Thomas Wright
tgw4@illinois.edu

Variations on BM25

The BM25 (also known as Okapi BM25, Okapi being the name of the information retrieval system at London's City University where BM25 was first used) is a very powerful and commonly used algorithm in information retrieval. This function scores and ranks the results of documents compared to a given query. The BM25 function has become the standard function that many search engines and test retrieval functions use to do their ranking. It does a very good job of ranking, performing at near human levels on several test collections of documents [4]. Though it has been considered one of the go-to scoring systems to use for this application for many years, there have been several variations on it created by researchers over the years in an attempt to improve its scoring accuracy, as well as some variants that have been created for more specific targeted uses, rather than the general purpose nature of BM25. This paper will give an overview of the major BM25 variations to date, and note how they differ from the standard BM25.

BM25 grew out of probabilistic retrieval research work done in the 1970s and 1980s by Stephen Robertson, Karen Jones, and others. BM stands for "Best Match" and as the results of the research advanced, the number kept increasing, until they settled on BM25 being the best method. There were notable predecessors such as BM11 and BM15, which still live on in BM25, as it can emulate them by setting the BM25 parameters to certain values[9]. In fact, the original BM25 was a bit different from the one used most often today, mainly in the area of calculating the inverse document frequency (IDF). BM25 originally just considered it a binary variable, whether a term was present or not, but this evolved into a count of the term frequency rather than just a 1 or 0. Other changes to BM25 included being able to raise some of the parameter to various exponential powers, but research concluded that this did not improve performance over just using them at the 1st power.

<u>General purpose BM25 implementations:</u>

BM25+

When a document is longer than the average document length in a collection, that document is penalized in the scoring system. Since the $\frac{\textit{document length}}{\textit{average of all document lengths}}$ term is in the denominator of the equation, as the document length reaches large multiples of the average document length, it can cause a very long document which does contain query terms to be scored very low, nearly the same as a short document which does not contain query terms. To address this, BM25+ adds a constant δ to the term frequency component of the calculation, so that regardless of the length of a document, a document which matches query terms will always contribute at least that fixed amount to the score.

BM25L

This variation on BM25 has similar goals to BM25+. It also looks to address the issue of very long documents being over-penalized in the calculations. A bit different approach is taken than the one in BM25+ however. In BM25L, like BM25+, there is also a δ constant added in to the equation, but in this case, it is added to each occurrence of the count of terms in the document component (c_{td}). BM25L does appear to outperform standard BM25 on several standard document test collections, and works especially well on web collections compared to BM25, and does not have a higher computation cost than BM25 [7].

ATIRE BM25

A different inverse document frequecy (IDF) function is used for ATIRE BM25. The standard BM25 IDF function (aka the Robertson-Spark Jones IDF) can become negative if a term is in more than half of the documents in the collection if $d_{ft} > N/2$ (if document frequency d_{ft} is greater than half of the total number of documents N) since this will result in taking the log of a number less than one, which results in a negative value. In this case, if a query consists of only 1 term, if that term is found in more than half of the collections' documents, it could be ranked below documents that didn't contain the term. ATIRE BM25 seeks to remedy this by changing the IDF function to the Robertson-Walker IDF version, which approaches zero as d_{ft} approaches N, but does not go negative. In this way, documents containing the query term will be ranked above those that do not.

BM25-adpt

Standard BM25 uses a fixed k_1 parameter in all calculations. This k_1 parameter controls the scale and shape of the term frequency normalization component of the calculation. Rather than it being fixed, BM25-adpt uses an adaptive k_1 which is calculated for each term to try to improve the performance over standard BM25. BM25-adpt also includes a different IDF term, which also considers the randomness of the repeated occurences of a term. It is claimed that this new IDF works as well as the standard IDF and testing on various document test collections shows BM25-adpt to have a small but demonstrable advantage over standard BM25 [6].

BM25C, BM25Q, and BM25T

These BM25 variants also avoid a fixed k_1 parameter as BM25-adpt above does, but they do it in different ways. BM25C uses a k_1 specific to the document collection, BM25Q's k_1 is specific to the query, and BM25T, like BM25-adpt has the k_1 specific to the term, but BM25-T uses a log-logarithmic calculation to determine the k_1 value. BM25C, BM25Q and BM25T can give as good as or better results than standard BM25, but they are not as robust, and require a good set of training data to ensure k1 can be set correctly[8].

<u>Specific use-case BM25 variants:</u>

BM25F

This version is used when processing more structured documents. When looking at docuemnts with identifiable fields, such as When looking at documents with identifiable fields, such as title, abstract, summary, body, anchor text, etc. By giving priority to terms that show up in multiple fields, such as a word showing up in the title, abstract, and the body of the document, BM25F can give improved ranking results over BM25 which just treats the document as one entity.

BM25H

This derivative of BM25 is specifically targeted towards web searches on an internet browser. It includes a client-side plug-in to the browser that allows the scoring function to know the user's recent browsing history. This introduces what is called a "temporal document frequency" which uses recently browsed pages to compute the imporance of keywords. This allows discovery of new relvant keywords [3]. Initial testing showed that the performance of BM25H was significantly better than BM25 for these types of searches with users' histories.

Lucene BM25

The Lucene open source engine uses a modified BM25 version by default. The first difference between it and standard BM25 is that the adds a constant 1 to the IDF calculation before taking the logarithm, so that it cannot become negative. An even bigger difference is that is compresses the document length used in the calculations down to a single byte, giving only 256 different values. This allows pre-calculation of the document length component in advance, ostensibly reducing calculation effort at calculation time. However, it has been found that comparing the computation time of this default behavior of Lucene and one modified to use the standard document length calculations yielded no appreciable savings in calculation time by using this compression. It has also been suggested that Lucene implementations could benefit from storing the exact document lengths rather than compressing them, as this can benefit other modules that interface with Lucene [2].

In conclusion, BM25 has been the standard scoring and ranking function for many information retrieval tasks for decades. Several variations have been created that try to improve on the results that can be obtained from BM25. In certain circumstances, such as with very long documents, or queries with only a few terms that occur in a majority of the documents, they have been shown to have small but statistically significant gains over the standard BM25 algorithm. And in the specific cases where BM25F and BM25H can be used, they can give noticeably better results than a standard BM25 implementation. For general use outside of these specific circumstances and data cases, any of the BM25 versions give similarly good results. In fact, in their conclusion, Kamphuis et al ask “Does it matter” (which BM25 version you choose), and they say “no, it does not.” as there is not a significant difference in the results [2]. So if you need to use a scoring/ranking function over a general collection of documents that don’t fit the special cases noted in the specific BM25 implementations above, using any of them should give you almost equally good results.

References:

1. G. Lan, Y. Ge and J. Kong, "Research on Scoring Mechanism Based on BM25F Model," *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 2019, pp. 1782-1786, doi: 10.1109/ITAIC.2019.8785547.

2. Kamphuis C., de Vries A.P., Boytsov L., Lin J. 2020. Which BM25 Do You Mean? A Large-Scale Reproducibility Study of Scoring Variants. In: Jose J. et al. (eds) Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science, vol 12036. Springer, Cham. DOI:https://doi.org/10.1007/978-3-030-45442-5_4

3. Karkali, Margarita & Plachouras, Vassilis & Stefanatos, Constantinos & Vazirgiannis, Michalis. (2012). Keeping keywords fresh: A BM25 variation for personalized keyword extraction. ACM International Conference Proceeding Series. 10.1145/2169095.2169099.

4. Trotman, Andrew and Keeler, David. 2011. Ad hoc IR: not much room for improvement. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11). Association for Computing Machinery, New York, NY, USA, 1095–1096. DOI:https://doi.org/10.1145/2009916.2010066

5. Trotman, Andrew, Puurual, Antii, and Burgess, Blake. 2014. Improvements to BM25 and Language Models Examined. In Proceedings of the 2014 Australasian Document Computing Symposium (ADCS '14). Association for Computing Machinery, New York, NY, USA, 58–65. DOI:https://doi.org/10.1145/2682862.2682863

6. Yuanhua Lv and ChengXiang Zhai. 2011. Adaptive term frequency normalization for BM25. In Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM '11). Association for Computing Machinery, New York, NY, USA, 1985–1988. DOI:https://doi.org/10.1145/2063576.2063871

7. Yuanhua Lv and ChengXiang Zhai. 2011. When documents are very long, BM25 fails! In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11). Association for Computing Machinery, New York, NY, USA, 1103–1104. DOI:https://doi.org/10.1145/2009916.2010070

8. Yuanhua Lv and ChengXiang Zhai. 2012. A Log-Logistic Model-Based Interpretation of TF Normalization of BM25. In: Baeza-Yates R. et al. (eds) Advances in Information Retrieval. ECIR 2012. Lecture Notes in Computer Science, vol 7224. Springer, Berlin, Heidelberg. DOI:https://doi-org.proxy2.library.illinois.edu/10.1007/978-3-642-28997-2_21

9. Unknown Author. The BM25 Weighting Scheme. https://xapian.org/docs/bm25.html