INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

AIML B25 – PGCP

Real-World RAG System

Group -3

Srinivas Gundu
Kaveri Pulibandla
Deeksha Pandey
Rahul Pelluri

Project Mentor: Gopi Chand, Lokesh Madasu
Project Supervisor: Manish Shrivastava

May 24, 2025

TABLE OF CONTENTS:

1. INTRODUCTION:

Large Language Models (LLMs) possess impressive general capabilities but frequently suffer from providing outdated information or fabricating facts, a phenomenon known as hallucination. This inherent limitation significantly impacts their reliability and accuracy. Retrieval Augmented Generation (RAG) emerges as a promising technique to mitigate these issues and enhance the quality and accuracy of LLM responses, particularly within specific domains. RAG achieves this by combining the strengths of pretraining with retrieval-based methods. A key advantage of RAG is its capacity for rapid deployment of applications tailored for organizations and domains; this is possible without the need to retrain the LLM parameters, provided that relevant documents for queries are available.

This project proposes to construct an effective Retrieval-Augmented Generation (RAG) pipeline designed to deliver accurate answers to user queries. Simultaneously, it aims to undertake a systematic evaluation to understand the impact and performance of different Large Language Models when they are augmented with retrieved information. The goal is to identify current challenges and suggest avenues for future improvement.

2. PROBLEM STATEMENT:

Despite the notable promise of Retrieval-Augmented Generation (RAG) in addressing the limitations of standalone Large Language Models (LLMs), there remains a significant gap in systematic evaluation of how different LLM architectures respond to retrieval augmentation. The performance of RAG systems can vary widely depending on the model used and the configuration of the pipeline, yet there is a lack of comprehensive understanding of the underlying factors that influence this variability. Consequently, it is difficult to identify performance bottlenecks or develop generalizable best practices for building effective RAG systems.

Another key challenge stems from the nature of retrieved content in real-world scenarios. Retrieved documents often contain irrelevant information, noise, or even factual inaccuracies. Such content can negatively affect the LLM’s response quality. In some cases, LLMs may either ignore useful retrieved information or, more problematically, be misled by erroneous documents—prioritizing retrieved content over more accurate internal knowledge. This phenomenon, known as unreliable generation, presents a critical obstacle in building trustworthy and robust RAG systems.

Furthermore, the implementation of a RAG pipeline involves several interdependent processing stages, each with multiple design choices that impact overall performance. These include decisions about query classification, document chunking, embedding selection, retrieval and reranking strategies, and summarization techniques. Currently, there is no consensus or standardized framework for optimally configuring these components, making RAG system development both complex and domain specific.

This project aims to address these challenges by constructing a modular RAG pipeline and empirically evaluating the performance of various LLMs within this framework. The objective is to gain deeper insights into how retrieval augmentation interacts with LLMs, identify key performance determinants, and provide actionable recommendations for improving the reliability and effectiveness of RAG-based QA systems.

3. DATASET DESCRIPTION:

This project will primarily leverage two key benchmark datasets specifically designed for RAG evaluation: the **Retrieval-Augmented Generation Benchmark (RGB)** and **RAGBench**.

3.1 RAGBench

RAGBench: This is described as a **comprehensive, large-scale benchmark dataset** intended for training and evaluating RAG systems.

• It contains **100,000 examples**.

• These examples span **five distinct industry-specific domains**: Biomedical Research, General Knowledge, Legal, Customer Support, and Finance.

AI based generative QA system

• The data is **sourced from real-world industry corpora**, such as user manuals, ensuring its relevance for practical applications.

• RAGBench supports **various RAG task types**.

• It includes evaluation metrics such as context relevance, context utilization, answer faithfulness (referred to as Adherence in the TRACe framework), and answer completeness.

• The dataset provides **explainable labels** to facilitate holistic evaluation and offer actionable feedback for improving production applications.

3.2 RGB Benchmark

Retrieval-Augmented Generation Benchmark (RGB): This corpus was specifically established for evaluating RAG systems in both English and Chinese.

• To mitigate bias from the LLMs' internal knowledge, RGB instances are constructed using **latest news information**.

• The corpus is organised into **four separate testbeds** based on fundamental abilities required for RAG: **Noise**

Robustness, **Negative Rejection**, **Information Integration**, and **Counterfactual Robustness**.

• RGB includes a total of **600 base questions**, with an additional 200 questions for evaluating information integration and another 200 for counterfactual robustness. These are equally divided between English and Chinese instances.

External documents are retrieved from the Internet using a Search API.

4. METHODOLOGY:

The development of the proposed Retrieval-Augmented Generation (RAG) system will follow a structured and empirical approach, grounded in current research and best practices. The system architecture will be composed of modular components that reflect key stages in the RAG pipeline, including Query Classification, Retrieval, Reranking, Repacking, and Summarization. Each stage will be designed to support configurability and experimentation to identify optimal configurations across multiple evaluation criteria.

A critical focus of the methodology will be the systematic exploration of implementation choices available at each stage. This includes:

- **Query Classification:** Determining whether a query requires retrieval-augmented generation, which helps in minimizing unnecessary retrieval operations and optimizing system efficiency.
- **Document Chunking**: Investigating chunking strategies such as fixed-size segmentation, sliding windows, and dynamic chunking (e.g., small-to-big chunking) to maintain contextual coherence and improve retrieval performance. Metadata enrichment for each chunk will also be explored to support enhanced semantic matching.
- **Embedding Models:** Evaluating state-of-the-art embedding techniques, such as LLM-Embedder and the BAAI/bge family of models, for encoding both queries and document chunks. These models will be assessed based on semantic matching accuracy and computational efficiency.
- **Vector Databases:** Selecting and configuring high-performance vector databases (e.g., Milvus) for efficient storage and retrieval of embeddings. Consideration will be given to scalability, query latency, and integration compatibility.

- **Retrieval Techniques:** Implementing and comparing dense retrieval, sparse retrieval (e.g., BM25), and hybrid approaches. Additionally, advanced query transformation techniques, such as query rewriting, decomposition, and pseudo-document generation (e.g., HyDE), will be investigated to improve retrieval recall and precision.
- **Reranking Methods:** Enhancing initial retrieval outputs through reranking strategies that prioritize relevance. Models such as monoT5, monoBERT, RankLLaMA, and TILDE will be evaluated for their effectiveness in improving downstream generation quality.
- **Repacking Strategies:** Reordering and structuring retrieved documents before generation using techniques like forward, reverse, and sides configurations. These strategies aim to position critical information optimally within the input context.
- **Summarization:** Applying summarization to reduce context length while preserving informativeness. Both extractive and abstractive methods (e.g., Recomp, LongLLMLingua) will be examined for their potential to enhance generation efficiency and accuracy.
- **Generator Model:** Utilizing advanced LLMs for answer generation. The generator may be fine-tuned using a mix of relevant and distractor documents to improve robustness against irrelevant or misleading context, while reinforcing accurate use of relevant information.

Throughout the development lifecycle, an iterative experimentation process will be employed to assess combinations of these components. The evaluation will be guided by domain-specific datasets and a comprehensive set of metrics (as described in Section 5), ensuring that both effectiveness and efficiency are systematically measured.

5. EVALUATION METRICS:

5.1 RAG-Specific Metrics

The project will employ a comprehensive suite of metrics to evaluate both the performance of the RAG system components and the capabilities of the LLMs within the RAG framework.

The TRACe evaluation framework from RAGBench will be a core component for evaluating the quality of the retriever and the response generator. TRACe measures four dimensions:

• **Relevance**: Assesses the quality of the retriever's output with respect to the query. It measures whether the provided context includes specific information needed to answer the question accurately.

• **Utilization**: Measures how effectively the generation model uses the retrieved information in its response.

- **Adherence**: Measures how well the LLM's output adheres to the information in the source context. It is synonymous with faithfulness or groundedness and indicates whether the response is strictly based on the provided context without

introducing hallucinations.

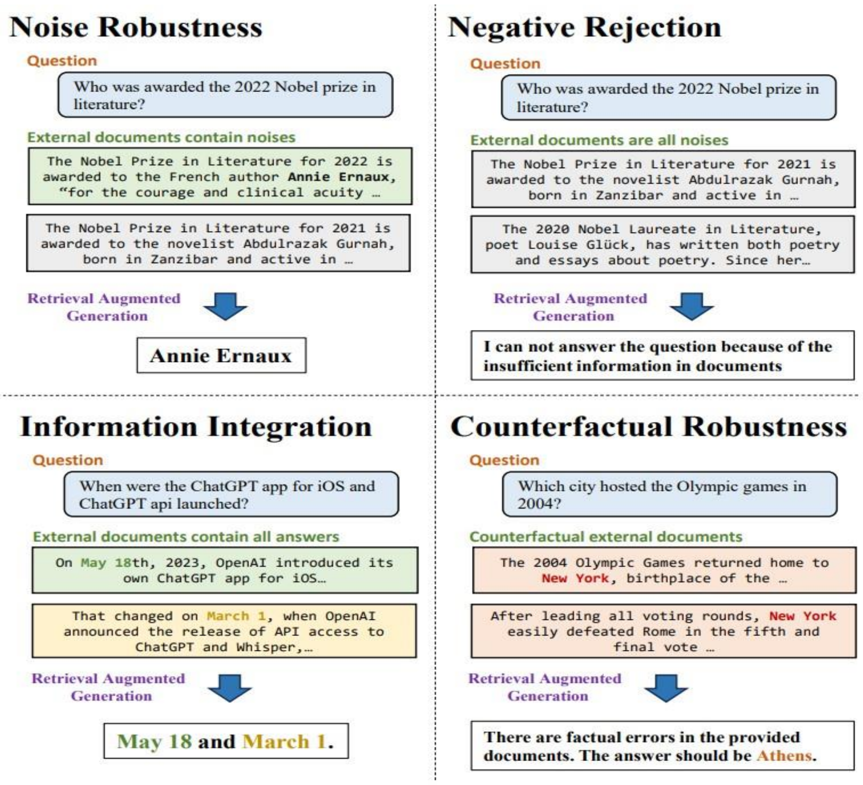
- **Completeness**: Measures how well the response incorporates all the relevant information identified in the context.

These metrics are designed to provide granular and actionable insights into the RAG system's performance. RAGBench provides ground-truth labels for Adherence, Relevance, and Utilization, often derived using an LLM annotator (such as GPT-4).

5.2 LLM Abilities Evaluation

Additionally, evaluation will draw upon the fundamental abilities defined in the RGB benchmark and their corresponding metrics:

Abilities of LLM for RAG


Noise Robustness
Question
External documents contain noises
Retrieval Augmented Generation

Who was awarded the 2022 Nobel prize in literature?

The Nobel Prize in Literature for 2022 is awarded to the French author **Annie Ernaux**, "for the courage and clinical acuity …

The Nobel Prize in Literature for 2021 is awarded to the novelist Abdulrazak Gurnah, born in Zanzibar and active in …

Annie Ernaux

Negative Rejection
Question
External documents are all noises
Retrieval Augmented Generation

Who was awarded the 2022 Nobel prize in literature?

The Nobel Prize in Literature for 2021 is awarded to the novelist Abdulrazak Gurnah, born in Zanzibar and active in …

The 2020 Nobel Laureate in Literature, poet Louise Glück, has written both poetry and essays about poetry. Since her…

I can not answer the question because of the insufficient information in documents

Information Integration
Question
External documents contain all answers
Retrieval Augmented Generation

When were the ChatGPT app for iOS and ChatGPT api launched?

On May 18th, 2023, OpenAI introduced its own ChatGPT app for iOS…

That changed on March 1, when OpenAI announced the release of API access to ChatGPT and Whisper,…

May 18 and March 1.

Counterfactual Robustness
Question
Counterfactual external documents
Retrieval Augmented Generation

Which city hosted the Olympic games in 2004?

The 2004 Olympic Games returned home to New York, birthplace of the …

After leading all voting rounds, New York easily defeated Rome in the fifth and final vote …

There are factual errors in the provided documents. The answer should be Athens.

Figure 1: Illustration of 4 kinds of abilities required for retrieval-augmented generation of LLMs.

• **Noise Robustness:** The LLM's ability to extract useful information even when noisy documents are present. This is evaluated using Accuracy.

AI based generative QA system

• **Negative Rejection:** The LLM's ability to decline answering when the required knowledge is not found in the retrieved documents. This is evaluated using the Rejection Rate.

• **Information Integration:** The ability to answer complex questions by synthesizing information from multiple retrieved documents. This is evaluated using Accuracy.

• **Counterfactual Robustness:** The ability to identify known factual errors within retrieved documents (especially when warned) and generate the correct answer using the LLM's internal knowledge. This is evaluated using Error Detection Rate and Error Correction Rate.

Evaluation will be conducted across the diverse domains and task types present in the selected datasets to provide a holistic assessment.

6. DEPLOYMENT PLAN:

To ensure accessibility and usability, the system will be deployed using:

• **API Layer**: FastAPI – backend endpoints for model inference.

• **Frontend UI**: Streamlit – interactive interface for user queries.

• **Hosting Platforms**:

- Heroku or Google Cloud Run for serverless deployment.
- GitHub for code repository and CI/CD pipelines.

7. CHALLENGES:

Several challenges are anticipated during the development of this RAG system:

• **Hallucinations:** While RAG reduces hallucinations, they can still occur if the retrieval fails or the generator misinterprets the context.

• **Tuning Optimal Configuration:** Finding the best combination of data collection strategies, chunking methods, embedding models, retrieval techniques, and fine-tuning parameters requires extensive iterative evaluation.

• **Monitoring and Continuous Improvement:** Implementing effective tracking of system performance, prompt usage, user feedback, and model drift to continuously improve the system.

• **Data Diversity and Bias:** Ensuring the collected data is sufficiently diverse to handle various query types and is free from biases or misinformation that could be reflected in the generated content.

• **Capturing Contextual Nuances and Style:** Accurately and consistently capturing the subtle nuances of a person's writing style and personality is complex.

• **LLM Reliability for Structured Output:** Ensuring LLMs reliably produce structured data during synthetic data generation or self-querying steps may require specific techniques (e.g., JSON mode).

• **Judge LLM Biases:** Awareness and mitigation of potential biases in LLMs used for evaluation.

8. SUMMARY OF RESEARCH PAPERS:

8.1 RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems

- Introduces a domain-specific, explainable benchmark tailored for evaluating RAG systems in real-world industry contexts. · Proposes the TRACe evaluation framework, focusing on four core metrics: faithfulness (adherence), relevance, completeness, and utilization.
- Highlights the importance of explainability and diagnostic feedback in evaluating RAG performance and guiding practical improvements.
- Demonstrates the significance of benchmarking on diverse domains (biomedical, legal, finance, etc.), enabling fine-grained analysis of model behavior.
- Reinforces the need for robust annotation strategies using large LLMs (e.g., GPT-4) to establish high-quality evaluation datasets.

8.2 RGB: Retrieval-Augmented Generation with Backward Reasoning

- Proposes a novel Backward Reasoning paradigm that reverses the conventional RAG workflow: generation is performed first, followed by retrieval validation.
- This dual-step architecture increases factual robustness by allowing the system to fact-check generated responses using retrieved evidence.
- Defines four challenging testbeds: Noise Robustness, Negative Rejection, Information Integration, and Counterfactual Robustness.
- Showcases the effectiveness of the RGB benchmark in evaluating LLMs’ behavior under controlled perturbations and adversarial conditions.
- Provides actionable strategies for improving model reliability when handling noisy or misleading retrievals.

8.3 Best Practices in Retrieval-Augmented Generation (EMNLP 2024)

- Offers a systematic study of over 150 RAG configurations across multiple tasks, LLMs, and domains.
- Emphasizes the critical role of chunking strategies, particularly highlighting the trade-offs between fixed-size and semantic-based chunking.
- Identifies the impact of reranking methods and repacking formats on the reliability and relevance of generated responses.
- Benchmarks popular vector databases and embedding models, presenting guidance on infrastructure-level optimization.
- Highlights the value of prompt engineering and adaptive response formatting in enhancing LLM alignment with retrieved content. · Concludes with a set of consolidated best practices for real-world RAG system design.

9. DELIVERABLES:

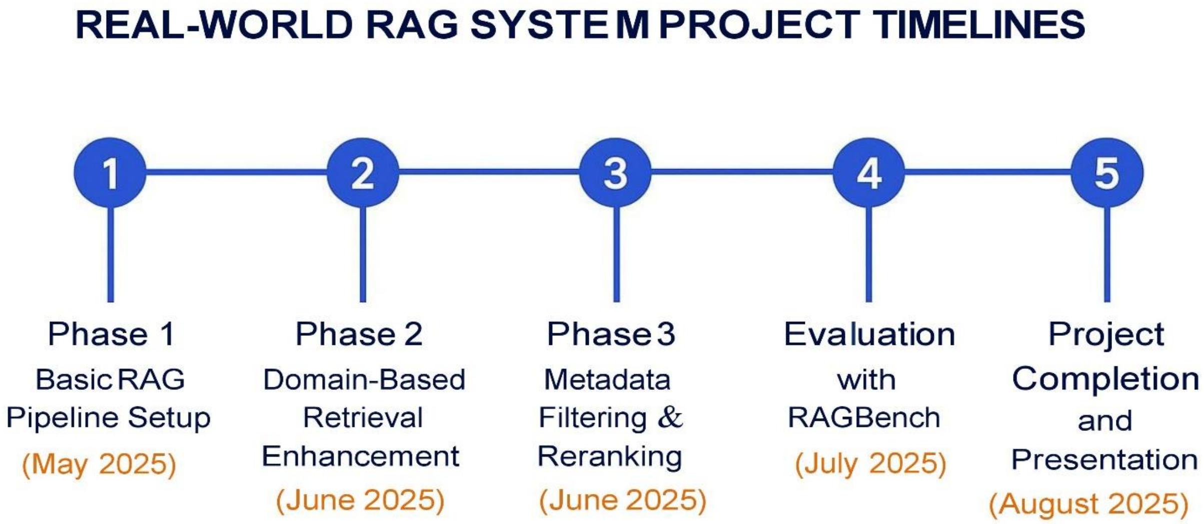
1. **Technical Report**: Full pipeline architecture, ablation studies.

2. **Code**: GitHub repo with modular RAG implementation.

3. **Presentation**: Demo video + performance benchmarks.

4. **Research Summary**: 3-page synthesis of key findings from papers.

10. PROJECT TIMELINE:


REAL-WORLD RAG SYSTEM PROJECT TIMELINES
1
2
3
4
5
Phase 1
Basic RAG Pipeline Setup
(May 2025)
Phase 2
Domain-Based Retrieval Enhancement
(June 2025)
Phase 3
Metadata Filtering & Reranking
(June 2025)
Evaluation with RAGBench
(July 2025)
Project Completion and Presentation
(August 2025)

11. REFERENCES:

1. RAGBench: Explainable Benchmark for Retrieval Augmented Generation Systems
2. Searching for Best Practices in Retrieval Augmented Generation
3. Benchmarking Large Language Models in Retrieval Augmented Generation