IRise: Study of A Comprehensive Retrieval System Boosted by Neural Search

Devrim Çavuşoğlu [1] Ezgi Çavaş [1]

[1]Department of Computer Engineering, METU

Motivation/Introduction

The Big Problem With the exponential growth of information, retrieving relevant documents quickly and accurately has become a critical challenge in information retrieval.

The Small Problem Traditional retrieval models, such as TF-IDF and BM25, may fall short in capturing semantic similarities, leading to less relevant results.

Solution Attempt Our project, IRise, integrates traditional retrieval models with modern text embedding techniques to enhance retrieval performance. By using a two-stage process: initial retrieval followed by re-ranking. With this, we aim to balance retrieval speed and relevancy.

Implementation Details

Dataset and Preprocessing The dataset is BEIR's split of MSMARCO. The test split is used and all preprocessing done on that split.

The dataset is composed of 8.8 million documents. There are 9.2 thousand query relevance judgements for 43 queries. The topics for the queries vary from science to knowledge of vocabulary.

Indexing and Initial Ranking In order to index the dataset, PyTerrier and Whoosh are utilized. The indexing type is the default one, which creates a direct index alongside the inverted index. During indexing, an English stopword list and Porter's stemmer are applied to the corpus. The tokenizer is the default one for the English language.

The initial ranking was conducted by a search in the inverted index by utilizing vector space models (VSMs). We used TF-IDF/BM25 models as effective VSMs in the initial retrieval.

Re-ranking For the re-ranking stage, we used a text embedding model, particularly GTE-base [1] capable to capture semantic similarity between two contexts. This stage is used to boost the performance of the system re-ranking the top 200 results of the initial retrieval. It is important to note that the top 200 results are calculated on-the-fly per query. The reason stems from the fact that the initial rankings are retrieved quite fast.

Methodology and Practical Implementation

Our system workflow is summarized below:

- Initial results are retrieved using a TF-IDF/BM25 model for efficiency.
- These results are then re-ranked using a text embedding model to improve relevancy.
- The number of initial results (top 200) balances speed and performance but can be adjusted.

A high-level system architecture is given in Figure 1.

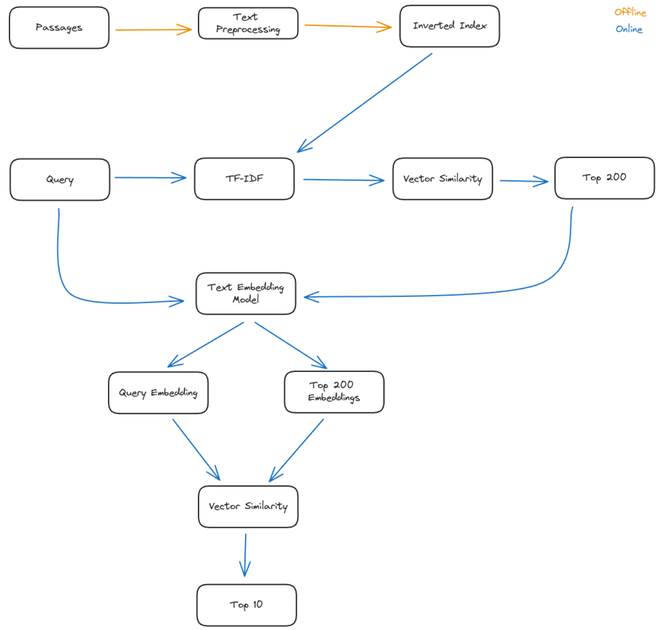

Figure 1. High-level system overview.

Evaluation

Explored various vector space models (VSMs) such as TF-IDF and BM25. Combined VSM scores using weighted averages (controlled by lambda). Tested different text preprocessing techniques for the inverted index, including stop-word removal and stemming.

Performance comparison between initial retrieval (TF-IDF/BM25) and the re-ranked results (embedding model). Improvements in relevancy metrics and retrieval times.

Model	P@10	MRR	MAP	nDCG@10	Speed (s/it)
TF-IDF	14.52	25.44	0.7	11.44	8.54
BM25	55.00	77.70	7.90	43.55	11.65
TF-IDF(Re-ranked)	56.67	86.19	7.91	49.20	**11.05**
BM25(Re-ranked)	**79.05**	**91.67**	**12.93**	**67.18**	14.86

Table 1. Evaluation results.

Conclusion

We proposed a methodology and an implementation to search through document sets where the order of magnitude is in millions.

- The two-stage retrieval process effectively improves relevancy while maintaining acceptable retrieval speeds.
- Adjusting the number of initial results (top 200) allows for a trade-off between speed and performance.
- Future work includes exploring more advanced embedding models and optimizing preprocessing techniques for better results.

References

[1] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.