Lower-Bounding Term Frequency Normalization

Yuanhua Lv
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
ylv2@uiuc.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.uiuc.edu

ABSTRACT

In this paper, we reveal a common deficiency of the current retrieval models: the component of term frequency (TF) normalization by document length is not lower-bounded properly; as a result, very long documents tend to be overly penalized. In order to analytically diagnose this problem, we propose two desirable formal constraints to capture the heuristic of lower-bounding TF, and use constraint analysis to examine several representative retrieval functions. Analysis results show that all these retrieval functions can only satisfy the constraints for a certain range of parameter values and/or for a particular set of query terms. Empirical results further show that the retrieval performance tends to be poor when the parameter is out of the range or the query term is not in the particular set. To solve this common problem, we propose a general and efficient method to introduce a sufficiently large lower bound for TF normalization which can be shown analytically to fix or alleviate the problem. Our experimental results demonstrate that the proposed method, incurring almost no additional computational cost, can be applied to state-of-the-art retrieval functions, such as Okapi BM25, language models, and the divergence from randomness approach, to significantly improve the average precision, especially for verbose queries.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory

Keywords

Term frequency, lower bound, formal constraints, data analysis, document length, BM25+, Dir+, PL2+

1. INTRODUCTION

Optimization of retrieval models is a fundamentally important research problem in information retrieval because an improved retrieval model would lead to improved performance for all search engines. Many effective retrieval models have been proposed and tested, such as vector space models [16, 18], classical probabilistic retrieval models [13, 8, 14, 15], language models [12, 20], and the divergence from randomness approach [1]. However, it remains a significant challenge to further improve these state-of-the-art models and design an ultimately optimal retrieval model.

In order to further develop more effective models, it is necessary to understand the deficiencies of the current retrieval models [4]. For example, in [18], it was revealed that the traditional vector space model retrieves documents with probabilities different from their probabilities of relevance, and the analysis led to the pivoted normalization retrieval function which has been shown to be substantially more effective than the traditional vector space model. In this work, we reveal a common deficiency of existing retrieval models in optimizing the TF normalization component and propose a general way to address this deficiency that can be applied to multiple state-of-the-art retrieval models to improve their retrieval accuracy.

Previous work [4] has shown that all the effective retrieval models tend to rely on a reasonable way to combine multiple retrieval signals, such as term frequency (TF), inverse document frequency (IDF), and document length. A major challenge in developing an effective retrieval model lies in the fact that multiple signals generally interact with each other in a complicated way. For example, document length normalization is to regularize the TF heuristic which, if applied alone, would have a tendency to overly reward long documents due to their high likelihood of matching a query term more times than a short document. On the other hand, document length normalization can also overly penalize long documents [18, 4]. What is the best way of combining multiple signals has been a long-standing open challenge. In particular, a direct application of a sound theoretical framework such as the language modeling approach to retrieval does not automatically ensure that we achieve the optimal combination of necessary retrieval heuristics as shown in [4].

To tackle this challenge, formal constraint analysis was proposed in [4]. The idea is to define a set of formal constraints to capture the desirable properties of a retrieval function related to combining multiple retrieval signals. These constraints can then be used to diagnose the deficiency of an existing model, which in turn provides insight into how to improve an existing model. Such an axiomatic approach has been shown to be useful for motivating and developing more effective retrieval models [6, 7, 2].

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

In this paper, we follow this axiomatic methodology and reveal a common deficiency of the current retrieval models in their TF normalization component and propose a general strategy to fix this deficiency in multiple state-of-the-art retrieval models. Specifically, we show that the normalized TF may approach zero when the document is very long, which often causes a very long document with a non-zero TF (i.e., matching a query term) to receive a score too close to or even lower than the score of a short document with a zero TF (i.e., not matching the corresponding query term). As a result, the occurrence of a query term in a very long document would not ensure that this document be ranked above other documents where the query term does not occur, leading to unfair over-penalization of very long documents.

The root cause for this deficiency is that the component of TF normalization by document length is not lower-bounded properly, i.e., the score "gap" between the presence and absence of a query term could be infinitely close to zero or even negative. In order to diagnose this problem, we first propose two desirable constraints to capture the heuristic of lower-bounding TF in a formal way, so that it is possible to apply them to any retrieval function analytically. We then use constraint analysis to examine several representative retrieval functions and show that all these retrieval functions can only satisfy the constraints for a certain range of parameter values and/or for a particular set of query terms. Empirical results further show that the retrieval performance tends to be poor when the parameter is out of the range or the query term is not in the particular set.

Motivated by this understanding, we propose a general and efficient methodology for introducing a sufficiently large lower bound for TF normalization, which can be applied directly to current retrieval models. Constraint analysis shows analytically that the proposed methodology can successfully fix or alleviate the problem.

Our experimental results on multiple standard collections demonstrate that the proposed methodology, incurring almost no additional computational cost, can be applied to state-of-the-art retrieval functions, such as Okapi BM25 [14, 15], language models [12, 20], and the divergence from randomness approach [1], to significantly improve their average precision, especially when queries are verbose. Due to its effectiveness, efficiency, and generality, the proposed methodology can work as a "patch" to fix or alleviate the problem in current retrieval models, in a plug-and-play way.

2. RELATED WORK

Developing effective retrieval models is a long-standing central challenge in information retrieval. Many different retrieval models have been proposed and tested, such as vector space models [16, 18], classical probabilistic retrieval models [13, 8, 14, 15], language models [12, 20], and the divergence from randomness approach [1]; a few representative retrieval models will be discussed in detail in Section 3.1. In our work, we reveal and address a common "bug" of these retrieval models (i.e., TF normalization is not lower-bounded properly), and develop a general plug-and-play "patch" to fix or alleviate this bug.

Term frequency is the earliest and arguably the most important retrieval signal in retrieval models [15, 18, 12, 20, 17, 1, 4, 10]. The use of TF can be dated back to Luhn's pioneer work on automatic indexing [9]. It is widely recognized that linear scaling in term frequency puts too much weight on repeated occurrences of a term. Thus, TF is often upper-bounded through some sub-linear transformations [15, 18, 12, 20, 1, 2, 10] to prevent the contribution from repeated occurrences from growing too large. Particularly, in Okapi BM25 [14, 15], it is easy to show that there is a strict upper bound $(k_1 + 1)$ for TF normalization. However, the other interesting direction, *lower-bounding TF*, has not been well addressed before. Our recent work [11] appears to be the first study that notices the inappropriate lower-bound of TF in BM25 through empirical analysis, but there is no theoretic diagnosis of the problem. Besides, the approach proposed in [11] is not generalizable to lower-bound TF normalization in retrieval models other than BM25. In this paper, we extend [11] to show analytically and empirically that lower-bounding TF is necessary for all representative retrieval models and develop a general approach to effectively lower-bound TF in these retrieval models.

Document length normalization also plays an important role in almost all existing retrieval models to fairly retrieve documents of all lengths [18, 4], since long documents tend to use the same terms repeatedly (higher TF). For example, both Okapi BM25 [14, 15] and the pivoted normalization retrieval model [18] use the pivoted length normalization schema [18], which uses the average document length as the pivoted length to coordinate the normalization effects for documents longer than this pivoted length and documents shorter than it. The PL2 model, a representative of the divergence from randomness models [1], also uses the average document length to control document length normalization. A common deficiency of all these existing length normalization methods is that they tend to force the normalized TF to approach zero when documents are very long. As a result, a very long document with a non-zero TF could receive a score too close to or even lower than the score of a short document with a zero TF, which is clearly unreasonable. Although some exiting studies have attempted to use a sub-linear transformation of document length (e.g., the squared root of document length [3]) to heuristically replace the original document length in length normalization, they are not guaranteed to solve the problem and often lose to standard document length normalization such as the pivoted length normalization in terms of retrieval accuracy. Our work aims at addressing this inherent weakness of traditional document length normalization in a more general and effective way.

Constraint analysis has been explored in information retrieval to diagnostically evaluate existing retrieval models [4, 5], introduce novel retrieval signals into existing retrieval models [19], and guide the development of new retrieval models [6, 2]. The constraints in these studies are basic and are designed mostly based on the analysis of some common characteristics of existing retrieval formulas. Although we also use constraint analysis, the proposed constraints are novel and are inspired by our empirical finding of a common deficiency of the existing retrieval models. Moreover, although some existing constraints (e.g., LNCs and TF-LNC in [4, 5]) are also meant to regularize the interactions between TF and document length, they tend to be loose and cannot capture the heuristic of lower-bounding TF normalization. For example, the modified Okapi BM25 satisfies all the constraints proposed in [4, 5], but it still fails to lower-bound TF normalization properly. In this sense, the proposed two new constraints are complimentary to existing constraints [4, 5].

Model	$G\left(c(t,Q)\right)$	$F\left(c(t,D),\|D\|,td(t)\right)$
BM25	$\frac{(k_3+1)\cdot c(t,Q)}{k_3+c(t,Q)}$	$\frac{(k_1+1)c(t,D)}{k_1(1-b+b\cdot\|D\|/avdl)+c(t,D)}\cdot\log\frac{N+1}{df(t)}$
PL2	$c(t,Q)$	$\begin{cases}\frac{tfn_t^D\cdot\log_2\left(tfn_t^D\cdot\lambda_t\right)+\log_2 e\cdot\left(1/\lambda_t-tfn_t^D\right)+0.5\log_2\left(2\pi\cdot tfn_t^D\right)}{tfn_t^D+1} & \text{if } tfn_t^D>0 \text{ and } \lambda_t>1\\ 0 & \text{otherwise}\end{cases}$
Dir	$c(t,Q)$	$\log\left(\frac{\mu}{\|D\|+\mu}+\frac{c(t,D)}{(\|D\|+\mu)p(t\|C)}\right)$
Piv	$c(t,Q)$	$\begin{cases}\frac{1+\log(1+\log(c(t,D)))}{1-s+s\cdot\|D\|/avdl}\cdot\log\frac{N+1}{df(t)} & \text{if } c(t,D)>0\\ 0 & \text{otherwise}\end{cases}$

Table 1: Document and query term weighting components of representative retrieval functions.

Notation	Description
$c(t,D)$	Frequency of term t in document D
$c(t,Q)$	Frequency of term t in query Q
N	Total number of docs in the collection
$df(t)$	Number of documents containing term t
$td(t)$	Any measure of discrimination value of term t
$\|D\|$	Length of document D
$avdl$	Average document length
$c(t,C)$	Frequency of term t in collection C
$p(t\|C)$	Probability of a term t given by the collection language model [20]

Table 2: Notation

3. MOTIVATION OF LOWER-BOUNDING TF NORMALIZATION

In this section, we discuss and analyze a common deficiency (i.e., lack of appropriate lower bound for TF normalization) of four state-of-the-art retrieval functions, which respectively represent the classical probabilistic retrieval model (Okapi BM25 [14, 15]), the divergence from randomness approach (PL2 [1]), the language modeling approach (Dirichlet prior smoothing [20]), and the vector space model (pivoted normalization [18, 17]).

An effective retrieval function is generally comprised of two basic separable components: a within-query scoring formula for weighting the occurrences of a term in the query and a within-document scoring formula for weighting the occurrences of this term in a document. We will represent each retrieval function in terms of these two separable components to make it easier for us to focus on studying the document side weighting:

$$S(Q,D)=\sum_{t\in Q}G\left(c(t,Q)\right)\cdot F\left(c(t,D),|D|,td(t)\right) \quad (1)$$

where $S(Q,D)$ is the total relevance score assigned to document D with respect to query Q, and $G(\cdot)$ and $F(\cdot)$ are within-query scoring function and within-document scoring function respectively. In Table 1, we show how this general scheme can be used to represent all the four major retrieval models. Other related notations are listed in Table 2. Note that most of the notations were also used in some previous work, e.g., [4], and will be adopted throughout our paper.

3.1 Deficiency of Existing Retrieval Functions

3.1.1 Okapi BM25 (BM25)

The Okapi BM25 method [14, 15] is a representative retrieval function that represents the classical probabilistic retrieval model. The BM25 retrieval function is summarized in the second row of Table 1. Following work [4], we modify the original IDF formula of BM25 to avoid the problem of possibly negative IDF values. The within-document scoring function of BM25 can be re-written as follows:

$$F_{BM25}\left(c(t,D),|D|,td(t)\right)=\frac{(k_1+1)\cdot tfn_t^D}{k_1+tfn_t^D}\cdot\log\frac{N+1}{df(t)} \quad (2)$$

where k_1 is a parameter, and tfn_t^D is the normalized TF by document length using pivoted length normalization [18].

$$tfn_t^D=\frac{c(t,D)}{1-b+b\frac{|D|}{avdl}} \quad (3)$$

where b is the slope parameter in pivoted normalization.

When a document is very long (i.e., $|D|$ is much larger than $avdl$), we can see that tfn_t^D could be very small and approach 0. Consequently, F_{BM25} will also approach 0 as if t did not occur in D. It can be seen clearly in Figure 1 (1): when $|D_2|$ becomes very large, the score difference between D_2 and D_1 appears to be very small. This by itself, would not necessarily be a problem, but the problem is that, the occurrence of t in a very long document D fails to ensure D to be ranked above other documents where t does not occur. It suggests that the occurrences of a query term in very long documents may not be rewarded properly by BM25, and thus those very long documents could be overly penalized, which as we will show later, is indeed true.

3.1.2 PL2 Method (PL2)

The PL2 method is a representative retrieval function of the divergence from randomness framework [1]. In this paper, we use the modified PL2 formula derived by Fang et al. [5] instead of the original PL2 formula [1]. The only difference between this modified PL2 function and the original PL2 function is that the former essentially ignores non-discriminative query terms. It has been shown that the modified PL2 is more effective and robust than the original PL2 [5]. The modified PL2 (still called PL2 for convenience in the following sections) is presented in the third row of Table 1, where $\lambda_t=\frac{N}{c(t,C)}$ is the term discrimination value, and tfn_t^D is the normalized TF by document length:

$$tfn_t^D=c(t,D)\cdot\log_2\left(1+c\cdot\frac{avdl}{|D|}\right) \quad (4)$$

where $c>0$ is a retrieval parameter.

We can see that, when a document is very long, tfn_t^D could be very small and approach 0, which is very similar to the corresponding component in BM25. What is worse is that, when tfn_t^D is sufficiently small, the within-document score F_{PL2} will be a **negative** number surprisingly. However, as shown in Table 1, even if the term is missing, i.e., $c(t,D)=0$, F_{PL2} can still receive a default *zero* score. This interesting observation is illustrated in Figure 1 (2). It suggests that a very long document that matches a query term

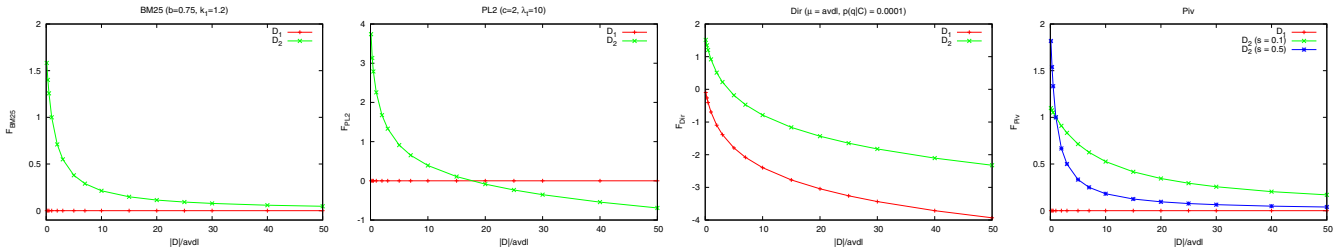

Figure 1: Comparison of the within-document term scores, i.e., $F(\cdot)$, of documents D_1 and D_2 w.r.t. query term t against different document lengths, where we assume $c(t, D_1) = 0$ and $c(t, D_2) = 1$. Here, x-axis and y-axis stand for the length of documents and the within-document term scores respectively.

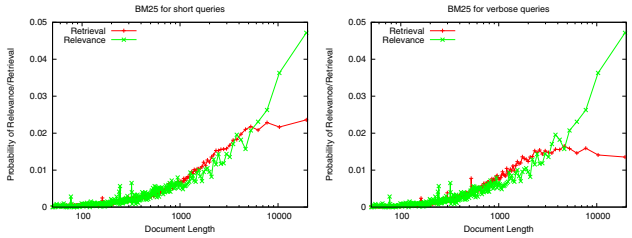

Figure 2: Comparison of retrieval and relevance probabilities against all document lengths when using BM25 on short (left) and verbose (right) queries.

may be penalized even more than another document (the length can be arbitrary) that does not match the term; consequently, those very long documents tend to be overly penalized by PL2.

3.1.3 Dirichlet Prior Method (Dir)

The Dirichlet prior method is one of the best performing language modeling approaches [20]. It is presented in the fourth row of Table 1, where μ is the Dirichlet prior.

It is observed that, the within-document scoring function F_{Dir} is monotonically decreasing with the document length variable. And when a document D_2 is very long, say $50 * avdl$, even if it matches a query term, the within-document score of this term could still be arbitrarily small. And this score could be smaller than that of any average-length document D_1 which does not match the term. This is shown clearly in Figure 1 (3). Thus, the Dirichlet prior method can also overly penalize very long documents.

3.1.4 Pivoted Normalization Method (Piv)

The pivoted normalization retrieval function [17, 4] represents one of the best performing vector space models. The detailed formula is shown in the last row of Table 1, where s is the slope parameter. Similarly, the analysis of the pivoted normalization method also shows that it tends to overly penalize very long documents, as shown in Figure 1 (4).

3.2 Likelihood of Relevance/Retrieval

Our analysis above has shown that, *in principle*, all these retrieval functions tend to overly penalize very long documents. Now we turn to seeking empirical evidence to see if this common deficiency hurts document retrieval *in practice*.

Inspired by Singhal et al.'s finding that a good retrieval function should retrieve documents of all lengths with similar chances as their likelihood of relevance [18], we compare the retrieval pattern of different retrieval functions with the relevance pattern. We follow the binning analysis strategy proposed in [18] and plot the two patterns against all document lengths on WT10G in Figure 2, where the bin size is set to 5000. Due to the space reason, we only plot BM25 as an example. But it is observed that other retrieval functions have similar trends as BM25. The plot shows clearly that BM25 retrieves very long documents with chances much lower than their likelihood of relevance. This empirically confirms our previous analysis that very long documents tend to be overly penalized.

4. FORMAL CONSTRAINTS ON LOWER-BOUNDING TF NORMALIZATION

A critical question is thus how we can regulate the interactions between term frequency and document length when a document is very long so that we can fix this common deficiency of current retrieval models?

To answer this question, we first propose two desirable heuristics that any reasonable retrieval function should implement to properly lower bound TF normalization when documents are very long: (1) there should be a sufficiently large gap between the presence and absence of a query term, i.e., the effect of document length normalization should not cause a very long document with a non-zero TF to receive a score too close to or even lower than a short document with a zero TF; (2) a short document that only covers a very small subset of the query terms should not easily dominate over a very long document that contains many distinct query terms.

Next, in order to analytically diagnose the problem of over-penalizing very long documents, we propose two formal constraints to capture the above two heuristics of lower bounding TF normalization so that it is possible to apply them to any retrieval function analytically. The two constraints are defined as follows:

LB1: Let Q be a query. Assume D_1 and D_2 are two documents such that $S(Q, D_1) = S(Q, D_2)$. If we reformulate the query by adding another term $q \notin Q$ into Q, where $c(q, D_1) = 0$ and $c(q, D_2) > 0$, then $S(Q \cup \{q\}, D_1) < S(Q \cup \{q\}, D_2)$.

LB2: Let $Q = \{q_1, q_2\}$ be a query with two terms q_1 and q_2. Assume $td(q_1) = td(q_2)$, where $td(t)$ can be any reasonable measure of term discrimination value. If D_1 and D_2 are two documents such that $c(q_2, D_1) = c(q_2, D_2) = 0$, $c(q_1, D_1) > 0$, $c(q_1, D_2) > 0$, and $S(Q, D_1) = S(Q, D_2)$, then $S(Q, D_1 \cup \{q_1\} - \{t_1\}) < S(Q, D_2 \cup \{q_2\} - \{t_2\})$, for all t_1 and t_2 such that $t_1 \in D_1$, $t_2 \in D_2$, $t_1 \notin Q$ and $t_2 \notin Q$.

The first constraint LB1 captures the basic heuristic of 0-1 gap in TF normalization, i.e., the gap between presence and absence of a term should not be closed by document length normalization. Specifically, if a query term does not occur in document D_1 but occurs in document D_2, and both documents receive the same relevance score from matching other query terms, then D_1 should be scored lower than D_2, no matter what are the length values of D_1 and D_2. In other words, the occurrence of a query term in a very long document should still be able to differentiate this document from other documents where the query term does not occur.

In fact, when $F(0, |D|, td(t))$ is a document-independent constant, LB1 can be derived from a basic TF constraint, TFC1 [4]. Here, $F(0, |D|, td(t))$ is the document weight for a query term t not present in document D, i.e., $t \in Q$ but $t \notin D$. This property is presented below in Theorem 1.

THEOREM 1. *LB1 is implied by the TFC1 constraint, if the within-document weight for any missing term is a document independent constant.*

Proof: Let Q be a query. Assume D_1 and D_2 are two documents such that $S(Q, D_1) = S(Q, D_2)$. We reformulate query Q by adding another term $q \notin Q$ into the query, where $c(q, D_1) = 0$ and $c(q, D_2) > 0$. If D_2' is another document, which is generated by replacing all the occurrences of q in D_2 with a non-query term $t \notin Q \cup \{q\}$, then $c(q, D_2') = 0$ and $S(Q, D_1) = S(Q, D_2) = S(Q, D_2')$. Due to the assumption that the document weight for the missing term q is a document independent constant, it follows that $S(Q \cup \{q\}, D_1) = S(Q \cup \{q\}, D_2')$. Finally, since $|D_2'| = |D_2|$ and $c(q, D_2') = 0 < c(q, D_2)$, according to TFC1, we get $S(Q \cup \{q\}, D_1) = S(Q \cup \{q\}, D_2') < S(Q \cup \{q\}, D_2)$. □

However, when the document weights for missing terms are document dependent, LB1 will not be redundant in the sense that it cannot be derived from other constraints such as the proposed LB2 and the seven constraints proposed in [4]. For example, the Dirichlet prior retrieval function, as shown in Table 1, has a document-dependent weighting function for a missing term, which is $\log \frac{\mu}{|D|+\mu}$. As will be shown later, the Dirichlet prior method violates LB1, although it satisfies LB2 and most of the constraints proposed in [4].

The second constraint LB2 states that if two terms have the same discrimination value, a repeated occurrence of one term is not as important as the first occurrence of the other. LB2 essentially captures the intuition that covering more *distinct* query terms should be rewarded sufficiently, even if the document is very long. For example, given a query Q = {"computer", "virus"}, if two documents D_1 and D_2 with identical relevance scores with respect to Q both match "computer", but neither matches "virus", then if we add an occurrence of "virus" to D_1 to generate D_1' and add an occurrence of "computer" to D_2 to generate D_2', we should ensure that D_1' has a higher score than D_2'. This intuitively makes sense because D_1' is more likely to be related to computer virus, while D_2' may be just about other aspects of computer.

LB1 and LB2 are two necessary constraints to ensure that very long documents would not be overly penalized. When either is violated, the retrieval function would likely not perform well for very long documents and there should be room to improve the retrieval function through improving its ability of satisfying the corresponding constraint.

5. CONSTRAINT ANALYSIS ON CURRENT RETRIEVAL MODELS

5.1 Okapi BM25 (BM25)

BM25 satisfies TFC1 [4], and the within-document weight for any missing term is always 0. Therefore, BM25 satisfies LB1 unconditionally according to Theorem 1.

We now examine LB2. Due to the sub-linear property of TF normalization, we only need to check LB2 in the case when $c(q_1, D_1) = 1$, since when $c(q_1, D_1) > 1$, it is even harder to violate the constraint. Consider a common case when $|D_1| = avdl$. It can be shown that the LB2 constraint is equivalent to the following constraint on $|D_2|$:

$$|D_2| < \left(\frac{2k_1 + 2}{(k_1)^2 \cdot b} + 1\right) \cdot avdl \tag{5}$$

This means that LB2 is satisfied only if $|D_2|$ is smaller than a certain upper bound. Thus, a sufficiently long document would violate LB2. Note that the upper bound of $|D_2|$ is a monotonically decreasing function with both b and k_1. This suggests that a larger b or k_1 would lead BM25 to violate LB2 more easily, which is confirmed by our experiments.

5.2 PL2 Method (PL2)

In Fang et al.'s work [5], the TFC1 constraint is regarded equivalent to that "the first partial derivative of the formula w.r.t. the TF variable should be positive", which has been shown to be satisfied by the modified PL2 [5]. However, the PL2 function is not continuous when the TF variable is zero, and what is worse is that,

$$\lim_{c(t,D)\to 0} F_{PL2}\left(c(t, D), |D|, td(t)\right) < F_{PL2}\left(0, |D|, td(t)\right) = 0 \tag{6}$$

which shows that even the modified PL2 still fails to satisfy TFC1. So we cannot use Theorem 1 for PL2.

We thus check both LB1 and LB2 directly. Since the optimal setting of parameter c is usually larger than 1 [4], we consider a common case when $|D_1| = \frac{c}{3} \cdot avdl$. Similar to the analysis on BM25, we only need to examine LB2 for $c(q_1, D_1) = 1$. The LB1 constraint is approximately equivalent to

$$|D_2| < \frac{c}{2^{\exp\left(-\frac{2}{\lambda_t} - 1.84\right)} - 1} \cdot avdl \tag{7}$$

and LB2 is approximately equivalent to

$$|D_2| < \frac{c}{2^{\exp\left(0.27\log(\lambda_t) - \frac{2.27}{\lambda_t} - 2.26\right)} - 1} \cdot avdl \tag{8}$$

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

We can see that both LB1 and LB2 set an upper bound for document length, suggesting that a very long document would violate both LB1 and LB2. However the upper bound introduced by LB1 is always larger than that introduced by LB2. So we focus on LB2 in the following sections.

The upper bound of document length in LB2 is monotonically increasing with both c and λ_t. It suggests that, when c is very small, there is a serious concern that long documents would be overly penalized. On the other hand, a more discriminative term also violates the constraint more easily. These analyses are confirmed by our experiments.

5.3 Dirichlet Prior Method (Dir)

With Dir, the within-document weight for a missing term is $\log \frac{\mu}{|D|+\mu}$, which is document dependent. So Theorem 1 is not applicable to the Dirichlet method. We thus need to examine LB1 and LB2 directly.

First, we only check LB1 at the point of $c(q, D_2) = 1$, which is the easiest case for LB1 to be violated. By considering the common case that $|D_1| = avdl$, the LB1 constraint is equivalent to the following constraint on $|D_2|$:

$$|D_2| < avdl + \frac{1}{p(q|C)}\left(1 + \frac{avdl}{\mu}\right) \tag{9}$$

It shows that the Dirichlet method can only satisfy LB1 if $|D_2|$ is smaller than a certain upper bound, suggesting again that a very long document would violate LB1. And this upper bound is monotonically decreasing with both $p(q|C)$ and μ. On the one hand, a non-discriminative (i.e., large $p(q|C)$) term q violates LB1 easily; for example, if $\mu \cdot p(q|C) = 1$, the upper bound appears to be as low as $(2 * avdl + \mu)$. Thus, the Dirichlet method would overly penalize very long documents more for verbose queries. On the other hand, a large μ would also worsen the situation according to Formula 9. These are all confirmed by our experimental results.

Next, we turn to check LB2, which is equivalent to

$$\frac{n + 1 + \mu \cdot p(q|C)}{n + \mu \cdot p(q|C)} < \frac{1 + \mu \cdot p(q|C)}{\mu \cdot p(q|C)} \tag{10}$$

where $n \in \{1, 2, \cdots\}$. Interestingly, this inequality is always satisfied, suggesting that the Dirichlet method satisfies LB2 *unconditionally*. We thus expect that the Dirichlet method would have some advantages in the cases when other retrieval functions tend to violate LB2.

5.4 Pivoted Normalization Method (Piv)

It is easy to show that the pivoted normalization method also satisfies LB1 unconditionally.

We now examine LB2. Similar to the analysis on BM25, we only need to check LB2 in the case of $c(q_1, D_1) = 1$. By considering a common case when $|D_1| = avdl$, we see that LB2 is equivalent to the following constraint on $|D_2|$:

$$|D_2| < \left(\frac{0.899}{s} + 1\right) \cdot avdl \tag{11}$$

This means that LB2 is satisfied only if $|D_2|$ is smaller than a certain upper bound. And this upper bound is a monotonically decreasing function with s. So, in principle, a larger s would lead the pivoted normalization method to violate LB2 more easily, which can also explain why the optimal setting of s tends to be small [4]. Of course, if s is set to zero, LB2 would be satisfied, but that would be to turn off document length normalization completely, which would clearly lead to non-optimal retrieval performance.

6. A GENERAL APPROACH FOR LOWER-BOUNDING TF NORMALIZATION

The analysis above shows analytically that all the state-of-the-art retrieval models would tend to overly penalize very long documents. In order to avoid overly penalizing very long documents, we need to lower-bound TF normalization to make sure that the "gap" of the within-document scores $F(c(t, D), |D|, td(t))$ between $c(t, D) = 0$ and $c(t, D) > 0$ is sufficiently large. However, we do not want that the addition of this new constraint changes the implementations of other retrieval heuristics in these state-of-the-art retrieval functions, because the implementations of existing retrieval heuristics in these retrieval functions have been shown to work quite well [4].

We propose a general heuristic approach to achieve this goal by defining an improved within-document scoring formula F' as shown in Equation 12, where δ is a pseudo TF value to control the scale of the TF lower bound, and l is a pseudo document length which is document-independent. In this new formula, a retrieval model-specific, but document-independent value $F(\delta, l, td(t)) - F(0, l, td(t))$ would serve as an ensured "gap" between matching and missing a term: if $c(t, D) > 0$, the component of TF normalization by document length will be lower-bounded by such a document-independent value, no matter how large $|D|$ would be.

$$F'(c(t, D), |D|, td(t)) = \begin{cases} F(c(t, D), |D|, td(t)) + F(0, l, td(t)) & \text{if } c(t, D) = 0 \\ F(c(t, D), |D|, td(t)) + F(\delta, l, td(t)) & \text{otherwise} \end{cases} \tag{12}$$

It is easy to verify that $F'(c(t, D), |D|, td(t))$ is able to satisfy all the basic retrieval heuristics [4] that are satisfied by $F(c(t, D), |D|, td(t))$: first, it is trivial to show that, if $F(c(t, D), |D|, td(t))$ satisfies TFCs, $F'(c(t, D), |D|, td(t))$ will also satisfy them; secondly, $F(\delta, l, td(t))$ and $F(0, l, td(t))$, as two special points of $F(c(t, D), |D|, td(t))$, satisfy the TDC constraint in exactly the same way as $F(c(t, D), |D|, td(t))$, so does $F'(c(t, D), |D|, td(t))$; finally, since the newly introduced components are document-independent, they raise no problem for LNCs and TF-LNC.

The proposed methodology is very efficient, as it only adds a retrieval model specific but document-independent value to those standard retrieval functions. For a query Q, we only need to calculate $|Q|$ such values, which can even be done offline. Therefore, our method incurs almost no additional computational cost.

Finally, we can obtain the corresponding lower-bounded retrieval function through substituting $F'(c(t, D), |D|, td(t))$ for $F(c(t, D), |D|, td(t))$ in each retrieval function,

Take BM25 as an example. Obviously $F'(0, |D|, td(t)) = 0$. In $F(\delta, l, td(t))$, since l is a constant document length variable used for document length normalization, its influence can be absorbed into the TF variable δ, we thus set $l = avdl$ simply. Then, we obtain $F(\delta, avdl, td(t)) = \frac{(k_1+1)\delta}{k_1+\delta} \log \frac{N+1}{df(t)}$. Clearly parameter k_1 can also be absorbed into δ, and the above formula is simplified again as $\delta \log \frac{N+1}{df(t)}$. Finally, we derive a lower-bounded BM25 function, namely **BM25+**, as shown in the following Formula 13.

$$\sum_{t \in Q \cap D} \frac{(k_3 + 1)c(t, Q)}{k_3 + c(t, Q)} \times \left[\frac{(k_1 + 1)c(t, D)}{k_1\left(1 - b + b\frac{|D|}{avdl}\right) + c(t, D)} + \delta\right] \times \log \frac{N + 1}{df(t)} \tag{13}$$

$$\sum_{t \in Q \cap D} c(t, Q)\left[\log\left(1 + \frac{c(t, D)}{\mu \cdot p(t|C)}\right) + \log\left(1 + \frac{\delta}{\mu \cdot p(t|C)}\right)\right] + |Q| \cdot \log \frac{\mu}{|D| + \mu} \tag{14}$$

$$\sum_{t \in Q \cap D} c(t, Q)\left[\frac{1 + \log\left(1 + \log(c(t, D))\right)}{1 - s + s\frac{|D|}{avdl}} + \delta\right] \log \frac{N + 1}{df(t)} \tag{15}$$

$$\sum_{t\in Q\cap D,\lambda_t>1} c(t,Q)\left[\frac{tfn_t^D\log_2\left(tfn_t^D\cdot\lambda_t\right)+\log_2 e\cdot\left(\frac{1}{\lambda_t}-tfn_t^D\right)+\frac{\log_2\left(2\pi\cdot tfn_t^D\right)}{2}}{tfn_t^D+1}+\frac{\delta\log_2\left(\delta\cdot\lambda_t\right)+\log_2 e\cdot\left(\frac{1}{\lambda_t}-\delta\right)+\frac{\log_2(2\pi\delta)}{2}}{\delta+1}\right] \quad (16)$$

Similarly, we can derive a lower-bounded Dirichlet prior method (**Dir+**), a lower-bounded pivoted normalization method (**Piv+**), and a lower-bounded PL2 (**PL2+**), which are presented in Formulas 14, 15, and 16 respectively.

Next, we check LB1 and LB2 on these four improved retrieval functions.

6.1 Lower-Bounded BM25 (BM25+)

It is trivial to verify that BM25+ still satisfies LB1 unconditionally. To examine LB2, we apply an analysis method that is consistent with our analysis for BM25 in Section 5.1. The LB2 constraint on BM25+ is equivalent to

$$\frac{k_1}{k_1+2}<\frac{(k_1+1)\cdot 1}{k_1\left(1-b+b\frac{|D_2|}{avdl}\right)+1}+\delta \quad (17)$$

which can be shown to be satisfied unconditionally if

$$\delta\geq\frac{k_1}{k_1+2} \quad (18)$$

Clearly, if we set δ to a sufficiently large value, BM25+ is able to satisfy LB2 unconditionally, which is also confirmed in our experiments that BM25+ works very well when we set $\delta=1$.

6.2 Lower-Bounded PL2 (PL2+)

We only need to check LB2 on PL2+, since it is easier to violate than LB1. With a similar analysis strategy as used for analyzing PL2, the LB2 constraint on PL2+ is equivalent to

$$|D_2|<\frac{c\cdot avdl}{2^{\exp\left(\left(0.27-\frac{2\delta}{\delta+1}\right)\log(\lambda_t)-\frac{2.27-\frac{2}{\delta+1}}{\lambda_t}-2.26-g(\delta)\right)}-1} \quad (19)$$

where $g(\delta)=\frac{(2\delta+1)\log\delta-2\delta+\log(2\pi)}{\delta+1}$. Due to space limit, we cannot show all the derivation details in this section.

We can see that, given a δ, the right side of the Formula 19 (i.e., the upper bound of $|D_2|$) is minimized when $\lambda_t=\frac{2.27\delta+0.27}{1.73\delta-0.27}$. This suggests that, in contrast to PL2, the upper bound of $|D_2|$ is not monotonically decreasing with λ_t. This interesting difference is shown clearly in Figure 3. Thus, if we set δ to an appropriate value to make the minimum upper bound still large enough (e.g., larger than the length of the longest document), PL2+ would not violate LB2.

6.3 Lower-Bounded Dirichlet Method (Dir+)

It is easy to show that Dir+ also satisfies LB2 unconditionally. We analyze Dir+ in the same way as analyzing Dir, and obtain the following equivalent constraint of LB1:

$$|D_2|<avdl+\frac{1+\delta}{p(t|C)}\left(1+\frac{avdl}{\mu}\right)+\frac{\delta}{\mu\cdot p^2(t|C)}\left(1+\frac{avdl}{\mu}\right) \quad (20)$$

We can see that, although Dir+ does not guarantee that LB1 is always satisfied, it indeed enlarges the upper bound of document length as compared to Dir in Section 5.3, and thus makes the constraint harder to violate. Generally, if we set δ to a sufficiently large value, the chance that very long documents are overly penalized would be reduced.

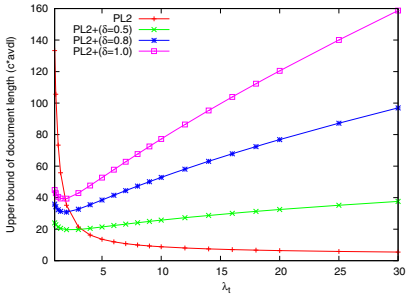

Figure 3: Comparison of upper bounds of document length in PL2 and PL2+ to satisfy LB2.

	Terabyte	WT10G	Robust04	WT2G
queries	701-850	451-550	301-450 601-700	401-450
#qry(with qrel)	149	100	249	50
avg(ql_short)	3.13	4.24	2.74	2.46
avg(ql_verb)	11.55	11.61	15.47	13.86
#total_qrel	28,640	5,981	17,412	2,279
#documents	25205k	1692k	528k	247k
avdl	949	611	481	1056
std(dl)/avdl	2.63	2.31	1.19	2.14

Table 3: Document set characteristic

6.4 Lower-Bounded Pivoted Method (Piv+)

It is easy to verify that Piv+ also satisfies LB1. Regarding LB2, similar to our analysis on Piv, the LB2 constraint on Piv+ is equivalent to

$$\log(1+\log 2)<\frac{1}{1-s+s\frac{|D_2|}{avdl}}+\delta \quad (21)$$

which is always satisfied if

$$\delta\geq\log(1+\log(2))\approx 0.53 \quad (22)$$

This shows that Piv+ can be able to satisfy LB2 unconditionally with a sufficiently large δ.

7. EXPERIMENTS

7.1 Testing Collections and Evaluation

We use four TREC collections: WT2G, WT10G, Terabyte, and Robust04, which represent different sizes and genre of text collections. WT2G, WT10G, and Terabyte are small, medium, and large Web collections respectively. Robust04 is a representative news dataset. We test two types of queries, short queries and verbose queries, which are taken from the title and the description fields of the TREC topics respectively. We use the Lemur toolkit and the Indri search engine (http://www.lemurproject.org/) to carry out our experiments. For all the datasets, the preprocessing of documents and queries is minimum, involving only Porter's stemming. An overview of the involved query topics, the average length of short/verbose queries, the total number of relevance judgments, the total number of documents, the

Query	Method	WT10G		WT2G		Terabyte		Robust04	
		MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
Short	BM25	0.1879	0.2898	0.3104	0.4840	0.2931	0.5785	0.2544	0.4353
	BM25+	**0.1962**^{4}	0.3040	0.3172^{1}	0.4820	**0.3004**^{1}	0.5685	**0.2553**	0.4357
	BM25+ ($\delta = 1.0$)	0.1927^{3}	0.3010	**0.3178**^{1}	0.4840	0.2997^{4}	0.5718	0.2548	0.4349
Verbose	BM25	0.1745	0.3250	0.2484	0.4380	0.2234	0.5221	0.2260	0.4036
	BM25+	**0.1850**^{1}	0.3360	**0.2624**^{3}	0.4400	0.2336^{4}	0.5309	0.2274	0.4056
	BM25+ ($\delta = 1.0$)	0.1841^{1}	0.3340	0.2565^{1}	0.4340	**0.2339**^{4}	0.5329	**0.2275**	0.4052

Table 4: Comparison of BM25 and BM25+ using cross validation. Superscripts 1/2/3/4 indicate that the corresponding MAP improvement is significant at the 0.05/0.02/0.01/0.001 level using the Wilcoxon test.

average document length, and the standard deviation of document length in each collection are shown in Table 3.

We employ a 2-fold cross-validation for parameter tuning, where the query topics are split into even and odd number topics as the two folds. The top-ranked 1000 documents for each run are compared in terms of their mean average precisions (MAP), which also serves as the objective function for parameter training. In addition, the precision at top-10 documents (P@10) is also considered. Our goal is to see if the proposed general heuristic can work well for improving each of the four retrieval functions.

7.2 BM25+ VS. BM25

In both BM25+ and BM25, we train b and k_1 using cross validation, where b is tuned from 0.1 to 0.9 in increments of 0.1, and k_1 is tuned from 0.2 to 4.0 in increments of 0.2. Besides, in BM25+, parameter δ is trained using cross validation, where δ is tuned from 0.0 to 1.5 in increments of 0.1, but we also create a special run in which δ is fixed to 1.0 empirically (labeled as BM25+ ($\delta = 1.0$)). The comparison results of BM25+ and BM25 are presented in Table 4.

The results demonstrate that BM25+ outperforms BM25 consistently in terms of MAP and also achieves P@10 scores better than or comparable to BM25. The MAP improvements of BM25+ over BM25 are much larger on *Web* collections than on the *news* collection. In particular, the MAP improvements on all Web collections are statistically significant. This is likely because there are generally more very long documents in Web data, where the problem of BM25, i.e., overly-penalizing very long documents, would presumably be more severe. For example, Table 3 shows that the standard deviation of the document length is indeed larger on the three Web collections than on Robust04.

Another interesting observation is that, BM25+, even with a fixed $\delta = 1.0$, can still work effectively and stably across collections and outperform BM25 significantly. This empirically confirms the constraint analysis results in Section 6.1 that, when $\delta > \frac{k_1}{k_1+2}$, BM25+ can satisfy LB2 unconditionally. It thus suggests that the proposed constraints can even be used to guide parameter tuning.

We further plot the curves of MAP improvements of BM25+ over BM25 against different δ values in Figure 4, which demonstrates that, when δ is set to a value around 1.0, BM25+ works very well across all collections. Therefore, δ can be safely "eliminated" from BM25+ by setting it to a default value 1.0.

Regarding different query types, we observe that BM25+ improves more on verbose queries than on short queries. For example, the MAP improvements on Web collections are often more than 5% for verbose queries and are around 2% for short queries. We hypothesize that BM25 may overly-penalize very long documents more seriously when queries are verbose, and thus there is more room for BM25+ to boost the performance. To verify our hypothesis, we collect the optimal settings of b and k_1 for BM25 in Table 5, which show that the optimal settings of b and k_1 are clearly *larger* for verbose queries than for short queries. Recall that our constraint analysis in Section 5.1 has shown that the likelihood of BM25 violating LB2 is monotonically increasing with parameters b and k_1. We can now conclude that BM25 indeed tends to overly penalize very long documents more when queries are more verbose.

Query	WT10G		WT2G		Terabyte		Robust04	
	b	k_1	b	k_1	b	k_1	b	k_1
Short	0.3	1.0	0.2	0.8	0.3	1.0	0.4	0.6
Verbose	0.6	2.0	0.6	1.6	0.4	1.8	0.7	1.2

Table 5: Optimal settings of b and k_1 in BM25.

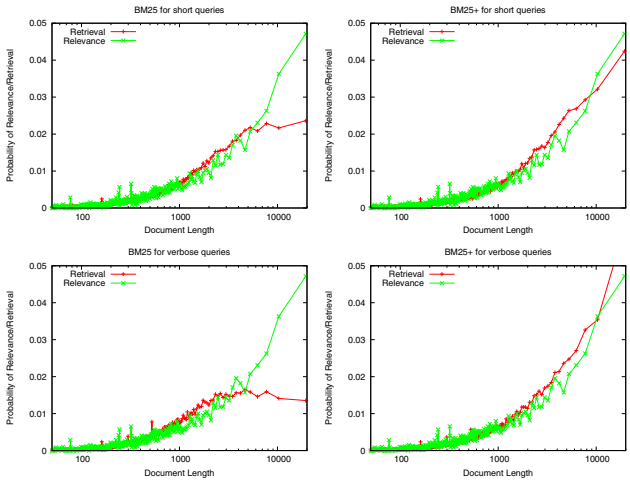

Figure 5: Comparison of retrieval and relevance probabilities against all document lengths when using BM25 (left) and BM25+ (right) for retrieval. It shows that BM25+ alleviates the problem of BM25 that overly penalizes very long documents.

So far we have shown that BM25+ is more effective than BM25, but if it is really because BM25+ has alleviated the problem of overly-penalizing very long documents? To answer this question, we plot the retrieval pattern of BM25+ as compared to the relevance pattern in a similar way as we have done in Section 3.2. The pattern comparison is presented in Figure 5. We can see that the retrieval pattern of BM25+ is more similar to the relevance pattern, especially

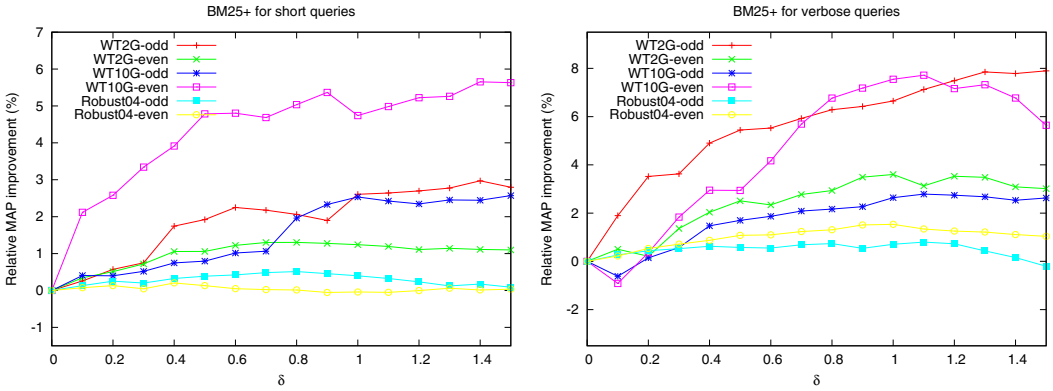

Figure 4: Performance Sensitivity to δ of BM25+, where y-axis shows the relative MAP improvements of BM25+ over BM25, and suffix '-even'/'-odd' indicates that only even/odd-number query topics are used.

Query	Method	WT10G	WT2G	Robust04
Short	PL2	0.1883	0.3231	0.2531
	PL2+	**0.1920**	0.3227	**0.2549**
	PL2+ ($\delta = 0.8$)	0.1912	**0.3255**	0.2540
Verbose	PL2	0.1695	0.2473	0.2185
	PL2+	$\mathbf{0.1886}^4$	0.2595	$\mathbf{0.2348}^4$
	PL2+ ($\delta = 0.8$)	$\mathbf{0.1886}^4$	$\mathbf{0.2639}^2$	0.2347^4

Table 6: Comparison of PL2 and PL2+ using cross validation. Superscripts 1/2/3/4 indicate that the corresponding MAP improvement is significant at the 0.05/0.02/0.01/0.001 level using the Wilcoxon test.

Query	WT10G	WT2G	Robust04
Short	9	23	9
Verbose	2	3	2

Table 7: Optimal settings of c in PL2.

for the retrieval of very long documents. This suggests that BM25+ indeed retrieves very long documents more fairly.

7.3 PL2+ VS. PL2

In both PL2+ and PL2, we train parameter c using cross validation, where c is tuned from 0.5 to 25 (27 values). Besides, in PL2+, parameter δ is also trained using cross validation, where δ is tuned from 0.0 to 1.5 in increments of 0.1. Also we create a special run of PL2+ in which δ is fixed to 0.8 empirically without training. The comparison results of PL2+ and PL2 are presented in Table 6.

The results show that PL2+ outperforms PL2 consistently, and even if we fix $\delta = 0.8$, PL2+ can still achieve stable improvements over PL2. Specifically, PL2+ improves significantly over PL2 for about **10**% on verbose queries, yet it only improves slightly on short queries; PL2+ appears to be less sensitive to the genre of collections, since it also improves significantly over PL2 on *news* data (verbose queries). We hypothesize that, PL2 may overly-penalize very long documents seriously on verbose queries but works well on short queries, and thus there is more room for PL2+ to improve the performance on verbose queries than on short queries. To verify it, we collect the optimal settings of c in PL2 and show them in Table 7. We can see that the optimal settings of c are "huge" for short queries as compared to that for verbose queries, presenting an obvious contrast. As a result, recalling the upper bound of document length in Formula 8, verbose queries would be more likely to violate LB2 even if a document is not very long (e.g., a news article), while short queries would only have a very small chance to violate LB2 even if a document is very long. Again, we can see that our constraint analysis is consistent with empirical results.

Query	Method	WT10G	WT2G	Robust04
Short	Dir	0.1930	0.3088	0.2521
	Dir+	0.1961	0.3112^2	$\mathbf{0.2530}^1$
	Dir+ ($\delta = 0.05$)	$\mathbf{0.1967}^1$	$\mathbf{0.3123}^3$	0.2525
Verbose	Dir	0.1790	0.2742	0.2329
	Dir+	$\mathbf{0.1874}^3$	0.2867^1	0.2440^4
	Dir+ ($\delta = 0.05$)	0.1871^3	$\mathbf{0.2871}^2$	$\mathbf{0.2440}^4$

Table 8: Comparison of Dir and Dir+ using cross validation. Superscripts 1/2/3/4 indicate that the corresponding MAP improvement is significant at the 0.05/0.02/0.01/0.001 level using the Wilcoxon test.

7.4 Dir+ VS. Dir

In both Dir+ and Dir, we train parameter μ using cross validation, where μ is tuned in a parameter space of 12 values from 500 to 10000. Besides, in Dir+, parameter δ is also trained, the candidate values of which are from 0.0 to 0.15 in increments of 0.01. Similarly, we also create a special run in which δ is fixed to 0.05 empirically without training. The comparison of Dir+ and Dir is presented in Table 8.

Overall, we observe that Dir+ improves over Dir consistently and significantly across different collections, and even if we fix $\delta = 0.05$ without training, Dir+ can still outperform Dir significantly in most cases. Note that, similar to BM25+ and PL2+, Dir+ works more effectively on verbose queries, which is consistent with our constraint analysis that Dir is more likely to overly penalize very long documents when a query contains more non-discriminative terms. In addition, we further compare Dir+ and Dir thoroughly by varying μ from 500 to 10000. It shows that Dir+ is consistently better than Dir no matter how we change the μ value.

Moreover, comparing Table 8 with Table 4 and 6, we can see that Dir works clearly better on verbose queries than BM25 and PL2. One possible explanation is that Dir satisfies LB2 unconditionally, but BM25 and PL2 do not.

7.5 Piv+ VS. Piv

In both Piv+ and Piv, we train s using cross-validation, where s is tuned from 0.01 to 0.25 in increments of 0.02.

Query	Method	WT10G	WT2G	Robust04
Short	Piv	0.1870	0.2915	0.2410
	Piv+	0.1869	0.2945^1	0.2455^1
Verbose	Piv	0.1493	0.2148	0.2144
	Piv+	0.1493	0.2154	0.2150

Table 9: Comparison of Piv and Piv+ using cross validation. Superscripts 1 indicates that the corresponding MAP improvement is significant at the 0.05 level using the Wilcoxon test.

Query	WT10G	WT2G	Robust04
Short	0.05	0.01	0.05
Verbose	0.05	0.11	0.19

Table 10: Optimal settings of s in Piv.

Besides, in Piv+, parameter δ is also trained, the candidate values are from 0.0 to 1.5 in increments of 0.1. The comparison results of Piv+ and Piv are presented in Table 9.

Unfortunately, Piv+ does not improve over Piv significantly in most of the cases, which, however, is also as we expected: although there is an upper bound of document length for Piv to satisfy LB2 (as shown in Formula 11), this upper bound is often very large because the optimal setting of parameter s is often very small as presented in Table 10. Nevertheless, Piv+ would work much better than Piv when s is large, as observed in our experiments.

7.6 Summary

Our experiments demonstrate empirically that, the proposed general methodology can be applied to state-of-the-art retrieval functions to successfully fix or alleviate their problem of overly-penalizing very long documents.

We have derived three effective retrieval functions, BM25+ (Formula 13), PL2+ (Formula 16), and Dir+ (Formula 14). All of them are as efficient as but more effective than their corresponding standard retrieval functions, i.e., BM25, PL2, and Dir, respectively. There is an extra parameter δ in the derived formulas, but we can set it to some default values (i.e., $\delta = 1.0$ for BM25+, $\delta = 0.8$ for PL2+, and $\delta = 0.05$ for Dir+), which perform quite well. The proposed retrieval functions can potentially replace its corresponding standard retrieval functions in all retrieval applications.

8. CONCLUSIONS

In this paper, we reveal a common deficiency of the current retrieval models: the component of term frequency (TF) normalization by document length is not lower-bounded properly; as a result, very long documents tend to be overly-penalized. In order to analytically diagnose this problem, we propose two desirable formal constraints to capture the heuristic of lower-bounding TF, and use constraint analysis to examine several representative retrieval functions. We find that all these retrieval functions can only satisfy the constraints for a certain range of parameter values and/or for a particular set of query terms. Empirical results further show that the retrieval performance tends to be poor when the parameter is out of the range or the query term is not in the particular set. To solve this common problem, we propose a general and efficient method to introduce a sufficiently large lower bound for TF normalization which can be shown analytically to fix or alleviate the problem.

Our experimental results on standard collections demonstrate that the proposed methodology, incurring almost no additional computational cost, can be applied to state-of-the-art retrieval functions, such as Okapi BM25 [14, 15], language models [20], and the divergence from randomness approach [1], to significantly improve the average precision, especially for verbose queries. Our work has also helped reveal interesting differences in the behavior of these state-of-the-art retrieval models. Due to its effectiveness, efficiency, and generality, the proposed methodology can work as a "patch" to fix or alleviate the problem in current retrieval models, in a plug-and-play way.

9. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-0713581, CNS-0834709, and CNS 1028381, by NIH/NLM grant 1 R01 LM009153-01, a Sloan Research Fellowship, and a Yahoo! Key Scientific Challenge Award.

10. REFERENCES

[1] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20:357–389, October 2002.

[2] S. Clinchant and E. Gaussier. Information-based models for ad hoc ir. In *SIGIR '10*, pages 234–241, 2010.

[3] R. Cummins and C. O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artif. Intell. Rev.*, 28:51–68, June 2007.

[4] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *SIGIR '04*, pages 49–56, 2004.

[5] H. Fang, T. Tao, and C. Zhai. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29:7:1–7:42, April 2011.

[6] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR '05*, pages 480–487, 2005.

[7] H. Fang and C. Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR '06*, pages 115–122, 2006.

[8] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35:243–255, 1992.

[9] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1:309–317, October 1957.

[10] Y. Lv and C. Zhai. Adaptive term frequency normalization for bm25. In *CIKM '11*, 2011.

[11] Y. Lv and C. Zhai. When documents are very long, bm25 fails! In *SIGIR '11*, pages 1103–1104, 2011.

[12] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, 1998.

[13] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society of Information Science*, 27(3):129–146, 1976.

[14] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, pages 232–241, 1994.

[15] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC '94*, pages 109–126, 1994.

[16] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[17] A. Singhal. Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24:2001, 2001.

[18] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR '96*, pages 21–29, 1996.

[19] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *SIGIR '07*, pages 295–302, 2007.

[20] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.