Pre-trained Large Language Models for Question-Answering

Ziyue Yin[1] Weisheng Zhang[2] Paul Weng[3]

[1]Undergraduate Student in Data Science, Class of 2026, Duke Kunshan University [2]Undergraduate Student in Computer Science, Class of 2026, Duke Kunshan University
[3]Associate Professor of Electrical and Computer Engineering, Duke Kunshan University

Abstract

- **Large language models (LLMs)**, pre-trained on large text corpora, are powerful tools for natural language processing.
- This project evaluates methods to adapt LLMs for efficient question-answering using **Retrieval-Augmented Generation (RAG)**.
- Implemented in Langchain with *all-mpnet-base-v2* as the **embedding model** and *Llama3-8B* for **text generation**.
- RAG allows continuous knowledge updates, enabling the model to learn from external documents.
- Addressed issues like content repetition, unstable responses, and irrelevant queries by developing a relevancy check, adopting hierarchical content selection, and fine-tuning the embedding model.
- The improved workflow **demonstrates significant improvements** in generating accurate and contextually appropriate answers.

Basics

This project integrates *Llama3-8B* with **Retrieval-Augmented Generation (RAG)** from Langchain to enable continuous knowledge updates.

RAG embeds both input queries and external documents in a shared vector space, retrieving relevant content to generate responses. This combination allows the model to continuously learn from external documents, making it well-suited for our goal of improving question-answering performance.

Initial testing showed issues such as content repetition, failure to retrieve key information, and inability to assess query relevance. To address these, we implemented relevancy checks, improved content selection, and fine-tuned the embedding model.

Evaluation Metrics

All the evaluation metrics below are scaled to a 0-1 range.

Retrieval Performance Metric

- **Hit Rate**: Evaluates retrieval performance by determining if top-k retrieved documents contain relevant documents.

Generation Performance Metrics

As introduced in Figure 1, four metrics are implemented to evaluate the generated answers: **Faithfulness**, **Context Precision**, **Context Recall**, and **Answer Relevancy**.

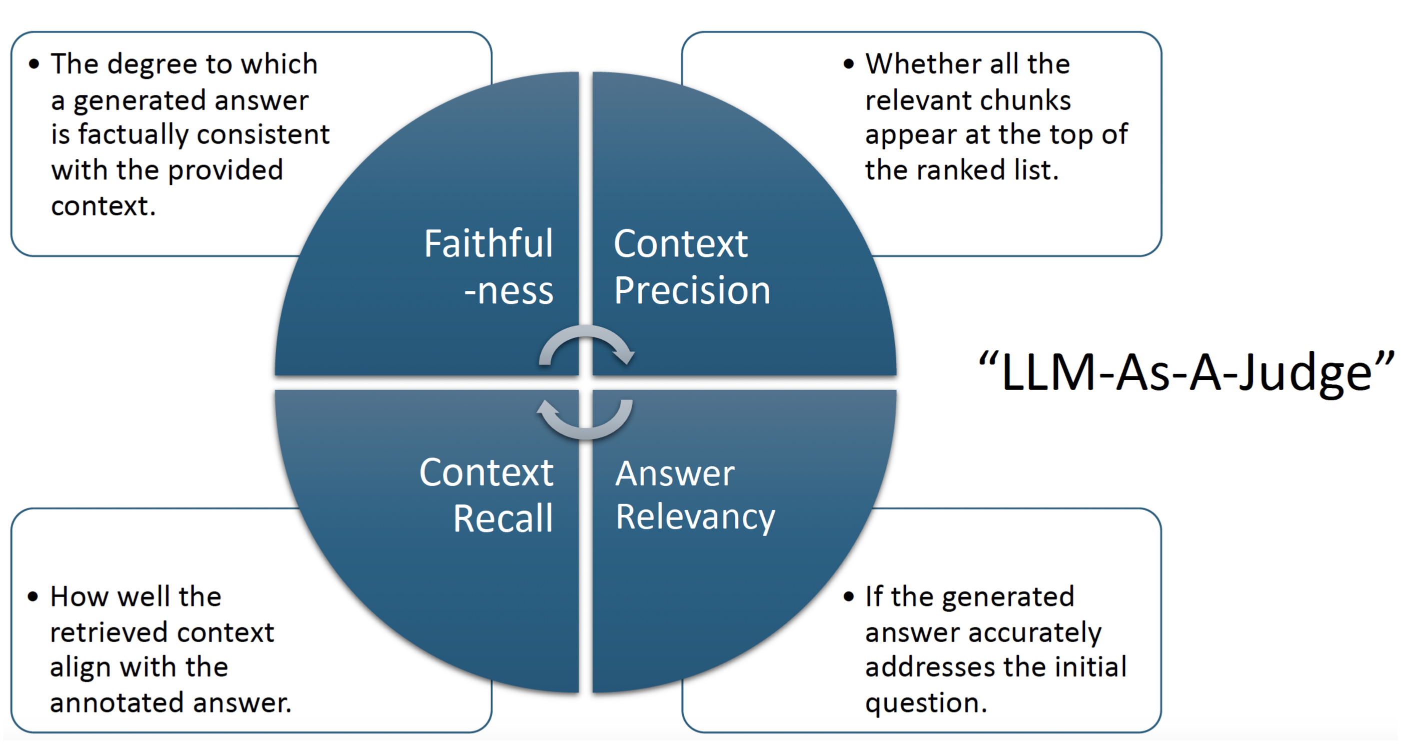

Figure 1. Four evaluation metrics as indicators of generation performance.

Methodology

Overall Mechanism

The overall workflow of the proposed model is depicted in Figure 2.

The process begins by assessing the relevance of a given query to the reference document. If the query is relevant, a hierarchical approach is used to identify the most pertinent section of the document. This selected section then serves as the input for the RAG baseline, which utilizes optimized hyperparameters and a fine-tuned embedding model to generate the answer.

The best-performing configuration combines the hierarchical selection approach with the fine-tuned *all-mpnet-base-v2* embedding model, using hyperparameters $max\_length = 650$, $chunk\_size = 2500$, and $chunk\_overlap = 200$.

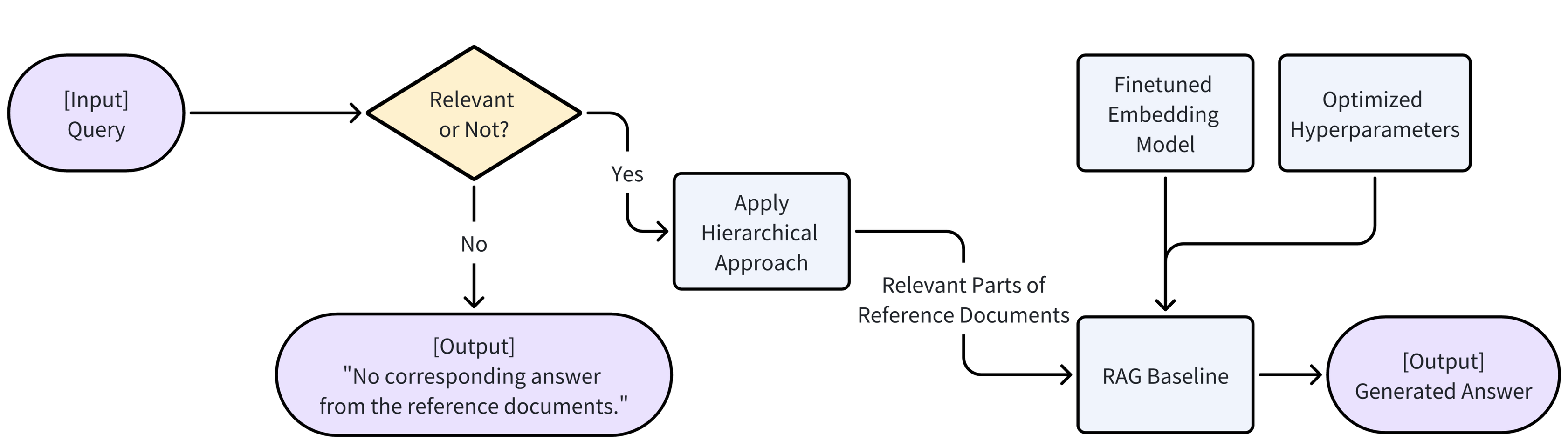

Figure 2. The overall model workflow.

昆山杜克大学
DUKE KUNSHAN UNIVERSITY

Methodology (continued)

Relevancy Check

- Implemented a relevancy check to improve efficiency by filtering out irrelevant queries, reducing unnecessary computational costs.
- Calculated cosine similarity between the query and retrieved documents.
- Set a threshold of 0.6 for similarity to determine query relevance, balancing performance and accuracy.

Hierarchical Approach

- Applied to manage large and structured reference documents efficiently, ensuring precise content selection.
- If a query is deemed relevant, the document is broken into smaller segments, and only the most pertinent sections are selected.
- This approach enhances the precision of RAG, allowing the model to focus on specific content, improving both the accuracy and contextual appropriateness of generated answers.

Hyperparameter Configuration

- Tested 75 different hyperparameter configurations to find the optimal balance.
- Considered the impact of max_length, $chunk\_size$, and $chunk\_overlap$:
  - Large max_length or $chunk\_overlap$ can cause repetition.
  - Inappropriate $chunk\_size$ may worsen information loss.
- Selected values that ensure comprehensive, accurate, and non-redundant responses.

Finetuning Embedding Model

- Used a manually verified dataset with $(query, context)$ pairs for finetuning.
- Evaluated the embedding model, *all-mpnet-base-v2*, before and after finetuning (see Figure 3).
- Observed improvements in most performance metrics, with a slight decrease in Answer Relevancy.

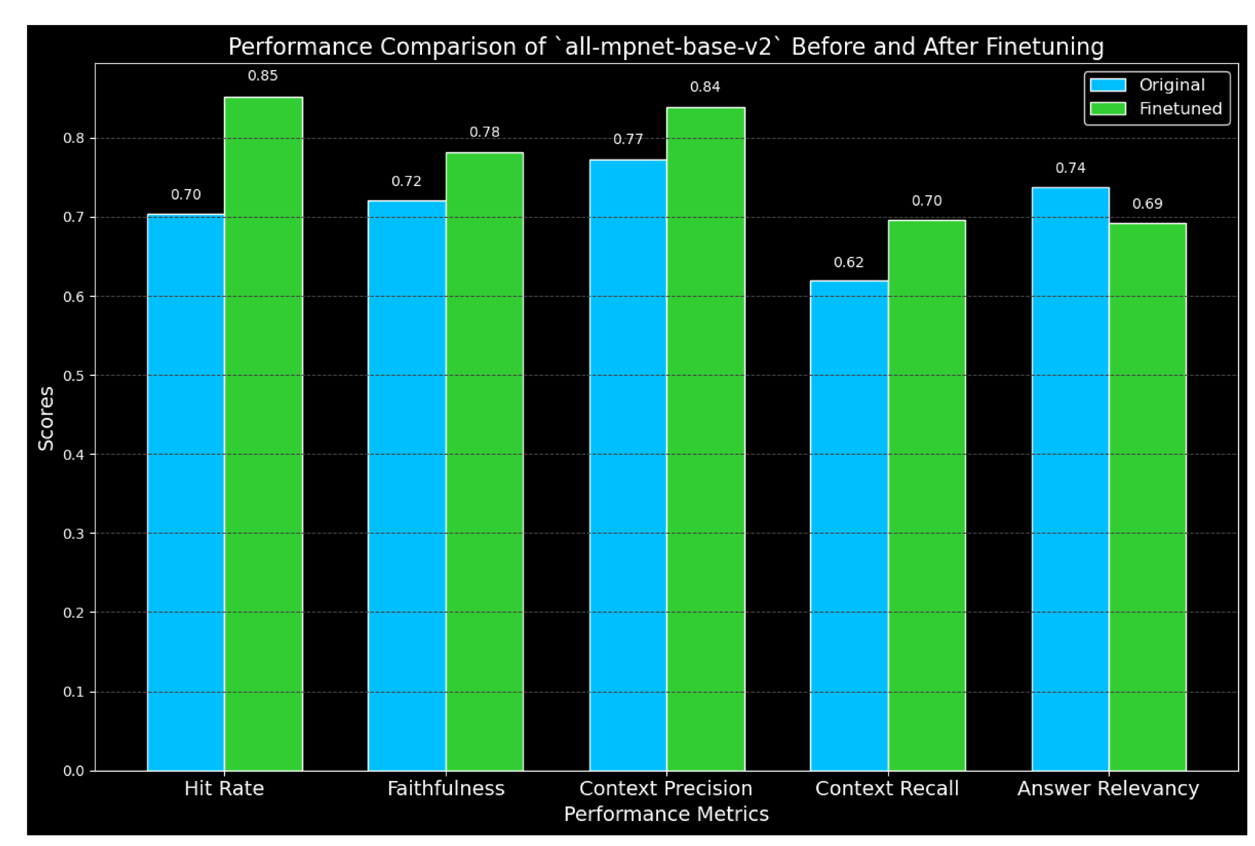

Figure 3. Visualization of Performance of *all-mpnet-base-v2* with and without finetuning

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. Overall, the final model enhanced the model's ability to retrieve and focus on relevant contexts, albeit with a minor trade-off in the final answer's overall relevance.

Future work involves exploring optimizing the hierarchical content selection further and incorporating real-time knowledge updates 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 Siegfreid 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, Ph.D.**.

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

References

[1] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert.
Ragas: Automated evaluation of retrieval augmented generation.
arXiv, 2023.
arXiv:2309.15217.

[2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang.
Retrieval-augmented generation for large language models: A survey.
arXiv, 2024.
arXiv:2312.10997.

[3] S. Xu.
Pulsar on kubernetes: Enabling kafka-on-pulsar (kop) with pulsar operators.
Medium, April 26 2023.

[4] W. X. Zhao, K. Zhou, 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, and J.-R. Wen.
A survey of large language models.
arXiv, 2023.
arXiv:2303.18223.