Vivekanand Education Society's Institute of Technology
Department of Computer Engineering

Group No : 7 Date : 03.08.2023

BE Project Synopsis (2024-25) - Sem VII

RAG (Retrieval Augmented Generation)

Mentor Name : Dr. Mrs. Gresha Bhatia
Deputy H.O.D. , CMPN

Sustainable Development Goal : Industry,Innovation & Infrastructure

Aryan Girish Raje
V.E.S.I.T
d2021.aryan.raje@ves.ac.in

Arya Girish Raje
V.E.S.I.T
d2021.arya.raje@ves.ac.in

Prasad Kishor Lahane
V.E.S.I.T
d2021.prasad.lahane@ves.ac.in

Ishita Sudhir Marathe
V.E.S.I.T
2021.ishita.marathe@ves.ac.in

Domain: **AI, Deep Learning and Data Warehousing and Mining**

Abstract

This research focuses on the development and implementation of a chatbot utilizing Retrieval-Augmented Generation (RAG) for improved conversational skills. The system generates real-time, contextually appropriate responses, drawing from a comprehensive dataset that includes a knowledge base and external sources. By blending retrieval-based and generative techniques, the chatbot delivers precise and informative responses. The proposed system seeks to enhance user experience across the domain of railways,ranging towards its applications and benefits such as customer support, information sharing, and personalized interactions. The results contribute to the field of conversational AI by improving communication effectiveness and user satisfaction.

Introduction

Retrieval augmented generation, or RAG, is an architectural approach that can improve the efficacy of large language model (LLM) applications by leveraging custom data. This is done by retrieving data/documents relevant to a question or task and providing them as context for the LLM. RAG has shown success in supporting chatbots and Q&A systems that need to maintain up-to-date information or access domain-specific knowledge.

In the modern railway industry, the efficient management and analysis of vast amounts of operational data, maintenance records, safety reports, and customer feedback are critical for ensuring smooth operations and high-quality service. With the increasing complexity and volume of data, traditional methods of data analysis and reporting often fall short in providing timely, actionable insights.

To address these challenges, this project explores the application of Retrieval Augmented Generation (RAG) to the railway sector. RAG is an advanced artificial intelligence technique that combines retrieval-based methods with generative language models to enhance information retrieval and content generation. By leveraging this approach, we aim to improve the accuracy, relevance, and comprehensiveness of reports and responses related to railway operations.

The primary objectives of this project are:

1. **Enhanced Reporting:** Develop a system that utilizes RAG to automatically generate detailed and insightful reports based on historical and real-time railway data. This includes operational performance, maintenance status, and safety incidents.
2. **Advanced Query Handling:** Create a tool that can answer complex queries about railway operations by retrieving relevant data and generating coherent, contextually accurate responses.
3. **Data Summarization:** Implement a solution that can summarize large volumes of textual and numerical data, providing stakeholders with concise and actionable insights.

Problem Statement

Organizations in customer service, healthcare, and education often encounter limitations with traditional chatbots due to their restricted knowledge and context capabilities. To address this issue, an advanced Retrieval-Augmented Generation (RAG) chatbot is needed, which can pull relevant information from large databases and generate precise, contextually appropriate responses.

The railway industry, characterized by its complexity and scale, faces significant challenges in managing and analyzing large volumes of data generated from various sources such as operational records, maintenance logs, safety reports, and customer feedback. These challenges hinder the ability to generate timely, accurate, and actionable insights, which are crucial for effective decision-making and operational efficiency

This approach seeks to enhance user experience, adapt to different fields, and manage high volumes of queries effectively. Challenges include ensuring data accuracy, maintaining context throughout interactions, data summarization, complex query handling, manual reporting limitations and data fragmentation The ultimate aim is to boost user satisfaction, operational efficiency, and access to current information.

Proposed Solution

We propose a Retrieval-Augmented Generation (RAG) chatbot to enhance customer support, healthcare, education, and internal knowledge management. By combining information retrieval with natural language generation, it will provide accurate, context-aware responses.

To address these challenges, this project proposes the development of a Retrieval Augmented Generation (RAG) system tailored for the railway sector. The RAG system aims to enhance the management, analysis, and reporting of railway data by integrating advanced retrieval and generative capabilities.

Retrieval System:

- **Objective:** Efficiently index and search railway data (operational records, maintenance logs, safety reports, customer feedback).
- **Functionality:** Fetch relevant documents based on user queries for targeted data retrieval.

Generative Model Integration:

- **Objective:** Use a pre-trained language model to generate accurate and coherent responses.
- **Functionality:** Produce detailed reports, answer complex queries, and summarize data using the context from the retrieval system.

Application Development:

- **Objective:** Develop a user-friendly application integrating retrieval and generative models.
- **Functionality:** Support real-time query handling, report generation, and data summarization.

Testing and Validation:

- **Objective:** Ensure the system's accuracy and effectiveness in real-world scenarios.
- **Functionality:** Perform rigorous testing and adjust based on user feedback for optimal performance.

Methodology

1. Create external data

The new data outside of the LLM's original training data set is called external data. It can come from multiple data sources, such as a APIs, databases, or document repositories. The data may exist in various formats like files, database records, or long-form text. Another AI technique, called embedding language models, converts data into numerical representations and stores it in a vector database. This process creates a knowledge library that the generative AI models can understand.

2. Retrieve relevant information

The next step is to perform a relevancy search. The user query is converted to a vector representation and matched with the vector databases. For example, consider a smart chatbot that can answer human resource questions for an organization. If an employee searches, "How much annual leave do I have?" the system will retrieve annual leave policy documents alongside the individual employee's past leave record. These specific documents will be returned because they are highly-relevant to what the employee has input. The relevancy was calculated and established using mathematical vector calculations and representations.

3. Augment the LLM prompt

Next, the RAG model augments the user input (or prompts) by adding the relevant retrieved data in context. This step uses prompt engineering techniques to communicate effectively with the LLM. The augmented prompt allows the large language models to generate an accurate answer to user queries.

4. Update external data

The next question may be—what if the external data becomes stale? To maintain current information for retrieval, asynchronously update the documents and update embedding representation of the documents. You can do this through automated real-time processes or periodic batch processing. This is a common challenge in data analytics—different data-science approaches to change management can be used.

The following diagram shows the conceptual flow of using RAG with LLMs.

Block Diagram

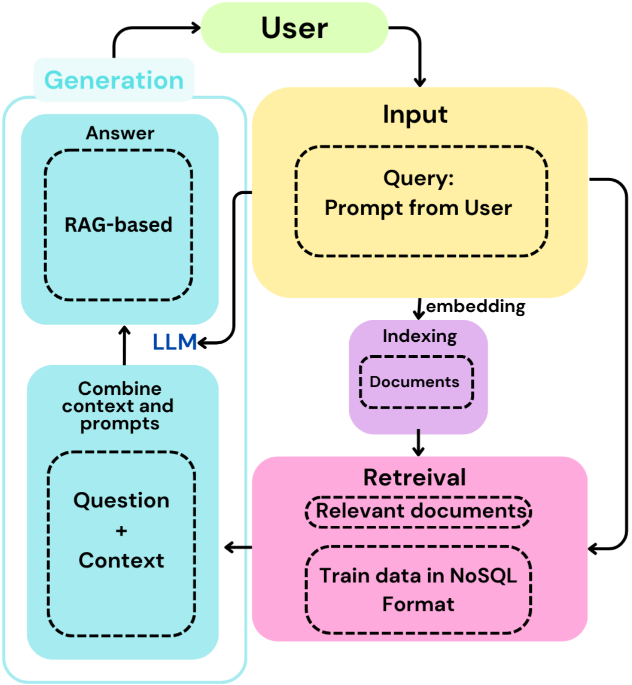

FIGURE 1.1: METHODOLOGY BLOCK DIAGRAM

Hardware , Software and tools Requirements

Building and implementing a RAG that provides advantages for classic LLM models requires prerequisite knowledge of NLP and basic system requirements.

Hardware Requirements:

1.System:

Processor- 12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz

Installed RAM16.0 GB (15.7 GB usable)

System type 64-bit operating system, x64-based processor

2. CPUs/GPUs: 2.4 GHz (Base) - 4.2 GHz (Max) - 4 Cores - 8 Threads - 8 MB Cache

3. RAM: 8/16 GB RAM

4. Storage: 1 TD + 256 SSD | Cloud Storage

5. Real-Time Data Sources: APIs for Indian Railways data

Software Requirements:

1. Large Language Model: OpenAI, Llama2/3, Hugging Face, etc

2. Python Programming Language: Python 3.8.X+, Jupyter, Anaconda

3. Data Preprocessing Libraries: Pandas (Python Data Analysis), Numpy (Numerical Python)

4. Real-Time Data Streaming Tools: APIs for Indian Railways Data

5. Visualization Tools: MatPlotLib (graph plotting library)

6. Web Application Framework: Flask, Streamlit. etc

7. Database Management System: NoSQL, GraphQL etc

8. Operating System: Windows OS

Proposed Evaluation Measures

1.1. Precision at K (P@K):

- Definition: Measures the proportion of relevant documents in the top K retrieved documents.
- Purpose: Ensures that the most relevant documents are ranked highly by the retrieval component.

1.2. Recall at K (R@K):

- Definition: Measures the proportion of relevant documents retrieved out of all relevant documents available.
- Purpose: Assesses the system's ability to retrieve all relevant documents within the top K results.

1.3. Mean Average Precision (MAP):

- Definition: Average of precision scores at each relevant document's position.
- Purpose: Provides a single-figure measure of retrieval performance considering the precision of relevant documents in the ranked list.

1.4. Normalized Discounted Cumulative Gain (NDCG):

- Definition: Measures the gain of retrieving relevant documents based on their position in the ranked list, with a discount factor.
- Purpose: Reflects the usefulness of retrieving relevant documents earlier in the list.

2. Generation Performance

2.1. Exact Match (EM):

- Definition: Measures the percentage of generated responses that exactly match the reference answers.
- Purpose: Evaluates the accuracy of the generated text.

2.2. F1 Score:

- Definition: Harmonic mean of precision and recall for the generated responses.

- Purpose: Provides a balance between precision and recall, suitable for evaluating responses where exact matches are less common.

2.3. BLEU Score:

- Definition: Measures the overlap of n-grams between the generated text and reference answers.
- Purpose: Assesses the quality of the generated text compared to human-written references.

2.4. ROUGE Score:

- Definition: Measures recall-oriented metrics such as the overlap of n-grams, word sequences, and word pairs between generated and reference text.
- Purpose: Evaluates the quality of the generated text in terms of recall and coverage.

3. System Effectiveness

3.1. Query Coverage:

- Definition: Measures the proportion of user queries for which the system provides a relevant answer.
- Purpose: Assesses how well the system handles a diverse range of user queries.

3.2. Response Completeness:

- Definition: Evaluates whether the responses cover all aspects of the query or question.
- Purpose: Ensures that the system provides comprehensive answers.

3.3. Response Coherence:

- Definition: Measures how logically consistent and relevant the generated responses are to the input queries.
- Purpose: Evaluates the quality and relevance of the responses generated by the system.

4. User Experience

4.1. User Satisfaction:

- Definition: Surveys or feedback collected from users about their experience with the system.

- Purpose: Provides insights into user satisfaction with the system's performance and usability.

4.2. Response Time:

- Definition: Measures the time taken by the system to retrieve and generate responses.
- Purpose: Assesses the system's efficiency and performance in real-time scenarios.

5. Failure Analysis

5.1. Failure Point Rate:

- Definition: Measures the frequency of specific failure points (e.g., missed documents, incorrect answers).
- Purpose: Identifies and quantifies common issues and areas for improvement.

5.2. Error Classification:

- Definition: Categorizes errors based on the identified failure points (e.g., missing content, incorrect specificity).
- Purpose: Helps in understanding the types of errors occurring and prioritizing fixes.

Conclusion

The development and implementation of a Retrieval Augmented Generation (RAG) system for the railway sector represents a significant advancement in how data is managed, analyzed, and reported. By combining sophisticated retrieval techniques with powerful generative language models, this project addresses critical challenges faced by the railway industry in handling large volumes of diverse data.By integrating advanced retrieval and generative capabilities, the RAG system not only enhances the efficiency of data processing but also supports a more proactive and responsive approach to managing railway operations. This innovative solution is poised to transform how the railway sector leverages data, ultimately contributing to more effective and efficient railway operations and improved service quality.

References

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", 22 May 2020 (v1), 12 Apr 2021 (this version, v4)

[2] Isamu Isozaki, "Literature Review on RAG(Retrieval Augmented Generation) for Custom Domains", Nov 26, 2023

[3] Kieran Pichai, "A Retrieval-Augmented Generation Based Large Language Model Benchmarked On a Novel Dataset", November 2023, Journal of Student Research, DOI:10.47611/jsrhs.v12i4.6213, License :CC BY-NC-SA 4.0

[4] Shouvik Sanyal, Alam Ahmad, Hafiz Wasim Akram, "An Analysis of Performance of Indian Railways", January 2021, DOI:10.1504/IJLSM.2021.10043738

[5] Mohd Arshad, Muqeem Ahmed, "Prediction of Train Delay in Indian Railways through Machine Learning Techniques ", February 2019, International Journal of Computer Sciences and Engineering 7(2):405-4117(2):405-411,
DOI:10.26438/ijcse/v7i2.405411.

[6] Mohd Arshad, Muqeem Ahmed, "Train Delay Estimation in Indian Railways by Including Weather Factors Through Machine Learning Techniques", September 2019, DOI:10.2174/2666255813666190912095739.