Biasco Anna Marika [865873]
Novacco Alessia [918550]

Search Engine for Medical Information Retrieval

Introduction

The goal of the project is to develop a search engine for medical information retrieval capable of processing natural language queries. Using the NFCorpus dataset, the system retrieves relevant medical documents in response to health-related queries. The system is evaluated given a set of qrels, relevance judgments linking queries to relevant documents.
The analysis includes a detailed examination of query performance and the impact of document preprocessing.

Pre-processing and dataset analysis

Pre-processing

Before beginning the analysis of the dataset, we performed pre-processing to standardize and clean the data, ensuring it was suitable for analysis.
To prepare the NFCorpus dataset for analysis and retrieval tasks, the following steps were implemented:

1. Text Standardization: Converted all text to lowercase for uniformity.
2. Data Cleaning:
    - Removed HTML tags using BeautifulSoup.
    - Stripped URLs using regex.
    - Removed numeric characters to focus on textual content.
    - Removed stopwords, punctuation, numbers, spaces, and repeated characters, keeping only the roots of meaningful words.
3. Tokenization: Split the text into individual tokens with SpaCy.
4. Stopword Removal and Stemming:
    - Removed common stopwords using SpaCy's English stopword list.
    - Applied Porter stemming to reduce words to their root forms.

This pre-processing pipeline ensured clean and standardized data.

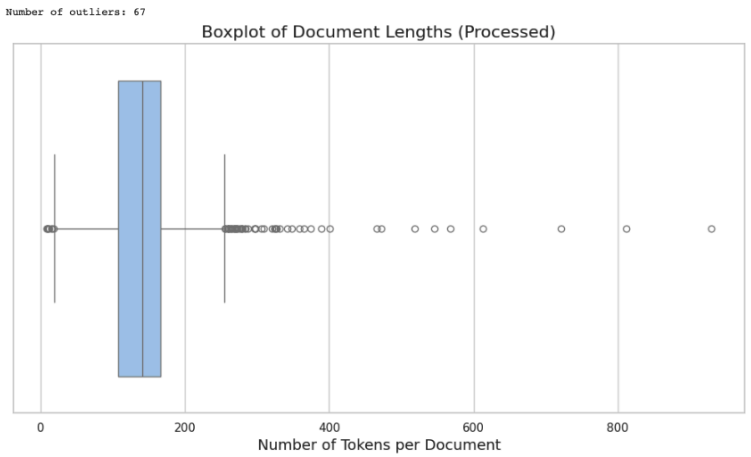
Analysis of the dataset

The NFCorpus dataset consists of:

- Queries: 3,244 health-related questions written in non-technical English.
- Documents: 9,964 medical documents with complex, terminology-heavy language.
- Relevance Judgments: 169,756 automatically extracted qrels to support the evaluation of retrieval systems.

Analysis' steps:

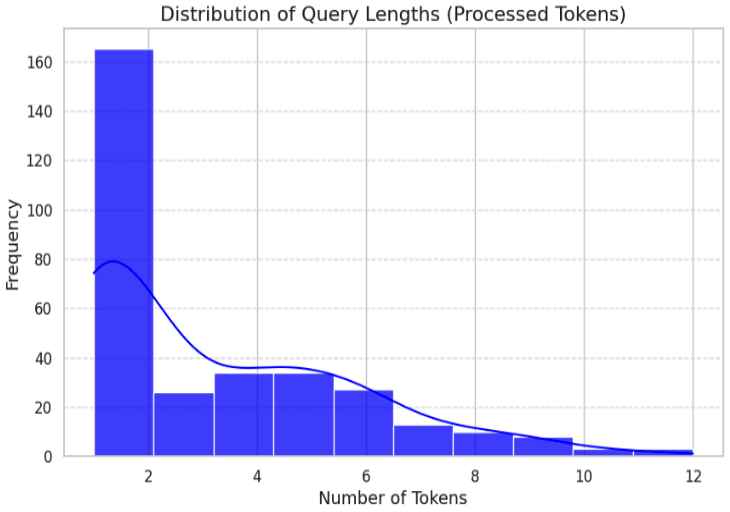
1. Document Length Analysis:
    - Calculated the distribution of document lengths in tokens.
    - Computed document lengths before and after preprocessing.
    - Identified outliers, including extremely short and long documents, using statistical techniques and visualizations like histograms.
2. Query Length Analysis:
    - Measured the average query length and analyzed its distribution.
    - Examined the prevalence of short, ambiguous queries, which can complicate retrieval tasks.
3. Relevance Judgments Analysis:
    - Assessed the number of relevant documents per query using the qrels data.
    - Evaluated the distribution of relevance labels to identify underrepresented queries and terms.
4. Visualizations:
    - Generated word clouds to highlight frequent terms in documents and queries.
    - Plotted histograms to show document and query length distributions, focusing on the presence of possible outliers.
    - Created heatmaps to illustrate term frequency and relevance score distributions.
    - We applied these visualizations to all the documents and then to the most relevant ones in order to see if the distributions and word clouds would change.



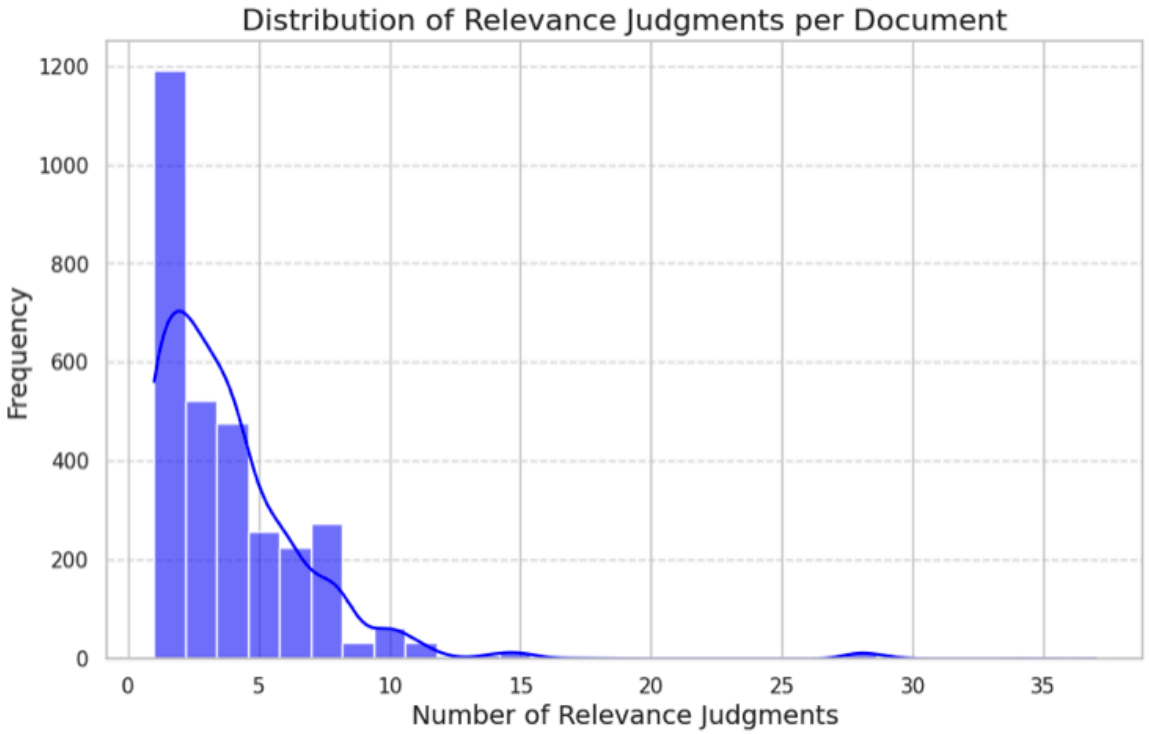
Analysis

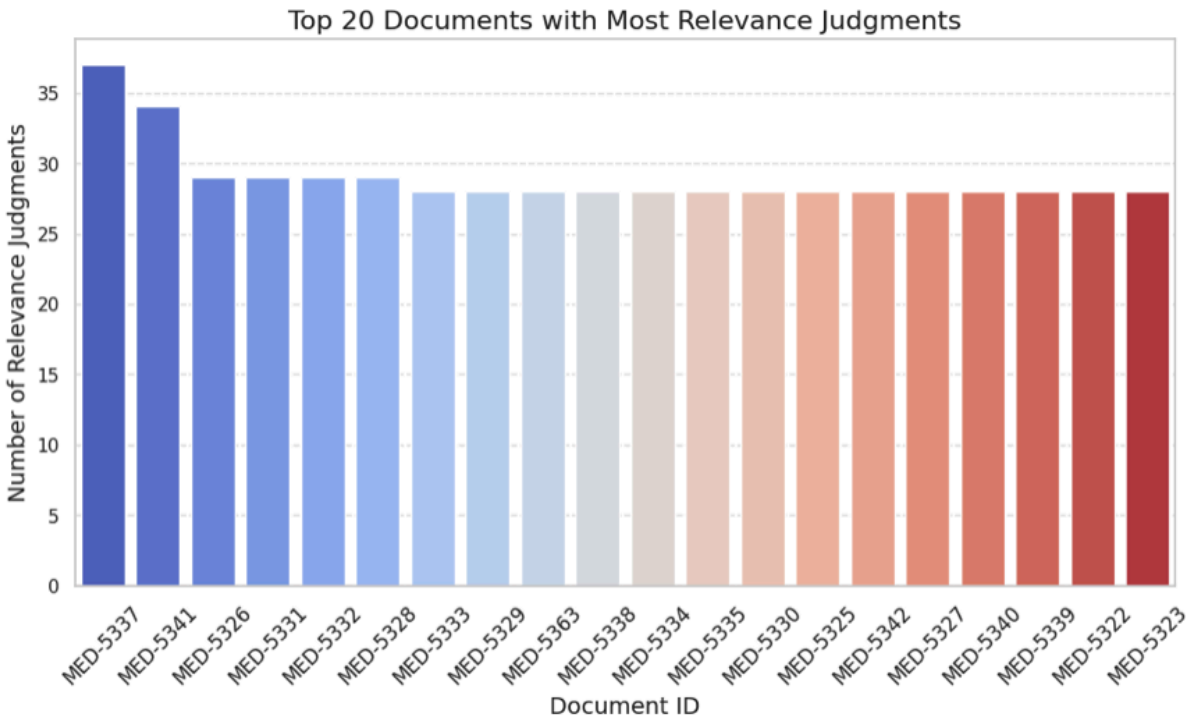
1. Document Lengths:
    - Most documents fall within a standard range of token counts, but outliers (shorter or longer documents) could skew retrieval performance.
    - Longer documents may contain more relevant information but can also introduce noise if not properly indexed.
2. Query Lengths:
    - Short queries, while common, often lack contextual information, leading to ambiguous interpretations and potential retrieval errors.

- Queries with more tokens tend to produce better matches due to added specificity.
- Queries are often concise, posing potential challenges for effective retrieval.



3. Qrels:
   - Some queries are underrepresented in terms of relevant documents, which could hinder the evaluation of retrieval performance for those queries.
   - The distribution of qrels indicates that certain queries are well-supported by the dataset, while others lack sufficient relevance data, highlighting areas for improvement.





Retrieval pipelines

Indexing

The first step in building the retrieval pipeline involved indexing the dataset. Two indexing configurations were implemented:

1. Full-text Corpus: This included both the titles and abstracts of the documents, offering richer contextual information for retrieval.
2. Title-only Corpus: This configuration limited the indexed data to document titles.

PyTerrier was employed for indexing, which allowed efficient data organization and retrieval. Index statistics such as the number of tokens and postings were recorded to analyze the impact of different configurations.

During indexing, stemming was applied to reduce words to their root forms, minimizing vocabulary size while maintaining semantic relevance. Different algorithms for stemming were applied in order to evaluate the best one. Given that NFCorpus is a medical dataset with documents in English, we focused on stemmers that can handle English effectively. We evaluated the following stemmers:

- porter: The most widely used English stemmer, suitable for reducing words to their linguistic roots. Commonly used in general text retrieval tasks.
- weakporter: A less aggressive version of the Porter stemmer, which may retain more medically significant terms compared to the standard Porter.
- none: Testing with no stemming is essential for a baseline to observe the effect of stemming.

The Weak Porter Stemmer was selected as the optimal choice after evaluating other stemming strategies. The Weak Porter stemmer performs less aggressive stemming compared to the standard Porter stemmer. This allows it to preserve more distinctions between different word forms while still significantly reducing the vocabulary size compared to using no stemmer. This approach provides a well-balanced trade-off between precision and recall, making it particularly suitable for retrieval tasks where maintaining linguistic nuance is important without compromising too much on query-document alignment.

Stemmer	Number of Documents	Number of Terms	Number of Postings	Number of Fields	Number of Tokens
No Stemmer	3633	24109	312553	1	473003
Porter	3633	17034	292012	1	473003
Weakporter	3633	19889	299937	1	473003

Retrieval Models

Two retrieval models were implemented and evaluated:

1. **TF-IDF (Term Frequency-Inverse Document Frequency)**: This model assigns higher weights to terms that are frequent in a document but rare across the entire corpus. It is effective for distinguishing unique terms in shorter text segments like titles.
2. **BM25 (Best Matching 25)**: A probabilistic ranking model that enhances relevance scoring by considering term frequency saturation and document length normalization. BM25 excels in ranking longer documents where term distributions vary significantly.

The retrieval pipelines were constructed using these models, enabling evaluation of their performance under different indexing configurations. Queries were processed similarly to the corpus, ensuring consistency in tokenization and stemming.

For each pipeline, the following steps were conducted:

1. Queries were input into the retrieval model.
2. Relevance scores for each document were computed based on the query.
3. Top-ranked documents were returned, forming the basis for performance evaluation.

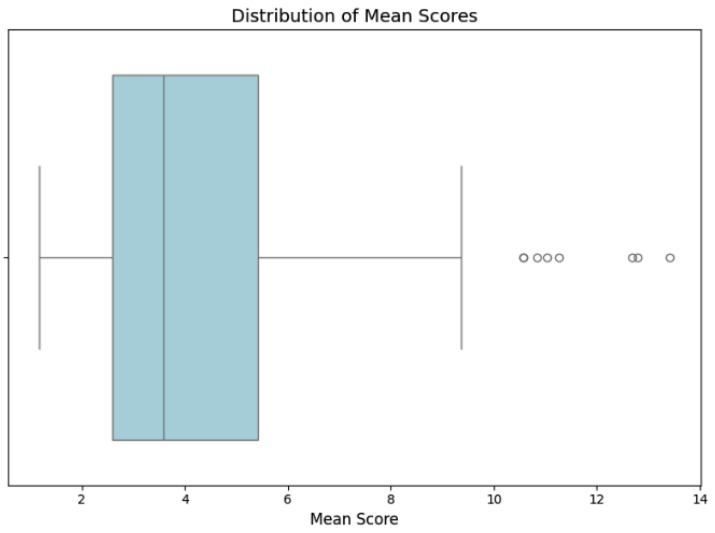
Results and Evaluation

Full-text Corpus

BM25 outperformed TF-IDF across all evaluation metrics:

- MAP (Mean Average Precision)
- NDCG (Normalized Discounted Cumulative Gain)
- P@10 and Recall@10: Higher proportions of relevant documents retrieved in the top results.

Model Name	MAP	NDCG	P@10	Recall@10
TF-IDF	0.143	0.289	0.226	0.145
BM25	0.143	0.289	0.225	0.144



Title-only Corpus

Both models exhibited reduced performance compared to the full-text corpus due to the lack of contextual information. Minimal differences were observed between TF-IDF and BM25 metrics, with both struggling to achieve high precision.

Model name	MAP	NDCG	P@10	recall@10
TF-IDF	0.096	0.198	0.164	0.106
BM25	0.096	0.197	0.164	0.106

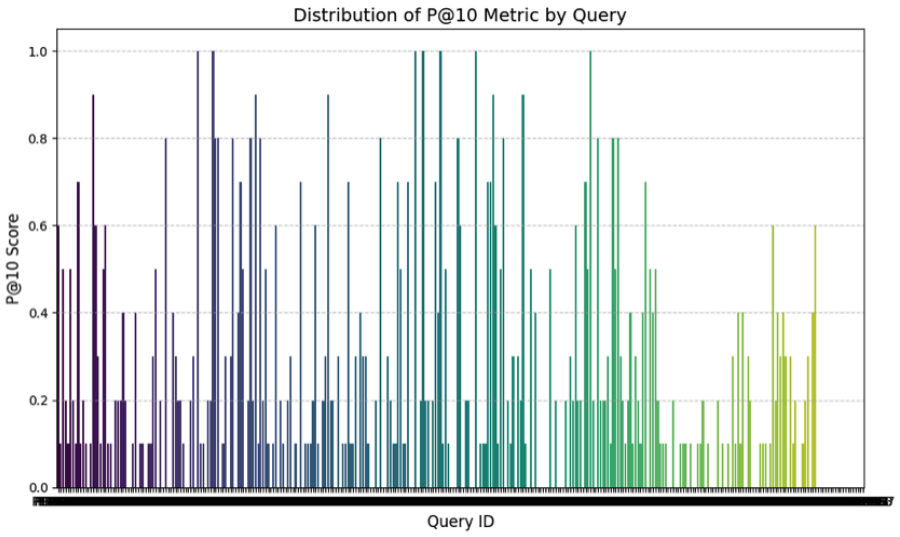
Query Performance Analysis

Poor-performing Queries

- Queries with low MAP (< 0.001) were analyzed:
    - Some queries were ambiguous or overly broad, providing insufficient context for meaningful document retrieval. Such queries often failed to align with specific documents in the corpus, like “junk food” or “dietary guidelines from dairies to berries”.
    - Some queries were highly specific with niche terms. These queries often targeted information sparsely represented in the dataset, making it challenging for the retrieval models to locate relevant documents (for example "what do you think of dr jenkins take on paleolithic diets")
    - The title-only indexing configuration exacerbated poor performance for queries requiring rich contextual information, as the limited textual content in titles could not adequately match the intent of complex queries.

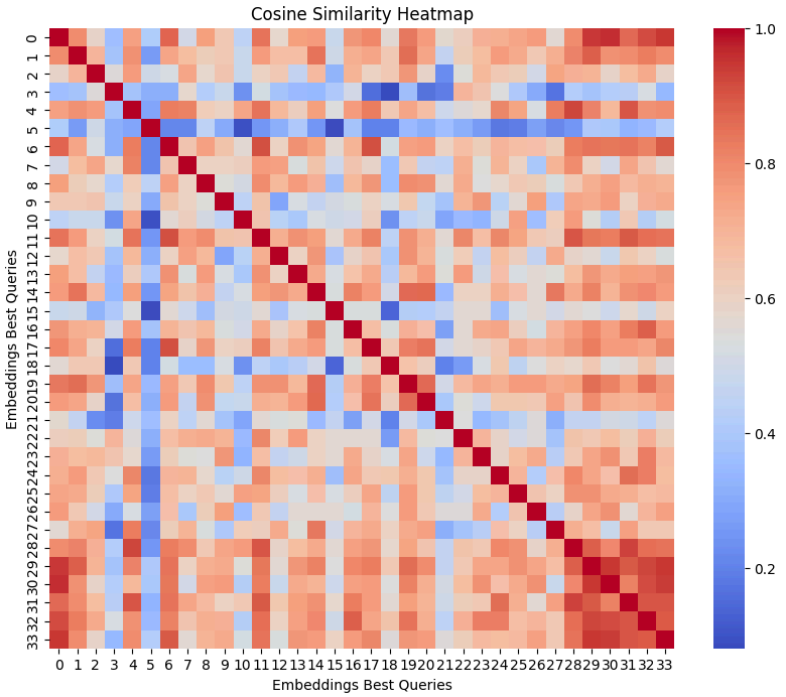
High-performing Queries

- Queries with P@10 > 0.6 exhibited high precision due to:
    - High-performing queries often contained precise terminology directly aligned with the medical themes of the NFCorpus dataset (“BPH” which means Benign Prostatic Hyperplasia P@10 =0.8).
    - Queries with a clear scope and well-defined intent were more likely to retrieve relevant documents effectively, minimizing ambiguity (“how should i take probiotics?” , P@10 =0.9).



Semantic Similarity

Word2Vec embeddings were used to compute cosine similarities between high-performing queries, identifying semantically related pairs. This method revealed clusters of semantically related queries: related queries often shared overlapping topics, such as dietary factors (e.g., "vegetarian diets" and "effect of carrageenan") or disease prevention strategies (e.g., "probiotics for immunity" and "caloric restriction effects").



Comparative Observations

- Poor-performing queries highlight the need for clear and concise query formulation, as well as the importance of sufficient contextual data in the corpus.
- High-performing queries underscore the benefits of precise terminology and detailed document indexing, particularly when abstracts were included.

Improving retrieval performance

To improve retrieval performance, we address query ambiguity by expanding the most effective retrieval pipeline, which includes both 'title' and 'abstract' ('text').
The code uses the Qwen model to generate three expanded versions of each query in a DataFrame. A tailored prompt was designed for the NFCorpus dataset, emphasizing medical information retrieval. It incorporates dataset-specific context, clarifies intent while preserving meaning.

Each of the three expanded query versions, along with the original queries, was tested on various models:

- TF-IDF and BM25: These are baseline models used for comparison.
- BM25 with RM3: This model uses the BM25 algorithm enhanced by RM3 query expansion for improved retrieval.

- BM25 with Neural Re-ranking using BERT: This model combines BM25 with a neural re-ranking approach powered by BERT for more accurate results.

The following are the results.

Results obtained with the first query expansion generated by Qwen:

Name	MAP	NDCG	P@10	Recall@10
TF-IDF	0.107	0.282	0.168	0.113
BM25	0.109	0.284	0.172	0.116
BM25 + RM3	0.121	0.318	0.186	0.126
BM25 + BERT	0.016	0.150	0.021	0.009

Results obtained with the second query expansion generated by Qwen:

Name	MAP	NDCG	P@10	Recall@10
TF-IDF	0.086	0.254	0.142	0.092
BM25	0.088	0.256	0.143	0.092
BM25 + RM3	0.100	0.284	0.158	0.107
BM25 + BERT	0.010	0.141	0.016	0.002

Results obtained with the third query expansion generated by Qwen:

Name	MAP	NDCG	P@10	Recall@10
TF-IDF	0.092	0.272	0.149	0.099
BM25	0.095	0.274	0.150	0.098
BM25 + RM3	0.113	0.304	0.170	0.106
BM25 + BERT	0.008	0.144	0.011	0.002

Results obtained with the original queries:

Name	MAP	NDCG	P@10	Recall@10
TF-IDF	0.143	0.289	0.226	0.145
BM25	0.143	0.289	0.225	0.144
BM25 + RM3	0.164	0.363	0.241	0.162
BM25 + BERT	0.063	0.181	0.084	0.062
BM25 + BioBERT	0.058	0.173	0.081	0.061

The First Extended Version outperformed the other expansions across all metrics, likely due to better alignment with the corpus's vocabulary and structure.
The Second and Third Extended Versions showed progressively worse results, indicating that the added terms became less relevant and strayed from the corpus's semantic space. This decline may stem from the Qwen model's generated phrases becoming less coherent

as the prompt required exactly three expansions per query, potentially introducing redundant or irrelevant terms.
Query expansion using Qwen likely increases recall but reduces precision, as additional terms may introduce noise. The performance of BM25 + RM3 with expanded queries showed a slight benefit, but not as significant as with the original queries, suggesting that the extended terms diluted the relevance signal.

Both **TF-IDF** and **BM25** produced nearly identical results across all metrics, especially with the original queries. This indicates that term-based models perform similarly in this domain. However, query expansion with Qwen did not notably improve these models, as they rely on lexical overlap and struggle with capturing semantic relationships in specialized medical texts.
BM25 + RM3 outperformed the baseline across all metrics. RM3 query expansion, which reweights terms based on pseudo-relevance feedback, improved NDCG and recall by introducing semantically relevant terms, better aligning with document content and improving ranking quality.
BM25 + BERT (neural re-ranking) produced surprisingly poor results. The performance drop was notable in MAP and P@10, likely because the short abstracts and titles lacked sufficient context for BERT to make effective semantic matches. Additionally, query expansion using Qwen may have introduced irrelevant or overly general terms, making it harder for BERT to determine relevance.

We conducted another test using the original queries, but instead of BERT, we used **BioBERT**—a biomedical language representation model specifically designed for biomedical text mining. However, despite being fine-tuned on medical documents, BioBERT did not improve the results. This could be due to its limited ability to generalize to the specific sub-domain of our corpus. Differences in terminology, context, or document structure (e.g., abstracts versus full papers) may have constrained its effectiveness.

As a last attempt, we tried expanding search queries with an additional technique by replacing words with their synonyms from **WordNet**, a lexical database. It begins by splitting each query into individual words. For each word, the code retrieves its synonyms from WordNet. If synonyms are found, the word is replaced with one of them; otherwise, the original word remains. The expanded query is then reconstructed by joining the words together.

Model	MAP	NDCG	P@10	Recall@10
TF-IDF	0.069	0.142	0.104	0.077
BM25	0.068	0.142	0.103	0.076
BM25 + RM3	0.086	0.207	0.117	0.086

The results of this experiment are worse compared to those obtained with the original queries, which means that it's very difficult to reformulate queries with such a specific vocabulary as medical terminology.

Conclusions

Medical abstracts contain highly specialized vocabulary, which models like BM25 and RM3 are well-suited to handle due to their effective term-weighting mechanisms. In contrast, neural models such as BERT and BioBERT tend to underperform in this context, likely because they require additional fine-tuning on domain-specific data to fully capture the nuances of the medical domain. Even though BioBERT is pre-trained on biomedical text, its performance in these experiments highlights that it may not generalize well to the specific sub-domain or structure of the corpus, such as differences in how information is presented in abstracts versus full documents.

The experiments show that original queries consistently produce better results, particularly with the BM25 + RM3 pipeline. This suggests that the original query terms are highly aligned with the content of relevant documents, maximizing precision and recall. On the other hand, extended queries, while potentially enriching the search space, often introduce noise or less relevant terms, leading to diminished performance.

Overall, BM25 combined with RM3 proves to be the most robust and effective approach for retrieving information from a corpus consisting solely of titles and abstracts.