novel syntactically-based noise reduction technique for long queries and the index. We presented a new theoretically-driven way of combining query independent evidence, in the form of prior probabilities, which we tested in Named Page Finding. In the Expert Search task, we further enhanced our understanding of our model for expert search, and through experimentation, generated some very promising results. Overall, our participation in TREC 2006 includes parts of our ongoing research in weighting models and retrieval performance enhancing techniques, which are effectively combined as part of the DFR framework, and easily implemented in our Terrier retrieval platform. The good results reported in our participation pave the way for further research.

6. 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 C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, Ed., *CLEF 2001*, 257–264, 2002.

[3] 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.

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

[5] D. Hawking, T. Upstill, and N. Craswell. Towards better Weighting of Anchors. In *Proceedings of SIGIR'2004*, 512–513, Sheffield, UK, 2004.

[6] W. Kraaij, T. Westerveld, and D. Hiemstra. The Importance of Prior Probabilities for Entry Page Search. In *Proceedings of SIGIR'2002*, 27–34, Tampere, Finland, 2002.

[7] C. Lioma and I. Ounis. Examining the Content Load of Part-of-Speech Blocks for Information Retrieval. In *Proceedings of COLING/ACL 2006*, Sydney, Australia, 2006.

[8] C. Lioma and I. Ounis. Light Syntactically-Based Index Pruning for Information Retrieval. In *Proceedings of ECIR-2007*, Rome, Italy, 2007.

[9] C. Lioma and I. Ounis. A Syntactically-Based Query Reformulation Technique for Information Retrieval. *Information Processing and Management* (In Press), 2007.

[10] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC 2005: Experiments in Terabyte and

Parameter	Equation(s)	uogTB06QET1	uogTB06QET2	uogTB06S50L	uogTB06SS10L	uogTB06SSQL
c_f for the first-pass retrieval	(1) & (5)	0.8	1.3	1.0	-	-
c_f for the expanded queries	(1) & (5)	6.0	9.7	2.4	-	-
w_{anchor}	(1) & (5)	0.0	0.1	0.1	-	-
w_{title}	(1) & (5)	2.0	1.0	2.5	-	-
c	(1) & (4)	-	-	-	2.16	2.16
exp_doc	(14) & (15)	3	3	4	5	5
exp_term	(14) & (15)	30	20	24	20	20

Table 9: The parameter values used in our submitted TREC 2006 Terabyte track Adhoc task runs.

Enterprise tracks with Terrier. In *Proceedings of TREC-2005,* Gaithersburg, USA, 2005.

[11] C. Macdonald, V. Plachouras, H. Ben, C. Lioma, and I. Ounis. University of Glasgow at WebCLEF 2005: Experiments in Per-Field Normalisation and Language Specific Stemming. C. Peters, F. Gey, J. Gonzalo, H. Mueller, G. Jones, M. Kluck, B. Magnini, and M. de Rijke, Ed., *CLEF 2005*, 898–907, 2006.

[12] C. Macdonald and I. Ounis. Combining Fields in Known-Item Email Search. In *Proceedings of SIGIR'2006*, 675–676, Seattle, USA, 2004.

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

[14] 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.

[15] D. Metzler, T. Strohman, Y. Zhou, and W. B. Croft. Indri at TREC 2005: Terabyte Track. In *Proceedings of the TREC-2005*, Gaithersburg, USA, 2005.

[16] 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

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

[18] V. Plachouras and I. Ounis. Multinomial Randomness Models for Retrieval with Document Fields. In *Proceedings of ECIR-2007*, Rome, Italy, 2007.

[19] V. Plachouras, B. He, and I. Ounis. University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. In *Proceedings of the TREC-2004,* Gaithersburg, USA, 2004.

[20] V. Plachouras, I. Ounis, and G. Amati. The Static Absorbing Model for the Web. *Journal of Web Engineering*, 165–186, 2005.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* Cambridge University Press, 1992.

[22] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In *Proceedings of TREC-4*. Gaithersburg, USA, 1995

[23] 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.

[24] 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.

[25] 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.

[26] X. L. Zhu and S. Gauch. Incorporating Quality Metrics in Centralized/Distributed Information Retrieval on the World Wide Web. In *Proceedings of SIGIR'2000*, 288–295, Athens, Greece, 2000.

Parameter	Equations(s)	Value
PL2F w_{body}	(1) & (3)	1.0000
PL2F w_{title}	(1) & (3)	2.6568
PL2F w_{anchor}	(1) & (3)	0.5153
PL2F c_{body}	(1) & (3)	1.0153
PL2F c_{title}	(1) & (3)	11.9652
PL2F c_{anchor}	(1) & (3)	9.1145
ML2 c_{body}	(9) & (4)	(0.3300) 0.2514
ML2 c_{title}	(9) & (4)	(9.8468) 43.3551
ML2 c_{anchor}	(9) & (4)	(5.6892) 7.4939
pBiL2 c_p	(13) & (4)	0.0500
pBiL2 $dist$	(13)	5

Table 10: The parameter values used in our TREC 2006 Terabyte track Named-page task runs. Figures in brackets were before being retrained using the correct logarithm function base.

Parameter	Equation(s)	Value
pInL2 c_p	(12) & (4)	0.5
InL2 c	(2) & (4)	0.124
Norm2 c_{pro} with InL2	(22)	3.690
InL2F c_{body}	(2) & (3)	0.171
InL2F c_{title}	(2) & (3)	1.131
InL2F c_{atext}	(2) & (3)	2.598
InL2F w_{body}	(2) & (3)	0.772
InL2F w_{title}	(2) & (3)	1.320
InL2F w_{atext}	(2) & (3)	1.334
Norm2 c_{pro} with InL2F	(22)	12.0000
QE exp_term	(14)	10
QE exp_doc	(14)	3

Table 11: The parameter values used in our submitted runs to the TREC 2006 Enterprise Track, Expert Search task.