Documents Ranking using Retrieval Augmented Generation

Submitted in partial fulfillment of the requirements
for the award of the degree of

Master of Technology
in
Information Technology
With specialization in MLIS

Submitted by

Shahil Kumar
(Enrollment No. MML2023008)

Under the Supervision of
Dr. Muneendra Ojha

DEPARTMENT OF INFORMATION TECHNOLOGY
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD
Prayagraj-211015, India

SEPTEMBER 26, 2024

<u>CANDIDATE DECLARATION</u>

I, Manu Pande, MML2023005, certify that this thesis work titled **"Documents Ranking using Retrieval Augmented Generation”**, submitted towards fulfillment of MASTER’S THESIS report of M.Tech. IT at Indian Institute of Information Technology Allahabad, is an authenticated record of our original work carried out under the guidance of Dr. Muneendra Ojha. Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Shahil Kumar

Shahil Kumar
MML2023008
Machine Learning and Intelligent Systems

Acknowledgements

I reserve a special place in my heart for my beloved parents, whose unwavering love, unwavering support, and unwavering belief in my abilities have been the bedrock upon which my dreams have flourished. Their persistent support, sacrifices, and unshakable trust in my abilities have been the driving factors behind my quest for knowledge and academic pursuits.

I owe a tremendous debt of gratitude to my esteemed supervisor, **Dr. Muneendra Ojha** , whose guidance and advice have been the compass guiding me through the many twists and turns of this thesis. His stimulating conversations, insightful feedback, kind advice, and boundless forbearance have challenged me to push the boundaries of my capabilities and inspired me to strive for academic excellence. I am very thankful for the trust you put in me and the chances you gave me to grow both professionally and personally. I am grateful beyond words for the opportunity to have worked under your guidance, and I hope my thesis serves as a fitting tribute to your hard work, knowledge, and encouragement.

Finally, I want to thank the Faculty who helped me grow as a scholar.

Shahil Kumar

ABSTRACT

This study explores the effectiveness of embedding-based document re-ranking for biomedical information retrieval using different tokenization strategies. We implement large language models (LLMs) to generate semantic embeddings for both documents and queries and re-rank documents based on similarity scores. Our approach systematically compares general-purpose tokenizers with domain-specific ones to assess their impact on re-ranking performance. Additionally, we fine-tune an LLM model on a biomedical corpus to enhance domain-specific relevance. The evaluation includes standard information retrieval metrics, comparing the performance of our approach with baseline methods. By testing various tokenizers and assessing their effect on document ranking, this work aims to optimize re-ranking processes and provide insights into improving search relevance in biomedical applications.

Contents

List of Abbreviations

LLM	Large Language Model
RAG	Retrieval Augmented Generation
HyDE	Hypothetical Data Embeddings
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
MS MARCO	MicroSoft MAchine Reading COmprehension

Chapter 1

Introduction

1.1 Introduction

The rapid expansion of textual data across various domains has introduced significant challenges in information retrieval (IR). Conventional methods, such as Term Frequency-Inverse Document Frequency (TF-IDF) and learning-to-rank models, often struggle with understanding complex queries and providing results that are both relevant and contextually meaningful. Moreover, these traditional approaches fail to adequately address challenges such as ambiguous queries and the need for contextual reasoning.

Retrieval-Augmented Generation (RAG) represents a transformative approach that combines the robust retrieval capabilities of dense embedding-based search with the generative strengths of large-scale language models (LLMs). By leveraging dense retrieval mechanisms, RAG identifies a subset of relevant documents from a large corpus, which are then utilized by the generative model to produce responses or rank documents effectively. This dual capability enables RAG to excel in open-domain applications, bridging the gap between IR and natural language generation.

1.1.1 Retrieval-Augmented Generation (RAG)

RAG is a hybrid framework that integrates two powerful paradigms: *retrieval* and *generation*. The process involves two primary steps:

1. **Dense Retrieval:** A retriever, often based on neural embeddings, fetches a set of candidate documents or passages relevant to the input query. Dense retrieval mechanisms like DPR (Dense Passage Retrieval) or FAISS provide a scalable and efficient method for retrieving semantically relevant documents.

2. **Generative Response:** The retrieved documents are fed into a generative language model, such as GPT-3 or T5, which synthesizes a response or ranks the documents based on contextual relevance to the query.

RAG overcomes the limitations of traditional retrieval systems by leveraging the generative model's ability to fill knowledge gaps and reformulate queries dynamically. This approach is particularly effective in scenarios where queries are ambiguous or require contextual understanding beyond surface-level features.

1.1.2 Hypothetical Document Embedding (HyDE)

HyDE extends the capabilities of retrieval systems by generating hypothetical answers to the input query and then evaluating the relevance of documents in relation to the generated answer. The key steps in HyDE are as follows:

1. **Hypothetical Answer Generation:** A generative model hypothesizes an answer to the input query based on prior knowledge and context. This answer is treated as a pseudo-query.

2. **Embedding and Comparison:** Both the hypothetical answer and the candidate documents are encoded into dense vector embeddings. The similarity between the embeddings is used to score and rank the documents.

By hypothesizing answers, HyDE effectively bridges the semantic gap between the query and the documents, especially in cases where the original query is incomplete or ambiguous. This technique has been shown to enhance the relevance of ranked documents in complex retrieval tasks.

1.1.3 Motivation and Problem Statement

Ranking documents effectively is a cornerstone of IR systems, especially in knowledge-intensive domains such as academic research, technical documentation, and enterprise search. Traditional ranking methods, such as BM25 or machine learning-based rerankers, rely heavily on surface-level features and fail to capture semantic nuances. On the other hand, the emergence of transformer-based models, such as GPT-3 and BERT, has enabled systems to understand and generate human-like responses, paving the way for novel ranking strategies.

While RAG and HyDE provide robust solutions, further enhancements are required to improve ranking performance, particularly in terms of capturing deeper semantic relationships, reducing noise in retrieval, and optimizing the integration of hypothetical embeddings with generative models.

Chapter 2

Literature Review

2.1 Literature Table

Authors	Paper	Findings
Z. Qin, R. Jagerman, K. Hui, et al. [1]	Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting	This paper introduces "pairwise ranking prompting" for document ranking. It takes two documents and a query, and the LLM is asked to rank the two based on relevance to query. It uses off the shelf LLMs without domain-specific fine tuning. Context is given to the LLM about the task which is - ranking documents making use of few-shot learning.
B. Nouriinanloo and M. Lamothe [2]	Re-Ranking Step by Step: Investigating Pre-Filtering for Re-Ranking with Large Language Models	This paper introduces an LLM-based pre-filtering step before re-ranking, where passages or documents are filtered out based on their relevance to a query. The pre-filtering helps reduce the noise that could misguide the re-ranking process. Then use of LLM is made for these best candidates to re-rank them. This led to development of re-rankers which were much smaller yet effective
S. Zhuang, B. Liu, B. Koopman, and G. Zuccon [3]	Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking	This paper explores using open-source LLMs to estimate how likely a document is to be relevant to a given query. Higher likelihood means higher rank. The models perform effectively in zero-shot settings, where no task-specific training is provided. The paper also shows that additional instruction fine-tuning may hinder effectiveness
A. Drozdov, H. Zhuang, Z. Dai, et al. [4]	PaRaDe: Passage Ranking using Demonstrations with Large Language Models	This research addresses the limitations of zero-shot learning. Incorporates few-shot demonstrations into prompt. Demonstrations are pairs of passages and their relevance scores. The paper argues that presenting the LLM with difficult examples yield better results in ranking tasks. Difficult here means queries or passages that present more challenge to LLM to correctly rank

Table 2.1: Literature Table

Chapter 3

Methodology

This section outlines the proposed framework for document ranking using Retrieval-Augmented Generation (RAG) enhanced with Hypothetical Document Embedding (HyDE). The methodology integrates dense retrieval mechanisms with generative models to improve ranking accuracy and relevance. The system is designed to address key challenges in semantic understanding, contextual reasoning, and efficient ranking.

3.0.1 System Architecture

The proposed methodology consists of the following core components:

1. **Query Preprocessing:** The input query is preprocessed to standardize its format and tokenize it for further processing.
2. **Dense Retrieval:** Candidate documents are retrieved from a large corpus using dense embedding-based similarity search.
3. **Hypothetical Embedding Generation (HyDE):** Hypothetical answers are generated for the query, which are encoded into embeddings for refined ranking.
4. **Generative Reranking:** A large-scale language model scores and ranks the candidate documents by evaluating their relevance to both the query and the hypothetical embeddings.
5. **Final Ranking:** The ranked list of documents is produced as the output.

The overall workflow is illustrated in Fig. 3.1.

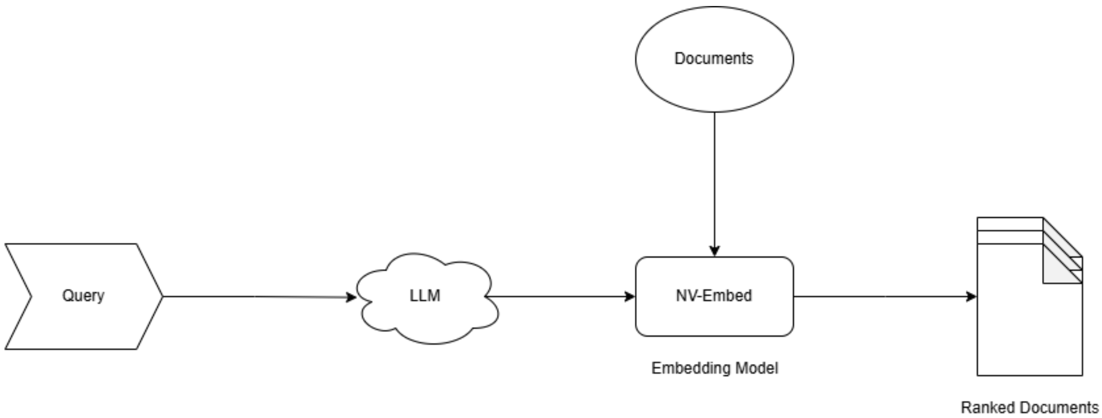

Figure 3.1: System architecture of the proposed methodology.

3.0.2 Dense Retrieval

Dense retrieval forms the first stage of the pipeline, where the goal is to identify a subset of candidate documents from a large corpus that are semantically relevant to the input query. This is achieved using dense vector representations generated by transformer-based models such as BERT or DPR. The steps include:

1. **Embedding Generation:** The input query and corpus documents are encoded into high-dimensional vectors using a pre-trained dense retriever.

2. **Similarity Computation:** Cosine similarity is computed between the query vector and document vectors.

3. **Top-K Retrieval:** The top-K documents with the highest similarity scores are retrieved for further processing.

This stage ensures that only the most relevant documents are passed to the next phase, reducing computational overhead.

3.0.3 Hypothetical Document Embedding (HyDE)

The HyDE technique is incorporated to enhance the retrieval system by generating hypothetical answers to the query and utilizing these answers to refine the ranking. The steps involved are:

1. **Hypothetical Answer Generation:** A generative model, such as GPT-3, generates a hypothetical response to the input query. This response acts as a pseudo-query.

2. **Embedding Creation:** Both the hypothetical answer and the candidate documents are encoded into dense vector embeddings.

3. **Relevance Scoring:** Cosine similarity is computed between the embeddings of the hypothetical answer and the candidate documents. This generates an additional relevance score for each document.

The HyDE component is particularly useful for addressing cases where the query is ambiguous or lacks sufficient context, as it hypothesizes missing information to improve document ranking.

3.0.4 Generative Reranking

In this stage, a large-scale generative language model is employed to score and rerank the documents based on their contextual relevance. The steps include:

1. **Input Construction:** The retrieved documents and hypothetical embeddings are provided as input to the generative model along with the query.

2. **Contextual Scoring:** The model generates a contextual score for each document by evaluating its relevance to the query and the hypothetical answer.

3. **Ranking Adjustment:** The initial ranking is adjusted based on the contextual scores to produce the final ranked list.

This step ensures that the ranking reflects not just surface-level similarity but also deeper semantic and contextual relevance.

3.0.5 Implementation Details

The proposed methodology was implemented using state-of-the-art tools and frameworks. Key details include:

- **Dense Retriever:** The DPR (Dense Passage Retriever) model from Hugging Face was used for encoding queries and documents.

- **Generative Model:** The GPT-3 model was utilized for hypothetical answer generation and generative reranking.
- **Embedding Framework:** The embeddings were normalized and processed using PyTorch for efficient similarity computation.
- **Optimization:** Techniques such as batching and GPU acceleration were used to optimize performance on large datasets.

3.0.6 Evaluation Pipeline

The evaluation of the proposed methodology involves benchmarking its performance on standard datasets. The evaluation metrics include:

- **Precision@K:** Measures the proportion of relevant documents in the top-K results.
- **Mean Reciprocal Rank (MRR):** Evaluates the ranking by considering the position of the first relevant document.
- **Normalized Discounted Cumulative Gain (nDCG):** Assesses the quality of the ranking by accounting for the relevance and order of documents.

A detailed analysis of the results and comparisons with baseline methods is presented in Section V.

3.0.7 Workflow Summary

To summarize, the proposed methodology leverages dense retrieval, hypothetical embeddings, and generative reranking to achieve high-quality document rankings. By integrating these components into a unified pipeline, the system addresses key challenges in information retrieval and provides significant improvements in ranking accuracy and relevance.

Chapter 4

Results

The comparison of base models with their fine-tuned versions can be seen in this section. Both fine-tuned versions on an average show better performance (accuracy) than their respective base models

4.1 Gemini_PMC

The performance evaluation metrics of the fine-tuned model are as follows:

Figure 4.1: Comparison of the Gemini_PMC model with the base model

4.2 GPT2_PMC

The performance evaluation metrics of the fine-tuned model are as follows:

Figure 4.2: Comparison of the GPT2_PMC model with the base model

Showing the table from Open Medical-LLM Leaderboard [5]

Metric	**GPT2_PMC**	**GPT2**
Average	27.67	26.97
MedMCQA	31.96	31.89
MedQA	27.97	27.73
MMLU Anatomy	18.52	20.00
MMLU Clinical Knowledge	21.51	20.75
MMLU College Biology	25.00	25.69
MMLU College Medicine	21.39	20.81
MMLU Medical Genetics	30.00	31.00
MMLU Professional Medicine	19.12	19.49
PubMedQA	53.6	45.40

Table 4.1: Model performance comparison using accuracy percentage as metric

Chapter 5

Conclusions and Future Scope

5.1 Conclusions

The proposed methodology of fine-tuning pre-trained LLMs for cardiovascular diseases has been successfully implemented. The fine-tuned models, Gemini_PMC and GPT2_PMC, have shown improved performance on the medical dataset, outperforming the base models. The results demonstrate the effectiveness of fine-tuning large language models for medical domain-specific tasks, highlighting the potential of LLMs in healthcare applications.

5.2 Future Scope

1. In future a better curated set of data should be used for fine-tuning the models. As of today there is a lack of resource available for the specific field of cardiovascular diseases.

2. Other PEFT techniques such as LoRA, and many other adapter based methods can be used for faster fine-tuning of the models.

3. The models can be further evaluated on a larger and more diverse set of medical datasets to assess their capabilities. An approach that may be used is by involving medical professionals to rate responses to the questions set by them.

4. The proposed methodology can be extended to other medical domains, such as oncology, neurology, and radiology, to develop specialized LLMs for various healthcare applications.

5. The future work can also include the development of multimodal LLMs that combine text, image, and other data modalities to enhance the performance of medical models.

6. Involving LLMs into healthcare has a huge risk associated with it. To mitigate this risk, the models can be further tuned on an extensive set of ethical guidelines. This won't eliminate the risk but will keep the model in check to produce reponses that can cause potential damage to health without a medical expert's supervision.

References

[1] Z. Qin, R. Jagerman, K. Hui, H. Zhuang, J. Wu, L. Yan, J. Shen, T. Liu, J. Liu, D. Metzler, X. Wang, and M. Bendersky, “Large language models are effective text rankers with pairwise ranking prompting,” 2024. [Online]. Available: https://arxiv.org/abs/2306.17563

[2] B. Nouriinanloo and M. Lamothe, “Re-ranking step by step: Investigating pre-filtering for re-ranking with large language models,” 2024. [Online]. Available: https://arxiv.org/abs/2406.18740

[3] S. Zhuang, B. Liu, B. Koopman, and G. Zuccon, “Open-source large language models are strong zero-shot query likelihood models for document ranking,” 2023. [Online]. Available: https://arxiv.org/abs/2310.13243

[4] A. Drozdov, H. Zhuang, Z. Dai, Z. Qin, R. Rahimi, X. Wang, D. Alon, M. Iyyer, A. McCallum, D. Metzler, and K. Hui, “Parade: Passage ranking using demonstrations with large language models,” 2023. [Online]. Available: https://arxiv.org/abs/2310.14408

[5] A. Pal, P. Minervini, Andreas Geert Motzfeldt, and Beatrice Alex, “openlifescienceai/’open_medical_llm_leaderboard’,” 2024. [Online]. Available: https://huggingface.co/spaces/openlifescienceai/open_medical_llm_leaderboard