Building a Simple Information Retrieval System using BM25 and GPT-3 and evaluated in the CISI collection

Author: Monique Louise de Barros Monteiro – moniquelouise@gmail.com

1. Introduction

In this work, we build a baseline PoC information retrieval (RI) system with BM25 [1] algorithm (and some of its variants [2]). The CISI dataset [3] was used to evaluate the algorithm.

Instead of implementing the algorithm from scratch, the use of already implemented open-source libraries was the best choice, due to time constraints. Also, open-source libraries are generally well tested and optimized.

2. Uses of ChatGPT

ChatGPT was extensively used in this PoC development. Specifically:

- during the inception phase, ChatGPT was asked about Information Retrieval in general, BM25 algorithm and BM25 implementations;
- during development and validation, ChatGPT was asked about 1) the CISI dataset; 2) ways of assessing BM25 performance; 3) state-of-the-art results with CISI and BM25 and 4) ideal grid search intervals for BM25 hyperparameters;
- the code, developed with Jupyter notebooks in Google Colab environment, was initially documented in Portuguese. Here, ChatGPT was used to aid in the translation to English, requiring only minor changes.

The chats with ChatGPT are available at [4]. The document contains transcripts of the main interactions with ChatGPT, as well as an analysis of some information provided by the agent. Some transcripts are in Portuguese, while others are in English. This helps to check both the agent's language detection capabilities and its fluency in non-English languages.

3. Experiments

This proof-of-concept experiment was implemented in Python language. Two Python BM25 libraries were validated and compared: Rank-BM25 [5] and Pyserini [6].

Rank-BM25 is meant to be “a two line search engine” according to its documentation. It offers “a collection of algorithms for querying a set of documents and returning the ones most relevant to the query”. The following algorithms have been implemented: Okapi BM25, BM25L and BM25+. The experiments conducted in this PoC tested the three variants. The PoC code with Rank-BM25 is available at [7].

Pyserini is “a Python toolkit for reproducible information retrieval research with sparse and dense representations”. Here, BM25 is classified as a sparse retrieval mechanism, as it uses bag-of-words representations instead of dense representations such as word vectors. The PoC code with Pyserini is available at [8].

For each library, a grid search was conducted to tune the main BM25 hyperparameters: **k1** and **b**. According to [9], *k1* is used to control the degree of term frequency use: a value of 0 corresponds to a binary model (no term frequency), and a large value corresponds to using raw term frequency. Also, according to [9] *, b* determines the scaling by document length: b = 1 corresponds to fully scaling the term weight by the document length, while b = 0 corresponds to no length normalization.

We used *mean average precision* (MAP) as the main metric to choose the best performing model/implementation and hyperparameters. We also calculated recall and F1 for each combination of implementation and hyperparameters.

For both libraries, we need to define a priori the number of retrieved documents. For Rank-BM25, we consider the 95th percentile of the retrieved documents, by analyzing the respective document scores. Here, although ChatGPT recommendation for using the 75th percentile, the experiment was run several times with different percentiles to check for the most suitable one, showing that using the 95th percentile led to the best results. For Pyserini, we inspected the dataset to find the maximum number of returned documents and use it as the (fixed) number of hits. In the real world, we recommend the definition of the number of hits based on user input, as commonly the true dataset will not be fully available.

In order to run the experiments, it's only necessary to upload the Jupyter notebooks (.ipynb files [7], [8]) and the dataset files (CISI.ALL, CISI.QRY and CISI.REL available at [10]) to Google Colab.

4. Results and discussions

4.1. Algorithms Performance

We summarize the results Table 1.

Implementation	MAP	Recall	F-1	Hyperparameters
Rank-BM25 - Okapi BM25	10.72%	11.10%	10.91%	k1 = 1.2, b = 1.0
Rank-BM25 - BM25L	9.05%	9,79%	9.41%	k1 = 0.5, b = 0.5
Rank-BM25 - BM25+	11.64%	12.11%	11.87%	**k1 = 3.0, b = 0.75**
Pyserini	**12.07%**	**53.36%**	**19.68%**	**k1 = 3.0, b = 0.75**

Table 1 - Best results (MAP) for from grid search for each BM25 implementation and hyperparameters combination

As we can see in Table 1, Pyserini gives better performance than Rank-BM25. The best MAP value achieved by Pyserini was slightly better than the MAP value achieved by Rank-

BM25. Also, a significant improvement was made to recall (around 41 percent points!), leading to an improvement of almost 8 percent points to F-1. Another interesting result is that the same values for k1 and b (3.0 and 0.75, respectively) led to the best MAP values in each library, suggesting a strong sensitivity to term frequencies and document lengths for this specific dataset.

We also evaluated the use of a fixed number of 155 returns for Rank-BM25, but the results were worse, not even reaching 8% for MAP, which suggests that for that library the heuristic of returning the number of documents according to the percentiles of the scores generates better results.

Regarding the variants implemented by Rank-BM25, BM25+ gave the best results in Rank-BM25 implementation. According to [2], it brings a reasonable improvement when compared to pure Okapi BM25, because no matter how long the document is, a single occurrence of a search term contributes at least a constant amount to its retrieval score [2]. It is implemented by making small adjustments to the formula for calculating each document score. By the way, BM25 implementations vary in the exact formula for document scoring, as the algorithm has received several improvements since its creation.

Finally, as pointed by [2], a stemming step may lead to improvements in the algorithm performance. Due to time constraints, a stemming preprocessing step was not implemented here, but it certainly would be worth further investigation for the CISI dataset.

4.2. ChatGPT Performance

4.2.1. Chat about other Libraries

As said before, ChatGPT was asked about BM25 Python libraries/implementations.

When queried about Gensim's support to BM25 (removed from version 4), ChatGPT initially gave a wrong answer, but it soon corrected itself when the same question was asked again - showing its "memory" of previous interactions. On the other hand, it "mixed" Portuguese with English. These interactions are shown in [4].

Further, regarding "irspack" – another Python library cited by the agent – the link provided by ChatGPT seems to be broken. Furthermore, the official documentation of the tool (https://irspack.readthedocs.io/en/stable/) suggests that its focus is on recommendation systems. Although this does not necessarily mean that its implementation for BM25 is inferior or not suitable for search systems, no information about it could be found in the documentation, at least compared to the ease of use of the two libraries evaluated in this PoC.

Also, in a previous interaction, ChatGPT mentioned the Whoosh library (https://whoosh.readthedocs.io/en/latest/index.html). However, according to the library's documentation, its interface is somewhat cumbersome to use given the time available for implementing this PoC.

Finally, in a previous interaction, scikit-learn was also mentioned as a library that implements BM25. Although it is possible to implement the algorithm using objects such as TfidfVectorizer from scikit-learn, it does not have a direct implementation of the BM25 algorithm.

4.2.2. Chat about state-of-the-art values for CISI

When ChatGPT was asked about the best metrics achieved for MAP for the CISI dataset, the references returned by it did not clarify the use of the CISI dataset with BM25 algorithm.

For example, it said that according to the paper "Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search" by Zamani and Croft, “they reported a MAP score of 0.6826 for the CISI dataset”. However, that paper makes no reference to the CISI database.

Also, the reference "KMax-Selective Deep Learning for Information Retrieval" by Zhang et al., cited by ChatGPT, could not be found. Interestingly, on the website of the SIGIR 2018 conference (https://sigir.org/sigir2018/accepted-papers/), there are several authors with the surname Zhang, as well as a paper with "Max-k" in the title. So, it seems that ChatGPT, being a generative language model (and consequently creative too!), "invented" this paper.

5. Conclusions

This work evaluated BM25 baseline algorithm for the CISI dataset for information retrieval. Two libraries were compared against each other (Rank-BM25 and Pyserini), leading to similar results in MAP metric, with a slight advantage to Pyserini.

We extensively used ChatGPT during development. It showed excellent results in general – a good “memorization” of dialogue status/context and near human performance. However, this experiment also showed that we should take ChatGPT responses with critical eyes. It is capable of occasionally giving wrong or contradictory answers to the same question. Finally, it may “lie” by “creating” fake bibliographic references and facts (e.g.: reference to an inexistent academic paper). Specifically, the reference to an inexistent paper may be a strong indicator for the potential to the creation of fake news by the tool.

Bibliography

[1] K. Spärck Jones, S. Walker and S. E. Robertson, "A probabilistic model of information retrieval: Development and comparative experiments," 2000.

[2] A. Trotman, A. Puurula and B. Burgess, "Improvements to BM25 and Language Models Examined," in *ADCS '14: Proceedings of the 2014 Australasian Document Computing Symposium*, 2014, pp. 58-65.

[3] University of Glasgow, "CISI Collection," University of Glasgow, [Online]. Available: https://ir.dcs.gla.ac.uk/resources/test_collections/cisi/. [Accessed 22 February 2023].

[4] M. Monteiro, "Queries to ChatGPT," 22 February 2023. [Online]. Available: https://github.com/monilouise/cisi_bm25/blob/main/Queries_ChatGPT.ipynb. [Accessed 22 February 2023].

[5] D. Brown, "Rank-BM25: A two line search engine," 2022. [Online]. Available: https://github.com/dorianbrown/rank_bm25. [Accessed 22 February 2023].

[6] Castorini, "Pyserini," February 2023. [Online]. Available: https://github.com/castorini/pyserini. [Accessed 22 February 2023].

[7] M. Monteiro, "Rank-BM25 Library," 22 February 2023. [Online]. Available: https://github.com/monilouise/cisi_bm25/blob/main/CISI_BM25_Rank_BM25.ipynb. [Accessed 22 February 2023].

[8] M. Monteiro, "Pyserini Library," 22 February 2023. [Online]. Available: https://github.com/monilouise/cisi_bm25/blob/main/CISI_BM25_Pyserini.ipynb. [Accessed 22 February 2023].

[9] C. D. Manning, P. Raghavan and H. Schütze, in *An Introduction to Information Retrieval*, Cambridge University Press, 2009, p. 233.

[10] M. Monteiro, "Building a Simple Information Retrieval System using BM25 and GPT-3 and evaluated in the CISI collection," 22 February 2023. [Online]. Available: https://github.com/monilouise/cisi_bm25. [Accessed 22 February 2023].