When Documents Are Very Long, BM25 Fails!

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

We reveal that the Okapi BM25 retrieval function tends to *overly penalize very long documents*. To address this problem, we present a simple yet effective extension of BM25, namely **BM25L**, which "shifts" the term frequency normalization formula to boost scores of very long documents. Our experiments show that BM25L, with the same computation cost, is more effective and robust than the standard BM25.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

BM25, BM25L, term frequency, very long documents

1. MOTIVATION

The Okapi BM25 retrieval function [2, 3] has been the state-of-the-art for nearly two decades. BM25 scores a document D with respect to query Q as follows:

$$\sum_{q \in Q \cap D} \frac{(k_3+1)c(q,Q)}{k_3 + c(q,Q)} \cdot f(q,D) \cdot \log \frac{N+1}{df(q)+0.5} \quad (1)$$

where $c(q,Q)$ is the count of q in Q, N is the total number of documents, $df(q)$ is the document frequency of q, and k_3 is a parameter. Following [1], we use a modified IDF formula in BM25 to avoid its problem of possibly negative IDF values.

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

$$f(q,D) = \frac{(k_1+1)c(q,D)}{k_1\left(1-b+b\frac{|D|}{avdl}\right)+c(q,D)} = \frac{(k_1+1)c'(q,D)}{k_1 + c'(q,D)} \quad (2)$$

where $|D|$ represents document length, $avdl$ stands for average document length, $c(q,D)$ is the raw TF of q in D, and b and k_1 are two parameters. $c'(q,D)$ is the normalized TF by document length using pivoted length normalization [4].

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

SIGIR'11, July 24–28, 2011, Beijing, China.
ACM 978-1-4503-0757-4/11/07.

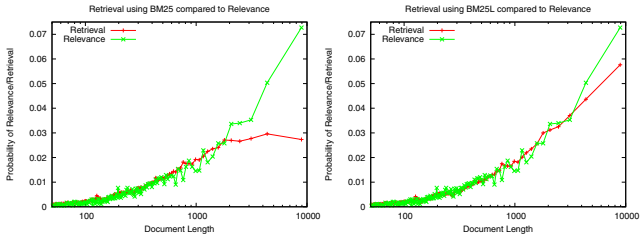

Figure 1: Comparison of retrieval and relevance probabilities against all document lengths.

When a document is very long, we can see that $c'(q,D)$ could be very small and approach 0. Consequently, $f(q,D)$ will also approach 0 as if q did not occur in D. That is, the presence of q in a very long document D fails to differentiate D clearly from other documents where q is absent. This 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 can be overly penalized. (See Figure 1 for empirical evidence of this problem.)

2. BOOSTING VERY LONG DOCUMENTS

In order to avoid overly-penalizing very long documents, we need to add a constraint in TF normalization to make sure that the "score gap" of $f(q,D)$ between $c'(q,D)=0$ and $c'(q,D)>0$ be sufficiently large. However, we do not want that the addition of this new constraint violates those existing properties of BM25 [2], which have been shown to work quite well. Thus what we want is an improved sub-linear TF normalization formula $f'(q,D)$ that has all the following characteristics: (I) It is zero for $c'(q,D)=0$; (II) it increases monotonically as $c'(q,D)$ increases but to an asymptotic maximum; (III) it decreases monotonically as $c'(q,D)>0$ decreases but to *an asymptotic minimum that is sufficiently large.* Here (I) and (II) are characteristics of the TF normalization formula of the original BM25 [2].

One heuristic way to achieve this goal is to define $f'(q,D)$ as follows:

$$f'(q,D) = \begin{cases} \frac{(k_1+1)\cdot[c'(q,D)+\delta]}{k_1+[c'(q,D)+\delta]} & \text{if } c'(q,D)>0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

which is essentially a shifted version of $f(q,D)$ by addition of a shift parameter $\delta > 0$. It is easy to verify that $f'(q,D)$ still satisfies both properties (I) and (II). Moreover, $f'(q,D)$ has a *positive* lower bound $(k_1+1)\delta/(k_1+\delta)$ for $c'(q,D)>0$. In this sub-linear function $f'(q,D)$, the score increase from the addition of δ is decreasing as $c'(q,D)$ increases. Therefore,

Method	Terabyte		WT10G		WT2G		Robust04		TREC8		AP8889	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
BM25	0.2942	0.5703	0.2099	0.3031	0.3198	0.4620	0.2543	0.4345	0.2557	0.4580	0.2631	0.4071
BM25L	**0.2999**	0.5703	**0.2154**	0.3072	**0.3260**	0.4780	0.2553	0.4390	0.2571	0.4560	0.2650	0.4152

Table 1: Comparison of optimal performance. δ is fixed to 0.5 in BM25L. Bold font indicates that the corresponding MAP improvement is statistically significant using the Wilcoxon test ($p < 0.05$).

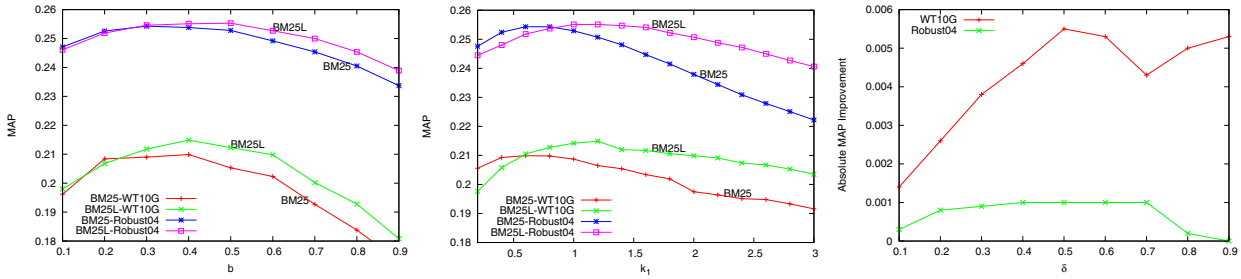

Figure 2: Performance Sensitivity to b (left), k_1 (middle), and δ (right).

$f'(q, D)$ tends to favor small $c'(q, D)$ values more, which would intuitively boost very long documents.

Finally, substituting $f'(q, D)$ into Equation 1 to replace $f(q, D)$, we obtain a new retrieval function, namely **BM25L**.

3. EXPERIMENTS

We compared BM25L with BM25 using six TREC collections: Terabyte, WT10G, and WT2G are web datasets with queries 701-850, 451-550, and 401-450 respectively; Robust04, TREC8, and AP8889 represent news datasets with queries 301-450&601-700, 401-450, and 51-150 respectively. The preprocessing included Porter stemming and removal of standard InQuery stopwords. For both BM25 and BM25L, we set $k_3 = 1000$ and tuned $b \in [0.1, 0.9]$ and $k_1 \in [0.2, 3.0]$ to optimize MAP. The parameter δ in BM25L was empirically set to **0.5** unless otherwise specified.

3.1 Retrieval Pattern VS. Relevance Pattern

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 [4], we also compare the retrieval pattern of BM25 with the relevance pattern. We follow the binning analysis strategy proposed in [4] and plot the two patterns against all document lengths on WT10G in Figure 1 (left). It confirms our previous analysis that *BM25 retrieves very long documents with chances much lower than their likelihood of relevance.*

We also plot the retrieval and relevance patterns for BM25L in Figure 1 (right). As expected, BM25L indeed alleviates the problem of over-penalizing very long documents clearly.

3.2 Experimental Results

We first compare the optimal performance of BM25 and BM25L in Table 1. We observe that BM25L outperforms the well-tuned BM25 consistently in both MAP and P@10. Besides, BM25L works better on web collections than on news collections. This is likely because there are generally more very long documents in web collections, where the problem of BM25, i.e., overly-penalizing very long documents, would presumably be more severe.

Parameter b controls the influence of document length in TF normalization. We draw the sensitivity curves of BM25 and BM25L to b on WT10G and Robust04 in Figure 2 (left), where k_1 is optimized for each method. It shows that BM25L is more robust than BM25; if we increase b, the MAP of BM25 drops more quickly than BM25L. It intuitively makes sense in that, increasing b in BM25 would overly penalize very long documents even more. Overall, BM25L works effectively with $b \in [0.3, 0.6]$.

We also draw in Figure 2 (middle) the sensitivity curves of BM25 and BM25L through varying k_1, where b is optimized for each method. We can see that BM25 drops dramatically when increasing k_1, while BM25L appears more stable. This observation is expected: according to Formula 2, for a small $c'(q, D)$ in very long documents, the larger k_1 is, the smaller $f(q, D)$ will be; as a result, increasing k_1 would exacerbate the problem of BM25. However, the $f'(q, D)$ of BM25L is guaranteed to have a document-independent lower bound $(k_1+1)\delta/(k_1+\delta)$ to avoid overly penalizing very long documents. Additionally, the optimal k_1 of BM25L is also larger than that of BM25 due to the effect of "shifting"; setting $k_1 \in [1.0, 2.0]$ usually works well for BM25L.

In previous experiments, we empirically set the shift parameter $\delta = 0.5$ in BM25L. To examine how δ affects the performance of BM25L, we plot the absolute MAP improvements against different values of δ in Figure 2 (right). It shows that the simple shifting strategy can successfully improve performance steadily and setting $\delta = 0.5$ works well for both web and news datasets. Besides, it confirms our finding that BM25L works more effectively on web data.

4. CONCLUSIONS

We proposed BM25L, a simple yet effective extension of BM25, to overcome the problem of BM25 which tends to over-penalize very long documents. Our experiments show that BM25L, with the same computation cost, is more effective and robust than BM25. BM25L can potentially replace the standard BM25 in all retrieval applications.

5. REFERENCES

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

[2] 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.

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

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