

Pre-trained Large Language Models for Question-Answering

Ziyue Yin¹ Weisheng Zhang² Paul Weng³

¹Undergraduate Student in Data Science, Class of 2026, Duke Kunshan University

²Undergraduate Student in Computer Science, Class of 2026, Duke Kunshan University

³Associate Professor of Electrical and Computer Engineering, Duke Kunshan University

Abstract

- Retrieval-Augmented Generation (RAG) in large language models (LLMs) enables querying external documents for continuous knowledge updates.
- Current RAG methods often face challenges like irrelevant queries, incorrect responses, and content repetition [5].
- This project aims to adapt LLMs using RAG for efficient question-answering by focusing on **Relevacy Prediction, Hierarchical Approach, Hyperparameter Tuning, and Finetuning Embedding Model**.
- The improved workflow **demonstrates significant enhancements** in generating accurate and contextually relevant answers.

Background

- Large Language Models (LLMs)** are powerful tools for various natural language processing tasks, pre-trained on massive datasets to generate human-like responses [3].
- Retrieval-Augmented Generation (RAG)** allows LLMs to access external knowledge and update responses based on new information, as shown in Figure 1.

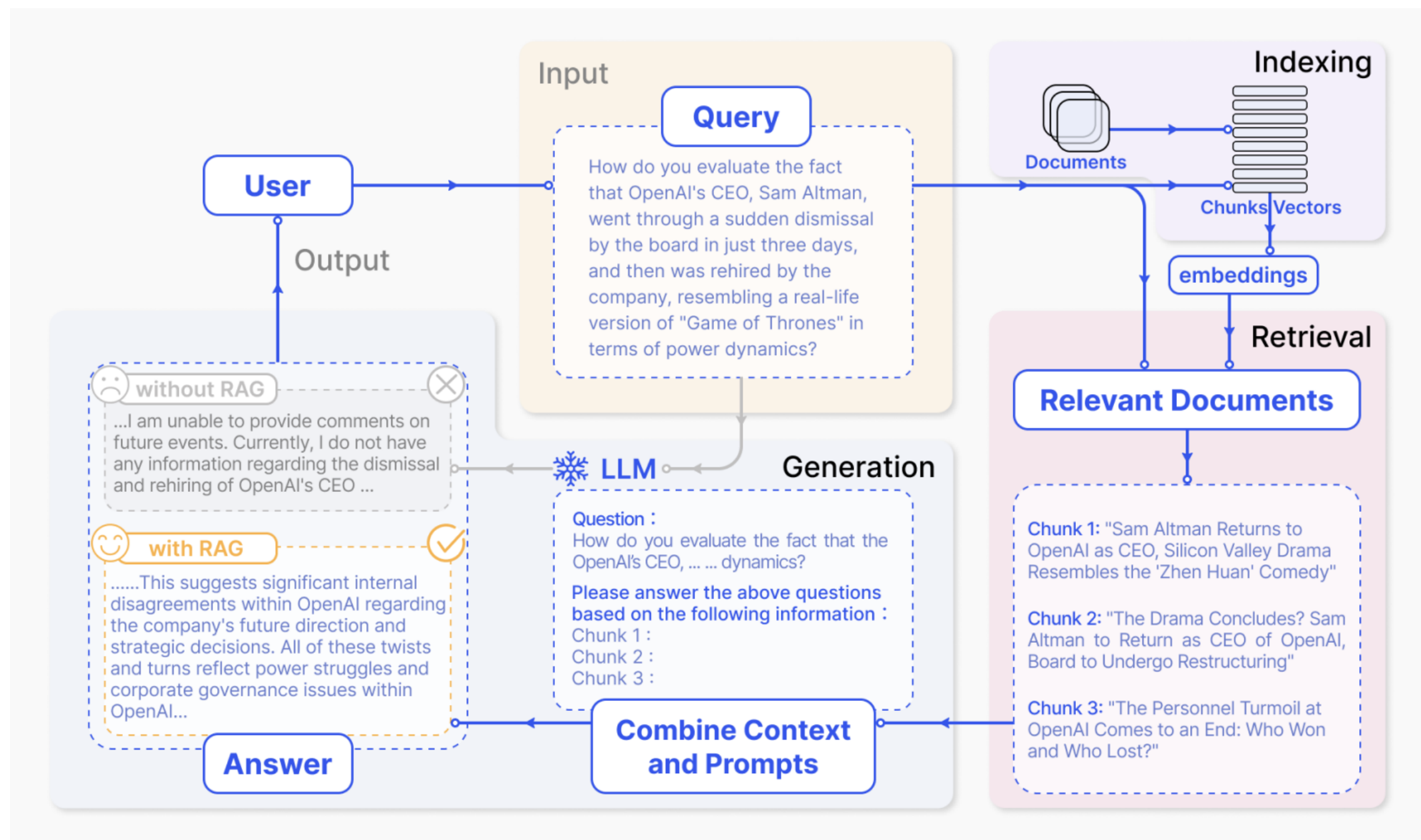


Figure 1. A representative instance of the RAG process applied to question answering [5].

Problem Statement

Commonly Observed Issues When Applying RAG

- Irrelevant Queries:** The retrieval mechanism sometimes fails to filter out irrelevant queries, leading to unnecessary computational effort and reduced efficiency.
- Incorrect Responses:** The system occasionally retrieves incorrect or incomplete chunks of information, resulting in inaccuracies in the generated answers.
- Content Repetition:** The model often repeats information redundantly within the same response, affecting the quality and coherence of the output.

Research Goal

This project aims to identify key factors that impact the performance of the RAG method in the context of the DKU Bulletin. To address those issues, the performance will be assessed on the following evaluation metrics with a 0-to-1 scale:

- Retrieval Performance Metric: **Hit Rate**, which evaluates whether top-k retrieved documents contain relevant content.
- Generation Performance Metrics [2]: Shown in Figure 2, **Faithfulness**, **Context Precision**, **Context Recall**, and **Answer Relevancy** are used to measure the quality of generated answers.

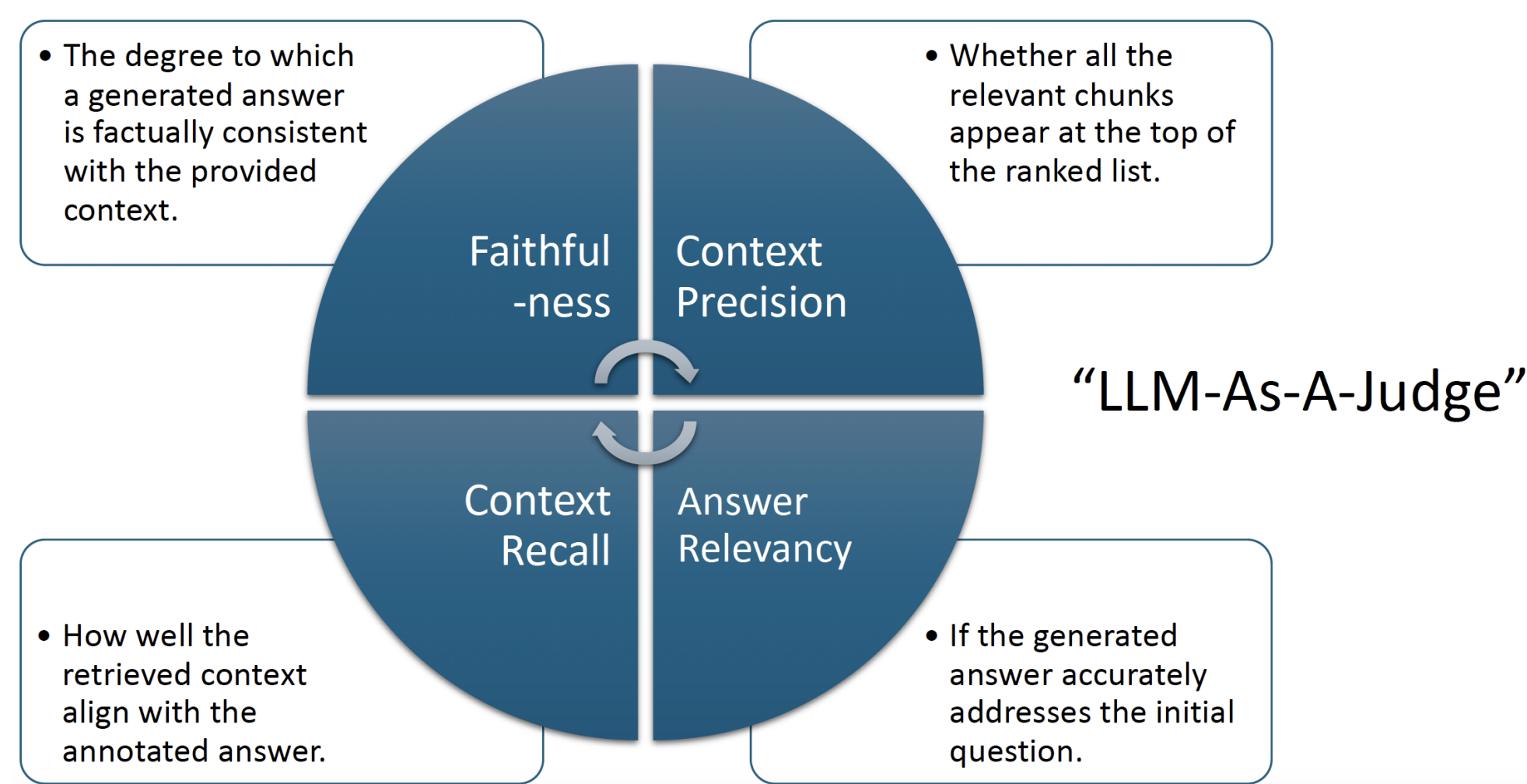


Figure 2. Four evaluation metrics as indicators of generation performance.

Methodology

Overall Mechanism

The overall workflow of the proposed model is depicted in Figure 3.

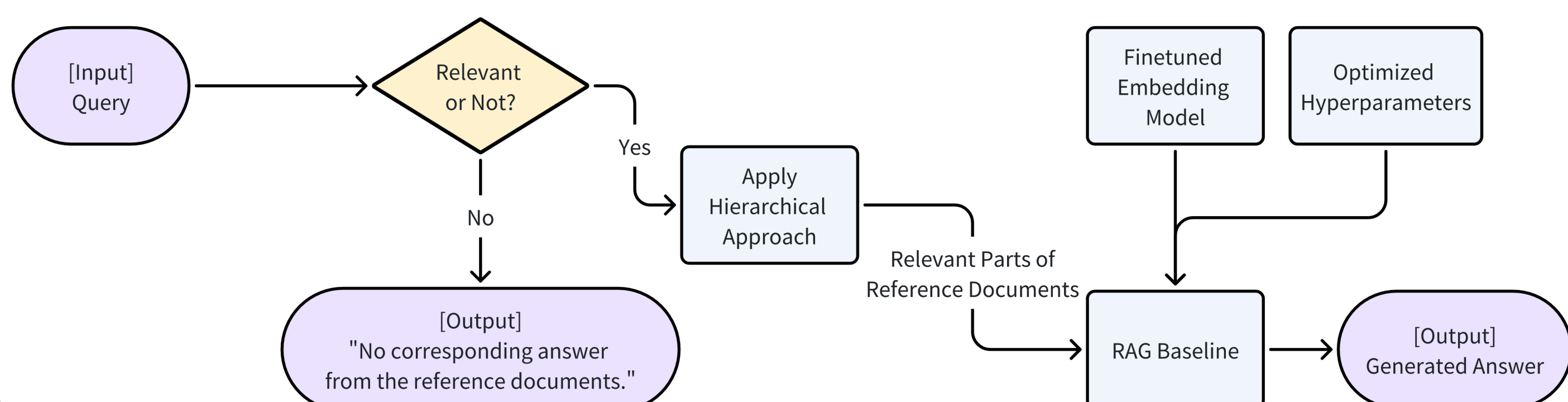


Figure 3. The overall model workflow.

Methodology (continued)

Relevancy Prediction

- Addressed Issues:** Filtering irrelevant queries that reduce efficiency and increase computational costs.
- The cosine similarity between the query and retrieved documents is calculated as a benchmark, with a threshold of 0.6.

Hierarchical Approach

- Addressed Issues:** Efficiently handling large documents and selecting precise content relevant to the query.
- As illustrated in Figure 4, on each level of the hierarchy, we test our query embedding against that level's summary embedding. If relevant, continue down the hierarchy for that specific part of the document.

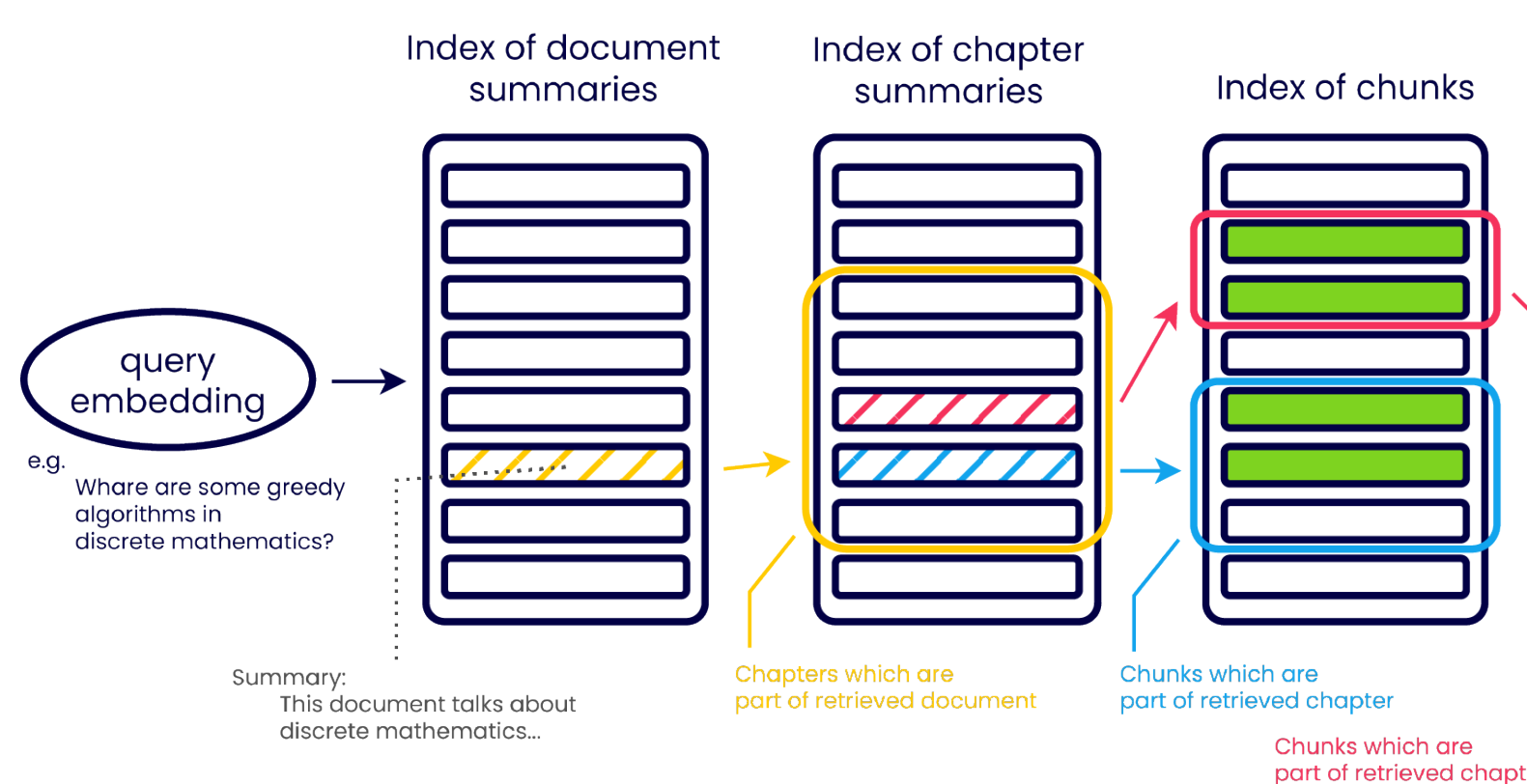


Figure 4. Progressive narrowing in hierarchical approach [1].

Hyperparameter Tuning

- Addressed Issues:** Balancing between accurate, non-redundant responses and avoiding information loss.
- Tested 75 sets of different combinations of hyperparameters.
 - Large length limit of generated answers or overlapped chunks led to repetition.
 - Inappropriate size of chunks caused information loss.
- Best hypermeter combination was selected to prevent redundancy while ensuring comprehensive answers.

Finetuning Embedding Model [4]

- Addressed Issues:** Improving relevance and contextual accuracy of retrieved content.
- Used a manually verified dataset with (*query, context*) pairs to finetune the embedding model, *all-mpnet-base-v2*.

Experimental Results

- Experiments were implemented in Langchain, with *all-mpnet-base-v2* as the embedding model and *Llama3-8B* for text generation.
- Figure 5 reveals the performance with the embedding model *all-mpnet-base-v2*, before and after fine-tuning.
 - Improvements were noted in most performance metrics, though with a slight decrease in Answer Relevancy.

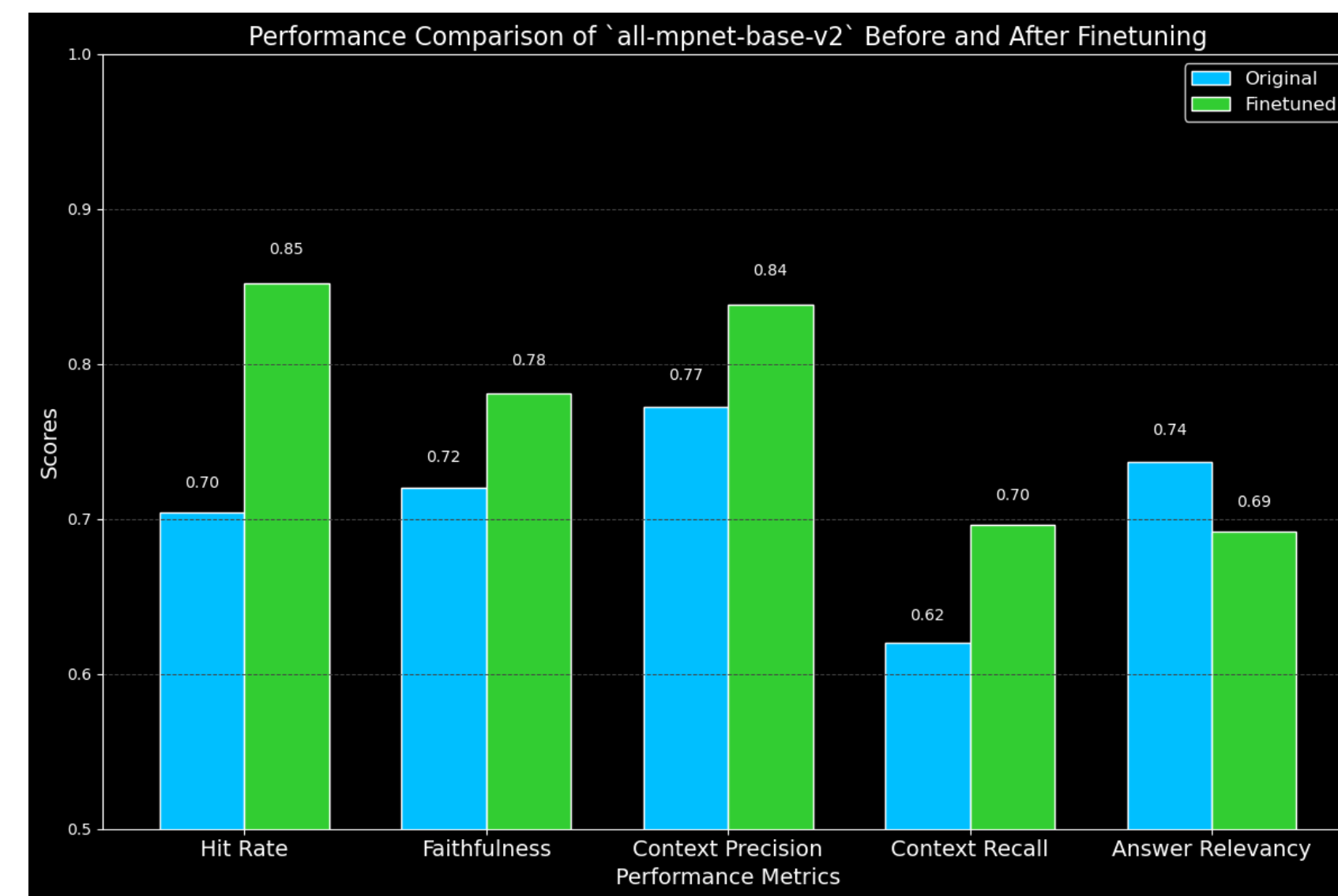


Figure 5. Performance with(out) finetuning.

Conclusion & Future Work

This project successfully enhanced the performance of LLM-based question-answering through the use of RAG, relevancy checks, a hierarchical approach, and fine-tuning. The final model demonstrated an improved ability to retrieve and focus on relevant contexts, although some challenges with answer relevance remain.

Future work will focus on using a more advanced RAG framework and exploring more comprehensive methods for optimizing content selection, aiming to make the model more adaptive and context-aware.

Acknowledgement

We sincerely thank our colleagues for their invaluable support throughout this SRS project:

- Yuwen Zhang** played a pivotal role in constructing the RAG baseline and developing the hierarchical approach, serving as a key leader within the group.
- Sean Allen Siegfried Bugarin** contributed to the evaluation metrics and dataset creation.
- Yuhan Wei** introduced innovative proposals for enhancing evaluation metrics.
- Runchu Wu** contributed to the optimization of hyperparameter configurations.
- Rongfan Liu** offered crucial guidance for improving the retrieval model.

We are grateful for the guidance and supervision delivered by **Prof. Paul Weng**.

This project was funded by the **DKU Office of Academic Services** and the **DKU Summer Research Scholars (SRS) Program**.

References

- Pixion. Rag strategies: Hierarchical index retrieval. Accessed: 2024-10-25.
- L. Espinosa-Anke S. Schockaert S. Es, J. James. Ragas: Automated evaluation of retrieval augmented generation. *arXiv:2309.15217*, 2023.
- J. Li T. Tang X. Wang Y. Hou Y. Min B. Zhang J. Zhang Z. Dong Y. Du C. Yang Y. Chen Z. Chen J. Jiang R. Ren Y. Li X. Tang Z. Liu J.-R. Wen W. X. Zhao, K. Zhou. A survey of large language models. *arXiv:2303.18223*, 2023.
- S. Xu. Pulsar on kubernetes: Enabling kafka-on-pulsar (kop) with pulsar operators. *Medium*, 2023-04-26.
- X. Gao K. Jia J. Pan Y. Bi Y. Dai J. Sun M. Wang H. Wang Y. Gao, Y. Xiong. Retrieval-augmented generation for large language models: A survey. *arXiv:2312.10997*, 2024.