AI based RAG Capstone NSE talentsprint IIIT Hyd

Rahul, Sridhar, Bhanu, Santosh

February, 2025

Mentors and ProjectSupervisor:

- Gopichand, Lokesh
- Dr. Manish Shrivastava

Contents

1 Introduction and Problem Description

RAG systems are prone to hallucinations as the generator model struggles to retrieve relevant information from the context. Tuning an optimal system for a particular RAG application involves iterative evaluation of multiple configurations.

1.1 Dataset

We used the Hugging Face Ragbench dataset, available at https://huggingface.co/datasets/rungalileo/ragbench. This dataset is designed for evaluating Retrieval-Augmented Generation (RAG) systems and contains a variety of question-answer pairs along with relevant documents and evaluation metrics. We also used Dataset available in https://github.com/chen700564/RGB, we verified and evaluate LLM abilities required for RAG: Noise Robustness, Negative Rejection, Information Integration, Counterfactual Robustness The dataset is split into training, validation, and test sets:

1.2 Data Preprocessing

The ragbench dataset is already preprocessed and structured for RAG evaluation. It includes the following key fields:

- question: The input query
- documents: A sequence of relevant documents for each query
- response: The generated response
- documents_sentences: Sentences from the relevant documents
- response_sentences: Sentences from the generated response

- sentence_support_information: Information about how each response sentence is supported by the documents
- adherence_score: A boolean indicating overall adherence of the response to the documents

1.3 Models Used

We fine-tuned and evaluated several models for the task. The following table provides an overview of the models used:

Model	Training Notebook
Mistral, QWEN, DEEPSEEL	$\text{RAG}_{C}apstone_{g}rp2/Rahul_{l}lm_{R}AG_{g}rp2_{A}IML_{C}apstone.ipynb$
T5 Small and T5 base	Being small transformer models, were just used initially
GPT 3o mini	Due to high cost, only used for learning

Table 1: Models used for training and testing

1.4 RAG Implementation

The RAG implementation uses vector databases and embeddings for retrieving relevant documents. The following steps were implemented:

- Load Documents: Load the data into a pandas DataFrame.
- Chunking: Split documents into smaller chunks.
- Embedding: Generate embeddings for each chunk using Hugging Face embeddings.
- Vector Database: Store embeddings in a FAISS vector database.
- Retrieval: Retrieve relevant documents based on user queries.
- LLM Response: Generate responses using the LLM and retrieved documents.

1.5 Evaluation Metrics

We computed various evaluation metrics to assess the performance of the RAG system:

- Context Relevance: Measures the relevance of retrieved documents.
- Context Utilization: Measures the extent to which retrieved information is used by the LLM.
- Completeness: Measures how comprehensive the LLM's response is compared to the original answer.
- Adherence: Measures the factual consistency of the LLM's response.
- Noise Robustness: Measures the system's ability to handle noisy or irrelevant retrieved documents.
- Negative Rejection: Measures the system's ability to reject incorrect or unsupported queries.
- Counterfactual Robustness: Measures the system's resistance to counterfactual information.
- Information Integration: Measures the system's ability to integrate information from multiple sources.

  Noise Filtering: Noise filtered out early in process. Data cleaning from noise is very important. We have added Noise to text, so that evaluation of metrics can be verified with different noise level.

1.6 Evaluation Results

The evaluation results show the performance of the RAG system across different datasets and LLMs. Key metrics include context relevance, utilization, completeness, and adherence.

- Overall Performance: The RAG system demonstrates good performance, with high context relevance and utilization scores.
- Model Comparison: LLAMA3 and Qwen 0.5B models generally outperform DeepSeek in terms of response quality and adherence.
- Dataset Analysis: Performance varies across datasets, with some datasets showing better results due to the quality and relevance of the documents.

1.7 Observations and Further Reading

Key observations include the importance of context relevance and utilization in achieving high-quality responses. Further improvements can be made by fine-tuning the LLMs and optimizing the retrieval process. Models should have high precision in distinguishing relevant from irrelevant data. An ideal Noise Robustness value could be ¿90 percent accuracy in filtering out noisy or irrelevant information, similarly for negative rejection should reject answers when retrieval yields insufficient or poor-quality data. The rejection rate should be more than 95 percent when the retrieval system returns low-confidence or irrelevant documents. Information integration, ideally have more than 90 percent accuracy in synthesizing data from multiple sources. Counterfactual Robustness could be more than 95 percent in handling and rejecting flawed external information.

Links for Reading
1.https://arxiv.org/pdf/2309.01431
2. https://encr.pw/BlZfK
3. https://arxiv.org/pdf/2501.06713
4. https://arxiv.org/pdf/2408.09017

2 Conclusion

It was a great learning experience fine tuning these transformer models to perform a specific task. The nuances of encoder-decoder vs decoder-only models and how it could influence the performance was a great learning experience. There is still a significant amount of work needed to effectively apply RAG to LLMs. To ensure accurate and reliable responses from LLMs, it is crucial to exercise caution and carefully design for RAG

There were engineering issues that these large models caused and we had to figure out ways to deal with them along the way. This task has increased our understanding multifold.