Evaluation of parameter values of BM25 Retrieval Function for different datasets

Lavanya Sharma
Univeristy of Waterloo
Waterloo, Canada
l9sharma@uwaterloo.ca

ABSTRACT

One of the popular traditional retrieval functions based on the bag-of-words model is BM25. It ranks a set of documents depending on the query terms appearing in each document. However, it has free parameters whose values must be set for optimum retrieval. Most of the time, default values of these parameters are used. The retrieval performance can be improved significantly by tuning these parameters properly for that dataset. In this study, the aim is to find a range of optimum parameters for BM25 retrieval function for four famous datasets: MS MARCO Document, MS MARCO Passage, FEVER, and Robust04. Also, the effect and trend of different parameters (b and k1 in BM25) on these datasets are studied. Performance measures such as Mean Average Precision (MAP), Normalized Cumulative Gain (NDCG), and Recall are used to evaluate the performance of different parameters of BM25 retrieval function. The experimental results demonstrate that each dataset is sensitive to parameter tuning. The study shows that datasets like MS MARCO Passage and MS MARCO Document are more sensitive to the tuning of the b parameter and demonstrate good results when the value of b is higher. While FEVER and Robust04 datasets prefer lower values of both b and k1 for good results.

KEYWORDS

Information Retrieval, MS MARCO, FEVER, Robust04, BM25, Parameter Optimization, MAP

1 INTRODUCTION

Information retrieval system responds to each new query with results that would satisfy the querying user's information need. The probabilistic relevance framework for document retrieval led to the development of one of the most successful text-retrieval algorithms, BM25 [5]. It is characterized by a notion of relevance between query and document pair. It estimates the probability of relevance and ranks documents in descending order of probability of relevance.

In a classic two-stage ranking system, term-based models like BM25 remain a go-to for the initial retrieval, as they provide baselines that are both efficient (through the inverted index) and are actually hard to beat [8].

A common technique is to first retrieve the top-k documents for a query by using BM25 (phase 1) and then apply another advanced technique to re-rank the top-k documents (phase 2). A good implementation of phase 1 can enhance the end-to-end performance of the system by enriching the candidate set for phase 2 [2]. Since BM25 is used in phase1 for ranking, the best performance of BM25 becomes essential for ranking purposes.

The good performance of BM25 mainly depends on two parameters, k1 and b. For most document collections, default values for k1 and b should be suitable. But the optimal values really depend on the collection being used. Finding good values for a collection is a matter of adjusting and checking again and again. Significant gains in relevance can be obtained by properly optimizing the parameters.

The main goal of this study is to find optimal BM25 parameters for mainly four famous collections namely, MS MARCO doc, MS MARCO passage, robust04, and FEVER. In addition to finding optimal parameter values for these datasets, the goal will be to find some additional insights about the datasets and also answer some questions like what is the effect of different parameters on these datasets, are these datasets inclined to certain parameters more, how well does BM25 work on different types of datasets.

2 RELATED WORK

It is required to maximize the relevance of the BM25 ranking model to the model parameters and it can be associated with an optimization problem. Though, optimizing standard IR measures is not easy as these functions are not smooth and they don't have gradients [4]. For these reasons, it is not easy to apply standard optimization techniques. Hence, experiment evaluation is an alternative option. One such experiment evaluation was conducted where BM25 parameters k1 and b were manually tuned for the .gov and OHSUMED collections [3]. Also, large-scale datasets like MS MARCO have used BM25 with parameters $b = 0.87$ and $k1 = 4.68$ for retrieving top-k documents [1].

Taylor et al. [6] did optimization of the parameters of ranking function (BM25) to improve the performance for end-to-end retrieval. They also suggested that size of training data set can affect parameters of the ranking function. Zeng at al. [9] used BM25PRF, a variant of BM25 and tuned parameters for a range of values. For both k1 and b the range was from 0.1 to 0.9 with increments of 0.1 and compared the results with default values of BM25PRF.

3 INFORMATION RETRIEVAL WITH BM25

BM25 is a standard retrieval model which measures the relevance of a document to a query by aggregating scores between terms from the query and terms from the document. The 2-parameter model of BM25 is quite popular. The parameter k1 controls the term frequency saturation rate and parameter b controls document length normalization.

Tuning of BM25 is an important requirement for high recall and precision. Though default values depend on different types of implementations of BM25. For example Elasticsearch uses default values of $k1 = 1.2$ and $b = 0.75$. While IR toolkits like Anserini use $k1 = 0.9$, $b = 0.4$ as default parameters. Best k1 and b values are

Table 1: File size and number of records for corresponding datasets

Name	*Number of records*	*File size*
MS MARCO Document	3,213,835	22GB
MS MARCO Passage	8,841,823	2.9GB
FEVER	185,445	1.6GB
Robust04	528,155	1.9GB

found by incrementally changing parameters and then evaluating the results for that dataset. The parameters are evaluated between their respective bounds. The parameter b is between 0 and 1 while k1 is usually evaluated from 0 to 3. Large b helps in penalizing irrelevant topics in search. Further, the optimum value of k1 can even be larger than 3 depending upon the length of the text. k1 is generally large for the longer text.

BM25 function scores each document in a corpus according to the document's relevance to a particular text query. For a query Q, with terms $q_1, ...q_n$, the BM25 score for document D is:

$$BM25(D, Q) = \sum_{i=1}^{n} IDF(q_i, D) \frac{f(q_i, D).(k1 + 1)}{f(q_i) + k1.(1 - b + b.\frac{|D|}{d_{avg}})} \quad (1)$$

where:

- $f(q_i, D)$ is the number of times term q_i occurs in document D.
- $|D|$ is the number of words in document D.
- d_{avg} is the average number of words per document.
- b and $k1$ are hyper parameters for BM25.

4 DATASETS

In this study, experiments are performed on four datasets. These are 1) MS MARCO document 2) MS MARCO passage 3) Robust04 and 4) FEVER dataset. The file size and number of records for the four datasets are presented in Table 1.

4.1 MS MARCO Document

MS MARCO (Microsoft Machine Reading Comprehension) document is a large-scale dataset focused on machine reading comprehension, question answering, and document/passage ranking.

MS MARCO document dataset is based on questions in the Question Answering Dataset. The dataset was formulated based on documents that answered the questions. There are 3.2 million documents and the dataset was created with the goal to rank documents based on their relevance.

4.2 MS MARCO Passage

MS MARCO passage dataset is based on passages and questions in the Question Answering Dataset. There are 8.8 million passages and the dataset was created with the goal to rank passages based on their relevance. This dataset is one of the largest relevance datasets ever.

4.3 FEVER

We used pre-processed Wikipedia Pages (June 2017 dump) as the evidence corpus and this is provided by the FEVER (Fact Extraction and Verification) task, together with the large training dataset with 185,445 claims generated by altering sentences extracted from Wikipedia. The dataset is labeled as Supported, Refuted, and NotEnoughInfo with necessary evidence for the judgment. The Wikipedia dump used consists of Wikipedia articles' introductions only, which are referred to as "paragraphs". The collection consists of a total of 5,396,106 paragraphs.

4.4 Robust04

Robust04 dataset is a combined collection consisting of the documents from the Financial Times, the Federal Register 94, the LA Times, and FBIS (i.e. TREC disks 4&5, minus the Congressional Record). It contains 250 topics. It consisted of 528,155 documents totaling 1.9 GB. The average document size was 3781 bytes.

5 SOFTWARE TOOL

Anserini [7] is an open-source information retrieval toolkit used for performing information retrieval experiments. It is built on Lucence that aims to bridge the gap between academic information retrieval research and real-world search applications. A python-based toolkit (Pyserini) which is based on Anserini is also widely used for performing information retrieval tasks using the BM25 function. Pyserini requires Python 3.6+ and Java 11. It is primarily designed to provide effective, reproducible, and easy-to-use first-stage retrieval in a multistage ranking architecture.

Anserini supports a number of pre-built indexes for common collections. Some pre-built Anserini indexes are hosted at the University of Waterloo's GitLab and are mirrored on Dropbox.

6 EXPERIMENT

The experiment focuses on the full retrieval of rank documents /passages based on their relevance to the query. Given a query q, most relevant 100 documents $D = d_1, d_2, d_3, ..., d_{100}$, were retrieved by BM25 using different parameters (b and k1).

The experiment was performed using Pyserini. We used the pre-built indexes for datasets, Robust04 and MS MARCO passage. For the other two datasets (MS MARCO document and FEVER) we built the indexes for datasets on our own. For both, the datasets Lucene indexes were built. Paragraph indexing was used for the FEVER dataset. After retrieving the index for each dataset, we generated the queries and qrels file for all four datasets. Then, a retrieval run was performed on the Dev queries stored.

For BM25 many possible variants can be used. Here, BM25 implementation of Lucence open-search library was used which is by default built-in Anserini. Anserini uses default value of $k1 = 0.9$ and $b = 0.4$. The k1 variable affects how much a single query term can contribute to the score. That is, how many occurrences in the document maximizes the possible score for a term. The b variable affects how much the document length is penalized, with 0 not penalizing long documents at all, and 1 being the maximum. Typically, the value of k1 is evaluated in the 0 to 3 range. Many experiments seem to show optimal k1 to be in a range of 0.5-2.0. k1 highly depends on the size of the document. It is suggested that k1 should generally

Table 2: The results of MAP, NDCG, and Recall@100 for different parameter values of b and k1 of BM25 are presented. Maximum MAP results are shown in boldface and light blue color. Whereas minimum MAP results are presented in light red color

			NDCG					**MAP**					**Recall@100**				
			k1					*k1*					*k1*				
			1.0	*2.0*	*3.0*	*4.0*	*5.0*	*1.0*	*2.0*	*3.0*	*4.0*	*5.0*	*1.0*	*2.0*	*3.0*	*4.0*	*5.0*
MS MARCO Doc	*b*	*0.1*	0.2663	0.2749	0.2743	0.2715	0.2660	0.1727	0.1783	0.1767	0.1747	0.1702	0.6466	0.6649	0.6686	0.6632	0.6538
		0.3	0.3198	0.3317	0.3336	0.3304	0.3261	0.2193	0.2267	0.2284	0.2253	0.2214	0.7165	0.7460	0.7487	0.7458	0.7422
		0.6	0.3522	0.3691	0.3731	0.3719	0.3686	0.2489	0.2609	0.2636	0.2625	0.2593	0.7516	0.7874	0.7978	0.7968	0.7938
		0.9	0.3586	0.3776	0.3848	0.3873	0.3855	0.2568	0.2703	0.2764	**0.2782**	0.2764	0.7510	0.7909	0.8018	0.8082	0.8067
MS MARCO Passage	*b*	*0.1*	0.2704	0.2422	0.2188	0.2012	0.1867	0.1778	0.1554	0.1374	0.1243	0.1137	0.6342	0.5894	0.5480	0.5159	0.4876
		0.3	0.2817	0.2573	0.2346	0.2165	0.2019	0.1876	0.1670	0.1490	0.1351	0.1238	0.6495	0.6166	0.5788	0.5454	0.5200
		0.6	0.2900	0.2686	0.2489	0.2489	0.2169	**0.1931**	0.1751	0.1591	0.1591	0.1331	0.667	0.6368	0.6067	0.6067	0.5555
		0.9	0.2901	0.2694	0.2506	0.2338	0.2200	0.1925	0.1748	0.1596	0.1458	0.1350	0.669	0.6419	0.6138	0.5882	0.5645
FEVER	*b*	*0.1*	0.6310	0.5891	0.5415	0.5041	0.4734	**0.5450**	0.4975	0.4457	0.4063	0.3749	0.8980	0.8794	0.8523	0.8285	0.8060
		0.3	0.6255	0.5798	0.5325	0.4934	0.4617	0.5374	0.4845	0.4328	0.3903	0.3572	0.9002	0.8854	0.8596	0.8400	0.8189
		0.6	0.5587	0.4959	0.4513	0.4146	0.3839	0.4632	0.3946	0.3486	0.3109	0.2804	0.8758	0.8418	0.8111	0.7855	0.7600
		0.9	0.4560	0.3719	0.3182	0.2791	0.2507	0.3579	0.2770	0.2285	0.1936	0.1691	0.8039	0.7219	0.6565	0.6077	0.5675
Robust04	*b*	*0.1*	0.3830	0.3753	0.3593	0.3452	0.3325	0.2081	0.1973	0.1836	0.1723	0.1623	0.4028	0.3909	0.3681	0.3510	0.3371
		0.3	0.3906	0.3814	0.3646	0.3503	0.3385	**0.2149**	0.2034	0.1894	0.1778	0.1679	0.4125	0.3972	0.3753	0.3559	0.3440
		0.6	0.3868	0.3779	0.3625	0.3484	0.3354	0.2110	0.2009	0.1887	0.1777	0.1677	0.4121	0.3977	0.3762	0.3582	0.3425
		0.9	0.3639	0.3500	0.3339	0.3226	0.3130	0.1936	0.1815	0.1700	0.1612	0.1537	0.3934	0.3706	0.3480	0.3344	0.3224

trend toward larger numbers when the text is a lot longer and more diverse. For the inverse situation, it's suggested to set k1 on the lower side. For the current experiment and BM25 tuning of each dataset, we evaluated the values of k1 from 1 to 5 with a step size of 1. Similarly, for b it's suggested to keep the value of b higher when documents touch on several different topics in a broad way and otherwise smaller. Therefore, for b, the values were evaluated for 0.1, 0.3, 0.6 and 0.9. After the run was completed, we evaluated the results using trec_eval.

7 EVALUATION MEASURES

We evaluated the retrieved documents using the official TREC evaluation tool, trec_eval. The main evaluation metrics used for evaluation were MAP (Mean Average Precision), NDCG (Normalized Discounted Cumulative Gain) and Recall.

7.1 MAP

MAP considers the rank position of each relevant document. It requires many relevant documents for each query in the text collection. MAP provides a single figure measure of quality across recall levels for top k documents. The equation for MAP is given as follows:

$$MAP(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} Precision(R_{jk}) \tag{2}$$

where:

- Q is the set of relevant documents
- R_{jk} is the set of ranked retrieval results from the top results until you get to the document d_k

7.2 NDCG

NDCG is used for multiple levels of relevance. It assumes that highly relevant documents are more useful than marginally relevant documents. Also, if the ranked position of the relevant document is lower then it is less likely to be examined. Here, graded relevance is used as a measure of usefulness or gain. Gain is reduced for lower rank. NDCG is a measure of ranking quality and its formula is given by:

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj}\sum_{m=1}^{k}\frac{2^{R(j,m)-1}}{log_2(1+m)} \tag{3}$$

where:

- z_{kj} is the normalization factor

7.3 Recall

In order to determine the effectiveness of a query, two document sets are compared: 1) the set of documents returned by the query, Res and 2) the set of relevant documents for the topic contained in the collection, Rel. Therefore, recall is given by

$$Recall = \frac{|Rel \cap Res|}{|Rel|} \tag{4}$$

For ranked retrieval, the notion of recall is extended by considering top k documents.

$$Recall@k = \frac{|Rel \cap Res[1..k]|}{|Rel|} \tag{5}$$

8 RESULTS AND DISCUSSION

The results of MAP, NDCG, and Recall@100 for different parameter values of b and k1 of BM25 are evaluated and are presented in Table 2. For each dataset, the maximum MAP result is shown in boldface. The results show that a significant improvement can be achieved by manual tuning of parameters for all datasets. Maximum improvement was observed for FEVER dataset with MAP score varying from 0.1691 (b=0.9, k1=5.0) to 0.5450 (b=0.1, k1=1.0). This was followed by the MS MARCO document dataset where the MAP score varied from 0.1702 to 0.2782. Minimum improvement was observed for the MS MARCO passage dataset. MS MARCO passage dataset achieved the lowest MAP score among all the four datasets of 0.1931 even after parameter tuning.

We observe for the MS MARCO document dataset, the maximum MAP score is 0.2782 which is achieved when $b = 0.9$ and $k1 = 4$. The corresponding value of NDCG is 0.3873 and recall@100 is 0.8082.

We observe that maximum values are achieved for a very high value of k1 and b. We also observe for the MS MARCO document dataset, the highest values on MAP, NDCG, and recall are achieved when $b = 0.9$ suggesting that documents touch on several different topics broadly and are benefited by larger b as the irrelevant topics are penalized. Also, the value of k1 is pretty high suggesting that text is longer and more diverse. The minimum MAP score is achieved when $b = 0.1$ and $k1 = 5$.

We observe another interesting trend that as the value of b increases, the MAP score also gets better. And for low values of b, the MAP score decreases. This suggests that the value of b has a great effect in getting better values for the MS MARCO document dataset and higher values of b means better scores. For k1, we see that there is only a slight change in scores as the value of k1 increases. In general, we can conclude that optimum parameter values for the MS MARCO document dataset are around when $b = 0.9$ and $k1 = 4$.

MS MARCO passage dataset has a maximum MAP score of 0.1931 when $b = 0.6$ and $k1 = 1$. The corresponding values of NDCG and recall@100 are 0.2900 and 0.6667 respectively. Here, we observe a different effect of k1 and b as compared to the previous dataset. We see that as k1 increases, the scores decline for all values of b. This suggests that text is less diverse and shorter as compared to the previous dataset. This is true as we are dealing with passages here. The MAP score tends to be higher when $k1 = 1$. For b, as the value increases, the MAP score improves, though this change is less drastic as compared to the previous dataset. The minimum score of 0.1137 is obtained when $b = 0.1$ and $k1 = 5$. This suggests this dataset is more inclined to have a smaller k1 and higher value of b. For MS MARCO passage dataset optimal parameters are achieved when k1 is around 1 and b is on the higher end.

FEVER dataset has higher MAP, NDCG, and recall values as compared to the previous two datasets. The maximum MAP score achieved for FEVER dataset is 0.5450 when $k1 = 1$ and $b = 0.1$. Corresponding values of NDCG and recall@100 are 0.6310 and 0.8980. Since we observe high values when both k1 and b are low, this suggests that our text here is not long and diverse. Here, the topics are not irrelevant and hence are not penalized.

The different BM25 parameters for the FEVER dataset have a different trend as compared to the previous two datasets. Here, we observe that as we increase the value of b, the score decreases. This is different as compared to MS MARCO document and passage datasets. Here, lower values of b are better. Similarly, a lower value of k1 is much more preferable for this dataset. There is a steady difference as both values of b and k1 increase. The scores decreases steadily. And we observe that the worst score is 0.1691 when $k1 = 5$ and $b = 0.9$. The difference between the best score and the worst score is too high. This dataset definitely prefers lower values of k1 and b and therefore optimum values are when k1 and b are as low as possible.

Table 3: BM25 parameters for maximum MAP score

Dataset	*b*	*k1*	*Maximum MAP Score*
MS MARCO Document	0.9	4.0	0.2782
MS MARCO Passage	0.6	1.0	0.1931
FEVER	0.1	1.0	0.5450
Robust04	0.3	1.0	0.2149

Table 4: Corresponding Recall and NDCG values when MAP score was maximum

Dataset	*Recall@100*	*NDCG*
MS MARCO Document	0.8082	0.3873
MS MARCO Passage	0.6667	0.2900
FEVER	0.8980	0.6310
Robust04	0.4125	0.3906

Robust04 dataset performed best when $k = 1$ and $b = 0.3$ with MAP value of 0.2149. The corresponding NDCG and recall@100 values are 0.3906 and 0.4125. Here, we observe a similar trend as we observed for the FEVER dataset.

Since k1 and b are low, therefore we can conclude the text for robust04 is small and less diverse. The topics are also more relevant. The general trend of parameter values we observe here is that as k1 and b value increases, the scores decrease. For the robust04 dataset too, the minimum score values are obtained when the value of k1 and b are the largest. In general, we can say that optimum parameter values for the robust04 dataset are when both k1 and b values are low, that is, when k1 is around 1 and when b is around 0.3.

In general, we observe, all four datasets show performance improvement when their parameter values are changed. Both MS MARCO document and passage have similar trends such that optimum performance is attained for higher values of b. Whereas, for FEVER and Robust04 datasets performance improvement is observed for smaller values of b and k1. The maximum MAP scores and their corresponding b and k1 values are presented in Table 3. The corresponding recall and NDCG values with respect to maximum MAP score is also given in Table 4.

9 CONCLUSION

The goal of the study was to find the effect of different parameter values on the BM25 document ranking algorithm on different datasets. Also, we tried to find the optimal parameter values that could be attained for the MS MARCO document, MS MARCO passage, FEVER, and Robust04 datasets. The study showed that parameters like b and k1 can influence retrieval efficiency significantly based on the dataset. Also, the results show that tuning these parameters to a specific setting is a promising idea and can improve retrieval effectiveness.

REFERENCES

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Craswell, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, and Tri Nguyen. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[2] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[3] Dave Novak. 2014. Anne Schuth and Floor Sietsma and Shimon Whiteson and Maarten De Rijke. In *European Conference on Information Retrieval*. Springer, 75–87.

[4] Stephen Robertson and Hugo Zaragoza. 2007. On rank-based effectiveness measures and optimization. *Information Retrieval* 10, 3 (July 2007).

[5] Stephan Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[6] Michael Taylor, Hugo Zaragoza, Nick Craswell, Stephen Robertson, and Chris Burges. 2006. Optimisation methods for ranking functions with multiple parameters. In *Proceedings of the 15th ACM international conference on Information and knowledge management*. 585–593.

[7] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1256.

[8] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically Examining the Neural Hype Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 1129–1132.

[9] Zhaohao Zeng and Tetsuya Sakai. 2019. BM25 Pseudo Relevance Feedback Using Anserini at Waseda University.. In *OSIRRC@ SIGIR*. 62–63.