MS Marco Document re-ranking

Team-002

Geoffrey Brenne
270421@uis.no

Célian Debethune
270412@uis.no

Paul Duffaut
270399@uis.no

ABSTRACT

Your abstract here ...

KEYWORDS

information retrieval, machine learning

1 INTRODUCTION

Today, when we are curious about something and we are looking for an answer to our question, we use a search engine. This one will return by order of relevance, the various sites in connection with our question. In our case, we will keep this principle but use it a little differently. Indeed, given a set of documents as well as a set of words placed in a query, we try to determine the ranking of the 1000 most relevant documents according to our query. Thus, in order to obtain these 1000 documents, our method consists in using the BM25 algorithm to associate a score for each document. BM25 is an algorithm that scans a set of documents and returns a score according to the frequency of the terms present in this document.

. **BM25 Formula:**

$$score(d,q) = \sum_{t \in q} \frac{c_{t,d} \times (1+k_1)}{c_{t,d} + k_1(1 - b + b\frac{|d|}{avgdl})} \times idf_t$$

Parameters:

- k_1: calibrating term frequency scaling
- b: document length normalization

In order to obtain an order on these documents with respect to a set of terms, this method seems perfectly adequate in view of the various reports already established. For example, Ivan Sekulic[3] or even Liana Ermakova[2] also rely on BM25 to obtain a ranking on their documents. The ease of implementation of this method according to its very convincing results makes it a suitable method that we have chosen to use. In addition, other studies have been done on other models that are more expensive or more difficult to implement, such as that of Simon Jaillet[1], who uses the SYGMART method.

2 PROBLEM STATEMENT

As explained in the introduction, the problem consists in ranking the documents of a database. We therefore want the documents at the top of the ranking to be the most relevant for the user. Our work focuses mainly on ranking and then reranking a set of documents according to a given query. Thus, our main program receives as input a query in string form and produces a text file containing the ranking of the 1000 most relevant documents according to our model. This document is of the form:

qid	Q0	docno	rank	score	tag
1	Q0	nhslo3844_12_012186	1	1.733152	mySystem
1	Q0	nhslo1393_12_003292	2	1.725810	mySystem
1	Q0	nhslo3844_12_002212	3	1.725227	mySystem
1	Q0	nhslo3844_12_012182	4	1.725227	mySystem
1	Q0	nhslo1393_12_003296	5	1.713744	mySystem

Table 1: Example of results in TrecRun format (from Trec-Tools documentation)

The doc_id (or docno) are the id of the documents in MS-MARCO: a large dataset containing queries and documents in text form. This dataset allow the creation and training of ranking systems, based or not on machine learning. As said before, our work will not focus on the preprocessing of the documents and on the creation of the evaluation methods of our model, requiring a lot of computing power, energy and time:

- For the preprocess, we use the Pyterrier module which contains an inverted index of the documents in the MS-MARCO dataset that have been preprocessed. We also use this module to obtain information about the dataset such as the average size of the documents, necessary for the BM-25 function.
- We will also use the BM-25 algorithm of the Pyterrier module for the production of the top 1000 after having made sure that our version developed from our knowledge obtains the same results. We thus make sure that we have understood the algorithm while having optimal performance.
- To evaluate our model, we use the Trec-Tools module, which allows us to evaluate rankings according to several criteria such as Precision@k, Recall@k, NDCG, which allows us to know how our model performs.

Our model consists of a first ranking (baseline method) which retrieves the first 1000 documents and a reranking (advanced method), a more expensive method but allowing to obtain better results. The reranking of the documents is based on the model (Doing After ....)

3 BASELINE METHOD

. The baseline method retrieves the 1000 most relevant documents according to the BM-25 model score from the index provided by pyterier. This model uses often used statistics such as the frequency of appearance of a term in a document and the inverse document frequency (idf) which allows to modulate the action of the most common words and thus probably the least discriminating in terms of relevance of the document with respect to a query. As indicated on the score formula given in the introduction, BM-25 is based on the terms present in the query to calculate the score of the latter in relation to a document. The more frequently a document contains the terms of the query, the better its score will be. Similarly, if a document is rarely used, it will have a high idf, and documents containing this term will have a higher score. b and k1 are called smoothing parameters, commonly k=1.2 and b=0.75. This algorithm is based on the assumption that a document containing the terms of the query (and thus similar to this query) is probably a document containing the answer to this question, or at least deals with the same subject and is thus able to provide some information to the user. We have implemented this algorithm using the inverted index of pyterrier.

Our program generates the ranking of the 1000 runs with the best scores in TrecRun format, and then gives this document and the qrels in TrecQrel format to the Trec Tools package, allowing us to obtain different evaluations such as P@k, MRR, MAP... Two versions of the program exist: the first one is a python implementation of our part of BM-25 using the classic formula (3min per query to get the top 1000 on our PCs). The second one uses directly the BM-25 algorithm of the pyterrier module, much faster (10 seconds for the 200 queries of the test dataset). After having made sure that we obtained the same results with these two versions, we used the pyterrier version to evaluate the baseline method on the dataset. We performed the measurements on the test dataset of the TREC-2019 reranking competition.

We can see that the results are very close to those obtained by the MS Marco Team who implemented the baseline method. The slight differences are probably due to the use of a different stemmer or a different list of stopwords. These results ensure that we have a good base, however they could be greatly improved by using a second method, by reranking.

The raw performance of the baseline method BM-25 is more than once out of three very low, and therefore making statistics on our results was relatively difficult, as the ranking regularly contains no relevant documents. Nevertheless we obtain results of the same order of magnitude as those obtained by other teams and are therefore confident about the implementation of the algorithm, as well as the functioning and the good use of the pyterrier and Trec Tools packages. We can now try to improve the performance of our program by implementing an advanced method that will rerank the documents obtained by the baseline method.

4 ADVANCED METHOD

Explain what you are taking as your advanced method(s), as well as why this is a promising attempt to outperform the baseline method, and why you are making specific implementation choices.

5 RESULTS

BM-25 on test dataset: relevent docs: 4102 retrieved docs: 193245 retrieved relevant docs: 2831

Table 2: BM-25 evaluation on test dataset

P@100	0.0702
MRR@100	0.1709166666666667
NDCG@100	0.10481228686418696
MAP@100	0.06249871459335219

The MRR obtained seems correct because it is very close to the one obtained by the MS MARCO team. However we are not sure if the other measurements are correct. Indeed they seem relatively low. This is probably due to an inconsistency in the way queries are evaluated with qrels. Some queries will not appear in the qrels and the results change drastically if we take them into account or not. For example, P@100 varies from 0.0702 to 0.3265.

6 DISCUSSION AND CONCLUSIONS

Summarize and discuss different challenges you faced and how you solved those. Include interpretations of the key facts and trends you observed and pointed out in the Results section. Which method performed best, and why? Speculate: What could you have done differently, and what consequences would that have had?

REFERENCES

[1] Jacques Chauché, Violaine Prince, Simon Jaillet, and Maguelonne Teisseire. 2003. Classification automatique de textes à partir de leur analyse syntaxico-sémantique. In *Actes de la 10ème conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*. 55–64.

[2] Liana Ermakova and Josiane Mothe. 2016. Document re-ranking based on topic-comment structure. In *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 1–10.

[3] Ivan Sekulić, Amir Soleimani, Mohammad Aliannejadi, and Fabio Crestani. 2020. Longformer for MS MARCO document re-ranking task. *arXiv preprint arXiv:2009.09392* (2020).

A DIVISION OF WORK DURING THE PROJECT