Term Frequency Normalisation Tuning for BM25 and DFR Models

Ben He and Iadh Ounis

Department of Computing Science,
University of Glasgow,
United Kingdom

Abstract. The term frequency normalisation parameter tuning is a crucial issue in information retrieval (IR), which has an important impact on the retrieval performance. The classical pivoted normalisation approach suffers from the collection-dependence problem. As a consequence, it requires relevance assessment for each given collection to obtain the optimal parameter setting. In this paper, we tackle the collection-dependence problem by proposing a new tuning method by measuring the normalisation effect. The proposed method refines and extends our methodology described in [7]. In our experiments, we evaluate our proposed tuning method on various TREC collections, for both the normalisation 2 of the Divergence From Randomness (DFR) models and the BM25's normalisation method. Results show that for both normalisation methods, our tuning method significantly outperforms the robust empirically-obtained baselines over diverse TREC collections, while having a marginal computational cost.

1 Introduction

An Information Retrieval (IR) system receives a query from the user and returns the supposedly relevant documents [8]. A crucial issue underlying an IR system is to rank the returned documents by decreasing order of relevance. For example, recent surveys on the Web show that users rarely look beyond the top returned documents [10]. Usually, ranking is based on a weighting model.

Almost all weighting models take the within document term frequency (tf), the number of occurrences of the given query term in the given document, into consideration as a basic factor for weighting documents. For example, the classical $tf \cdot idf$ weighting formula is the following:

$$w(t,d) = tf \cdot \log \frac{N}{df} \tag{1}$$

where $w(t,d)$ is the weight of document d for term t, N is the number of documents in the collection and df is the document frequency, which is the number of documents containing the term t.

D.E. Losada and J.M. Fernández-Luna (Eds.): ECIR 2005, LNCS 3408, pp. 200–214, 2005.

The above $tf \cdot idf$ formula is based on two basic principles of weighting:

- For a given term, the higher its frequency in the collection the less likely it is that it reflects much content.
- For a given term in a given document, the higher the within document term frequency (tf) is, the more information the term carries within the document.

However, the term frequency is dependent on the document length, i.e. the number of tokens in a document, and needs to be normalised by using a technique called *term frequency normalisation*.

In [11], Singhal et. al. gave the following two reasons for the need of the tf normalisation:

- The same term usually occurs repeatedly in long documents.
- A long document has usually a large size of vocabulary.

The two reasons above are based on the observation of term occurrences in the documents. As a consequence, a weighting model without employing a normalisation method, such as $tf \cdot idf$, could produce biased weights with respect to the document length, favouring long documents.

A classical method of the tf normalisation tuning is the *pivoted normalisation* approach proposed by Singhal et. al. [11]. The idea of the pivoted normalisation is to fit the document length distribution to the length distribution of relevant documents. However, since the document length distribution is collection-dependent, the optimal parameter settings on different collections are different. Therefore, it requires relevance assessment on each given collection. This refers to the so-called *collection-dependence problem*. According to the study in [4], there is indeed a need to re-calibrate the tf normalisation parameter for different collections.

For the collection-dependence problem, we have proposed a tuning method by measuring the normalisation effect [7]. The idea is to use a collection-independence measure, namely the normalisation effect, to indicate the optimal parameter settings on diverse collections. In [7], the method has been applied to the *normalisation 2* with the PL2 model. PL2 is one of the divergence from randomness (DFR) document weighting models [2]. Using the PL2 model, the relevance score of a document d for query Q is given by:

$$\begin{aligned} score(d,Q) &= \sum_{t \in Q} w(t,d) \\ &= \sum_{t \in Q} \frac{1}{tfn+1} \big(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda + \frac{1}{12 \cdot tfn} - tfn) \cdot \\ &\quad \log_2 e + 0.5 \cdot \log_2 (2\pi \cdot tfn) \big) \end{aligned} \tag{2}$$

where λ is the mean and variance of a Poisson distribution. $w(t,d)$ is the weight of document d for query term t.

The normalised term frequency tfn is given by the *normalisation 2*:

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

where l is the document length and avg_l is the average document length in the whole collection. tf is the original term frequency. c is the free parameter of the normalisation method. The experiments in [7] have shown that applying the tuning method by measuring the normalisation effect to the normalisation 2 achieves a robust performance across collections.

However, the tuning method of measuring the normalisation effect also suffers from the following two problems:

1. The tuning method can not be systematically applied to BM25's normalisation method. As one of the most well-established IR systems, Okapi uses BM25 to perform the document ranking, where the *idf* factor $w^{(1)}$ is normalised as follows [9]:

$$w(t,d) = w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} \tag{4}$$

   where $w(t,d)$ is the weight of document d for query term t. The sum of $w(t,d)$ of the query terms gives the final weight of document d. K is given by $k_1((1-b)+b\frac{l}{avg\_l})$, where l and avg_l are the document length and the average document length in the collection, respectively. For the parameters k_1 and k_3, the standard setting recommended in [12] are $k_1 = 1.2$ and $k_3 = 1000$. qtf is the number of occurrences of a given term in the query and tf is the within document term frequency of the given term. b is the free parameter of the BM25's term frequency normalisation method, which can be seen as:

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

   where tfn is the normalised term frequency.
   As mention in [7], the function defining the normalisation effect is not systematically applicable to BM25 because the parameter b is only valid within [0, 1]. However, it was also suggested that it is possible to tackle the problem by proposing an alternative normalisation effect function.
2. For each given collection, the tuning involves the use of real user queries, which is not very practical, especially when such real user queries are not readily available.

In this paper, we aim to tackle the above two problems by improving the tuning method of [7]. First, we propose a new function defining the notion of normalisation effect, which is applicable to the Okapi's BM25 weighting model. We also show that this new function still applies to the normalisation 2. Second, we employ a novel query simulation method that is inspired by the query-based sampling approach described in [3]. Thus, the queries, which are used for document length sampling in a given collection, are created by this simulation method.

In the remainder of this paper, we briefly introduce the tuning method by measuring the normalisation effect in Section 2. By refining and extending this method, we propose a new tuning method in Section 3. In Sections 4 and 5, we provide our experimental setting and evaluation results. Finally, we conclude our work and suggest future work in Section 6.

2 Term Frequency Normalisation Tuning by Measuring Normalisation Effect

The tuning method proposed in [7] is based on measuring the *normalisation effect*, whose optimal value was experimentally shown as a collection-independent constant. The underlying idea of the method is that the effect of a normalisation method, with respect to a specific parameter value, on the term frequency is correlated with the document length distribution in a collection. Since the document length distribution is collection-dependent, the constant optimal normalisation effect corresponds to different parameter values. Thus, the tuning method assumes a constant optimal normalisation effect across collections. For a given collection, it applies the parameter setting such that it gives this constant. The approach has two steps, namely the training step and the tuning step.

In the training step, it obtains the optimal normalisation effect, which was shown experimentally to be a collection-independent constant in [7], on a training collection (e.g. disk1&2 of the TREC collections) with a set of real user queries (e.g. TREC topics 51-200) and their relevance assessment.

In the tuning step, for a given new collection, it samples the document length distribution using a set of real user queries, and then applies the parameter value such that it gives the optimal normalisation effect with respect to the sampled document length.

Then, the normalisation effect (NE) is defined as:

$$NE = \tau \frac{NE_D(\alpha)}{NE_{D,max}(\alpha)} \tag{6}$$

where α is the parameter of the applied tf normalisation method, e.g. the parameter c of the normalisation 2 in Equation (3). τ is 1 if $NE'_D(\alpha) \geq 0$, and -1 if $NE'_D(\alpha) < 0$. $NE_{D,max}(\alpha)$ is the maximum $NE_D(\alpha)$ value with respect to all possible settings of α[1]. The relation $NE_D(\alpha)$ is given by:

$$NE_D(\alpha) = \frac{Var(T_d)}{\mu}, d \in D \tag{7}$$

where D is the set of documents containing at least one of the query terms. Thus, NE_D can be interpreted as the normalisation effect on the document set D. To restrict the size of the set D to a fixed number so that the variance $Var(T_d)$ is not biased by the size of data, similar to the pivoted normalisation approach [11], we divide D into 1000 bins by document length. Each bin contains an equal number of documents, and is represented as a document that has the length of the average document length within the bin. Thus, d represents a bin in D, i.e. it can be seen as a document representing a bin.

For example, assuming that there are 2000 documents in D. If these 2000 documents are divided into 1000 bins by document length, then, each bin contains two documents, and documents with similar length are in the same bin. In

[1] In [7], it has been proved that with the use of the normalisation 2, a unique maximum $NE_D(\alpha)$ value does exist.

this case, the first and second shortest documents are in the same bin, the third and fourth shortest ones are in the same bin and so forth.

Moreover, in Equation (7), Var stands for variance. μ is the mean of T_d for all bins in D, where T_d is defined by:

$$T_d = \frac{tfn}{tf} \tag{8}$$

In Equation (8), the normalised term frequency tfn is given by the applied normalisation method, e.g. the normalisation 2 introduced in Section 1. Note that T_d depends only on the applied method's parameter setting and the mean document length within the bin. In the rest of this paper, the notion of bin length refers to the mean document length of the bin.

Having defined the notion of normalisation effect, on a training collection, the approach measures the optimal NE value that is assumed to be a constant. For a new collection, it applies the parameter giving this constant.

The approach has been applied for the normalisation 2 and clearly outperformed the robust empirically-based default setting. However, as introduced in the previous section, it also suffers from the following two problems:

1. The approach can not be systematically applied for BM25 because with the use of BM25, the maximum $NE_D(\alpha)$ value does not exist. This refers to the so-called "out-of-range" problem.
2. The tuning step involves the use of real user queries, which is not practical when not enough real user queries are available.

In the next section, we tackle the first problem by replacing the definition of $NE_D(\alpha)$ in Equation (7) with a new definition, and tackle the second problem by proposing a novel and efficient query simulation method.

3 The New Tuning Approach

In this section, we tackle the two above mentioned problems. In Section 3.1, we tackle the "out-of-range" problem by proposing a new definition for the relation $NE_D(\alpha)$ in Equation (7). In our derivation, we show that this new definition can be applied to both the normalisation 2 and BM25's normalisation method. In Section 3.2, we propose a novel query simulation method. Using this query simulation method, we sample the document length distribution by the simulated queries. Thus, our approach does not involve the need of real user queries in the tuning process.

3.1 Tackling the "Out-of-Range" Problem

The "out-of-range" problem is due to the fact that the parameter b of BM25 ranges only within [0,1]. As a consequence, using the original function of relation $NE_D(\alpha)$ in Equation (7), the b value, giving the maximum $NE_D(b)$, can be out of the range of [0,1]. In this section, we propose a new normalisation effect function

by replacing the definition of relation $NE_D(\alpha)$ in Equation (7) with a new one, which can solve the "out-of-range" problem. Our new proposed definition for the relation $NE_D(\alpha)$ is the following:

$$NE_D(\alpha) = Var\big(\frac{T_d}{T_{d,max}}\big), d \in D \tag{9}$$

where D is the set of documents containing at least one of the query terms. d is a bin in D. $T_{d,max}$ is the maximum T_d among all the bins in D, which is the T_d of the bin with the smallest average document length (the smallest bin length), since $T_d = \frac{tfn}{tf}$ is a decreasing function of document length.

Next, we approximate $NE'_D(\alpha)$, the derivative of function $NE_D(\alpha)$. If this derivative is a monotonic decreasing function of both parameter c of the normalisation 2 and parameter b of BM25, then the unique maximum $NE_D(\alpha)$ value exists, and the new definition can be applied to both normalisation methods. However, according to the definition in Equation (9), it is cumbersome to derive $NE'_D(\alpha)$ directly. To simplify the derivation, we assume a continuous and uniform distribution of T_d from $T_{d,min}$ to 1. $T_{d,min}$ is the minimum T_d in D, which is the T_d of the bin with the largest bin length in D. Although this assumption might not stand in real applications, because we just want to approximate $NE'_D(\alpha)$ to see if it is a monotonic decreasing function of α, this assumption is still applicable. Using the above mentioned assumption, we obtain:

$$\begin{aligned} NE_D(\alpha) &= \sum_D \big(\frac{T_d}{T_{d,max}}\big)^2 - \frac{\big(\sum_D \frac{T_d}{T_{d,max}}\big)^2}{n} \\ &\approx \int_{T_{d,min}}^{1} \frac{T_d}{T_{d,max}} d(T_d) - \frac{\big(\int_{T_{d,min}}^{1} \frac{T_d}{T_{d,max}} d(T_d)\big)^2}{n} \\ &\approx \frac{1 - T_{d,min}^3}{3n} - \frac{(1 - T_{d,min})^2}{4n^2} \end{aligned} \tag{10}$$

and the derivative is:

$$\begin{aligned} NE'_D(\alpha) &\approx \big(\frac{1 - T_{d,min}^3}{3n} - \frac{(1 - T_{d,min})^2}{4n^2}\big)' \\ &= -\frac{T_{d,min}^2}{n} \cdot T'_{d,min} + \frac{(1 - T_{d,min})}{2n^2} \cdot T'_{d,min} \\ &= \frac{-nT_{d,min}^2 - T_{d,min} + 1}{2n^2} \cdot T'_{d,min} \end{aligned} \tag{11}$$

Using BM25's normalisation method, $T_{d,min}(\alpha)$ becomes

$$T_{d,min}(b) = \frac{1}{(1 - b) + b \cdot \frac{l_{max}}{avg\_l}}, (0 \le b \le 1)$$

and

$$T'_{d,min}(b) < 0$$

Using the normalisation 2, $T_{d,min}(\alpha)$ becomes

$$T_{d,min}(c) = \log_2(1 + c \cdot \frac{avg\_l}{l_{max}}), (c > 0)$$

and

$$T'_{d,min}(c) > 0$$

where l_{max} is the maximum bin length in D.

We can see that using both the normalisation 2 and BM25's normalisation method, $NE'_D(\alpha)$ is a monotonic decreasing function of the parameter of the applied normalisation method. Therefore, the curve of the function $NE_D(\alpha)$ has a bell shape. When $NE'_D(\alpha) = 0$ is satisfied, $NE_D(\alpha)$ is at the peak point of the bell and has its unique maximum value. This demonstrates that our definition for relation $NE_D(\alpha)$ in Equation (9) is applicable to both normalisation methods, i.e. BM25's normalisation method and the DFR normalisation 2.

3.2 Query Simulation for Document Length Sampling

The computation of the normalisation effect needs a set of queries to sample the document length in a given new collection. A possible solution is to use real user queries to obtain the optimal parameter setting for each given collection, which is not practical. Instead, in this paper, we employ a novel query simulation method to sample the document length.

The idea of the proposed query simulation method is to formulate a query with the informative terms from documents that are related to a particular topic. In this way, the simulated queries can be meaningful rather than consisting of stop-words, or unrelated terms. This method is similar to the query-based sampling approach described in [3]. The difference between the two approaches is that our method adopts a term weighting model to extract the most informative terms from the top-ranked documents to formulate a query, while the query-based sampling approach uses the top-ranked documents to get various collection samples.

To simulate a query consisting of *exp_term* query terms, our query simulation method follows the steps listed below:

1. Randomly choose a seed-term from the vocabulary.
2. Rank the documents containing the seed-term using a specific document weighting function, e.g. PL2 or BM25 introduced in Section 1.
3. Extract the $exp\_term - 1$ most informative terms from the *exp_doc* top-ranked documents using a specific term weighting model. *exp_doc* is a parameter of the query simulation method. At this stage, we can use any term weighting model from the literature. In this paper, we apply a particular divergence from randomness (DFR) term weighting model, i.e. Bo1. The reason for using Bo1 is that it is one of the best-performing and stable DFR term weighting models [1]. Using this model, the weight of a term t in the *exp_doc* top-ranked documents is given by:

$$w(t) = tf_x \cdot \log_2 \frac{1 + P_n}{P_n} + \log_2(1 + P_n) \quad (12)$$

where tf_x is the frequency of the query term in the *exp_doc* top-ranked documents. P_n is given by $\frac{F}{N}$, where F is the term frequency of the query term in the collection and N is the number of documents in the collection.
4. To avoid selecting a junk term as the seed-term, we consider the most informative one of the extracted terms in step 3 as the new seed-term. Note that the original seed-term is discarded at this stage.
5. Repeat steps 2 and 3 to extract the $exp\_term - 1$ most informative terms from the *exp_doc* top-ranked documents, which are ranked according to the new seed-term.
6. The simulated query consists of the new seed-term and the $exp\_term - 1$ terms extracted in Step 5.

Adopting the above query simulation method, our new tuning method does not involve the use of real queries.

3.3 The New Tuning Method

Replacing the relation $NE_D(\alpha)$ in Equation (7) with our definition in Equation (9), and adopting the query simulation method proposed in Section 3.2, the new tuning method for the tf normalisation parameter is summarised below:

1. In the training step, on a training collection with a set of training queries, we obtain the optimal parameter setting using relevance assessment, and compute the corresponding optimal NE value that is assumed to be a constant across collections.
2. In the tuning step, on a given new collection, we apply the parameter setting such that it gives the constant optimal NE value obtained in the training step. In this step, the normalisation effect NE is computed over the document length sampled by a set of queries, which are created using the query simulation method proposed in Section 3.2.

In the above algorithm, the NE value is computed using our new normalisation effect function proposed in Section 3.1. Moreover, for a given collection, the tuning process is performed prior to the retrieval process. There is no additional overhead in the retrieval process.

In the following two sections, we introduce our experimental setting and evaluate our new tuning method.

4 Experimental Setting

Our experiments of evaluating the proposed approach are done within the Terrier Information Retrieval (IR) framework developed at the University of Glasgow. Terrier is a modular platform for the rapid development of large-scale IR applications, providing indexing and retrieval functionalities. Terrier is based on the DFR framework. It can index various document collections, including the

Table 1. Details of the four TREC collections used in our experiments. The second row gives the number of topics associated to each collection. N is the number of documents in the given collection. σ_l is the standard deviation of document length in the collection

	disk1&2	disk4&5	WT2G	WT10G
TREC topics	51 - 200	301 - 450 and 601 - 700	401 - 450	451 - 500
N	741860	528155	247491	1692044
σ_l	862.4977	558.1173	2009.3760	2303.4063

standard TREC collections. It also provides a wide range of parameter-free weighting approaches and full-text search algorithms, aiming to offer a public test-bed for performing IR experiments. Further information about Terrier can be found at http://ir.dcs.gla.ac.uk/terrier.

In our experiments, we evaluate our new term frequency (tf) normalisation tuning method on diverse collections. The training collection is the disk1&2 of the classical TREC collections. The reason for using this training collection is that it has a relatively large number of training queries available, which are the TREC topics numbered from 51 to 200. Having obtained the optimal NE value on the training collection using the corresponding relevance assessment, we evaluate our approach on three diverse TREC collections, including the disk4&5 (minus the Congressional Record on disk4) of the classical TREC collection[2], and two TREC Web collections, i.e. the WT2G [6] and the WT10G [5] collections. The test queries are TREC topics that are numbered from 301 to 450 and from 601-700 for the disk4&5, from 401 to 450 for the WT2G, and from 451 to 550 for the WT10G, respectively. Although these collections come with a set of test queries, such real user queries may not be readily available in an operational environment of a search engine. Therefore, it is more practical to employ the query simulation method in the tuning step.

Table 1 lists the test TREC topics, the number of documents, and the standard deviation of document length in each collection. As expected, the document length distribution of the four collections is quite different. In particular, the two Web collections clearly have large standard deviation values of document length compared to the two classical collections. This indicates that the document length distribution of the Web and the classical collections are widely diverse. Therefore, the default parameter setting for the classical collections might not be appropriate for the Web collections. This suggestion is confirmed later in our experiments.

Each TREC topic consists of three fields, i.e. title, description and narrative. In this paper, we experiment with three types of queries with respect to the use of different topic fields, in order to check the impact of query length on the effectiveness of our new tuning method. The three types of queries are:

[2] Related information of disk1&2 and disk4&5 of the TREC collections can be found from the following URL: http://trec.nist.gov/data/docs_eng.html

– **Short queries**: Only the title field is used.
– **Normal queries**: Only the description field is used.
– **Long queries**: All the three fields (title, description and narrative) are used.

Our evaluation is done with the use of PL2 and BM25 (see Equations (2) and (4)), respectively. Therefore, we test our new tuning method on both the normalisation 2 and BM25's normalisation method (see Equations (3) and (5)).

Our baselines are the empirical default settings of the two applied normalisation methods. For BM25's normalisation method, we use $b = 0.75$ for the three types of queries, which is the empirically recommended default setting [12]. For the normalisation 2, we use the default setting applied in [1], which is $c = 1$ for short queries and $c = 7$ for long queries. Since [1] does not report experiments using normal queries, we use the optimal parameter setting on the training collection as the baseline, i.e. $c = 1.40$ for normal queries.

For each type of queries, on the training collection, we retrieve documents for the training queries using a particular weighting model, and obtain the optimal parameter setting of the normalisation method of the applied weighting model, using relevance assessment.

In all our experiments, standard stop-words removal and the Porter's stemming algorithm are applied. We used one AMD Athlon 1600 processor, running at 1.4GHz.

For the query simulation approach in the tuning step (Section 3.2), we apply the PL2 DFR model (see Equation (2)) for document ranking and the Bo1 DFR model for term weighting (see Equation (12)). Both models were proposed in [1].

On each collection, we simulate 200 queries to sample the document length. The parameter *exp_doc* is set to 10. For each query type, *exp_term*, the number of composing query terms, is randomly chosen between *avql* and $avql + 1$, where *avql* stands for the integer part of the average query length of the TREC queries associated to the training collection. For example, the average query length of the long queries associated to the training collection, i.e. disk1&2, is 35.64. Thus, *avql* is 35. On each collection, the length of a simulated long query is either 35 or 36. In the next section, we report our obtained results.

5 Description of Results

In the training step, on the training collection, we obtain the optimal parameter setting and the corresponding optimal normalisation effect NE using relevance assessment. The obtained results in the training step are listed in table 2.

Moreover, the experiments on the four collections confirm that for both the normalisation 2 and BM25's normalisation method, the corresponding unique maximum $NE_D(\alpha)$ value does exist. Our new normalisation effect function in Equation (9) is indeed applicable to both normalisation methods. The parameter values giving the maximum $NE_D(\alpha)$ value are listed in table 3.

Tables 4, 5 and 6 provide the evaluation results for short, normal and long queries, respectively. In the three tables, the values of the parameter b of BM25's normalisation method and parameter c of the normalisation 2 are obtained using our tuning approach. MAP_d and MAP_t are the mean average precision obtained

Table 2. The optimal NE values and the corresponding parameter values for the training collection with respect to the three types of queries

	Short	Normal	Long
BM25			
NE	+0.8571	-0.9878	-0.9307
b	0.35	0.65	0.75
PL2			
NE	-0.9595	+0.9792	-0.9874
c	7	1.40	1

Table 3. The parameter value that gives the unique maximum $NE_D(\alpha)$ with respect to the three types of queries for the four collections used in our experiments

	Short Queries	Normal Queries	Long Queries
disk1&2			
b	0.55	0.60	0.63
c	2.55	2.14	1.85
disk4&5			
b	0.59	0.61	0.65
c	1.70	1.53	1.27
WT2G			
b	0.49	0.57	0.67
c	2.95	2.05	1.29
WT10G			
b	0.49	0.60	0.69
c	2.71	1.60	0.99

Table 4. Evaluation results for short queries on the three collections

Collection	parameter	MAP_d	MAP_t	Δ (%)	Wilc.
BM25					
disk4&5	0.40	0.2418	0.2534	+4.80	5.271e-09*
WT2G	0.30	0.2601	0.3161	+21.53	3.598e-06*
WT10G	0.27	0.1868	0.2110	+12.96	2.995e-06*
PL2					
disk4&5	3.63	0.2570	0.2533	-1.44	2.115e-06*
WT2G	10.99	0.3099	0.3164	+2.10	0.0008*
WT10G	13.13	0.2092	0.2095	≈ 0	0.5746

using the default setting and our tuning method, respectively. Δ (%) is the improvement obtained by our tuning method in percentage. Wilc. stands for the significance values according to the Wilcoxon test. A significance value marked

Table 5. Evaluation results for normal queries on the three collections

Collection	parameter	MAP_d	MAP_t	Δ (%)	Wilc.
BM25					
disk4&5	0.66	0.2461	0.2478	+0.69	0.0005*
WT2G	0.59	0.2527	0.2630	+4.08	0.0104*
WT10G	0.58	0.1776	0.1872	+5.29	0.0004*
PL2					
disk4&5	1.06	0.2361	0.2337	-1.02	0.9676
WT2G	2.33	0.2406	0.2490	+3.49	0.0072*
WT10G	2.65	0.1779	0.1875	+5.40	0.0116*

Table 6. Evaluation results for long queries on the three collections

Collection	parameter	MAP_d	MAP_t	Δ (%)	Wilc.
BM25					
disk4&5	0.76	0.2857	0.2858	≈ 0	0.4652
WT2G	0.73	0.2805	0.2802	≈ 0	0.1402
WT10G	0.70	0.2311	0.2338	+1.17	0.0042*
PL2					
disk4&5	2.23	0.2703	0.2769	+2.44	0.0150*
WT2G	4.80	0.2523	0.2679	+6.18	0.2507
WT10G	5.58	0.2235	0.2288	+2.37	0.6702

Table 7. The computational cost of the tuning process on the three collections for evaluation. The cost is measured in seconds

	Short	Normal	Long
disk4&5			
BM25	182.079s	249.994s	412.694s
PL2	222.955s	266.397s	478.540s
WT2G			
BM25	114.103s	138.249s	215.423s
PL2	240.584s	209.395s	275.028s
WT10G			
BM25	360.879s	672.493s	934.130s
PL2	542.648s	597.100s	981.056s

with a star indicates a statistically significant difference at the 0.05 level. From the results, we have the following observations:

– The tuning method significantly outperforms our baselines in most cases, apart from the 7th row in table 4, where there is a 1.44 percent negative improvement.

Table 8. Results on the WT2G collection obtained by using the query simulation method and the real queries, respectively

Query Type	Real	Sim.	MAP_r	MAP_s
BM25				
Short	0.27	0.30	0.3181	0.3159
Normal	0.59	0.59	0.3161	0.3161
Long	0.75	0.73	0.2805	0.2802
PL2				
Short	10.91	10.99	0.3166	0.3164
Normal	2.19	2.33	0.2483	0.2490
Long	4.28	4.80	0.2698	0.2679

Table 9. Results on the WT10G collection obtained by using the query simulation method and the real queries, respectively

Query Type	Real	Sim.	MAP_r	MAP_s
BM25				
Short	0.24	0.27	0.2112	0.2110
Normal	0.58	0.58	0.1872	0.1872
Long	0.73	0.70	0.2320	0.2338
PL2				
Short	13.29	13.13	0.2095	0.2095
Normal	2.42	2.65	0.1879	0.1875
Long	4.35	5.58	0.2336	0.2288

- Our tuning method works better for the two Web collections than for the disk4&5 of the TREC collection. This confirms our suggestion in the previous section, i.e. the baseline parameter settings for the classical collections might not be appropriate for the Web collections. Consequently, our tuning approach outperforms our baselines on the Web collections.
- For the normalisation 2, it seems that our tuning method works the best for long queries, while it achieves comparable performance with the baseline for short and normal queries.
- On the contrary, for BM25's normalisation method, our tuning method works better for short queries, although its performance with normal and long queries is at least as good as the baseline.

We report also the efficiency of our tuning method. Table 7 provides the computational cost of our tuning process for the three types of queries. As shown in the table, the cost of the tuning process is insignificant. Note that on a particular collection and for a particular type of queries, we only need to run the tuning process once during the indexing process.

To test our query simulation method of Section 3.2, we compared the results obtained by using two different sampling methods, i.e. query simulation and the real provided TREC queries, on the three test collections. Because of the space

limitation, we only report the results on the WT2G and WT10G collections. In tables 8 and 9, the second and the fourth columns correspond to the parameter values and mean average precision obtained using the real queries, respectively; the third and fifth columns correspond to the results obtained by the query simulation method. As shown in the tables, we find almost no difference between the obtained results, excepting the result for long queries using PL2 on WT10G. In this case, the simulated queries perform slightly less compared to the real queries. Note that both sampling methods result in a better retrieval performance than our robust baselines (see tables 4, 5 and 6).

In summary, our new normalisation effect function (see Equation (9)) is applicable to both PL2 and BM25. Moreover, adopting our query simulation method of Section 3.2, the tuning step does not involve the use of real user queries. This simulation method successfully samples the document length leading to an optimised tuning of the tf normalisation parameter as shown in the obtained results. According to the experiments, the new tuning method achieves robust and effective retrieval performance over the three diverse TREC collections with a marginal computational cost.

6 Conclusions and Future Directions

In this paper, we have proposed a term frequency (tf) normalisation tuning method, which refines and extends our methodology proposed in [7]. We have applied a new normalisation effect function by changing the definition of relation $NE_D(\alpha)$, i.e. the normalisation effect on the set of documents with at least one query term, such that the application of the tuning method can be extended to BM25. We have also proposed a novel query simulation method to avoid the use of real user queries in the tuning step.

Using various and diverse TREC collections, we have evaluated our new tuning method using both the normalisation 2 and BM25's normalisation method. In particular, by extending the application of the tuning method to BM25, the flexibility of the methodology in [7] has been significantly enhanced.

Compared to the used robust baselines, which are the empirically-based recommended parameter settings of the two applied normalisation methods, our new tuning method achieves robust and effective retrieval performance. Indeed, the results show that our method is at least as good as the baselines, and significantly outperforms them in most cases. Moreover, the computational cost of our tuning process is marginal.

In the future, we will investigate further applications of the tuning method by measuring the normalisation effect. In particular, we are currently investigating the application of our tuning method in the context of XML retrieval and intranet search. Moreover, so far, the tuning method has only been evaluated for ad-hoc tasks. We plan to apply the tuning method to other tasks, such as topic-distillation and named-page finding tasks.

Acknowledgments

This work is funded by the Leverhulme Trust, grant number F/00179/S. The project funds the development of the Smooth project, which investigates the term frequency normalisation (URL: http://ir.dcs.gla.ac.uk/smooth). The experimental part of this paper has been conducted using the Terrier framework (EPSRC, grant GR/R90543/01, URL: http://ir.dcs.gla.ac.uk/terrier).

References

1. G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Department of Computing Science, University of Glasgow, 2003.
2. G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. In *ACM Transactions on Information Systems (TOIS)*, volume 20(4), pages 357 – 389, 2002.
3. J. Callan and M. Connell. Query-based sampling of text databases. In *ACM Transactions on Information Systems (TOIS)*, pages 97 – 130, Volume 19, Issue 2, April, 2001.
4. A. Chowdhury, M. C. McCabe, D. Grossman, and O. Frieder. Document normalization revisited. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 381–382, Tampere, Finland, 2002.
5. D. Hawking. Overview of the TREC-9 Web Track. In *Proceedings of the Nineth Text REtrieval Conference (TREC-9)*, pages 87 – 94, Gaithersburg, MD, 2000.
6. D. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 Web Track. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 131 – 150, Gaithersburg, MD, 1999.
7. B. He and I. Ounis. A study of parameter tuning for term frequency normalization. In *Proceedings of the Twelveth ACM CIKM International Conference on Information and Knowledge Management*, pages 10 – 16, New Orleans, LA, 2003.
8. C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Department of Computer Science, University of Glasgow, 1979.
9. S. Robertson, S. Walker, M. M. Beaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In *NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-4)*, pages 73 – 96, Gaithersburg, MD, 1995.
10. C. Silverstein, M. R. Henzinger, H. Marais, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.
11. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.
12. K. Sparck-Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. *Information Processing and Management*, 36(2000):779 – 840, 2000.