Setting Per-field Normalisation Hyper-parameters for the Named-Page Finding Search Task

Ben He and Iadh Ounis

Department of Computing Science
University of Glasgow
United Kingdom
{ben,ounis}@dcs.gla.ac.uk

Abstract. Per-field normalisation has been shown to be effective for Web search tasks, e.g. named-page finding. However, per-field normalisation also suffers from having hyper-parameters to tune on a per-field basis. In this paper, we argue that the purpose of per-field normalisation is to adjust the linear relationship between field length and term frequency. We experiment with standard Web test collections, using three document fields, namely the body of the document, its title, and the anchor text of its incoming links. From our experiments, we find that across different collections, the linear correlation values, given by the optimised hyper-parameter settings, are proportional to the maximum negative linear correlation. Based on this observation, we devise an automatic method for setting the per-field normalisation hyper-parameter values without the use of relevance assessment for tuning. According to the evaluation results, this method is shown to be effective for the body and title fields. In addition, the difficulty in setting the per-field normalisation hyper-parameter for the anchor text field is explained.

1 Introduction

In Information Retrieval (IR), it is a crucial issue to rank retrieved documents in decreasing order of relevance. A recent survey on the query logs from real Web search engine users concluded that the users rarely look beyond the top returned documents [9]. Therefore, it is important to rank the highly relevant documents at the top of the retrieved list. Usually, the document ranking is based on a *weighting model*. In particular, most weighting models apply a *term frequency (tf) normalisation* method to normalise term frequency, the number of occurrences of the query term in the document.

Various *tf* normalisation methods have been proposed in the literature, e.g. the pivoted normalisation [16] in the vector space model [15], the normalisation method of the BM25 weighting model [13], normalisation 2 [1] and normalisation 3 [1,8] in the Divergence from Randomness (DFR) framework [1]. All the above mentioned normalisation methods normalise term frequency according to document length, i.e. the number of tokens in the document. Each of the above

G. Amati, C. Carpineto, and G. Romano (Eds.): ECIR 2007, LNCS 4425, pp. 468–480, 2007.

mentioned normalisation methods involve the use of a hyper-parameter. The setting of these hyper-parameter values usually has an important impact on the retrieval performance of an IR system [2,7,8].

Recently, Robertson et al. and Zaragoza et al. proposed the per-field normalisation technique, which normalises term frequency on a per-field basis [14,18], by extending BM25's normalisation method [13]. The resulting field-based weighting model is called BM25F. Using BM25F, the retrieval process is performed on indices of different document fields, such as body, title, and anchor text of incoming links. Following [14,18], Macdonald et al. extended the PL2 DFR weighting model, by employing the per-field normalisation 2F [10]. Compared with *tf* normalisation on a single field, on one hand, per-field normalisation can significantly boost the retrieval performance, particularly for Web search [12,18]. On the other hand, per-field normalisation has a hyper-parameter for each document field used. Therefore, per-field normalisation has more hyper-parameters than *tf* normalisation on a single index of the whole collection, which requires a heavier training process to set the hyper-parameter values. Similarly to *tf* normalisation on a single field, the setting of the hyper-parameter values of per-field normalisation can significantly affect the retrieval performance. In particular, the optimal hyper-parameter setting of a document field, which provides the best retrieval performance, varies across different collections [12]. As a consequence, training is required on each given new collection to guarantee an effective retrieval performance.

In this paper, we study how the per-field normalisation hyper-parameter setting is related to the resulting retrieval performance. Our study follows Harter and Amati's idea that there is a linear relationship between term frequency and document length [1,6]. This linear relationship is indicated by the linear correlation between these two variables [1,6]. The study of this paper is focused on a typical Web search task, namely the named-page finding search task [17]. The main contributions of this paper are two-fold. First, we provide a better understanding in per-field normalisation. we suggest that the purpose of per-field normalisation is to adjust the linear relationship between term frequency and field length, i.e. the number of tokens in the field. This is our main argument. Experiments are conducted to study how per-field normalisation adjusts this linear relationship. Second, we devise and evaluate an automatic hyper-parameter setting method for the per-field normalisation hyper-parameters. The proposed method does not need relevance assessment for tuning, making it particularly practical in an operational and realistic setting.

The rest of this paper is organised as follows. Section 2 introduces the current main per-field normalisation techniques. Section 3 describes our main argument in this paper. Sections 4 and 5 describe the experimental methodology for investigating the linear relationship between field length and term frequency, and analyse the experimental results, respectively. Section 6 devises an automatic method for setting the per-field normalisation hyper-parameter values, which is evaluated is Section 7. Finally, Section 8 concludes the paper and suggests possible future work.

2 Per-field Normalisation

In the context of field-based retrieval, Robertson et al. proposed the idea of normalising term frequency on a per-field basis [14]. The extended BM25 field-based weighting model, called *BM25F*, assigns the relevance score of a document d for a query q as follows:

$$score(d,q) = \sum_{t \in q} w^{(1)} \frac{(k_1+1)tfn}{k_1+tfn} \frac{(k_3+1)qtf}{k_3+qtf} \tag{1}$$

where qtf is the query term frequency. k_1 and k_3 are parameters. The default setting is $k_1 = 1.2$ and $k_3 = 1000$ [13]. $w^{(1)}$ is the *idf* factor, which is given by:

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

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

The above BM25F's weighting function is the same as the one of BM25 [13]. Instead of normalising term frequency on a single index, BM25F applies a per-field normalisation method to assign the normalised term frequency *tfn* [18]:

$$tfn = \sum_f w_f \cdot tfn_f = \sum_f w_f \cdot \frac{tf_f}{(1-b_f) + b_f \cdot \frac{l_f}{avg\_l_f}} \tag{2}$$

where w_f is the weight of a field f, which reflects the relative contribution of a field to the document ranking. tfn_f is the normalised term frequency on field f. tf_f is the frequency of the query term in the field f of the document. b_f is the term frequency normalisation hyper-parameter of field f. l_f is field length, namely the number of tokens in field f of the document. avg_l_f is the average length of field f in the collection.

Moreover, following [18], Macdonald et al. extended the PL2 weighting model to cope with different document fields, within the Divergence from Randomness (DFR) probabilistic framework [1]. The idea of the DFR framework is to infer the informativeness of a query term in a document by measuring the divergence of the term's distribution in the document from a random distribution. The larger the divergence is, the more informative the query term is in the document. The *PL2F* field-based weighting model has the following weighting function:

$$\begin{aligned} score(d,q) = \sum_{t \in q} qtw \cdot \frac{1}{tfn+1} \big(& tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn) \cdot \log_2 e \\ & + 0.5 \cdot \log_2 (2\pi \cdot tfn) \big) \end{aligned} \tag{3}$$

where λ is the mean and variance of a Poisson distribution. It is given by $\lambda = tf_c/N$. tf_c is the frequency of the query term in the collection, and N is the number of documents in the collection. In PL2F, the normalised term frequency *tfn* is given by the so-called *Normalisation 2F* as follows:

$$tfn = \sum_f w_f \cdot tfn_f = \sum_f \big(w_f \cdot tf_f \cdot \log_2(1 + c_f \cdot \frac{avg\_l_f}{l_f})\big), (c_f > 0) \quad (4)$$

where c_f is the hyper-parameter of field *f*. tfn_f is the normalised term frequency on field f. l_f is the length of field f in the document, and avg_l_f is the average field length in the collection. w_f is the weight of field f. In the above Normalisation 2F, the term frequency is normalised in each field, and each field f has a hyper-parameter c_f with a field weight w_f. The above normalisation 2F is based on the assumption that term density is decreasing with document length [1].

In addition, the PL3 weighting model [1,8], which applies the Dirichlet priors for *tf* normalisation, can be extended in a similar way to deal with field-based retrieval. The resulting field-based PL3 weighting model has the same weighting function as PL2F in Equation (3). The normalised term frequency *tfn* is given by *Normalisation 3F* as follows:

$$tfn = \sum_f w_f \cdot tfn_f = \sum_f \big(w_f \cdot \frac{tf_f + \mu_f \cdot \frac{tf_{cf}}{l_{cf}}}{l_f + \mu_f} \cdot \mu_f\big) \quad (5)$$

where w_f and tf_f are the weight and term frequency of field f in the document, respectively. tfn_f is the normalised term frequency on field f. μ_f is the hyper-parameter of field f. l_{cf} is the number of tokens in field f in the whole collection. l_f is the field length in the document. tf_{cf} is the frequency of the query term in field f in the whole collection.

As shown by previous experiments, compared with applying *tf* normalisation on a single index, per-field normalisation is particularly effective for Web search, such as named-page finding [12,18]. However, per-field normalisation has an associated hyper-parameter for each document field. The setting of these hyper-parameters is a crucial issue, which has an important impact of the retrieval performance. In this paper, this issue is studied by following the idea of measuring the linear relationship between field length and term frequency [6,1]. Based on this idea, we further understand the purpose of per-field normalisation, as described in the next section.

3 The Purpose of Per-field Normalisation

In the context of *tf* normalisation on a single index, Harter [6] and Amati [1] suggested that document length and term frequency have a linear relationship. Such a linear relationship can be indicated by the linear correlation between these two variables. Following their idea, in the context of per-field normalisation, we suggest that field length and term frequency also have a linear relationship, which can be indicated by the linear correlation between them. In this paper, following the definition of correlation in [4], the linear correlation between field length and normalised term frequency is given by:

$$\rho(tfn_f, l_f) = \frac{COV(tfn_f, l_f)}{\sigma(tfn_f)\sigma(l_f)} \tag{6}$$

where tfn_f is the normalised term frequency on field f, and l_f is the field length. COV stands for covariance and σ stands for the standard deviation. Note that the use of a *tf* normalisation method changes term frequency. Therefore, the normalised term frequency, instead of term frequency, is considered in our study.

We suggest that the aim of *tf* normalisation on a document field is to adjust the linear relationship between field length and term frequency. Applying different hyper-parameter settings results in different correlation values, which indicate different degree of linear dependence between field length and term frequency. In our study, we investigate how per-field normalisation affects the correlation $\rho(tfn_f, l_f)$. In particular, from our experiments, we expect to find a pattern that may help in proposing an automatic method in setting the hyper-parameter values, without using relevance assessment for tuning.

4 Experimental Setting and Methodology

Our experiments are conducted using Terrier [11]. Two TREC Web test collections, namely the .GOV and the .GOV2 collections, are used in our experiments. These two collections are the only currently available ones for the named-page finding task. The .GOV collection is a 1.25 million pages crawl of the .gov domain. The .GOV2 collection is a later crawl of the .gov domain, which contains 25,205,179 Web documents and 426 Gigabytes of uncompressed data[1]. This collection has been employed in the TREC Terabyte track since 2004. In addition, a named-page finding task has been run in the Terabyte track since 2005. .GOV2 is currently the largest TREC test collection. The indices of these two collections are created with the body, anchor text and title fields, respectively. Porter's stemmer and standard stopword removal are applied.

The test queries used are the 525 topics used in the TREC 2002-2004 Web track named-page finding tasks [17] on the .GOV collection, and the 252 topics used in the TREC 2005 Terabyte track named-page finding task [3] on the .GOV2 collection. The evaluation measure used is mean reciprocal rank (MRR), which is the official measure in TREC for the named-page finding task [17].

In our experiments, we investigate the linear relationship between field length and normalised term frequency. This linear relationship is indicated by the linear correlation between these two variables. Three field-based weighting models in the literature, namely PL2F, BM25F and PL3F, are used in this study.

The first step of the experiments is to optimise the three weighting models used, which provides a basis for our study. For each of the field-based weighting models, we need to optimise six parameters, namely the hyper-parameters

[1] Information of the collections can found at http://ir.dcs.gla.ac.uk/test_collections/

Table 1. The optimal hyper-parameter settings and weights of the body, anchor text, and title fields, using three field-based weighting models, on the two collections used

Coll.	Body	Anchor	Title	Body	Anchor	Title	Body	Anchor	Title
	PL2F (c_f)			BM25F (b_f)			PL3F (μ_f)		
.GOV	0.8	15.0	5.5	0.85	0.10	0.45	300	50000	10
.GOV2	1.2	2.6	2.6	0.85	0.90	0.50	300	40	20
	PL2F (w_f)			BM25F (w_f)			PL3F (w_f)		
.GOV	1	1.1	4.6	1	8.1	12.4	1	0.4	18.2
.GOV2	1	3.4	2.6	1	6.0	8.6	1	1.1	12.0

and the weights of the three indexed document fields. Our optimisation process follows the one for BM25F applied in [18]. However, we apply manual data sweeping, instead of automatic optimisation, as applied in [18]. This is because in our previous experiments, we found that the manual data sweeping with a small enough granularity can usually lead to a better optimised retrieval performance than automatic optimisation. Following [18], we set the field weight of body to 1 to reduce the cost of optimisation. For the remaining five parameters, the optimisation process is described as follows:

1. On each field, we optimise the hyper-parameter of the field, while disabling the other two fields. The optimised hyper-parameter setting are obtained by multiple-step data sweeping with from-large-to-small granularities. Data sweeping is performed within a reasonable range of values. This range is [1, 32] for PL2F, (0, 1] for BM25F and (0, 100000] for PL3F. The minimal granularity is 0.1 for PL2F, 0.05 for BM25F and 10 for PL3F.
2. We optimise the field weights of body and title by a two-step two-dimensional data sweeping within [0, 20], while setting the hyper-parameter values to the ones optimised in the first step. The granularities in the two data sweeping steps are 1 and 0.1, respectively.

We only briefly describe the optimisation process, for lack space. The obtained optimised parameter values are provided in Table 1.

The second step of the experiments is to investigate the linear relationship between field length and normalised term frequency. This linear relationship is indicated by $\rho(tfn_f, l_f)$, the linear correlation between these two variables. In particular, we study how the optimised hyper-parameter values are related to $\rho(tfn_f, l_f)$. For the three different document fields used, we plot the hyper-parameter values against $\rho(tfn_f, l_f)$, in order to study how the linear relationship between tfn_f and l_f varies on different document fields. We also look at the proportion of the optimal $\rho(tfn_f, l_f)$ value to the maximum $\rho(tfn_f, l_f)$ value with respect to all possible hyper-parameter values. By doing this, we expect to find a pattern that may indicate the optimal hyper-parameter setting, which can lead to a practical approach for setting the hyper-parameter values. The analysis of the related experimental results are provided in the next section.

5 The Linear Relationship Between Field Length and Normalised Term Frequency

Table 1 contains the optimised hyper-parameter values and field weights after the data sweeping process. From Table 1, we observe that, across the two collections used, on one hand, for the body and title fields, the optimised hyper-parameter settings are relatively similar. On the other hand, for the anchor text field, the hyper-parameter settings are largely different across the two collections used (e.g. 40 vs. 50000 using PL3F). A possible explanation is that the weighting models used assume a random distribution of a query term in the collection. This assumption is usually true for written text, such as body and title. Differently from these two fields, the anchor text of a Web page is extracted from its incoming links. Eiron & McCurley concluded that the anchor text of a Web page usually has only one or two repeatedly occurring unique terms [5]. Consequently, in the anchor text field, the curve of a query term's distribution looks like a Beta(0.5, 2) distribution[2] [4], because a query term usually has a large number of occurrences in the anchor text of some Web pages, and does not appear at all in the anchor text of other Web pages. Therefore, the optimised hyper-parameter setting for the anchor text field is unpredictable and can be largely different across different collections.

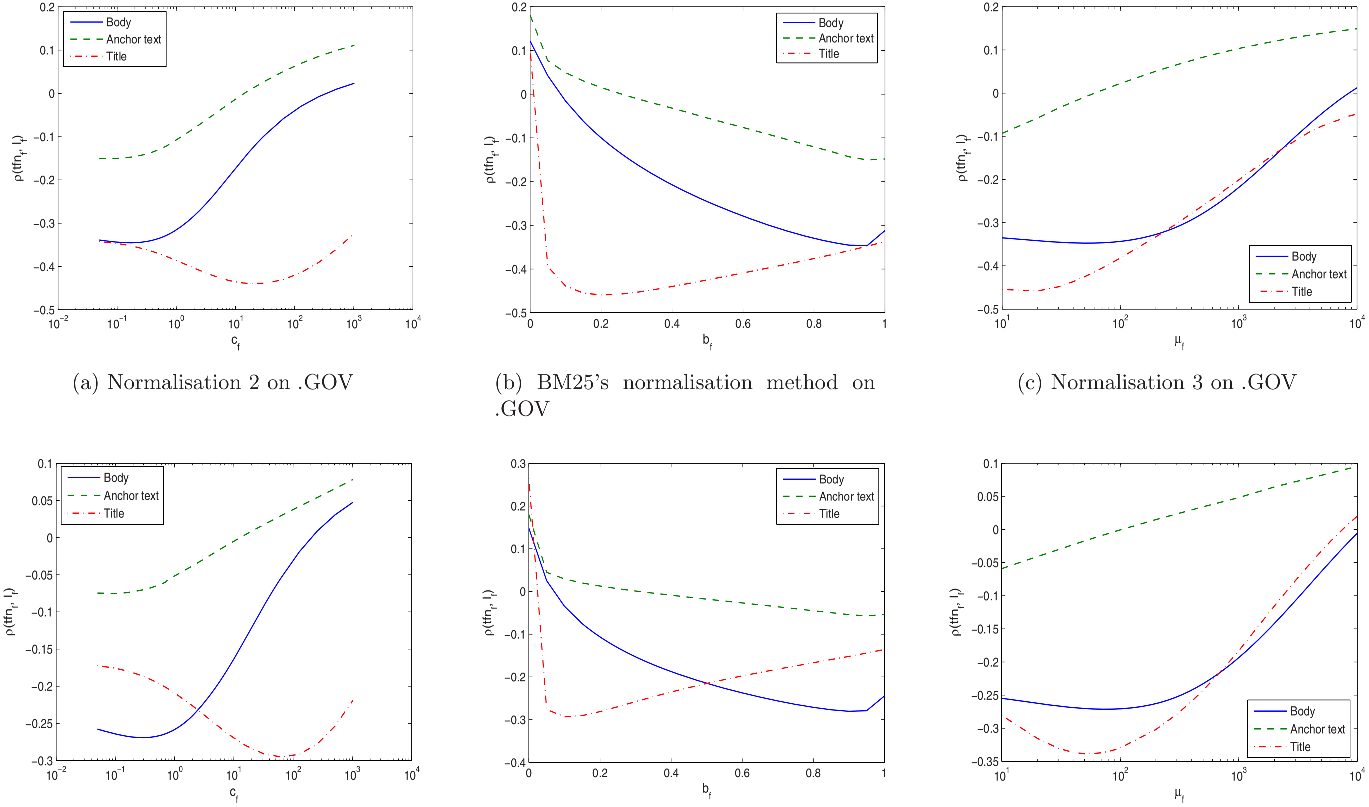
Next, we study the linear relationship between field length and normalised term frequency. Figure 1 (see page 475) plots the linear correlation between these two variables. From Figure 1, we find that the linear correlation $\rho(tfn_l, l_f)$ varies with the use of different hyper-parameter values. In particular, in all the cases, the plotted curve has a lowest point, which corresponds to the maximum negative $\rho(tfn_l, l_f)$ value. To further analysis the linear relationship between field length and normalised term frequency, Table 2 provides the resulting $\rho(tfn_f, l_f)$ values of the optimised hyper-parameter settings. The values in parenthesis are the $ratio_f$ that is given by:

$$ratio_f = \frac{\rho_{opt}(tfn_f, l_f)}{\rho_{max}(tfn_f, l_f)} \tag{7}$$

where $\rho_{opt}(tfn_f, l_f)$ is the $\rho(tfn_f, l_f)$ value given by the optimised hyper-parameter setting. $\rho_{max}(tfn_f, l_f)$ is the maximum negative $\rho(tfn_f, l_f)$ value that corresponds to the lowest points in the curves in Figure 1.

From Table 2, for the body and title fields, we find that the $\rho_{opt}(tfn_f, l_f)$ value seems to be proportional to the maximum negative $\rho(tfn_f, l_f)$ value. The $ratio_f$ for the body and title fields are similar to each other. For the anchor text field, we do not have the same observation, probably because of the repeatedly occurring tokens of the query terms in this field. Based on the observation from Table 2, in the next section, we devise an automatic method for estimating the hyper-parameter values of the body and title fields. For the anchor text field, we simply apply the optimised hyper-parameter setting after data sweeping.

[2] Beta(0.5, 2) distribution refers to a Beta distribution with $\alpha = 0.5$ and $\beta = 2$.



(a) Normalisation 2 on .GOV

(b) BM25's normalisation method on .GOV

(c) Normalisation 3 on .GOV

(d) Normalisation 2 on .GOV2

(e) BM25's normalisation method on .GOV

(f) Normalisation 3 on .GOV2

Fig. 1. The linear correlation $\rho(tfn_f, l_f)$ against the *tf* normalisation hyper-parameters on the two collections used

Table 2. The optimal $\rho(tfn_f, l_f)$ values and the corresponding $ratio_f$ for the three document fields. Mean ratio refers to the mean of the $ratio_f$ values over the three weighting models used. Side specifies on which side of the curves (in Figure 1) the optimal $\rho(tfn_f, l_f)$ value locates.

	ρ_{body} ($ratio_{body}$)	ρ_{anchor} ($ratio_{anchor}$)	ρ_{title} ($ratio_{title}$)
Coll.		PL2F	
.GOV	-0.3232 (0.9364)	0.002456 (-0.01629)	-0.4265 (0.9699)
.GOV2	-0.2543 (0.9447)	-0.03282 (0.4361)	-0.2332 (0.9688)
Side	Increasing	Increasing	Decreasing
Coll.		BM25F	
.GOV	-0.3389 (0.9763)	0.04932 (-0.3263)	-0.4325 (0.9414)
.GOV2	-0.2697 (0.9947)	-0.07142 (0.9393)	-0.2489 (0.7385)
Side	Decreasing	Decreasing	Increasing
Coll.		PL3F	
.GOV	-0.3097 (0.8918)	0.1658 (-1.0967)	-0.4546 (0.9886)
.GOV2	-0.2527 (0.9313)	-0.02301 (0.3898)	-0.3157(0.9312)
Side	Increasing	Increasing	Decreasing
Coll.		Mean ratio	
.GOV	0.9348	-0.4798	0.9666
.GOV2	0.9568	0.5884	0.8795

6 Method *prop* for Setting the Per-field Normalisation Hyper-parameter Values

In the previous section, across the two collections used, we found that on both the body and title fields, the optimal $\rho(tfn_f, l_f)$ value is proportional to the maximum negative $\rho(tfn_f, l_f)$ value. Therefore, on these two fields, we can estimate the hyper-parameter value, which gives a $\rho(tfn_f, l_f)$ value that is proportional to the $ratio_f$ value (see Table 2). Using the above suggested solution, we make the following hypothesis:

> Hypothesis H(*prop*): For the given body or title field, across different collections, the optimal hyper-parameter values provide a constant ratio of the optimal $\rho(tfn_f, l_f)$ value divided by the maximum negative $\rho(tfn_f, l_f)$ value.

The above Hypothesis H(*prop*) implies that, for a given body or title field, the optimal $\rho(tfn_f, l_f)$ value is proportional to the maximum negative $\rho(tfn_f, l_f)$ value. Using Hypothesis H(*prop*), for a given collection, we can estimate the hyper-parameter value that satisfies $\rho(tfn_f, l_f) = \rho_{max}(tfn_f, l_f) \cdot ratio_f$, where $\rho_{max}(tfn_f, l_f)$ is the maximum negative $\rho(tfn_f, l_f)$ value. $ratio_f$ is given by Equation (7). On the two collections used, the $ratio_f$ values of body and title are listed in Table 2. We denote the above described approach by method *prop*.

To apply method *prop*, we need to create a bidirectional mapping between a hyper-parameter and $\rho(tfn_f, l_f)$. Each $\rho(tfn_f, l_f)$ value should correspond to a unique hyper-parameter value, and vice versa. In fact, from Figure 1, we can see that a $\rho(tfn_f, l_f)$ value corresponds to two different hyper-parameter values: One is on the increasing side of the curve, and the other is on the decreasing side of the curve. Therefore, by looking at the curves in Figure 1, we identify at which side of the curve the optimal $\rho(tfn_f, l_f)$ value is located (see Table 2).

For the application of method *prop*, we need a training collection to obtain the assumed constant $ratio_f$. The training process for computing this constant $ratio_f$ needs to be done only once. For a given new collection, we do not need any associated relevance judgement. Finally, we apply the hyper-parameter setting that results in the $ratio_f$ value on the given query set, instead of optimising the hyper-parameter by maximising the retrieval performance using relevance judgement. For a given collection, the tuning process takes place before the retrieval process. There is no additional overhead during retrieval. We evaluate method *prop* in the next section.

7 Evaluation of Method *prop*

For evaluating method *prop*, we conduct a two-fold holdout evaluation on the two collections, namely .GOV and .GOV2. In each fold of the holdout evaluation, we use one collection for training, in order to compute the assumed constant $ratio_f$ value. The other collection is used for testing. The assumed constant $ratio_f$ value is the mean of the $ratio_f$ values of the three weighting models used, obtained on the training collection (see the mean $ratio_f$ values in Table 2). In addition to the test queries used in Section 5, we also experiment with the 181 latest TREC topics used in the TREC 2006 named-page finding task. Note that method *prop* is only applied for the body and title fields. For the anchor text field, we apply the optimised hyper-parameter setting obtained by data sweeping.

We compare the retrieval performance obtained using the hyper-parameter setting, estimated by method *prop*, with the optimised retrieval performance using data sweeping. We suggest that a large (>5%) and statistically significant difference between the obtained MRRs indicates a failure of method *prop* in estimating the hyper-parameter setting. Otherwise, we conclude that method *prop* is effective for the named-page finding retrieval task, when different document fields are used. The statistical test used is the sign test[3].

The evaluation results are listed in Table 3. In two cases, we observe that the estimated hyper-parameters result in MRRs that are higher than the optimised ones by data sweeping (see the MRR values in *italic* in Table 3). We suggest that this is because the optimisation procedure optimises the hyper-parameter of each document field separately. However, the optimised hyper-parameter setting of each individual field may not necessarily lead to the optimised retrieval performance, when different fields are summed up together. From Table 3, we observe that in all the nine cases, method *prop* provides a retrieval performance that is as good as the one optimised by data sweeping. We find no large (5%) difference

[3] For MRR, the sign test is more appropriate than the Wilcoxon test.

Table 3. Evaluation results for method *prop*. Columns body and title provides the hyper-parameter settings for the body and title fields, estimated by method *prop*. MRR(opt) and MRR(prop) are the MRRs obtained by data sweeping and by method *prop*, respectively. *diff.* is the difference between the two MRR values in percentage. p-value is given by the sign test.

Topics	body	title	MRR(opt)	MRR(prop)	*diff* (%)	p-value
	PL2F					
TREC 2002-2004 on .GOV	0.62	1.03	0.7294	0.7018	-3.78	3.03e-05
TREC 2005 on .GOV2	1.34	21.65	*0.4341*	*0.4522*	+4.17	0.0422
TREC 2006 on .GOV2	1.06	20.54	0.4736	0.4733	≈ 0	0.630
	BM25F					
TREC 2002-2004 on .GOV	0.81	0.63	0.7142	0.7145	≈ 0	0.497
TREC 2005 on .GOV2	0.69	0.17	0.4738	0.4522	-4.56	0.583
TREC 2006 on .GOV2	0.74	0.19	0.4405	0.4392	≈ 0	0.798
	PL3F					
TREC 2002-2004 on .GOV	170	2.40	0.6390	0.6140	-3.91	0.0115
TREC 2005 on .GOV2	291	27.23	*0.3721*	*0.3751*	≈ 0	0.00259
TREC 2006 on .GOV2	234	23.59	0.4470	0.4259	-4.72	1.03e-07

between MRR(opt) and MRR(prop). Therefore, we conclude that method *prop*, based on Hypothesis H(prop), is effective on the two collections used.

To summarise, in a practical setting, the assumed constant $ratio_f$ value is obtained on a training collection for once. For a given new collection, we recommend applying method *prop* for the body and title fields without the use of relevance assessment. For the anchor text field, we recommend applying an empirical hyper-parameter setting.

8 Conclusions and Future Work

In this paper, we have provided a better understanding of per-field normalisation, based on Harter and Amati's idea that there is a linear relationship between term frequency and document length. We argue that the purpose of per-field normalisation is to adjust the linear relationship between term frequency and field length. Based on this argument, we have conducted a study of setting the per-field normalisation hyper-parameters, based on experimentation on two TREC Web collections for named-page finding. From the experiments, we have the following important finding: For the body and title fields, using three different field-based weighting models, the optimal $\rho(tfn_f, l_f)$ value, given by the optimised hyper-parameter value, is proportional to the maximum negative $\rho(tfn_f, l_f)$ value across the two collections used. Another important finding is that the optimised hyper-parameter setting for the anchor text field are largely different across the two collections used. We suggest that this is because of the repeatedly occurring tokens of the query terms in the anchor text field. Based on the above findings, we proposed an automatic setting method for setting the per-field normalisation hyper-parameters, called method *prop*, for the body and title fields. The proposed method does not require relevance assessment for tuning. According to the evaluation results, method *prop* was shown to be effective on the two test collections used, with 958 associated test queries. For the application of method *prop* in practise, we recommend applying method *prop* for the document fields of written text, such as the body and title fields. For the anchor text field, we recommend applying an empirical hyper-parameter setting, obtained by training using relevance assessment.

The reported experiments in this paper were conducted for the named-page finding retrieval task, on two different TREC collections, including the large-scale .GOV2 collection. We have also conducted experiments for ad-hoc retrieval on various TREC test collections, from which we had similar observations with those in this paper. For lack of space, we only focus on the named-page finding retrieval task in this paper. In the future, We will further study if method *prop* is general enough to cope with other Web search tasks, in the context of field-based retrieval. Moreover, because of the repeatedly occurring terms in anchor text, it is difficult to estimate the hyper-parameter setting for this field. A possible solution is to apply an absolute discount on the term frequency in this field, before per-field normalisation is applied. We will also investigate this issue in future work.

References

1. G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow, 2003.
2. A. Chowdhury, M. C. McCabe, D. Grossman, and O. Frieder. Document normalization revisited. In *Proceedings of ACM SIGIR 2002*.
3. C. Clarke, F. Scholer, and I. Soboroff. Overview of the TREC-2005 Terabyte Track. In *Proceedings of TREC 2005*.
4. M. DeGroot. *Probability and Statistics*. Addison Wesley, 2nd edition, 1989.
5. N. Eiron and K. McCurley. Analysis of anchor text for web search. In *Proceedings ACM SIGIR 2003*, 2003. URL: http://mccurley.org/papers/anchor.pdf.
6. S. Harter. *A probabilistic approach to automatic keyword indexing*. PhD thesis, The University of Chicago, 1974.
7. B. He and I. Ounis. Term frequency normalisation tuning for BM25 and DFR model. In *Proceedings of ECIR 2005*, 2005.
8. B. He and I. Ounis. A study of the Dirichlet Priors for term frequency normalisation. In *Proceedings of ACM SIGIR 2005*, 2005.
9. B. Jansen and A. Spink. How are we searching the World Wide Web? a comparison of nine search engine transaction logs. *Information Processing & Management*, 42(1), 2006.
10. C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC 2005: experiments in Terabyte and Enterprise tracks with Terrier. In *Proceedings of TREC 2005*.
11. 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 ACM SIGIR OSIR Workshop 2006*.

12. V. Plachouras. *Selective Web Information Retrieval.* PhD thesis, University of Glasgow, 2006.
13. S.E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive. In *Proceedings of TREC 7*, 1998.
14. S.E. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings ACM CIKM 2004*.
15. G. Salton. *The SMART Retrieval System*. Prentice Hall, New Jersey, 1971.
16. A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of ACM SIGIR 1996.*
17. E. Voorhees. *TREC: Experiment and Evaluation in Information Retrieval.* The MIT Press, 2005.
18. H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S.E. Robertson. Microsoft Cambridge at TREC 13: Web and Hard Tracks. In *Proceedings of TREC 2004*.