METU Personal Tutor: A Chatbot for University Information Retrieval

Yusuf Sami Lök
Middle East Technical University
Ankara, Turkey
yusuf.lok@metu.edu.tr

Berk Açıkel
Middle East Technical University
Ankara, Turkey
berk.acikel@metu.edu.tr

Abstract

University students often face challenges when searching for specific information hidden within a vast amount of publicly available documents. This paper introduces *METU Personal Tutor*, a chatbot designed to efficiently retrieve and answer queries based on university-related data. We utilized BM25 and vector embeddings as retrieval mechanisms, combined to maximize document relevance. Retrieved documents were passed to Llama-3-70b for generating user-friendly answers. The dataset, scraped from METU's publicly accessible domains, was structured for efficient use. Experiments demonstrated the effectiveness of our approach, highlighting the strengths and limitations of different retrieval techniques. A Flask-based website was developed to host the chatbot, featuring basic authentication and a user-friendly interface.

1 Introduction

Accessing specific information from large collections of documents is a common challenge for university students and faculty members. In the case of METU, much of the relevant information is scattered across various subdomains and formats, such as web pages and PDFs. Because of this, searching for a question will likely take a lot of time. This paper addresses this problem by presenting *METU Personal Tutor*, a chatbot that combines retrieval-based and generation-based methods to answer user queries efficiently.

This project presents several key contributions that advance the field of information retrieval and chatbot systems: A comprehensive dataset was collected from METU's publicly accessible web pages and structured for efficient retrieval. BM25 (Robertson et al., 2009) and vector embedding methods were integrated for document retrieval. Llama-3-70b was implemented to generate human-like responses based on retrieved documents. Finally, a web-based interface was developed, featuring basic authentication and chatbot functionality.

The proposed system achieves scalability by avoiding fine-tuning large language models, instead relying on effective retrieval and structured data processing.

2 Related Work

Retrieval-Augmented Language Modeling is a popular way to improve language models by linking their outputs to external, relevant information. One key method in this area is In-Context RALM (Ram et al., 2023), which offers a simple way to enhance language models without changing their structure or requiring fine-tuning. Instead, it adds related documents to the input during use. This approach is easy to implement and improves performance of models with various datasets significantly. It also uses advanced document search and ranking techniques, making it ideal for situations where pre-trained models are used without any modification.

Another important method is the RAFT (Retrieval-Augmented Fine-Tuning) framework, designed for tasks that require domain-specific knowledge (Zhang et al., 2024). RAFT combines fine-tuning with retrieval to improve accuracy using specialized documents. It uses high-quality reference documents and applies a step-by-step reasoning approach to retrieve the most useful information. This helps it work well in specialized fields, ensuring accurate information retrieval and response generation.

Finally, the Self-Memory Framework (Cheng et al., 2024) introduces a new way of combining memory and generation. This process improves retrieval as the model generates better outputs over time. The model builds a memory pool from its outputs and uses a memory selector to improve future results. This approach achieves top performance on tasks like machine translation and sum-

marization, showing how self-generated memory can boost retrieval-augmented models.

3 Data

3.1 Data Collection

To collect data, we initially attempted to scrape all web pages under all subdomains of the `metu.edu.tr` domain, starting with `www.metu.edu.tr`. However, we soon realized that many subdomains contained irrelevant information, and have problematic loops, which slowed down the scraping process and can potentially affect the model's performance. Our initial solution to this was to blacklist irrelevant subdomains, but managing this list was impractical.

Next, we modified our approach by scraping only the `www.metu.edu.tr` website to extract a list of subdomains. This process identified 187 subdomains, of which 108 were relevant. We started to scrape all web pages from these relevant subdomains.

During our analysis of the scraped data, we noticed a significant number of Turkish web pages. To address this, we modified our script to exclude URLs containing the pattern `.metu.edu.tr/tr/`. Additionally, we found that many text elements, such as texts on buttons, provided no useful information and could confuse the retriever. To handle this, we updated our script to exclude text within HTML tags like `<a>`, `<li>`, and `<button>`.

We also included text from PDF documents in this version of the script using the PyPDF2 library. After implementing these changes, we ran the script again to generate the final dataset.

The final dataset consists of 75.9 MB of text data collected from 19,965 web pages.

3.2 Data Preparation

The scraped data was divided into chunks, and structured as a text file with a chunk per line. Each line of the text file starts by the URL of the webpage or the document, and continues with contents.

4 Our Method

The primary goal of our method was to retrieve relevant documents that could provide meaningful context for answering user queries, and then give it as input to our model, concatenated with the user query. To achieve this, we explored different retrieval mechanisms. Initially, we tried a baseline using a traditional retrieval method. However, we enhanced the system's capability by combining that method with a method based on vector embeddings, which showed significant improvement in retrieving semantically relevant documents. This hybrid approach combines the strengths of both methods to offer better performance in various scenarios. Figure 1 illustrates the flow of our method, where the user's query is passed through the hybrid retrieval system to fetch relevant documents. These documents, with the query, are then combined and given into the Llama-3-70b model to generate an accurate answer.

4.1 Retrieval Mechanisms

Efficient retrieval of relevant documents is a critical component of the METU Personal Tutor system. Two distinct methods were used to address this problem: BM25 and vector embeddings.

4.2 BM25

BM25 is a ranking algorithm built on Term Frequency-Inverse Document Frequency (TF-IDF). It evaluates the relevance of documents by considering term frequency, inverse document frequency, and document length normalization. This approach is particularly effective for identifying exact matches between query terms and document content. BM25's simplicity and robustness make it a popular choice for many information retrieval tasks. However, its reliance on exact term matching limits its ability to capture semantic details in queries.

4.3 Vector Embeddings

Vector embeddings are dense representations of text to measure semantic similarity between queries and documents. In this project, the multi-qa-mpnet-base-dot-v1 model was used to generate embeddings. Documents with the highest cosine similarity to the query were retrieved as relevant results. While vector embeddings are good at understanding contextual and semantic relationships, they are computationally more expensive and can miss exact matches that BM25 identifies.

4.4 Hybrid Retrieval Approach

To benefit from the strengths of both methods, a hybrid retrieval mechanism was implemented. The top-ranked documents from BM25 and vector embedding-based retrieval were combined. This approach ensured that both exact matches and semantically relevant documents were retrieved. Al-

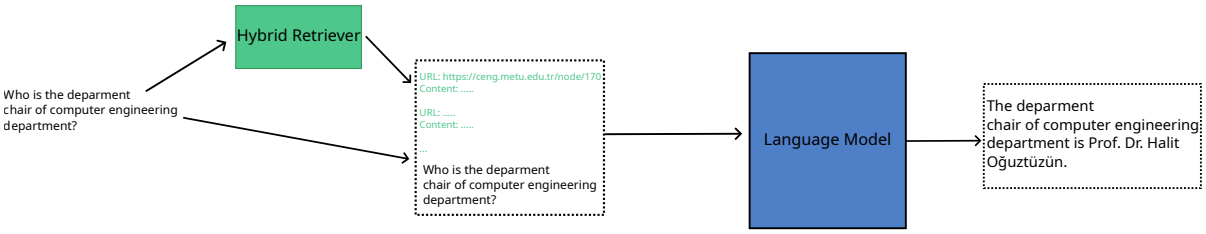

Figure 1: An example flow of our method.

though the hybrid method causes additional computational overhead compared to single-method retrieval, it significantly increases the relevance and diversity of the retrieved documents. Figure 2 illustrates our hybrid approach to the retrieval mechanism.

5 Experiments

5.1 Experimental Setup

Experiments were conducted to evaluate the performance of retrieval methods and their combination. The BM25 and embedding-based methods were tested individually and in combination.

5.2 Results

Preliminary results showed that:

- BM25 performed well on queries with exact term matches.
- Vector embeddings worked better in capturing semantic relationships.
- Combining both methods improved overall performance, by using documents with highest exact term matches or highest semantic similarity.

5.2.1 Evaluation Metrics

To evaluate the performance of our retrieval methods, we used the following metrics. ROUGE-1 F1 measures the overlap of unigrams between the generated and reference responses. It is useful for evaluating recall and precision at the word level. ROUGE-2 F1 is similar to ROUGE-1, but it measures the overlap of bigrams. This provides a deeper insight into the quality of the generated responses. ROUGE-L F1 measures the longest common subsequence between the generated and reference responses. It evaluates the fluency and syntactic structure of the generated text.

BLEU Score is a metric used to evaluate the quality of generated text by comparing n-grams in the candidate text to n-grams in a reference text. BERTScore F1 measures the semantic similarity between the generated and reference responses by embedding both texts using BERT and comparing their cosine similarities.

5.2.2 Evaluation Data

To evaluate our method's performance, we needed some sample questions with reference answers. Since there is not such a dataset for METU, we had to prepare it. We wrote sample questions and comprehensive answers to them in order to use for evaluation of our method.

5.2.3 Evaluation Results

Table 1 illustrates the performance of BM25, vector embeddings, and their combination across various evaluation metrics.

Metric	BM25	Vector Embedding	Combination
ROUGE-1 F1	0.547	0.442	**0.725**
ROUGE-2 F1	0.376	0.320	**0.545**
ROUGE-L F1	0.473	0.412	**0.635**
BLEU Score	0.248	0.209	**0.415**
BERTScore F1	0.911	0.891	**0.946**

Table 1: Comparisons of Retrieval Methods

The results indicate that the combination method consistently outperforms both BM25 and Vector Embedding across all metrics, showing its ability to use both exact term matching and semantic similarity for improved document retrieval and response generation.

6 Discussion

The hybrid retrieval approach demonstrated good results, outperforming both BM25 and vector embedding based methods individually. By combining exact term matching and semantic similarity, the system was able to retrieve more relevant documents and offer better response generation. This improvement shows the importance of using multiple retrieval strategies, as each method brings some advantages.

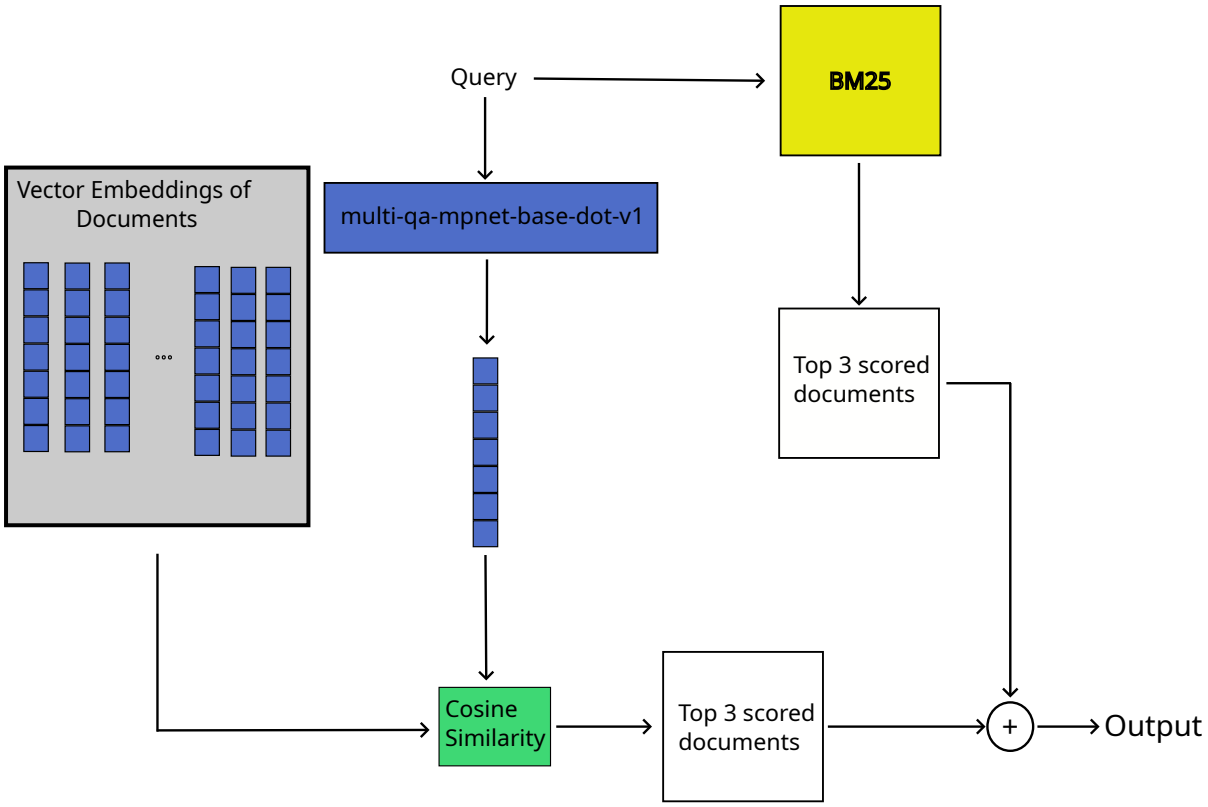

Figure 2: Hybrid Retrieval Mechanism

However, we faced several challenges during the implementation of the hybrid approach. While combining both methods enhanced performance, it also caused computational overhead. The additional computational cost of running both BM25 and vector embeddings for document retrieval could become a problem for large scale implementations. Future improvements may include implementing more advanced reranking methods to optimize the retrieval process and reduce resource consumption.

Making the system scalable may be challenging. As the number of documents grows, the retrieval and processing time may increase significantly, which may make the system unusable. To address this, techniques like indexing and caching could be tried to optimize scalability and reduce latency.

To ensure safety and privacy, it is crucial to exclude any private data from the documents. Although the data is collected from public web pages, manual analysis of the data may be required in case of any forgotten private information on public website. Furthermore, as the system relies on an external model for response generation, the leakage of query data is possible. To overcome this issue, the model may be run on a private server with necessary hardware.

In conclusion, despite the good results of hybrid retrieval approach, further improvements are needed for optimizing performance, enhancing scalability, and addressing safety and privacy problems to make the system more reliable and suitable for large scale real world applications.

7 Conclusion

METU Personal Tutor provides an effective solution for university related information retrieval. By combining BM25 and vector embeddings, we ensured accurate document retrieval, and Llama-3-70b generated accurate responses. Future work includes integrating more retrieval models, improving scalability, and addressing privacy concerns.

References

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2024. Lift yourself up: Retrieval-augmented text generation with self-memory. *Advances in Neural Information Processing Systems*, 36.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav

Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.