Multilingual Document Retrieval

HAY Team : Youssef Belghmi, Amine Belghmi, Hamza Morchid
Distributed Information Systems (CS-423), Department of Computer Science, EPFL, Switzerland

***Abstract*—This project presents a multilingual asymmetric document retrieval system designed to process a corpus of over 250,000 documents in seven languages. We focused on computational efficiency by retrieving the top 10 most relevant documents per query, balancing speed and accuracy. Key design choices, method selection, and performance results are discussed, with final results submitted to Kaggle for evaluation.**

I. Introduction

Document retrieval aims to identify and rank relevant documents based on a query. In a multilingual context, this task becomes complex due to linguistic diversity. This project tackles the problem by developing a multilingual document retrieval system that processes queries across seven languages. Our approach combines the use of TF-IDF, a widely used method in text retrieval that scores documents based on term frequency and inverse document frequency, and BM25, an advanced extension of TF-IDF that refines the relevance of results through saturation and document length adjustments. We balanced performance and speed through strategic design choices, detailed in this report.

II. Preprocessing

Given the multilingual nature of the corpus, preprocessing was essential for efficient and accurate retrieval. We started by dividing the corpus into language-specific subsets, which allowed each language to be treated independently, taking into account differences in vocabulary, grammar, and morphology between languages.

Key preprocessing steps included tokenization, which involves splitting text into individual tokens (or words), facilitating more precise text analysis. Following tokenization, we converted all text to lowercase to standardize capitalization, thereby improving consistency and reducing redundancy in the data. We also removed common, uninformative words—known as stopwords—using language-specific lists. This step is crucial in text retrieval, as these frequent words could otherwise overshadow more meaningful terms, diminishing retrieval accuracy. For most languages, we applied stemming to reduce words to their base form, allowing variations of a term to be unified. Language-specific stemmers were used, except for Arabic and Korean, where stemming was excluded due to morphological complexity.

This tailored preprocessing pipeline ensured that documents were thoroughly optimized for accurate, consistent, and efficient retrieval across all seven languages in the corpus. By addressing unique linguistic characteristics, the pipeline contributed to a more nuanced and effective document retrieval system that accommodates diverse language structures and vocabulary distributions.

III. TF-IDF Approach

Our initial document retrieval system is based on the Term Frequency-Inverse Document Frequency (TF-IDF) model, which ranks documents by assigning weights to terms according to their significance within individual documents and their rarity across the entire corpus. TF-IDF is particularly effective in multilingual contexts, as it balances simplicity and adaptability to diverse linguistic structures.

In this approach, we calculate the term frequency (TF) to highlight the importance of terms within a document and use the inverse document frequency (IDF) to downweight terms that appear frequently across many documents. This combination of local and global term significance allows us to score each term based on its relevance in a specific document relative to the corpus. The TF-IDF score for each term t in a document d is given by:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) = \frac{f_{t,d}}{N_d} \times \log\left(\frac{N}{df_t + 1}\right)$$

where $TF(t,d)$ represents the term frequency of t in document d, with $f_{t,d}$ being the number of times t appears in d and N_d the total number of terms in d, and $IDF(t)$ is the inverse document frequency of t across the corpus, where N is the total number of documents and df_t is the number of documents containing term t.

Using this score, we represent each document as a vector of weighted terms, enabling effective comparison of document relevance to queries. To optimize storage and computation, we built language-specific TF-IDF matrices as sparse matrices, with rows as documents and columns as language-specific terms. This efficient structure, storing only non-zero values, reduces memory usage while preserving linguistic nuances, enhancing retrieval accuracy. These matrices provide quick access to document vectors, allowing for fast similarity computations and ensuring scalability in our multilingual retrieval system.

IV. BM25 Approach

To enhance the accuracy of our document retrieval system, we extended our initial TF-IDF model by implementing BM25, an advanced weighting scheme that adjusts term frequencies and normalizes document length. BM25 refines the

scoring by addressing two key limitations of TF-IDF: term saturation and document length bias, making it especially effective in large, multilingual corpora.

BM25 incorporates a saturation function to limit the impact of highly frequent terms within documents, preventing them from disproportionately influencing relevance scores. Additionally, it adjusts for document length, ensuring that longer documents do not inherently receive higher scores. The BM25 score for each term t in a document d is calculated as follows:

$$BM25(t,d) = IDF(t) \times \frac{f_{t,d} \cdot (k_1 + 1)}{f_{t,d} + k_1 \cdot \left(1 - b + b \cdot \frac{N_d}{\text{avg}(N)}\right)}$$

where $f_{t,d}$ is the frequency of term t in document d, N_d is the length of document d, avg(N) is the average document length in the corpus, k_1 controls term frequency saturation, and b adjusts the influence of document length.

We constructed language-specific BM25 matrices using this formula, where each document is represented as a vector of weighted terms. Similar to our TF-IDF approach, these matrices are stored as sparse matrices, allowing efficient storage and computation. By leveraging BM25's adjustments, our system should achieves a higher level of accuracy in document retrieval, particularly in balancing relevance across documents of varying lengths and term distributions. This refined approach ensures that our multilingual document retrieval system is both scalable and precise across diverse languages.

V. Retrival Process

For each query in the test set, the system identifies the top 10 most relevant documents by calculating the similarity between the query and the documents in the corpus. Each query undergoes the same preprocessing steps as the documents, ensuring consistency in tokenization, lowercasing, stopword removal, and, where applicable, stemming. This alignment guarantees that terms in the query correspond to those in the document representations.

After preprocessing, the query is transformed into a vector representation using the vocabulary and weightings (TF-IDF or BM25) specific to the language of the query. This vectorized form enables direct comparison with the precomputed document vectors.

The similarity between the query vector and each document vector is then calculated, typically using cosine similarity, which provides a normalized measure of relevance. The documents are ranked based on their similarity scores, and the top 10 most relevant documents are selected as results.

VI. Results

The evaluation of our document retrieval system was performed using the Recall@10 metric, which measures the proportion of queries for which the relevant document of the query in the dev dataset appears among the first 10 documents retrieved. This metric is particularly useful for evaluating the effectiveness of document retrieval systems.

A. Results with TF-IDF

Using the TF-IDF approach, our system achieved an overall Recall@10 score of around 0.592. This indicates that for 59.2% of queries, the relevant document was successfully retrieved from the top 10 results. Although TF-IDF provided a solid baseline, the score suggests that there is room for improvement, possibly due to the simplicity of the weighting not fully capturing the importance of terms.

B. Results with BM25

To optimize the performance of the BM25 model, we conducted hyperparameter tuning on k_1, which controls term frequency saturation. The results of our tuning process are presented in the graph below, showing the impact of different k_1 values on the overall Recall@10 score.

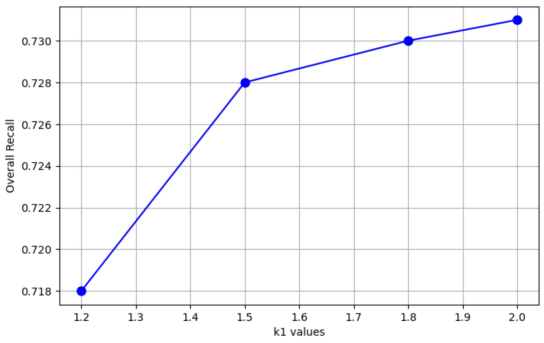

Figure 1. BM25 Recall@10 by k1 Value

While a k_1 value of 2.0 yielded the highest recall score, it may cause rapid saturation, reducing the influence of very frequent terms in future tests. To balance recall performance with term frequency saturation, we chose a slightly lower k_1 value of 1.8. This setting maintains near-optimal recall, close to that achieved with $k_1 = 2.0$, while allowing more linear term frequency weighting, potentially enhancing model robustness across varied query types in future evaluations.

The results of our evaluation showed a varied performance across the seven languages in our corpus. Specifically:

Language	Recall@10	Language	Recall@10
English	0.605	Italian	0.760
French	0.870	Arabic	0.685
German	0.635	Korean	0.645
Spanish	0.915	Overall	0.730

Figure 2. BM25 Recall@10 per language

We observed variation between languages, with the highest scores for Spanish (0.915) and French (0.870), likely

due to structural and lexical similarities in these Romance languages. Conversely, English had the lowest recall (0.605), possibly due to the large volume of documents, which made effective term weighting more challenging and may have impacted retrieval accuracy.

Our system achieved an overall Recall@10 of 0.730 across all languages, reflecting consistent retrieval capability. When we submitted the top 10 retrieved documents per query in the test dataset on Kaggle, we obtained an overall Recall@10 of 0.675. This result aligns with our dev dataset findings, indicating similar model performance across both datasets. This consistency reinforces the generalization and robustness of our approach, confirming its effectiveness for multilingual document retrieval.

VII. Other Methods

One approach we explored for implementing our asymmetric document search algorithm was to fine-tune a pretrained multilingual model on our training data to better match the embeddings of documents and queries. We used the Sentence-BERT model, specifically the `paraphrase-multilingual-MiniLM-L12-v2` from the sentence-transformers library [1]. This model maps sentences and paragraphs into a 384-dimensional dense vector space, making it suitable for tasks such as clustering and semantic search.

For fine-tuning, our dev training dataset provided, for each query, a positive document ID and multiple negative document IDs. We therefore decided to further train the model using a triplet-based approach, leveraging the TripletLoss function available in sentence-transformers. Given a triplet $(\text{anchor}, \text{positive}, \text{negative})$, the TripletLoss minimizes the distance between the anchor and the positive sample while maximizing the distance between the anchor and the negative sample. The loss function is defined as follows:

$$\mathrm{L} = \max\left(\|\text{anchor} - \text{pos.}\| - \|\text{anchor} - \text{neg.}\| + \text{margin}, 0\right)$$

where margin is a key hyperparameter that requires careful tuning to optimize model performance.

Unfortunately, we did not achieve significant improvements using this method, yielding only a Recall@10 score of 0.26 on the entire corpus. One potential reason for this low performance is the model's input length limitation of 128 tokens, which results in ignoring any part of the documents exceeding this length. While this model could perform well for symmetric semantic search tasks, it was less effective for our dataset, where document lengths often far exceeded the 128-token limit imposed by the model.

VIII. Summary

This project demonstrates how traditional retrieval techniques like TF-IDF and BM25 can be effectively tailored for a multilingual document retrieval system through language-specific preprocessing. Our approach highlights the strengths and limitations of each method: TF-IDF offers computational simplicity, while BM25 enhances precision by managing term saturation and document length more effectively, especially for languages with varied structural complexities.

The consistency of our Recall@10 scores across dev and test datasets underlines the robustness of our approach in adapting to linguistic diversity, validating the strategic choices made in preprocessing and weighting adjustments. However, our exploration of Sentence-BERT for enhanced semantic retrieval revealed the challenges of deep learning approaches in document retrieval, especially when document length and processing efficiency are critical.

References

[1] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," 11/2019. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2