A Log-Logistic Model-Based Interpretation of TF Normalization of BM25

Yuanhua Lv and ChengXiang Zhai

University of Illinois at Urbana-Champaign,
201 N Goodwin Ave, Urbana, IL 61801, USA
ylv2@uiuc.edu, czhai@cs.uiuc.edu

Abstract. The effectiveness of BM25 retrieval function is mainly due to its sub-linear term frequency (TF) normalization component, which is controlled by a parameter k_1. Although BM25 was derived based on the classic probabilistic retrieval model, it has been so far unclear how to interpret its parameter k_1 probabilistically, making it hard to optimize the setting of this parameter. In this paper, we provide a novel probabilistic interpretation of the BM25 TF normalization and its parameter k_1 based on a log-logistic model for the probability of seeing a document in the collection with a given level of TF. The proposed interpretation allows us to derive different approaches to estimation of parameter k_1 based solely on the current collection without requiring any training data, thus effectively eliminating one free parameter from BM25. Our experiment results show that the proposed approaches can accurately predict the optimal k_1 without requiring training data and achieve better or comparable retrieval performance to a well-tuned BM25 where k_1 is optimized based on training data.

Keywords: BM25, term frequency, log-logistic model, percentile term frequency normalization, automatic parameter tuning.

1 Introduction

The Okapi BM25 retrieval function [17,19] has been the state-of-the-art for nearly two decades, and its performance remains quite competitive for a wide range of tasks. The BM25 formula, as presented in [6], scores a document D with respect to query Q as follows:

$$\sum_{w \in Q \cap D} \frac{(k_3 + 1) \cdot c(w, Q)}{k_3 + c(w, Q)} \cdot dtf(w, D) \cdot \log \frac{N + 1}{df(w) + 0.5} \tag{1}$$

The meanings of the involved symbols are explained in Table 1.

A key component of BM25 contributing to its success is its sub-linear term frequency (TF) normalization formula:

$$dtf(w, D) = \frac{(k_1 + 1) \cdot c(w, D)}{k_1 \left(1 - b + b\frac{|D|}{avdl}\right) + c(w, D)} = \frac{(k_1 + 1) \cdot c'(w, D)}{k_1 + c'(w, D)} \tag{2}$$

R. Baeza-Yates et al. (Eds.): ECIR 2012, LNCS 7224, pp. 244–255, 2012.

Table 1. Notation

Notation	Description
$c(w, D)$	Raw term frequency of w in doc D
$c(w, Q)$	Raw term frequency of w in query Q
N	Number of docs in the collection
$df(w)$	Number of docs containing w, i.e., $\|C_w\|$
$\|D\|$	Length of doc D
$avdl$	Average doc length

where

$$c'(w, D) = \frac{c(w, D)}{1 - b + b\frac{|D|}{avdl}} \tag{3}$$

which is controlled by two parameters b and k_1. Parameter b has been interpreted as the "slope" in pivoted document length normalization [21], but it has so far been rather unclear how to interpret parameter k_1. Indeed, although the whole BM25 formula was derived using the classic probabilistic retrieval model, its TF normalization parameter appears to have no obvious corresponding probabilistic interpretation. The divergence from randomness (DFR) framework has been explored to explain the TF formula of BM25 [1], but there is no probabilistic interpretation for parameter k_1. In fact, it has been widely accepted that "the model tells us nothing about what kind of value to expect for k_1", as was pointed out by Robertson and Walker when proposing BM25 [17]. As a result, it is often hard to optimize k_1; it is usually set to a default value of 1.2 or manually tuned based on training data [17], which requires manual effort due to its parameter interaction with b, yet still without guarantee of optimality on future test data due to the possibility of overfitting.

In this paper, we provide a novel probabilistic interpretation of the TF normalization component of BM25 as the probability of seeing a document in the collection with a given level of TF. Our work is motivated by the observation that the parameter k_1 is related to the relative differences in score contributions from matching a term with different TF values. It is known that in general the score contribution should increase with TF, but the increasing rate should be decreasing as the already observed occurrences become larger, i.e., sub-linear growth, which is ensured through the BM25 formula. However, how exactly this score should increase with TF has not been well studied. We hypothesize that the optimal score for a document from matching a term with a TF value should depend on the counts of documents matching the term with more/less TF values. Specifically, the score from matching t occurrences is related to how many documents in the collection have at least t occurrences of the term. Intuitively, if many documents contain at least t occurrences of the term, it would not be surprising to see a document with t occurrences of the term, thus we should assign low scores to documents matching t occurrences of the term in such a case. As t becomes larger, there would be naturally fewer documents with at least t occurrences of the term, and thus the score would naturally become larger. Moreover, the numbers of documents containing at least t and $t + 1$ occurrences of the term would be very close to each other when t is very large, so the score would become relatively stable, i.e., the score increasing rate would decrease.

Following this intuition, we develop a *percentile TF normalization scheme* which connects TF normalization with the percentile rank of a TF value based on the counts of documents containing different levels of TF for a query term. In this scheme, we assume that, given a term, the distribution of its TF values over the collection can be described by the log-logistic model. Then we can estimate the score from matching t occurrences of the query term as the ratio of the number of documents matching the term with a TF value of at most t to the total number of documents containing this term, which is essentially the corresponding cumulative probability of the model. Interestingly, the proposed approach can cover the TF normalization formula of BM25 as its special case. As a result, it not only gives a *meaningful interpretation* to k_1 (as the scale parameter of the log-logistic model) but also makes it possible to automatically and adaptively estimate k_1 via model estimation based on the statistics of term occurrences. Note that the proposed approach requires no training data.

Our experiment results show that the log-logistic model-based interpretation can lead to accurate estimation of the optimal k_1 value without requiring any training data. It not only achieves better or comparable retrieval performance to an empirically tuned BM25 using training data, but also works more robustly. As a "by-product", we can effectively eliminate the k_1 parameter from BM25. Thus with our approach, we only need to tune a single parameter b, which needs less manul effort than tuning both b and k_1 simultaneously.

2 Related Work

The Okapi BM25 retrieval function [17,19] has been the state-of-the-art for nearly two decades. Quite a few studies have attempted to improve BM25 from various perspectives. Robertson et al. [18] proposed an important extension of BM25, called BM25F, for combining attributes across multiple fields. He and Ounis [8] evaluated different strategies for document length normalization under the framework of BM25. Some studies extended standard BM25 function to incorporate additional retrieval heuristics, e.g., term proximity [24,23]. Machine learning was also explored to tune and improve the BM25 retrieval function in a supervised way [25,22]. Recently, our work [15,14] revealed a previously unknown deficiency of BM25 (namely that the component of term frequency normalization by document length is not properly lower-bounded), and developed two efficient methods to fix this deficiency, resulting in two more effective retrieval formulas, BM25L and BM25+, respectively. In another recent work of ours [13], we proposed a heuristic approach for adaptive term frequency normalization in BM25. However, no previous work aims to interpret the TF normalization parameter k_1 of BM25 from the probabilistic perspective.

The within document term frequency (TF), which dated back to Luhn's pioneer work on automatic indexing [12], has been playing a critical role in modern information retrieval models [19,21,16,28,20,1,6,15,14,13]. It is widely recognized that linear scaling in term frequency puts too much weight on repeated occurrences of a term. So the term frequency should be normalized so as to dampen the contribution from repeated occurrences, which can often be achieved by using logarithm functions [21], adopting some special sub-linear transformations [19], or assuming some probabilistic models [16,28,1,27,5]. However, no TF normalization method considers the special characteristics of occurrence patterns for each individual term in probabilistic approaches.

Our use of the log-logistic model for modeling term distribution is not entirely novel, as some other studies (e.g., [4,5]) have used this distribution to model word burstiness [3] in information retrieval. However, the model parameters in their work were set heuristically and not really adapted to the special characteristics of the TF distribution of each term. Moreover, our study aims at a different goal, i.e., interpreting the sub-linear TF normalization formula of BM25.

In the general area of optimizing term weights, our work is also related to some recent work on learning optimal query term weights (see, e.g., [11] and [2]), but our work differs in that we focus on the optimization of the document-side weighting of a term, which is complementary with the query-side term weighting.

3 A Divergence-from-Randomness View of BM25

The TF component in BM25 was originally motivated based on the 2-Poisson model [17], but it did not lead to a rigorous probabilistic interpretation of this component. In this paper, we follow previous work [1] and attempt to interpret the TF component from the viewpoint of DFR [1]. As we will show later, such a view can lead to a novel probabilistic interpretation of BM25 TF as a normalized probability of seeing a document with a given level of TF.

We first borrow from [7,17,1] the notion of an elite set of a term w, i.e., C_w, which is defined as the set of documents containing w. With a DFR view, we can see that the component $idf(w)$ of BM25 is essentially the information content [9] of randomly picking a document that will fall in C_w, while the $dtf(w, D)$ component captures the relative weights of documents within C_w with respect to w. In other words, $idf(w)$ balances the weights between different elite sets, while $dtf(w, D)$ adjusts the relative weights of documents within the same elite set. In this sense, the functionalities of $idf(w)$ and $dtf(w, D)$ are similar to the two components $(-log_2Prob_1)$ and $(1-Prob_2)$ respectively in the DFR approach.

Note that the raw term frequency $c(w, D)$ depends on the document length. It is well-known that longer documents tend to have higher term frequency. Following BM25, we also choose the pivoted normalization method [21], as shown in Equation 3, to obtain the normalized version of the term frequency $c'(w, D)$.

Presumably, if the ranking of a document with $c'(w, D)$ occurrences of term w is relatively high in the elite set C_w in terms of the normalized TF, the contribution of the $c'(w, D)$ occurrences of this term is also high. Hence, given the term distribution of w over all documents in C_w, based on our analysis, a natural way to implement $dtf(w, D)$ would be to base it on the *percentage* of documents in C_w that no higher term frequency than D. Intuitively this implementation of $dtf(w, D)$ makes sense: if $c'(w, D)$ of D is larger than that of most of the documents in C_w, the relative weights of D, i.e., $dtf(w, D)$, would be very high no matter what is the absolute value of $c'(w, D)$, while if $c'(w, D)$ is small relative to other documents in C_w, $dtf(w, D)$ would be similarly small no matter how large the absolute value of $c'(w, D)$ is. What is more interesting is that this percentile TF normalization strategy is *term-specific*: the contribution of a TF value depends on the TF distribution of a term over the elite set.

Formally, we define the percentile-based $dtf(w, D)$ to be proportional to the probability that a randomly picked document from C_w will contain no more than $c'(w, D)$ occurrences of w:

$$dtf(w, D) \propto p(X \leq c'(w, D)) \tag{4}$$

where the random variable X takes on values $\{c'(w, D')|D' \in C_w\}$.

4 Log-Logistic Model-Based Interpretation of TF Normalization of BM25

A straightforward and non-parametric way to estimate the probability in Formula 4 is:

$$p(X \leq t) = \frac{|\{D'|D' \in C_w, c'(w, D') \leq t\}|}{|C_w|} \tag{5}$$

which, however, suffers from some potential problems if we use it directly as $dtf(w, D)$: (1) It is sometimes not very accurate because the absolute TF values are largely ignored. For example, given a query term w and two documents D_1 and D_2, if the ranking positions of D_1 and D_2 according to their TF values of w in a descending order are 10 and 20 respectively, the relative weights of D_1 and D_2 will be *fixed*, no matter $c'(w, D_1) - c'(w, D_2) = 0.1$ or $c'(w, D_1) - c'(w, D_2) = 10$. (2) It may not capture the *global* tendency of the entire TF distribution and is also hard to incorporate prior knowledge.

To address these problems, we propose to use a parametric model $F(x)$ to model the TF distribution over the elite set C_w. The probability $p(X \leq t)$ in this case is just the cumulative probability of the model. In our work, we use the log-logistic distribution (LL), mainly due to its burstiness property [4,5]:

$$p(X \leq t|k_1, \beta) = F(t|k_1, \beta) = \frac{t^\beta}{{k_1}^\beta + t^\beta} \tag{6}$$

where $t > 0$, $k_1 > 0$, and $\beta > 0$.

Interestingly, it can be easily proved that the sub-linear TF normalization function of BM25 is a special case of this log-logistic cumulative distribution function, as presented in Theorem 1.

Theorem 1. *The sub-linear TF normalization function of BM25 is equivalent to the cumulative distribution function of the log-logistic model $F(t|k_1, \beta = 1)$ normalized by the cumulative probability of this model at $t = 1$:*

$$\frac{(k_1 + 1)t}{k_1 + t} = \frac{F(t|k_1, \beta = 1)}{F(1|k_1, \beta = 1)} \tag{7}$$

Proof.

$$\frac{F(t|k_1,\beta=1)}{F(1|k_1,\beta=1)} = \frac{\frac{t}{k_1+t}}{\frac{1}{k_1+1}} = \frac{(k_1+1)t}{k_1+t} \tag{8}$$

Normalizing the cumulative distribution function by the cumulative probability at $t = 1$ can also be justified. As we have discussed, $p(X \leq t|k_1, \beta = 1) = F(t|k_1, \beta)$ only represents the relative weights of documents within the elite set C_w. Thus, in order to balance and aggregate the relevance scores with respect to different query terms, we need to adjust $p(X \leq t|k_1, \beta = 1)$ to the corresponding absolute values. Following the *dtf* formula in the standard BM25 retrieval function (Formula 2), $t = 1$ corresponds to the normalized TF for one occurrence in documents with average document length. This is why $F(1|k_1, \beta)$ is chosen as the normalization factor.

We can see that, our derivation provides a probabilistic interpretation of the well-performing TF normalization formula in BM25. The k_1 is shown to have a clear meaning, the *scale parameter* of the LL model, and thus it would be able to be estimated automatically through statistical model estimation.

5 Automatic Setting of k_1 Based on Log-Logistic Model

When $\beta = 1$, we call the LL model "one-parameter log-logistic distribution (LL1)". We use the log-moment estimation method [26] to fit this LL1 model to the TF distribution of w to estimate the parameter $k_1(w)$, which generally varies according to w. Although we have not found a closed-form solution, $k_1(w)$ can be calculated easily by solving the following minimization problem:

$$\hat{k_1}(w) = \arg\min_{k_1} \left(g(k_1) - \frac{1}{|C_w|} \sum_{D \in C_w} \log\left(c'(w,D)+1\right) \right)^2 \tag{9}$$

where

$$g(k_1) = \begin{cases} \frac{k_1}{k_1-1}\log(k_1) & \text{if } k_1 \neq 1 \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

Due to space limit, we cannot show all the derivation details.

Generally, we can solve this problem by applying a Newton-Raphson method, or we can even enumerate some k_1 values to find the approximately optimal one, since the optimal k_1 is usually within a range of $(0, 2]$ in our experiments. The above optimization problem can be solved efficiently, since it only relies on the distribution of term frequency which can be accessed very quickly thanks to the inverted index. Moreover, $\hat{k_1}$ can also be pre-computed easily offline.

After obtaining $\hat{k_1}(w)$, we can then construct an extended BM25 function that has essentially parameter k_1 eliminated.

$$\sum_{w \in Q \cap D} qtf(w,Q) \cdot \frac{(\hat{k_1}(w)+1) \cdot c'(w,D)}{\hat{k_1}(w) + c'(w,D)} \cdot idf(w) \tag{11}$$

Table 2. Overview of TREC collections and topics

Collection	Description	#Docs	#Queries
Robust04	Disk 4&5 (minus CR)	528,155	301-450 and 601-700
WT2G	Web collection (small)	247,491	401-450
WT10G	Web collection (large)	1,692,096	451-550
AP	Associated Press news documents 88-89	164,597	51-150

We call this BM25 with the proposed term-specific k_1 "BM25T". However, there would be some potential data sparseness issues with BM25T since it relies solely on the observed TF distribution of a particular term. For example, if a term w is rare and only occurs in a few documents, $\hat{k_1}(w)$ may be inaccurate. To address this problem, we average the $\hat{k_1}(w)$ values of all query terms in the same collection as a globally estimated k_1 for every query term to avoid large deviation caused by data sparseness. For example, we use the following $\hat{k_1}$ for every query term in the WT2G collection: $\hat{k_1} = \frac{1}{|\{w|w\in Q_{401\cdots450}\}|}\sum_{w\in Q_{401\cdots450}} \hat{k_1}(w)$. We label BM25 with such a collection-specific k_1 as "BM25C". Between BM25T and BM25C there is another run "BM25Q", which averages $\hat{k_1}(w)$ values for terms from the same query and applies this $\hat{k_1}(w)$ value to every term in this query. BM25Q takes a tradeoff between purely term-specific k_1 and purely term-independent k_1.

6 Experiments

6.1 Testing Collections and Evaluation

We used several standard TREC collections in our study for the evaluation purpose: Robust04, WT2G and WT10G. They represent different sizes and genres of heterogeneous text collections. Robust04 is a large news dataset. WT2G is a small Web collection, while WT10G is a relatively large one. Besides, we also used a homogeneous news collection AP as a cross-domain training data to evaluate the robustness of different methods. Queries were taken from the title field of the TREC topics. An overview of these collections is shown in Table 2.

We used the Lemur toolkit (4.10) [1] to implement all our algorithms. For all the datasets, the preprocessing of documents and queries included stemming with the Porter stemmer and stopwords removing using a total of 418 stopwords from the standard InQuery stoplist.

We used the standard BM25 function as our baseline, in which, both b and k_1 were well tuned on a training data set: we tuned b from 0.1 to 0.9 in increments of 0.1 and tuned k_1 from 0.2 to 3.0 in increments of 0.2. We also consider an upper bound run of BM25 (called "Oracle") where we train and test the model on the same dataset. We evaluated three variations of the proposed model-based TF normalization approach: BM25C, BM25Q, and BM25T, which automatically

[1] http://www.lemurproject.org/

Table 3. Comparison of the proposed automatic parameter tuning methods with "Oracle" which manually optimizes the parameter k_1 in BM25 on the test data

Collection		Oracle	BM25C	BM25Q	BM25T
WT2G	MAP	0.3198	0.3210	0.3198	**0.3232**
	P@10	0.4620	**0.4860**	0.4820	0.4740
WT10G	MAP	0.2099	0.2101	**0.2109**	0.2062
	P@10	0.3031	0.3062	**0.3093**	0.3021
Robust04	MAP	0.2543	**0.2543**	0.2539	0.2536
	P@10	0.4321	0.4373	0.4333	**0.4438**

tune k_1 in a collection-specific, query-specific, and term-specific way respectively. For the three proposed algorithms, we only tuned parameter b in the same way as tuning the baseline; in contrast, we tuned both b and k_1 for the baseline runs. In all experiments, k_3 was fixed to 1000 [10].

The top-ranked 1000 documents for each run were compared in terms of their mean average precisions (MAP), which also served as the objective function for parameter training. In addition, the precision at top-10 documents (P@10) was also considered in our evaluation for completeness.

6.2 Performance Comparison

In this section, we are seeking empirical evidence to answer the following two questions:

1. Can we estimate a near optimal k_1 using the automatic estimation method derived based on the log-logistic model *without* using any training data?
2. Is it beneficial to set k_1 *adaptively* to individual terms or individual queries?

We first compare our methods, i.e., BM25C, BM25Q, and BM25T, with the standard BM25. All methods are trained and tested on the same collection, and the parameter k_1 in the standard BM25 is well tuned to its optimal value manually. We can see that the BM25 run here represents the upper-bound of standard BM25, so we label it as "Oracle". Oracle uses the ground truth k_1 setting, while our methods use an automatically estimated k_1 setting. Our main goal here is to investigate if the proposed methods can automatically predict the optimal k_1 value. We report the experiment results in Table 3.

Comparing BM25C, BM25Q, and BM25T, we find that BM25C works mostly stably. This observation shows that, although we can potentially predict k_1 adaptively for individual terms (BM25T) or for individual queries (BM25Q) by exploiting term or query specific occurrence patterns, adaptive TF normalization tends to suffer from the data sparseness problem as we discussed previously in Section 5. As a result, leveraging the estimated k_1 values of multiple terms would be able to make the prediction more robust. Yet it does not necessarily mean that adaptive TF normalization is useless: we can see that BM25T works especially well on WT2G, which may suggest that adaptive TF normalization would

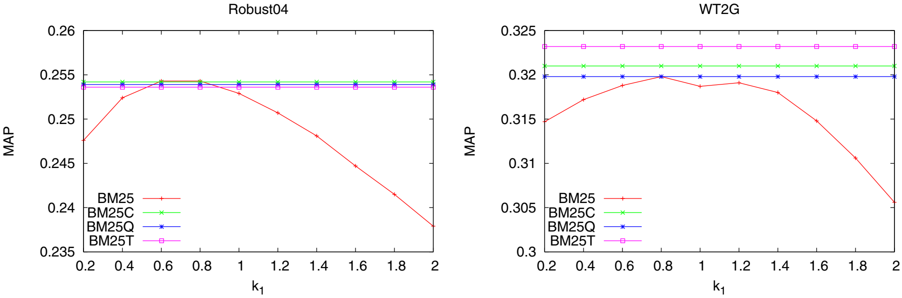

Fig. 1. Sensitivity of MAP to parameter k_1 on Robust04 (left) and WT2G (right)

Table 4. MAP comparison using cross validation. Both parameters b and k_1 are trained for BM25, while only b is trained for another three runs. Superscript + indicates the corresponding MAP improvements over BM25 are statistically significant.

Collection	BM25	BM25C	BM25Q	BM25T
WT2G	0.3091	0.3191^+	0.3198^+	$\mathbf{0.3216}^+$
WT10G	0.2059	0.2083	**0.2086**	0.2050
Robust04	0.2540	**0.2543**	0.2539	0.2536

potentially predict k_1 more accurately but its benefits can often be offset due to the data sparseness problem.

In general, BM25C works at least as effectively as Oracle. It suggests that we can automatically predict the optimal k_1 well without using any training data. Although Oracle uses the ground-truth parameter setting, BM25C still sometimes works more effectively than Oracle, which may be caused by the granularity of parameter tuning: we can only enumerate a set of discrete candidate k_1 values in the manual tuning process for Oracle, but the automatic predicted value based on Formula 9 can be more precise.

We are also interested in understanding how sensitive the parameter k_1 is and how likely a standard BM25 algorithm with manually tuned k_1 can achieve performance similar to our methods. We thus set the parameter b in each method to its corresponding optimal value, and then vary k_1 from 0.2 to 2.0 to examine the sensitivity of MAP to k_1. The sensitivity curves are shown in Figure 1. We can see that the standard BM25 is quite sensitive to k_1, and even with the optimal setting for k_1, the performance is still lower than or only comparable to our methods. This confirms again that the proposed automatic parameter tuning methods are effective.

Comparison of optimal retrieval performance alone only demonstrates the promising potential of the proposed methods. We further use a 2-fold cross-validation method to verify our observations, where the query topics are split into even and odd number topics as the two folds. The performance is then measured by combining all the test sets. The results are reported in Table 4.

Table 5. MAP comparison using a cross-domain training data AP. Both parameters b and k_1 are trained for BM25, while only b is trained for the another three runs. Superscript + indicates the corresponding MAP improvements over BM25 are statistically significant.

Collection	BM25	BM25C	BM25Q	BM25T
WT2G	0.3112	0.3210^+	0.3198^+	$\mathbf{0.3232}^+$
WT10G	0.1985	0.2083	**0.2091**	0.2055
Robust04	0.2415	$\mathbf{0.2543}^+$	0.2539^+	0.2516^+

Overall, the relative performance of BM25C, BM25Q, and BM25T is consistent with our previous observations. Moreover, BM25C works consistently better than standard BM25, suggesting that, our method, although with fewer parameters to tune, appears to perform more effectively and robustly.

In practice, we may not have appropriate training data in the right domain for all queries. So we are also interested in the robustness of a retrieval algorithm when the training and test sets are from different domains. In this experiment, we simulate such a scenario: a *homogeneous* news collection, i.e., AP, is chosen as the training data, while the three *heterogeneous* collections (i.e., WT2G, WT10G, and Robust04) are taken as the test data respectively. We compare the proposed three methods with a standard BM25 in this setting and present the results in Table 5. It shows that the proposed BM25C, BM25Q, and BM25T are clearly more effective than BM25, suggesting that our estimation method is more domain-insensitive. One possible explanation is that, the parameter k_1 in the standard BM25 often over-fits the training data; the proposed approach, however, can really adapt k_1 to different collections by exploiting their term occurrence patterns.

These results thus allow us to reach the following answers to the two questions we raised at the beginning of this section:

1. We can estimate an optimal k_1 using the automatic estimation methods derived based on the log-logistic model without using any training data.
2. Due to the data sparseness problem, setting k_1 adaptively to individual terms or individual queries is often not as robust as simply averaging the estimated k_1 values of multiple terms and setting k_1 to this average value for all query terms, although adaptive k_1 prediction would potentially help.

7 Conclusions and Future Work

In this paper, we provide a novel probabilistic interpretation of the BM25 TF normalization and its parameter k_1 based on a log-logistic model for the probability of seeing a document in the collection with a given level of TF. The proposed log-logistic TF model can cover the TF normalization formula of BM25 as its special case, and the parameter k_1 can be meaningfully interpreted as the scale parameter of the log-logistic model. In addition, the proposed interpretation allows us to derive different approaches to estimation of parameter k_1 based solely on the current collection without requiring any training data, thus effectively eliminating one free parameter from BM25. Our experiment results show that the proposed approaches can accurately predict the optimal k_1 without requiring training data and achieve better or comparable retrieval performance to a well-tuned BM25 where k_1 is optimized based on training data.

Our work also makes an important contribution in proposing a new retrieval heuristic, i.e., automatic TF normalization, which ties optimal TF normalization with interesting global statistics. Traditionally, the main global statistics exploited in a retrieval function is the IDF; our work shows that it is possible to further exploit other global statistics and develop more effective retrieval models.

There are many ways to further extend our work. Firstly, in our current study, we have only evaluated a one-parameter log-logistic model and a simple estimation method for modeling the distribution of term frequency, and it would be very interesting to investigate the interpretation of TF formula of BM25 using other appropriate models. Secondly, it would be interesting to further study how to interpret the whole TF normalization component (including document length normalization) in BM25 from a probabilistic viewpoint so that we can also optimize another important parameter in BM25, i.e., b. This is out the scope of this paper but is obviously an interesting future work.

Acknowledgments. We thank the anonymous reviewers for their useful comments. This material is based upon work supported by the National Science Foundation under Grant Number CNS 1027965, by MURI award FA9550-08-1-0265, and a Sloan Research Fellowship.

References

1. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. 20, 357–389 (2002)
2. Bendersky, M., Metzler, D., Bruce Croft, W.: Learning concept importance using a weighted dependence model. In: WSDM 2010, pp. 31–40 (2010)
3. Church, K.W., Gale, W.A.: Poisson mixtures. Natural Language Engineering 1, 163–190 (1995)
4. Clinchant, S., Gaussier, E.: Bridging Language Modeling and Divergence from Randomness Models: A Log-Logistic Model for IR. In: Azzopardi, L., Kazai, G., Robertson, S., Rüger, S., Shokouhi, M., Song, D., Yilmaz, E. (eds.) ICTIR 2009. LNCS, vol. 5766, pp. 54–65. Springer, Heidelberg (2009)
5. Clinchant, S., Gaussier, E.: Information-based models for ad hoc IR. In: SIGIR 2010, pp. 234–241 (2010)
6. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: SIGIR 2004, pp. 49–56 (2004)
7. Harter, S.P.: A Probabilistic Approach to Automatic Keyword Indexing. PhD thesis, The University of Chicago (1974)
8. He, B., Ounis, I.: On setting the hyper-parameters of term frequency normalization for information retrieval. ACM Trans. Inf. Syst. 25 (July 2007)

9. Hintikka, J.: On Semantic Information. In: Hintikka, J., Suppes, P. (eds.) Information and Inference, pp. 3–27. D. Reidel Pub. (1970)
10. Sparck Jones, K., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments. Information Processing and Management, 779–840 (2000)
11. Lease, M., Allan, J., Bruce Croft, W.: Regression Rank: Learning to Meet the Opportunity of Descriptive Queries. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 90–101. Springer, Heidelberg (2009)
12. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. IBM J. Res. Dev. 1, 309–317 (1957)
13. Lv, Y., Zhai, C.: Adaptive term frequency normalization for bm25. In: CIKM 2011, pp. 1985–1988 (2011)
14. Lv, Y., Zhai, C.: Lower-bounding term frequency normalization. In: CIKM 2011, pp. 7–16 (2011)
15. Lv, Y., Zhai, C.: When documents are very long, bm25 fails! In: SIGIR 2011, pp. 1103–1104 (2011)
16. Ponte, J.M., Bruce Croft, W.: A language modeling approach to information retrieval. In: SIGIR 1998, pp. 275–281 (1998)
17. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR 1994, pp. 232–241 (1994)
18. Robertson, S., Zaragoza, H., Taylor, M.: Simple bm25 extension to multiple weighted fields. In: CIKM 2004, pp. 42–49 (2004)
19. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: TREC 1994, pp. 109–126 (1994)
20. Singhal, A.: Modern information retrieval: a brief overview. In: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 24 (2001)
21. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: SIGIR 1996, pp. 21–29 (1996)
22. Svore, K.M., Burges, C.J.C.: A machine learning approach for improved bm25 retrieval. In: CIKM 2009, pp. 1811–1814 (2009)
23. Svore, K.M., Kanani, P.H., Khan, N.: How good is a span of terms?: exploiting proximity to improve web retrieval. In: SIGIR 2010, pp. 154–161 (2010)
24. Tao, T., Zhai, C.: An exploration of proximity measures in information retrieval. In: SIGIR 2007, pp. 295–302 (2007)
25. Taylor, M., Zaragoza, H., Craswell, N., Robertson, S., Burges, C.: Optimisation methods for ranking functions with multiple parameters. In: CIKM 2006, pp. 585–593 (2006)
26. Tison, C., Nicolas, J.M., Tupin, F.: Accuracy of fisher distributions and log-moment estimation to describe amplitude distributions of high resolution sar images over urban areas. In: IGARSS 2003, pp. 1999–2001 (2003)
27. Xu, Z., Akella, R.: A new probabilistic retrieval model based on the dirichlet compound multinomial distribution. In: SIGIR 2008, pp. 427–434 (2008)
28. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: SIGIR 2001, pp. 334–342 (2001)