LLM Lab 3

Shusrith S

PES1UG22AM155

March 9, 2025

1 Introduction

Document retrieval is a critical component of information retrieval systems. This report presents a comparative analysis of three retrieval techniques: BM25, ColBERT, and FAISS. The code provided implements these methods and evaluates their effectiveness on a small corpus of documents.

2 Dataset

The dataset consists of 10 short documents containing simple sentences about animals, nature, and daily activities. The query used for retrieval is "fox jumps over."

3 BM25

BM25 (Best Matching 25) is a ranking function used in probabilistic information retrieval. It scores documents based on term frequency and inverse document frequency (IDF), adjusting for document length. The key steps in its implementation are:

- Compute IDF for each term in the corpus.
- Compute BM25 scores for each document based on query terms.
- Rank documents by BM25 scores and return the top results.

Formula:

$$\text{BM25}(q,d) = \sum_{t \in q} IDF(t) \cdot \frac{TF(t,d)(k_1+1)}{TF(t,d) + k_1(1 - b + b \cdot \frac{|d|}{\text{avgdl}})} \tag{1}$$

where k_1 and b are hyperparameters, $|d|$ is the document length, and avgdl is the average document length.

4 ColBERT

ColBERT (Contextualized Late Interaction over BERT) leverages transformer-based embeddings for retrieval. The implementation follows these steps:

- Encode documents and query using the "all-MiniLM-L6-v2" Sentence Transformer.
- Compute cosine similarity between query and document embeddings.
- Rank documents by similarity and return the top results.

5 FAISS

FAISS (Facebook AI Similarity Search) is a library for fast nearest neighbor search. Two FAISS techniques are used:

5.1 FAISS Flat

This method computes L2 distances between embeddings and retrieves the closest matches. Steps include:

- Encode documents and query using Sentence Transformer.
- Compute L2 distance between query and document embeddings.
- Rank documents by L2 distance and return the top results.

5.2 FAISS IVF (Inverted File Index)

FAISS IVF partitions the vector space into clusters to improve efficiency. The steps include:

- Select cluster centroids from document embeddings.
- Assign each document to the closest centroid.
- Assign query to a cluster and retrieve nearest documents.

6 Results

The results for the top three retrieved documents using each method are as follows:

- **BM25**: `['A fox jumps over the fence.', 'The quick brown fox jumps over the lazy dog.', 'A child laughs with joy.']`
- **ColBERT**: `['A fox jumps over the fence.', 'The quick brown fox jumps over the lazy dog.', 'The dog plays in the yard.']`

- **FAISS Flat**: `['A fox jumps over the fence.', 'The quick brown fox jumps over the lazy dog.', 'The dog plays in the yard.']`
- **FAISS IVF**: `['A fox jumps over the fence.', 'The quick brown fox jumps over the lazy dog.', 'The dog plays in the yard.']`

Each method offers trade-offs between accuracy and efficiency. BM25 is strong in lexical matching, ColBERT excels in semantic understanding, and FAISS provides efficient large-scale retrieval.

7 Conclusion

This report presented an implementation and comparison of BM25, ColBERT, and FAISS retrieval methods. BM25 is effective for term-based ranking, while ColBERT enhances results using transformer embeddings. FAISS is suitable for large-scale applications requiring fast nearest neighbor search.