Explainable Retrieval-Augmented Generation (RAG)

Group Number:

**28**

**Team Members:**

* Abhishek Ranjan
* Narendra Vyas
* Ankita Mukherjee
* Dharshak Sagar K M

# 1. Problem Statement:

Large Language Models (LLMs) are increasingly used for tasks like question answering (QA). However, they often generate responses without any transparency regarding their reasoning or the data used to derive these answers. This limits trust and applicability in critical domains like healthcare, legal analysis, and education.

This project addresses the lack of transparency in generative QA systems by developing an **Explainable Retrieval-Augmented Generation (RAG)** system. By integrating a retriever to source relevant information and designing prompts to generate explanations alongside answers, we aim to create an interpretable system that balances factual accuracy and human-readable reasoning.

# 2. Objective:

To build an explainable Retrieval-Augmented Generation (RAG) system that:

1. **Generates high-quality answers** to descriptive questions using fine-tuned QA models.

2. **Provides clear explanations** of the reasoning and information sources used in generating these answers.

3. Compares and analyses the system’s performance against ground truth to evaluate its accuracy and interpretability.

# 3. Literature Review:

## 3.1 Retrieval-Augmented Generation (RAG)

RAG combines two components:

1. **Retriever**: Retrieves relevant external information for a query (e.g., BM25, SPLADE).

2. **Generator**: Uses the retrieved information to generate natural language responses.

This hybrid approach has been shown to improve factual accuracy by grounding answers in reliable sources.

## 3.2 Explainability in QA Systems

Modern LLMs lack inherent transparency. Methods like:

• **Prompt Engineering** (e.g., “Explain your answer”)

• **Attention Visualization** (e.g., highlighting source tokens)

are used to improve explainability, but integrating these techniques with RAG is underexplored.

## 3.3 Existing Systems

• **OpenAI Codex and ChatGPT**: Generate answers but do not explain reasoning.

• **T5 and BART Models**: Pretrained QA models, but limited in transparency.

• **BM25 and SPLADE Retrievers**: Effective for information retrieval but not optimized for end-to-end QA integration.

# 4. Proposed Methodologies:

## 4.1 Stage 1: Dataset Selection and Model Fine-Tuning

• **Dataset**: Choose a descriptive QA dataset (e.g., SQuAD, HotpotQA).

• **Fine-Tuning**: Adapt a pre-trained language model (e.g., T5, BERT) for QA tasks using selected data.

## 4.2 Stage 2: Explanation Prompting

• Design specialized prompts to elicit explanations alongside answers.

• Examples:

• Question: “What is photosynthesis?”

• Prompt: “Explain your answer in detail.”

• Validate the quality of explanations through testing.

## 4.3 Stage 3: RAG Retriever Implementation

• Implement **BM25** and **SPLADE** retrievers for retrieving context for each query.

• Integrate the retrieved information into the QA model to enhance answers and explanations.

## 4.4 Stage 4: Comparison and Analysis

• **Comparison**: Evaluate generated answers and explanations against ground truth.

• **Metrics**: Use BLEU, ROUGE, and qualitative human evaluations for accuracy and interpretability.

# 5. Expected Outcomes in Stages:

|  |  |
| --- | --- |
| **Stage** | **Outcome** |
| Dataset Selection & Fine-Tuning | A QA model fine-tuned for descriptive question answering. |
| Explanation Prompting | Enhanced responses with human-readable explanations. |
| RAG Retriever Implementation | An integrated retriever-generator system for QA. |
| Comparison and Analysis | Insights into differences between standard and RAG-based systems. |

# 6. Applicability in the Real World:

## 6.1 Domains of Impact

1. **Healthcare**: Explainable diagnoses based on medical records.

2. **Legal**: Transparent case summaries and analysis.

3. **Education**: Reliable, interpretable answers to complex student queries.

## 6.2 Broader Implications

By making QA systems more interpretable, we address concerns like:

• Bias in model outputs.

• Accountability in high-stakes decision-making.

# 7. Project Timeline:

|  |  |  |
| --- | --- | --- |
| **Stage** | **Timeframe** | **Key Deliverables** |
| Dataset Selection & Fine-Tuning | Week 0–2 | Fine-tuned QA model on descriptive datasets. |
| Explanation Prompting | Week 3 | Prompt templates and explanation-enabled responses. |
| RAG Retriever Implementation | Week 4–5 | Integrated retriever-generator QA pipeline. |
| Comparison and Analysis | Week 6–8 | Performance metrics and comparative insights. |

# 8. Conclusion:

This project aims to demonstrate the potential of combining retrieval-augmented generation with explainability techniques to enhance the interpretability and trustworthiness of QA systems. The outcomes could lay the foundation for broader adoption in critical applications, setting new standards for transparency in AI.