Tech Review: BM25 and Variants

Arindam Saha

saha2@illinois.edu

1. Introduction

Information Retrieval, in the field of Computer Science, is the processing of extracting relevant resources from a collection of resources [1]. In this modern age of technology, there is an abundant number of resources available on the web and it's expanding and evolving every day. To be able to extract relevant information from this massive sea of data requires sophisticated techniques. We need to be able to rank all such documents and present only the most relevant ones to the user. BM25 is one such ranking function used by search engines to estimate the relevance of documents to a given search query [2]. *BM* is an abbreviation of *best matching*. BM25 is the actual ranking function but it is often referred to as Okapi BM25, because Okapi was the first system to use it. It was named BM25 because it was the 25th iteration that worked.

2. BM25 and Variants

2.1 BM25

The motivation behind BM25 was to come up with a function to score a document D, given a query Q. An important term in the score function is the count of a particular word in the query, denoted by $c(w, D)$, where $\mathrm{w} \in \mathrm{Q}$. It is easy to see that the first occurrence of the word tells us a lot about the score, so does the second and so on. However, after a point, when we are pretty much certain that the word is quite important to the document D, its importance starts to wane off. For example, the 101st occurrence of the word does not necessarily convey more information that the 100th occurrence. So, the core idea behind BM25 is to modify the count, as such,

$$\frac{(\mathrm{k}+1)\mathrm{c}(\mathrm{w},\mathrm{D})}{\mathrm{k}+\mathrm{c}(\mathrm{w},\mathrm{D})}$$

where k is a parameter chosen by the user. It essentially caps the weight to $\mathrm{k} + 1$ and hence is an effective way of dealing with the problem stated earlier [3].

The full BM25 function, which is one of the most popular instantiations, is defined below (and includes document length normalization)

$$\text{score}(D, Q) = \sum_{i=1}^{n} \mathrm{IDF}(\mathrm{q_i}) \cdot \frac{\mathrm{f}(\mathrm{q_i}, \mathrm{D})(\mathrm{k_1} + 1)}{\mathrm{f}(\mathrm{q_1}, \mathrm{D}) + \mathrm{k_1} \cdot \left(1 - \mathrm{b} + \mathrm{b} \cdot \frac{|\mathrm{D}|}{\mathrm{avgdl}}\right)}$$

where,

$f(q_i, D)$ = number of times q_i occurs in document D

$|D|$ = number of words in document D

$avgdl$ = average document length in the collection

$k_1$and b are parameters usually chosen to be $k_1 \in [1.2, 2.0]$ and $b = 0.75$

From the above formula, we have the following special cases: BM15 $(b = 0, k_1 > 0)$, BM11 $(b = 1, k_1 > 0)$ and BM1 $(k_1 = 0)$. We can see that for documents of average length, BM25 = BM15 = BM11.

Over the years, researchers have proposed several tweaks to the original BM25 ranking function, in order to improve its performance. Here, we list some of them.

2.2 BM25L

Researchers have found that Okapi BM25 tends to overly penalize very long documents [5]. The function fails to generate a significant difference between very long documents that contain a query term and a document that does not contain a term. The change was to add a δ parameter that "shifts" TF normalization and establishes a lower bound when a query term occurs in a document, no matter how large the document is.

2.3 BM25+

This addresses the limitation of BM25 where the TF normalization by document length is not lower-bounded. Thus, like the issue being solved by BM25L, it tries to fix the fact that the ranking function is biased towards preferring shorter documents over long ones. The difference from BM25L is that here, we add a δ parameter to the whole TF component, instead of to the count i.e. $f(q_i, D)$.

2.4 BM25-adpt

This method applies a term-specific k_1, instead of a global k_1. The intuition is that a term-specific $k_1$will result in a better ranking scheme. The term-specific $k_1$values are calculated directly from the index. The formula was derived using information gain and divergence from randomness theory [6].

2.5 BM25T

This is a log-logistic method for calculating the term-specific $k_1$values. These $k_1$values are also calculated directly from the index and hence and they can just be applied to new collections.

There are extensions to this method, namely BM25C and BM25Q, to deal with issues in estimating k_1, when the document frequency is small.

2.6 BM25F

This version of BM25 considers that a document is made up of separate fields e.g., headlines, main text, anchor text, footer, etc. with each of them having varying degrees of importance.

3. Conclusion

BM25 is a state-of-the-art ranking function. It is widely used and has impressive performance. However, even though it has been in use for a while, researchers are still finding ways to improve it, as shown above, and specialize it for certain use cases. The above variants have all been tested on datasets and have been shown to outperform classic BM25. Hence, we should keep these variants in mind so that we know the tradeoffs between them and their strengths and weaknesses so that we pick the most appropriate one for our use case.

4. References

[1] https://en.wikipedia.org/wiki/Information_retrieval

[2] https://en.wikipedia.org/wiki/Okapi_BM25

[3] https://www.coursera.org/learn/cs-410/lecture/W0NZe/lesson-2-2-tf-transformation

[4] http://www.minerazzi.com/tutorials/okapi-bm25-model.pdf

[5] https://dl.acm.org/doi/pdf/10.1145/2009916.2010070

[6] http://www.cs.otago.ac.nz/homepages/andrew/papers/2014-2.pdf

[7] https://repository.ubn.ru.nl/bitstream/handle/2066/219374/219374.pdf?sequence=3&isAllowed=y

[8] https://www.mathworks.com/help/textanalytics/ref/bm25similarity.html