A Study of the Dirichlet Priors for Term Frequency Normalisation

Ben He
Department of Computing Science
University of Glasgow
Glasgow, United Kingdom
ben@dcs.gla.ac.uk

Iadh Ounis
Department of Computing Science
University of Glasgow
Glasgow, United Kingdom
ounis@dcs.gla.ac.uk

ABSTRACT

In Information Retrieval (IR), the Dirichlet Priors have been applied to the smoothing technique of the language modeling approach. In this paper, we apply the Dirichlet Priors to the term frequency normalisation of the classical BM25 probabilistic model and the Divergence from Randomness PL2 model. The contributions of this paper are twofold. First, through extensive experiments on four TREC collections, we show that the newly generated models, to which the Dirichlet Priors normalisation is applied, provide robust and effective performance. Second, we propose a novel theoretically-driven approach to the automatic parameter tuning of the Dirichlet Priors normalisation. Experiments show that this tuning approach optimises the retrieval performance of the newly generated Dirichlet Priors-based weighting models.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Retrieval models

General Terms

Experimentation, Performance, Theory

Keywords

Term frequency normalisation, weighting model, Dirichlet Priors

1. INTRODUCTION

Document ranking is a crucial issue in Information Retrieval (IR), which is usually based on a weighting model [9]. Almost all weighting models take term frequency (tf), the number of occurrences of a query term in a document, into consideration as a basic factor for document ranking.

However, the term frequency is dependent on the document length, i.e. the number of tokens in a document. In [11], Singhal et al. summarised the following two aspects of the effect of document length on tf:

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

As a consequence, tf needs to be normalised by using a technique called *term frequency normalisation*.

Many normalisation methods have been developed in the past, including the normalisation 2 [1] and BM25's normalisation component [10]. Moreover, as suggested in [1], the Dirichlet Priors, which have been applied to the IR language modeling approach [13], can also be applied to the tf normalisation. We will introduce these normalisation methods in the next section. In this paper, we denote the tf normalisation using the Dirichlet Priors as the *Dirichlet Priors normalisation*, and denote the weighting model, to which the Dirichlet Priors normalisation is applied, as the *Dirichlet Priors-based model*.

In this paper, we study the application of the Dirichlet Priors normalisation on a representative of two families of weighting models, including the classical BM25 probabilistic model [10] and the Divergence from Randomness (DFR) PL2 model [1]. In our extensive experiments, we show that the Dirichlet priors-based models achieve robust and effective retrieval performance over diverse TREC collections. Experiments also show that there is a justifiable need for the tuning of the parameter of the Dirichlet Priors normalisation. In particular, we propose a novel automatic theoretically-driven tuning methodology for the Dirichlet Priors normalisation. As mentioned above, there is a dependence between tf and the document length. Therefore, the purpose of the tf normalisation is to adjust the dependence between the normalised term frequency and the document length. In our tuning method, this dependence is interpreted as the correlation between the two variables. In our experiments, we show that the optimal parameter values, which give the best mean average precision, result in stable correlation measures. In particular, our tuning approach seems to be both collection-independent and query-independent.

The remainder of this paper is organised as follows. In Section 2, we introduce some related work, including the above mentioned normalisation methods and the weighting models to which these normalisation methods are applied. We apply the Dirichlet Priors normalisation to BM25 and PL2 in Section 3 and describe the proposed methodology for tuning the Dirichlet Priors normalisation in Section 4.

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

In Sections 5 and 6, we present our experimental setting and evaluation results. Finally, we conclude the work and suggest future directions in Section 7.

2. RELATED WORK

As one of the most established weighting models, BM25 computes the relevance score of a document d for a query Q by the following formula:

$$score(d,Q) = \sum_{t \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} \quad (1)$$

where qtf is the query term frequency; $w^{(1)}$ is the *idf* factor, which is given by:

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

N is the number of documents in the whole collection. N_t is the document frequency of term t.

K is:

$$k_1((1-b) + b\frac{l}{avg\_l})$$

l and avg_l are the document length and the average document length in the collection, respectively. The document length refers to the number of tokens in a document. k_1, k_3 and b are parameters. The default setting is $k_1 = 1.2$, $k_3 = 1000$ and $b = 0.75$ [10]. qtf is the number of occurrences of a given term in the query; tf is the within document frequency of the given term.

In the following derivation, we show how the BM25 model implicitly employs a tf normalisation component.

Let $tfn = \frac{tf}{(1-b)+b\cdot\frac{l}{avg\_l}}$, where tfn denotes the normalised term frequency, we obtain:

$$\begin{aligned} score(d,Q) &= \sum_{t \in Q} w^{(1)} \frac{k_1+1}{\frac{k_1}{tfn}+1} \frac{(k_3+1)qtf}{k_3+qtf} \\ &= \sum_{t \in Q} w^{(1)} \frac{(k_1+1)tfn}{k_1+tfn} \frac{(k_3+1)qtf}{k_3+qtf} \quad (2) \end{aligned}$$

Hence the term frequency normalisation component of the BM25 formula can be seen as:

$$tfn = \frac{tf}{(1-b) + b\cdot\frac{l}{avg\_l}} \quad (3)$$

The above BM25's normalisation component can be seen as a generalisation of Singhal et al.'s *pivoted normalisation* for normalising the $tf \cdot idf$ weight [11].

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 a query Q is given by:

$$\begin{aligned} score(d,Q) &= \sum_{t \in Q} qtf \cdot \frac{1}{tfn+1}\big(tfn \cdot \log_2 \frac{tfn}{\lambda} \quad (4) \\ &\quad +(\lambda - tfn)\cdot\log_2 e + 0.5\cdot\log_2(2\pi\cdot tfn)\big) \end{aligned}$$

where λ is the mean and variance of a Poisson distribution. qtf is the query term frequency.

The normalised term frequency tfn is given by the so-called *normalisation 2* [1, 7]:

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

where l is the document length and avg_l is the average document length in the whole collection. tf is the original within document term frequency. c is the free parameter of the normalisation method.

The Dirichlet Priors stand for the priors in a Dirichlet distribution, which is a generalisation of the Beta distribution in a multinomial case. In [13], Zhai & Lafferty have applied the Dirichlet priors in defining their language model for IR. It was also suggested by Amati that the Dirichlet Priors can be used for the tf normalisation [1]:

$$tfn = \frac{tf + \mu \cdot \frac{tf_c}{l_c}}{l+\mu} \cdot \mu \quad (6)$$

where tfn is the normalised term frequency. l is the document length. tf_c is the frequency of the given query term in the collection. l_c is the number of tokens in the whole collection. μ is the parameter of the Dirichlet Priors normalisation.

3. APPLICATIONS OF THE DIRICHLET PRIORS NORMALISATION

In this section, we apply the Dirichlet Priors normalisation to both PL2 (see Equation (4)) and BM25 (see Equation (1)).

As shown in the previous section, both BM25 and PL2 weighting models employ a tf normalisation component. By replacing the tf normalisation components of the two models with the Dirichlet Priors normalisation, we generate the following new models:

- *The Dirichlet Priors-based BM25 model (BM3)* is given as our derivation of BM25 in Equation (2), where the normalised term frequency tfn is given by the Dirichlet Priors normalisation in Equation (6). In the rest of this paper, we denote the Dirichlet Priors-based BM25 as BM3[1].

- *The Dirichlet Priors-based PL2 model (PL3)* is given as the PL2 model in Equation (4), where the normalised term frequency tfn is given by the Dirichlet Priors normalisation in Equation (6). We denote this new model as PL3.

In Section 6.1, through extensive experiments, we show that the newly generated Dirichlet Priors-based models lead to robust and effective retrieval performance. Experiments also show that the optimal parameter setting of the parameter μ, which gives the best mean average precision, varies on different collections and query types, indicating that there is a need for tuning the parameter. We describe the proposed tuning method in the next section.

[1]In [1], Amati denotes the Dirichlet Priors normalisation as the normalisation 3. In this paper, we follow his notation and denote a Dirichlet Priors-based model M as M3.

4. PARAMETER TUNING FOR THE DIRICHLET PRIORS NORMALISATION

In this section, we propose a novel theoretically-driven methodology for the automatic tuning of the Dirichlet Priors normalisation. Our proposed methodology is based on measuring the correlation ($corr$) [4] of the normalised term frequency (tfn) with the document length (l) for a given query term, which is given by:

$$corr(tfn, l) = \frac{COV(tfn, l)}{\sigma(tfn)\sigma(l)} \quad (7)$$

where COV stands for covariance. $\sigma(l)$ is the standard deviation of the length of the documents containing the given query term. $\sigma(tfn)$ is the standard deviation of the normalised term frequency of the given query term in all the documents containing the term.

As introduced in Section 1, the purpose of the tf normalisation is to smooth the dependence between tfn and document length, which can be represented by the above correlation formula. We believe that the tf normalisation can be seen as the tf density estimation of the document length. On different collections, the ideal tf density function should result in similar correlation of tfn with the document length. Therefore, the underlying hypothesis of our tuning method is the following:

> Hypothesis:
>
> On different collections, the optimal term frequency normalisation parameter setting provides similar average correlation of the normalised term frequency with the document length for a given set of query terms.

In the above hypothesis, the optimal parameter setting refers to the parameter value that provides the highest mean average precision. Based on the hypothesis, the tuning of the Dirichlet Priors normalisation becomes the issue of identifying the parameter setting that gives an optimal average correlation of tfn with l for the given set of query terms. The proposed tuning methodology can be described as follows:

1. On a training collection, we obtain the optimal parameter setting by using relevance assessment, and then compute the corresponding average $corr(tfn, l)$ value, which is a collection-independent and query-independent constant, based on our hypothesis.

2. On a given new collection, and for a given new query set, we apply the parameter setting that gives the optimal average $corr(tfn, l)$ obtained on the training collection.

Note that in the above tuning process, for a given set of query terms and a particular parameter value, the $corr(tfn, l)$ value is the average correlation measure for the given set of query terms. Ideally, we could tune the parameter setting for each query term, and apply different parameter settings for different query terms. However, as it is very expensive to carry out such a term-based parameter tuning mechanism, in this paper, we rather assume an optimal average correlation measure so that we do the tuning process for a set of query terms in a batch mode, and apply a unique parameter setting for all the terms in the given set of queries.

Table 1: The five weighting models involved in our experiments. The last three models are our baselines.

Model	Formula	Normalisation
BM3	Equation (2)	Equation (6)
PL3	Equation (4)	Equation (6)
BM25	Equation (2)	Equation (3)
PL2	Equation (4)	Equation (5)
$tf \cdot idf$	Equation (8)	implicitly applied

According to the study by Zhai & Lafferty [13], the optimal setting of the parameter μ of the Dirichlet Priors Smoothing changes with different collections and query sets. Based on the above hypothesis, our explanation is that the length distribution of documents containing a given query term varies with the change of data set, including the collection and query set. As a consequence, a particular correlation $corr(tfn, l)$ value corresponds to different μ values on different data sets, and therefore results in different optimal parameter values. In our experiments in Section 6.2, we show that our hypothesis holds on diverse TREC collections.

5. EXPERIMENTAL ENVIRONMENT

In our experiments, we evaluate the Dirichlet Priors-based models, i.e. BM3 and PL3, on diverse collections using three baselines, which are BM25, PL2 and the classical $tf \cdot idf$ weighting models. Table 1 lists the five models involved in our experiments.

Our implementation of the $tf{\cdot}idf$ baseline weighting model is a combination of the Okapi's tf [10] and Sparck-Jones' idf [12]:

$$score(d, Q) = \sum_{t \in Q} qtf \cdot \frac{k_1 tf}{tf + k_1(1 - b + b\frac{l}{l_{coll}})} \cdot \log_2(\frac{N}{N_t} + 1) \quad (8)$$

where qtf is the query term frequency of t. N is the number of documents in the whole collection and N_t is the document frequency of t. l and l_{coll} are the number of tokens in d and in the whole collection, respectively. k_1 and b are parameters and their default settings are $k_1 = 1.2$ and $b = 0.75$, respectively [10].

We experiment on four TREC collections to evaluate the Dirichlet Priors-based models. The four used collections are the disk1&2, disk4&5 (minus the Congressional Record on disk4) of the classical TREC collections[2], and the WT2G [6] and WT10G [5] Web collections. The test queries are TREC topics that are numbered from 51 to 200 for the disk1&2, 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 (see Table 2).

Table 2 lists the test TREC topics, the number of documents, and the standard deviation of document length in each collection. As shown in the last row, the document length distribution of the four collections is quite different, which indicates that the newly generated Dirichlet Priors-

[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

Table 2: 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_{lcoll} is the standard deviation of document length in the whole collection.

	disk1&2	disk4&5	WT2G	WT10G
Topics	51-200	301-450 and 601-700	401-450	451-550
N	741860	528155	247491	1692044
σ_{lcoll}	862.4977	558.1173	2009.3760	2303.4063

based models and the tuning method are evaluated on diverse collections.

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 models and the tuning method. The three types of queries are:

- **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.

Regarding the parameter setting of the baseline models, for the BM25 model, we use the default setting, which is $b = 0.75$, $k_1 = 1.2$ and $k_3 = 1000$ [10]; For the PL2 model, 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 disk1&2 as the baseline, i.e. $c = 1.4$ for normal queries; For the $tf \cdot idf$ model, we also apply the default setting that is $k_1 = 1.2$ and $b = 0.75$ [10].

For the Dirichlet Priors-based models, including BM3 and PL3, we test a series of values for the parameter μ, ranging from 0 (exclude 0) to 10,000, in order to extensively study the performance of the Dirichlet Priors-based models and show the need for the parameter tuning of the Dirichlet Priors normalisation.

Moreover, for testing our automatic tuning approach to the Dirichlet Priors normalisation, we use the disk1&2 as the training collection, and evaluate on the other three collections. An advantage of using this training collection is that it has a relatively large number of available training queries, which are the TREC topics numbered from 51 to 200. After obtaining the optimal $corr(tfn, l)$ value on the training collection using the corresponding relevance assessment, we evaluate our approach on the other three TREC collections. For the computation of the $corr(tfn, l)$ value, terms that appear in only one document are ignored in order to avoid a zero correlation.

Our baseline for the evaluation of the tuning method is the optimal setting on the training collection, which is an empirical setting. Moreover, we compare the performance of our tuning method with the best manually obtained parameter setting using relevance assessment.

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

Figure 1: The mean average precision (MAP) against the parameter μ for *short* queries on the four used TREC collections.

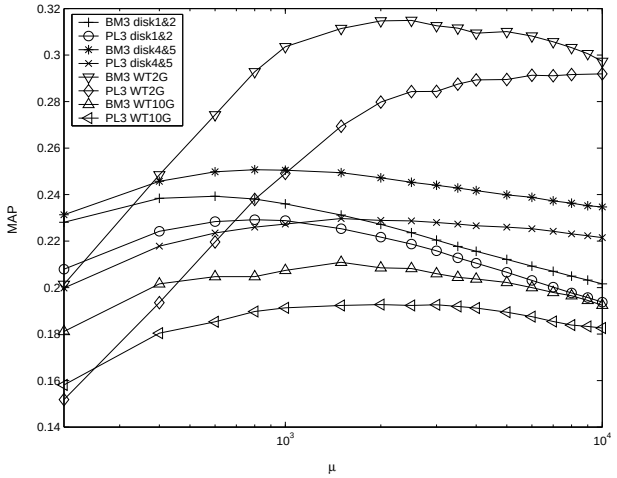

Table 3: The best manually obtained settings of the parameter μ, selected from a wide range of values, on four collections for three types of queries.

	Short	Normal	Long
BM3			
disk1&2	400	200	200
disk4&5	950	450	300
WT2G	2700	1400	650
WT10G	1500	400	500
PL3			
disk1&2	800	200	200
disk4&5	1600	400	400
WT2G	9700	2300	1200
WT10G	2600	600	900

6. DISCUSSION OF RESULTS

In Section 6.1, we start with presenting the results obtained using the Dirichlet Priors-based models with various parameter settings, showing the importance of our parameter tuning method. Then, we compare the performance of the Dirichlet Priors-based models with three baseline models. In Section 6.2, we show that our automatic tuning method achieves robust and effective performance. In particular, the tuning method's performance differs marginally from the best manually obtained setting using relevance assessment.

6.1 Performance of the Dirichlet Priors-based Models

Table 3 contains the manually obtained optimal parameter values on different collections with respect to the three query types. As we can see, on diverse collections, the optimal parameter values are quite different. Moreover, Figures 1, 2 and 3 sketch the plots of the parameter μ against mean average precision on the four used TREC collections for the three types of queries, respectively. In the figures, the curves of the parameter μ against mean average precision behave differently. Table 3 and Figures 1, 2 and 3 show that there is a justifiable need for the parameter tuning.

Figure 2: The mean average precision (MAP) against the parameter μ for *normal* queries on the four used TREC collections.

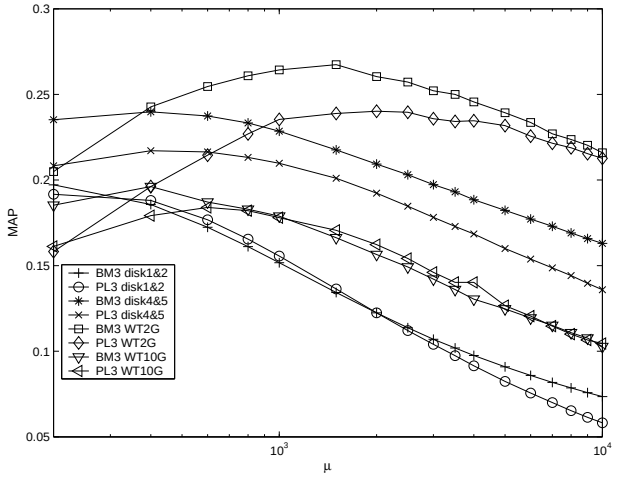

Figure 3: The mean average precision (MAP) against the parameter μ for *long* queries on the four used TREC collections.

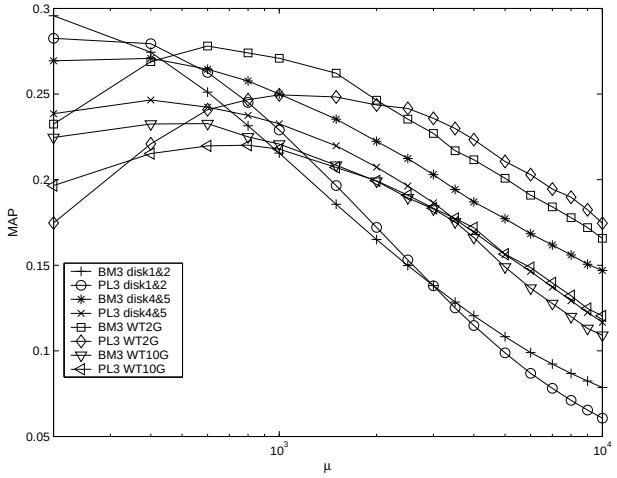

Table 4 provides the mean average precision obtained by using the baselines and the Dirichlet Priors-based models. From the table, we have the following observations:

- BM3 clearly outperforms BM25 for short queries, and achieves comparable performance for normal and long queries.
- Overall, PL2 provides slightly higher mean average precision than PL3, but PL3's performance is still shown to be robust and better than PL2 in some cases.
- The Dirichlet Priors-based models, i.e. BM3 and PL3, generally outperform $tf \cdot idf$ for short and normal queries, and provide comparable performance for long queries.

Overall, as shown by the results, the newly generated Dirichlet Priors-based models outperform the $tf \cdot idf$ model and achieve comparable performance with the robust BM25 and PL2 models.

Table 4: The obtained mean average precision on the four collections using the five models. The applied parameter settings for BM3 and PL3 are taken from Table 3.

	$tf \cdot idf$	BM25	PL2	BM3	PL3
Short queries					
disk1&2	.2214	.2226	.2338	.2395	.2293
disk4&5	.2431	.2418	.2570	.2508	.2301
WT2G	.2615	.2600	.3102	.3157	.2930
WT10G	.1866	.1868	.2092	.2109	.1933
Normal queries					
disk1&2	.1772	.1913	.1905	.1972	.1972
disk4&5	.2437	.2461	.2366	.2399	.2171
WT2G	.2407	.2528	.2407	.2679	.2410
WT10G	.1739	.1776	.1779	.1962	.1840
Long queries					
disk1&2	.2898	.2981	.2958	.2957	.2854
disk4&5	.2797	.2858	.2704	.2724	.2465
WT2G	.2772	.2807	.2520	.2790	.2523
WT10G	.2290	.2310	.2235	.2338	.2220

Table 5: The optimal parameter values and the corresponding correlation measures of the normalised tf with the document length on the training collection. The value marked with * is taken as the optimal constant $corr(tfn, l)$ of our tuning method.

	Short	Normal	Long
	BM3		
μ	400	200	200
$corr(tfn, l)$	-.1042*	-.1086	-.1253
	PL3		
μ	800	200	200
$corr(tfn, l)$	-.08581	-.1086	-.1253

6.2 Performance of the Automatic Tuning Approach

In this section, we start with presenting how we detect the optimal $corr(tfn, l)$ on the training collection[3], and then discuss the obtained evaluation results.

Table 5 contains the optimal parameter values and corresponding $corr(tfn, l)$, i.e correlation of the normalised term frequency with the document length on the training collection. Note that the provided correlation values are the mean of the correlation for each query term. We can see that the optimal parameter values for the two Dirichlet Priors-based models are very similar, and for different types of queries, the optimal parameter settings result in relatively similar correlation measures. Although for short queries, the optimal settings are not identical ($\mu = 800$ for BM3 and $\mu = 400$ for PL3), for both models, $\mu = 800$ and $\mu = 400$ differs marginally from each other in terms of mean average precision[4], indicating that the underlying hypothesis of our tuning method stands (see Section 4 for the hypothe-

[3]The notion of the "optimal $corr(tfn, l)$" refers to the $corr(tfn, l)$ value given by the optimal parameter setting. See step 1 of our tuning method in Section 4.

[4]For BM3, $\mu = 400$ and $\mu = 800$ provide mean average precision of .2384 and .2381, respectively. For PL3, $\mu = 400$ and $\mu = 800$ provide mean average precision of .2242 and .2292, respectively.

Table 6: The correlation measures for some query terms on the training collection for different parameter values.

μ	200	600	1000	4000	8000
airbus	-.1307	-.0752	-.0487	.0073	.0256
sanction	-.1184	-.1069	-.0963	-.0470	-.0157
contract	-.0863	-.0418	-.0129	.0742	.1144
merit	-.2231	-.2164	-.2026	-.1367	-.0959
iran	-.0456	-.0341	-.0291	-.0207	-.0203

sis). Therefore, for the tuning process, we take the optimal $corr(tfn, l)$ for short queries using BM3 as the optimal constant $corr(tfn, l)$ (see the value marked with a star in Table 3). We choose the optimal $corr(tfn, l)$ value for the short queries as the optimal constant correlation because of the fact that query terms in the titles are generally more informative than those in the descriptions and narratives. Therefore, terms in the titles can be more reliable than terms in other topic fields in inferring an optimal parameter setting. On a collection other than the training collection, we apply such a parameter setting that it gives this constant.

Table 6 presents some examples of the correlation measures on the training collection. As we can see, with respect to a particular parameter value, the correlation measures of different terms are diverse. Therefore, a term-based tuning approach might achieve higher precision/recall than just computing the mean of the correlation measures of query terms. However, as it is quite time-consuming to carry out a tuning process for each query term, we rather follow the proposed approach in this paper (see Section 4). Later we show that our approach achieves robust retrieval performance in the evaluation.

Tables 7 and 8 compare the mean average precision (MAP) obtained by using our tuning method with the MAP obtained by using the optimal values on the training collection. In the two tables, MAP_b and MAP_t stands for the mean average precision (MAP) obtained by using the baseline setting and the tuning method, respectively. μ stands for the parameter setting estimated by the tuning method. Δ is the percentage of improvement using the tuning method. A p-value marked with star indicates a significant difference between the results at 0.05 level according to the Wilcoxon test. As we can see, in most cases, our tuning method either significantly outperforms the baseline, or achieves comparable performance with the baseline.

Moreover, Tables 9 and 10 compare the performance of the tuning method with that of the manually optimal parameter setting obtained using relevance assessment. The notations in the two tables are the same as in Table 7. As can be seen from the tables, the performance of our tuning method is similar with the manual setting in most cases, and the difference of mean average precision is usually marginal. This indicates that the underlying hypothesis of our tuning method indeed holds (see Section 4 for the hypothesis). Overall, our tuning method provides effective and reliable retrieval performance over diverse TREC document and Web collections.

7. CONCLUSION AND FUTURE DIRECTION

In this paper, we have studied the application of the Dirichlet Priors to the term frequency normalisation. In particular, we have applied the Dirichlet Priors normalisation to a representative of two families of weighting models, i.e. the classical BM25 probabilistic model and the Divergence from Randomness PL2 model. By replacing the tf normalisation components of the two models with the Dirichlet Priors normalisation, the newly generated weighting models are shown to be robust and effective in our experiments.

Table 7: Results for BM3. This table compares the performance obtained by using the optimal setting on the training collection with that using the tuning method. The settings for μ are automatically obtained using our tuning method.

	μ	MAP_b	MAP_t	Δ	p-value
Short query					
disk4&5	668	.2490	.2499	+0.36	.02569*
WT2G	2266	.2692	.3151	+17.05	1.174e-05*
WT10G	1782	.2040	.2093	+2.60	.3253
Normal query					
disk4&5	578	.2352	.2377	+1.06	.7677
WT2G	1514	.2427	.2674	+10.18	.08932
WT10G	1168	.1853	.1745	-5.83	.7098
Long query					
disk4&5	610	.2694	.2646	-1.78	.1841
WT2G	1441	.2324	.2624	+4.30	.1503
WT10G	1212	.2246	.2155	-4.05	.6677

Table 8: Results for PL3. This table compares the performance obtained by using the optimal setting on the training collection with that using the tuning method. The settings for μ are automatically obtained using our tuning method.

	μ	MAP_b	MAP_t	Δ	p-value
Short query					
disk4&5	668	.2260	.2243	-0.75	.0887
WT2G	2266	.1935	.2833	+46.41	3.828e-07*
WT10G	1782	.1804	.1923	+6.60	.002791*
Normal query					
disk4&5	578	.2083	.2168	+4.08	.00288
WT2G	1514	.1582	.2388	+50.95	1.769e-06
WT10G	1168	.1792	.1746	-2.57	.9238
Long query					
disk4&5	610	.2385	.2421	+1.51	.1378
WT2G	1441	.1747	.2495	+42.82	2.461e-05
WT10G	1212	.2152	.2135	-0.79	.8997

A major contribution of this paper is the proposed novel theoretically-driven automatic tuning method for the Dirichlet Priors normalisation. The proposed approach interprets the dependence between the normalised term frequency and the document length as the correlation between the two variables. Experiments on the TREC collections show that the underlying hypothesis of our tuning approach holds. Evaluation results also show that the tuning method significantly outperforms the baseline and its performance differs marginally from the manual setting using relevance assessment.

There are some interesting future directions that will help in better understanding the tf normalisation. We plan to study the application of other smoothing methods, e.g. the Jelinek-Mercer smoothing [3, 8], to the tf normalisation.

Table 9: Results for BM3. This table compares the performance obtained manually using relevance assessment with that using the automatic tuning method.

	MAP_b	MAP_t	Δ	p-value
Short query				
disk4&5	.2508	.2499	-0.36	.07424
WT2G	.3157	.3151	-0.19	.7196
WT10G	.2109	.2093	-0.76	.3567
Normal query				
disk4&5	.2399	.2377	-0.92	3.709e-4*
WT2G	.2679	.2674	-0.19	.1812
WT10G	.1962	.1745	-11.06	.006093*
Long query				
disk4&5	.2724	.2646	-2.86	6.769e-4*
WT2G	.2790	.2624	-5.95	.0245*
WT10G	.2338	.2155	-7.83	.002763*

Table 10: Results for PL3. This table compares the performance obtained manually using relevance assessment with that using the automatic tuning method.

	MAP_b	MAP_t	Δ	p-value
Short query				
disk4&5	.2271	.2243	-1.23	6.137e-07*
WT2G	.2930	.2833	-3.31	.2732
WT10G	.1933	.1923	-0.52	.8031
Normal query				
disk4&5	.2169	.2168	≈ 0	.2573
WT2G	.2410	.2388	-0.91	.6816
WT10G	.1840	.1746	-5.22	.03436*
Long query				
disk4&5	.2465	.2421	-1.78	.005519*
WT2G	.2523	.2495	-1.11	.2127
WT10G	.2220	.2135	-3.83	.0175*

In particular, we will apply the proposed tuning method to these classical smoothing methods.

It will also be interesting to device a term-based tuning mechanism for the Dirichlet Priors normalisation. As suggested previously, a term-based tuning mechanism could achieve a better retrieval performance though it would have a high computational cost. A possible solution for lowering the overhead is to enable tuning only for the most informative terms in a query, while applying the default parameter setting for the rest of the query terms.

8. 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 experiments were conducted using Terrier's IR platform, version 1.0.0. It is a modular platform for the rapid development of large-scale IR applications, providing indexing and retrieval functionalities. More information can be found from http://ir.dcs.gla.ac.uk/terrier.

9. 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, October 2002.

[3] S. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310 – 318, San Francisco, CA, 1996.

[4] M. DeGroot. *Probability and Statistics*. Addison Wesley, 2nd edition edition, 1989.

[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. Tuning Term Frequency Normalisation for BM25 and DFR Models. In *Proceedings of the 27th European Conference on Information Retrieval (ECIR'05)*, pages 200 – 214, Santiago de Compostela, Spain, March, 2005.

[8] F. Jelinek and R. Mercer. Interpolated estimation of markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*, pages 381 – 402, Amsterdam, The Netherlands, 1980.

[9] C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Department of Computer Science, University of Glasgow, 1979.

[10] S. Robertson, S. Walker, 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.

[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, Zurich, Switzerland, 1996.

[12] K. Sparck-Jones. A statistical interpretation of term specificity and its application to retrieval. *Journal of Documentation*, (28):11 – 21, 1972.

[13] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334 – 342, New Orleans, LA, 2001.