Università degli Studi di Milano-Bicocca

Information Retrieval

2425-2-F1801Q110

Creating a search engine for Medical Information Retrieval

Authors:
Matteo Selvaggi: 866250
Marco Midali: 868082
Kristian Myklebust: 925327

1 Introduction

Medical information retrieval (MIR) is a specialized branch of information retrieval that focuses on extracting medical data from archives through advanced retrieval models and processing pipelines. Health-related content is among the most frequently searched topics on the internet, making MIR a crucial branch of information retrieval.

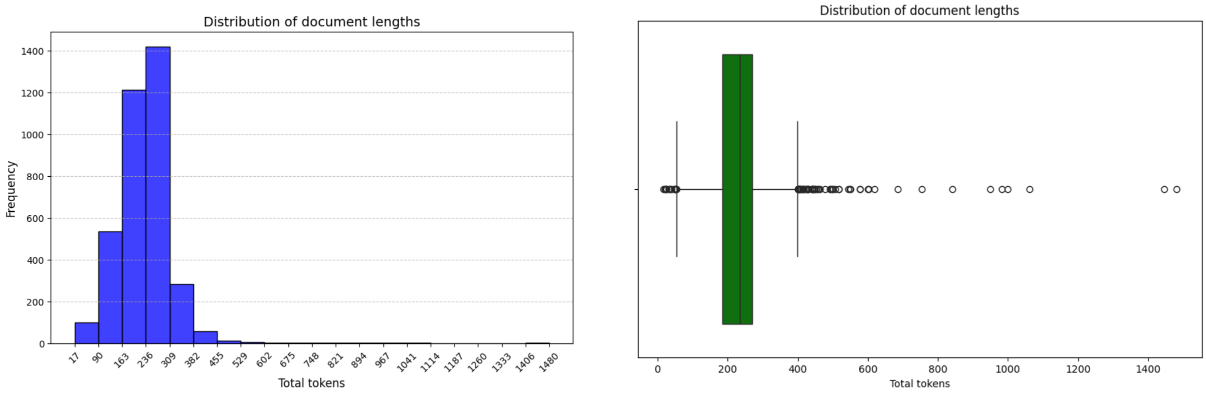
In this project, various retrieval models, pipelines and indexing strategies were explored in order to identify the optimal combination that had the best results on the **NFCorpus** dataset. NFCorpus is a full-text retrieval dataset for MIR, designed to assess the effectiveness of retrieval models and pipelines in this domain.

Furthermore, this project examined how a **neural model**, specifically **BERT**, impacted the results of the previously built retrieval models and pipelines when used for re-ranking, to determine if it had improved their performances.

2 Dataset Analysis

This paragraph analyzes the three components that make up the NFCorpus dataset: **medical documents**, **queries**, and **relevance judgments (qrels)** for each query.

2.1 Medical Documents Analysis



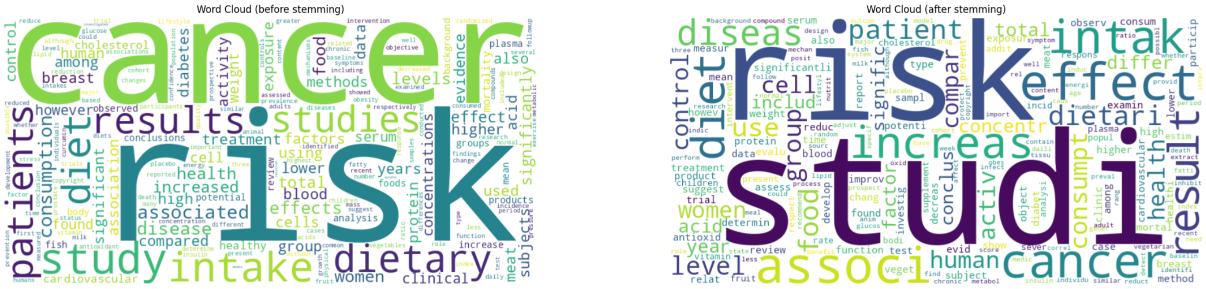
(a) Bar Chart.

(b) Box Plot.

Figure 1: Document length distribution using two different plots.

These two graphs illustrate the **length distribution** of the medical documents in terms of tokens. A simple form of tokenization was applied to the text of the documents, in particular the space-based tokenization. The bar chart shows that the majority of documents contain a total number of tokens ranging from 163 to 309, while the box plot makes it possible to visualize the **outliers** of the length distribution. Some documents contain very long texts with more than 1400 tokens, while others are extremely short, with a total length of only about a dozen tokens.

Now, let's examine the **term distribution** across the various documents.



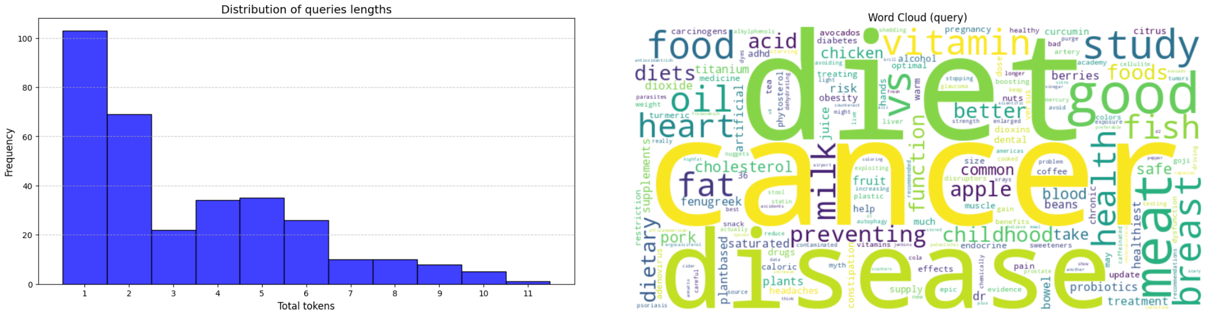
(a) Before Stemming.

(b) After Stemming.

Figure 2: Term distribution before and after stemming.

To extract the terms present in the documents, the text was normalized by converting all words to lowercase and removing stop words as well as words shorter than four characters. The first plot shows the most common terms between the various documents before Porter stemming. The terms *cancer* and *risk* appear most frequently in the corpus. After applying stemming, the term *studi* replaces cancer as one of the most frequent terms.

2.2 Queries Analysis



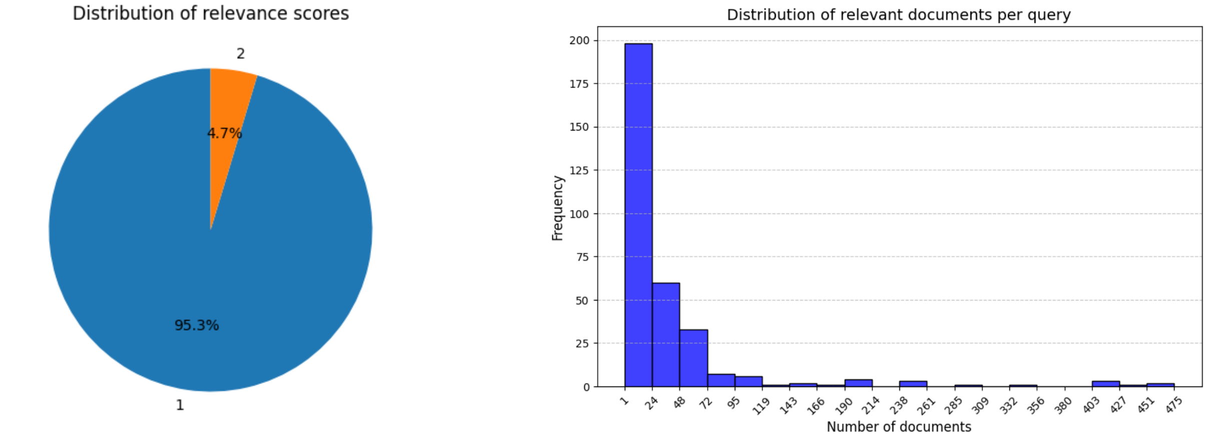
(a) Length Distribution.

(b) Term Distribution.

Figure 3: Query length and term distributions

The first plot illustrates the **length distribution** of the queries in terms of tokens, based on space-based tokenization. The majority of queries are composed by one or two tokens, while very few queries are well articulated. The second plot displays the **most common terms** found in the queries. People seem to often ask questions about topics related to diet, cancer, and diseases.

2.3 Relevance Judgements (Qrels) Analysis



(a) Relevance score distribution.

(b) Relevant documents per query distribution.

Figure 4: Relevance score and relevant documents per query distributions

The pie chart illustrates the **distribution of relevance scores** across the different qrels. It is evident that most documents have been assigned a relevance score of one, while only a few have a score of two, indicating that a small number of documents are considered perfectly relevant to a query. The second plot shows the **distribution of relevant documents** per query, with most queries having between 1 and 24 relevant documents, while some queries have over 400 relevant documents associated with them.

In this project, the **number of relevance judgments assigned to each document** was also analyzed. On average, a document has 4 relevance judgments assigned to it, with the extremes being a maximum of 37 relevance judgments and a minimum of 1.

3 The Methodological Approach

The first step in building a successful search engine is selecting the right indexing strategy. In this project, three different indexing strategies were considered. The first **indexed both the titles and texts** of the collection using the **Porter stemmer**, while the second applied the **weak Porter stemmer** to the same fields. The third strategy **indexed only the titles without applying any form of stemming** to preserve the critical information carried by each word in a title.

After selecting the indexing strategies, the retrieval models were chosen. The two retrieval models used in this project were:

1. **Tf-idf**: it measures the importance of a term within a document relative to its occurrence across a collection of documents. It assigns higher scores to terms that are frequent in a document but rare across others, helping to rank documents based on relevance.
2. **BM25**: improves upon TF-IDF by using non-linear scaling for term frequency, giving diminishing returns for repeated words. It also normalizes for document length, preventing longer documents from being unfairly ranked higher.

Two retrieval pipelines were created by combining the two retrieval models: **BM25 → TF-IDF** and **TF-IDF → BM25**. In these pipelines, the second model re-ranks only the top 100 documents retrieved by the first model. **Various combinations** of indexing strategies and retrieval pipelines/models were tested to identify the most effective approach, using **MAP**, **P@10**, **Recall@25**, and **NDCG** as evaluation metrics to assess the rank produced by the different combinations for the different queries.

Afterward, each combination underwent **neural re-ranking** using **BERT** as neural model to assess whether it produced better performances. BERT is a neural model that enhances search result ranking by understanding the context and semantics of queries and documents. Unlike TF-IDF and BM25, which rely on keyword matching, BERT uses a deep neural network to analyze words bidirectionally, capturing contextual meaning.

4 The Combinations and their Results

4.1 Porter stemmer (title + text)

Table 1: Before Neural Re-Rank

Model/Pipeline	map	P@10	recall@25	ndcg
Tf-Idf	0.339	0.231	0.185	0.299
BM25	0.340	0.233	0.185	0.299
Tf-Idf > BM25	0.396	0.233	0.186	0.259
BM25 > Tf-Idf	0.395	0.231	0.185	0.259

Table 2: After Neural Re-Rank

Model/Pipeline	map	P@10	recall@25	ndcg
Tf-Idf	0.176	0.080	0.085	0.185
BM25	0.177	0.081	0.082	0.184
Tf-Idf > BM25	0.209	0.097	0.100	0.167
BM25 > Tf-Idf	0.209	0.098	0.099	0.167

4.2 Weak porter stemmer (title + text)

Table 3: Before Neural Re-Rank

Model/Pipeline	map	P@10	recall@25	ndcg
Tf-Idf > BM25	0.397	0.233	0.184	0.255
BM25 > Tf-Idf	0.395	0.232	0.185	0.257

Table 4: After Neural Re-Rank

Model/Pipeline	map	P@10	recall@25	ndcg
Tf-Idf > BM25	0.207	0.098	0.101	0.163
BM25 > Tf-Idf	0.206	0.097	0.101	0.164

4.3 No stemmer (title)

Table 5: Before Neural Re-Rank

Model/Pipeline	map	P@10	recall@25	ndcg
Tf-Idf > BM25	0.338	0.143	0.124	0.165
BM25 > Tf-Idf	0.341	0.143	0.124	0.166

Table 6: After Neural Re-Rank

Model/Pipeline	map	P@10	recall@25	ndcg
Tf-Idf > BM25	0.238	0.091	0.095	0.124
BM25 > Tf-Idf	0.238	0.091	0.095	0.124

4.4 Best combination

Before applying Neural Re-Ranking, the best results came from the retrieval pipelines using Porter or Weak Porter stemming while indexing both the title and text. However, after applying Neural Re-Ranking with BERT, the top-performing combinations were the pipelines that indexed only the title without stemming. Neural Re-Ranking was not effective, as it failed to improve and even worsened the original retrieval results.

5 Best and Worst Queries

Table 7: Worst Queries

Query	Score	#Relevant
weight gain	0.0014	15
uterine health	0.0016	13
lyme disease	0.0020	6
poisonous plants	0.0035	4
genetic manipulation	0.0039	4

Table 8: Best Queries

Query	Score	#Relevant
neurocysticercosis	0.9091	16
adenovirus	0.8473	15
BMAA	0.8357	25
carrageenan	0.7967	9
bronchiolitis obliterans	0.7833	57

These tables show the worst (on the left) and best queries (on the right) based on the mean of all the evaluation metrics. The poorly performing queries share common issues, e.g. queries like "weight gain" and "genetic manipulation" are too **ambiguous** and **cover different topics**, leading to less relevant results. Another issue is the **lack of relevant documents**, for instance query such as: "poisonous plants" and "uterine health" do not have sufficient coverage. Last but not least, some scientific terms such as "lyme disease" may occur with another name.

Table 9: Best Queries for P@10

Query	P@10
pesticides	1.0
neurocysticercosis	1.0
Infectobesity Adenovirus 36 and Childhood Obesity	1.0
BMAA	0.975
nuts	0.975

The queries in the table 9 seem to be related to topic such as: health, nutrition and environmental factors that affect human well-being. For instance, queries like: "pesticides" and "BMAA", relate to toxic substances that can have health implications. "neurocysticercosis" is a neurological disease caused by parasitic infection. The third query links viruses to obesity. The last: "nuts", could be related to nutrition and their impact on health. Therefore, the queries are related to each other, but each one focuses on different subdomains.

Precision@10 is crucial in this task for different reasons. One reason is the reliability of retrieved information, since high precision ensures that the top 10 results are relevant, reducing the risk of retrieving misleading or non-relevant information. It is also important for efficiency, since high precision means that the researchers do not need to sift through irrelevant data, saving time and improving productivity. In the end, precision also has an impact on decision making because it ensures that conclusions are drawn from the most accurate and relevant sources.

6 Conclusion

This report explored various combinations of retrieval models/pipelines and indexing strategies on NFCorpus, both before and after Neural Re-Ranking. The best results came from pipelines using Porter or Weak Porter stemming while indexing both the titles and texst, without re-ranking, suggesting its ineffectiveness. Additionally, the report analyzed the best and worst-performing queries, highlighting challenges related to ambiguity and underrepresentation in the dataset.