International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Enhancing Large Language Models with a Hybrid Retrieval-Augmented Generation System: A Comparative Analysis

Mrs. S. A. Behle
Project Guide, Department of Artificial Intelligence and Data Science, AISSMS IOIT Pune

Parth Barse
Department of Artificial Intelligence and Data Science, AISSMS IOIT Pune

Dheeraj Chingunde
Department of Artificial Intelligence and Data Science, AISSMS IOIT Pune

Rutuja Katkar
Department of Artificial Intelligence and Data Science, AISSMS IOIT Pune

Vansh Koul
Department of Artificial Intelligence and Data Science, AISSMS IOIT Pune

Abstract:
Organizations are increasingly challenged with maintaining data privacy when utilizing cloud-based AI services for natural language processing tasks. This project aims to tackle these issues by creating a secure, on-premise system that employs Retrieval-Augmented Generation (RAG) to allow organizations to query their internal data safely. Our solution combines transformer models with document retrieval techniques to produce contextually relevant answers to user queries, all while ensuring that data stays within the organization's local network. This method assists organizations in adhering to strict data privacy regulations such as GDPR and HIPAA. Built entirely in Python, our system is designed for scalability, flexibility, and seamless integration with existing infrastructure.

I. INTRODUCTION

The rise of remote work and digital transformation has significantly increased data generation within organizations. Alongside this growth, concerns about data privacy have also escalated, particularly when utilizing cloud-based AI solutions for natural language processing (NLP). These solutions often necessitate the transmission of sensitive data to third-party servers, which can put organizations at risk of data breaches and compliance issues. Our project seeks to address these challenges by creating a secure, on-premise system for querying organizational data through Retrieval-Augmented Generation (RAG). By processing all data on local servers, our solution enables organizations to leverage AI's capabilities while maintaining complete control over their sensitive information. The system is tailored to support various industries that demand strict data confidentiality, including finance, healthcare, and legal sectors.

II. LITERATURE SURVEY

II. LITERATURE SURVEY

1. Robust Multi-Model RAG Pipeline for Documents with Text, Table & Images (2024). Authors: Pankaj Joshi, Aditya Gupta, Pankaj Kumar, Manas Sisodia. In this research, the Multi-Model Retrieval Augmented Generation (MuRAG) pipeline is introduced and tested [1]. The process starts with extracting and representing text and images from documents with their relationships intact and then generating a consistent knowledge base. The pipeline was tested stringently across different question-answering datasets and proved to yield better retrieval and generation results. One of the strengths of this pipeline lies in its capacity to process the complex interactions between text and images, a feature that is pivotal in files that have both types of content [1].

2. Retrieval-Augmented Generation for Large Language Models: A Survey (2024). Authors: Researchers based at Tongji University and Fudan University, China. This paper offers an extensive review of RAG methods for large language models (LLMs), covering Naive, Advanced, and Modular RAG frameworks [2]. The authors present retrieval, generation, and augmentation approaches, assessing their efficiency in refining the accuracy, reliability, and traceability of LLMs. The review highlights the significance of utilizing external knowledge to enhance the performance of LLMs, especially in overcoming hallucinations and outdated information problems. The study finds that RAG techniques markedly enhance LLM performance, providing more credible and informed outputs [2].

3. Retrieval-Augmented Response Generation for Knowledge-Grounded Conversation in the Wild (2022). Authors: Ahn, Lee, Shim, and Park. The authors introduce a retrieval-augmented model that combines topic consistency, dual-matching reranking, and data-weighting to improve the relevance and diversity of the generated responses [3]. One key takeaway from the research is that the model performs better than baseline models in producing responses not only that are informative but also contextually applicable and rooted in external documents. This was shown through automatic as well as human testing, which established the usefulness of the model in producing correct and coherent conversation responses.

4. Automation of Text Summarization Using Hugging Face NLP (2024). Authors: Asmitha M, Aashritha Danda, Hemanth Bysani, Rimjhim Padam Singh, Sneha Kanchan. The article "Automation of Text Summarization Using Hugging Face NLP" (2024) analyzes several summarization models including BERT, GPT, Pegasus, and Hugging Face's mbart-large-cc25 [5]. With a focus on extractive and abstractive methods, the research finds that fine-tuning Hugging Face's model on the CNN/DailyMail dataset delivers the highest performance in producing short and contextually appropriate summaries. The assessment, based on ROUGE scores, determined that this model produced high summarization accuracy, which made the model a valuable asset in automating text summarization.

5. Transformers' Capacity for Implementing in Solving Language Processing Intricacies. Authors: Paras Nath Singh & Sagarika Behera. This study investigates the potential of transformer models like BERT, GPT, and BART in processing multiple NLP tasks [4]. The research shows the use of the models in part-of-speech (POS) tagging, sentiment analysis, machine translation, paraphrasing, and summarization, with well-known Python-based libraries such as PyTorch, TensorFlow, and Hugging Face. The main result of this study is that transformer-based models perform better than standard LSTM/RNN models with high accuracy levels (94%-98%) in multilingual sentiment analysis and provide fast and accurate NLP results across many tasks.

Comparative Analysis:

Feature	Existing Systems	Proposed System (YARVIS)
Data Modalities	Limited to text-image; some support for plain text (e.g., MuRAG)	Supports text and enhances document processing
Re-Ranking Mechanism	Basic dual-matching in conversational RAG	Hybrid re-ranking with dual-matching, topic relevance, and context continuity
Retrieval Efficiency	Standard retrieval with limited speed for large data	Vector-based database integration, improving response time and scalability
Multi-Turn Conversation Handling	Limited memory, affects context in lengthy exchanges	Stores user interaction history, enabling better continuity and relevance
Summarization	Primarily extractive models (e.g., Hugging Face)	Combination of extractive and abstractive techniques for comprehensive summaries
Response Accuracy and Relevance	Moderate accuracy with tendency to hallucinate in complex queries	Improved accuracy due to refined re-ranking and contextual understanding
User Interface	Basic or absent	Integrated UI with multi-turn history tracking and enhanced usability
Audio Output	Not available	Can be integrated in future for accessibility and voice-based interaction
Document Retrieval	Basic or limited	Advanced document retrieval using vector search (FAISS) and semantic similarity
Notification System	Not available	If data not available, system alerts admin to update the database
Data Security	Often cloud-based, less control over privacy	Data is stored and processed locally, ensuring full control and compliance
Admin Interface	Often missing or limited	Allows admin to add or update data directly through a secure local interface

Additional Comparison Points:

1. Contextual Consistency: Our system's re-ranking mechanism incorporates both topic relevance and interaction memory, which provides improved response continuity, especially in multi-turn scenarios. This approach is more consistent compared to prior models, which often lack robust memory handling.
2. Handling Complex Document Structures: Unlike most RAG systems that struggle with data in tabular or nested formats, our system effectively decodes and integrates data across multiple formats, ensuring a comprehensive understanding of complex documents.
3. Scalability for Large Datasets: Vector-based indexing enables our system to manage large datasets with reduced latency, a limitation in other systems that utilize traditional database structures. This results in faster retrievals and supports real-time applications.
4. Re-Ranking Depth: Existing RAG models primarily use basic ranking methods, often limiting performance in diverse contexts. Our hybrid approach, combining dual-matching and topic re-ranking, yields better relevance and accuracy in retrieval.
5. User Interaction Focus: Enhanced UI with a memory-based system offers users more interactive and contextually aware experiences compared to standard RAG models with limited UI features.

III. METHODOLOGY

Our project focuses on developing a secure, on-premise system that uses Retrieval-Augmented Generation (RAG) to enable organizations to query their internal data using natural language. The system is designed to operate entirely within an organization's internal network, ensuring data privacy and compliance with regulatory standards. The methodology involves several key phases to achieve this goal.

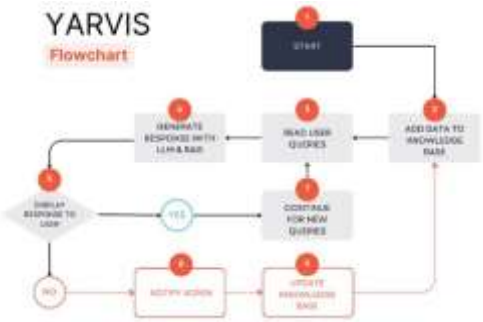
We begin by gathering and preparing data from various internal sources to ensure that it is ready for efficient querying. The system supports data from multiple sources, including relational databases like PostgreSQL and MySQL, flat files such as CSV, Excel, and JSON, and document repositories containing PDFs and Word files. Data cleaning is performed using Python libraries like pandas to handle inconsistencies, remove missing values, and eliminate duplicates. This ensures the data is standardized for further processing. Text normalization techniques, such as converting to lowercase, removing stopwords, and applying stemming, are used to maintain uniformity. The data is then tokenized with tools like spaCy or NLTK, and embeddings are generated using pre-trained models like Sentence-BERT to convert textual data into numerical vectors, optimizing them for search operations.

The next phase focuses on retrieving relevant documents that provide the necessary context for generating accurate responses. We leverage Elasticsearch to index all ingested documents, which enables fast, scalable full-text searches. Each document is stored with additional metadata, such as tags and timestamps, to improve search relevance. To support semantic search capabilities, data is stored in vector format using dense embeddings. For faster similarity searches, especially with large datasets, we utilize FAISS (Facebook AI Similarity Search), which enhances the system's ability to quickly identify the most relevant documents. User queries are converted into embeddings, and a nearest-neighbor search is performed to match the query with the closest documents, using cosine similarity to rank the results.

Once relevant documents are identified, the system generates a coherent, context-aware response using transformer models. The generative component leverages models like GPT-2, BART, or T5, sourced from Hugging Face's transformers library, which are fine-tuned on domain-specific data for improved relevance. The system integrates document retrieval and response generation in a two-stage process. First, the retriever identifies the top-k documents using Elasticsearch or FAISS, and then these documents are passed to the transformer model, which uses them as context for generating a detailed answer. Techniques such as beam search, top-k sampling, and temperature scaling are employed to enhance the quality and precision of the generated responses. The system also supports multi-turn conversations by maintaining dialogue history, enabling it to handle follow-up questions with more context-aware answers.

Given the sensitive nature of the data, robust security measures are implemented to protect the system. All data, both at rest and in transit, is encrypted using AES-256 encryption to prevent unauthorized access. Network communications are secured with SSL/TLS protocols. The system also incorporates role-based access control, ensuring that only authorized users can query sensitive datasets. Comprehensive logging is implemented to capture user activities, query logs, and system performance metrics, which are periodically reviewed for security audits and compliance.

For deployment, the system is packaged using Docker containers, which ensures consistency across various environments and simplifies the deployment process. Docker Compose is used to manage multi-container setups, including the RAG engine, database, and search engine. For organizations requiring scalability, the system can be deployed on Kubernetes clusters, allowing for horizontal scaling. This means additional instances of the RAG engine can be provisioned automatically to accommodate increased user demand. To optimize performance, we implement caching mechanisms for frequently accessed query results, and the system is designed to leverage GPU acceleration for faster model inference, which significantly improves response times, especially when handling large datasets.



IV. Project Architecture

The architecture of the system is designed to ensure that sensitive company data is kept private while still enabling advanced data querying through Retrieval-Augmented Generation (RAG). At its core, the system is built around a standalone binary that can be easily installed and deployed on local servers. This approach ensures that all data processing and AI queries are handled on-site, eliminating the need for sensitive information to be transmitted over the internet to external AI services.

The architecture is modular, consisting of several key components: a data ingestion layer that handles the collection and preprocessing of data, a retrieval engine that indexes the company's dataset for efficient querying, and a generation module powered by advanced language models. These components work together seamlessly to provide businesses with the capability to generate insightful responses to queries based on their proprietary data.

The system also includes a secure interface for interaction with the retrieval engine, allowing authorized users to submit queries while maintaining control over their data. Since the

entire solution is self-contained, companies are not dependent on third-party APIs, which not only helps in maintaining data privacy but also ensures full control over the infrastructure. Overall, the architecture is designed for scalability, security, and ease of deployment, making it suitable for businesses that require both confidentiality and advanced AI-driven insights.

V. CONCLUSION

To address the increasing demand for privacy-preserving AI, our solution presents a secure, on-premises Retrieval-Augmented Generation (RAG) solution that allows natural language querying of internal data without sacrificing privacy. By integrating transformer models with secure retrieval software, such as FAISS and Elasticsearch, the solution provides speedy, accurate answers in an overwhelming majority of document types—all executed locally to address GDPR and HIPAA compliance. Compared to cloud-based applications, our methodology has better multi-turn conversation management, document accuracy, contextual appropriateness, and user experience. With native encryption, access control, and containerized deployment, the system is scalable, secure, and appropriate for privacy-sensitive enterprise applications.

References

1. P. Joshi, A. Gupta, P. Kumar and M. Sisodia, "Robust Multi Model RAG Pipeline For Documents Containing Text, Table & Images," 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024, pp. 993-999, doi: 10.1109/ICAAIC60222.2024.10574972.

2. S. Vakayil, D. S. Juliet, A. J and S. Vakayil, "RAG-Based LLM Chatbot Using Llama-2," 2024 7th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2024, pp. 1-5, doi: 10.1109/ICDCS59278.2024.10561020.

3. Y. Ahn, S. -G. Lee, J. Shim and J. Park, "Retrieval-Augmented Response Generation for Knowledge-Grounded Conversation in the Wild," in IEEE Access, vol. 10, pp. 131374-131385, 2022, doi: 10.1109/ACCESS.2022.3228964.

4. P. N. Singh and S. Behera, "The Transformers' Ability to Implement for Solving Intricacies of Language Processing," 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, 2022, pp. 1-7, doi: 10.1109/ASIANCON55314.2022.9909423.

5. M, A. Danda, H. Bysani, R. P. Singh and S. Kanchan, "Automation of Text Summarization Using Hugging Face NLP," 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 2024, pp. 1-7, doi: 10.1109/INCET61516.2024.10593316.

6. M. Maryamah, M. M. Irfani, E. B. Tri Raharjo, N. A. Rahmi, M. Ghani and I. K. Raharjana, "Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access," 2024 16th International Conference on Knowledge and Smart Technology (KST), Krabi, Thailand, 2024, pp. 259-264, doi: 10.1109/KST61284.2024.10499652.

7. H. K. Chaubey, G. Tripathi, R. Ranjan and S. k. Gopalaiyengar, "Comparative Analysis of RAG, Fine-Tuning, and Prompt Engineering in Chatbot Development," 2024 International Conference on Future Technologies for Smart Society (ICFTSS), Kuala Lumpur, Malaysia, 2024, pp. 169-172, doi: 10.1109/ICFTSS61109.2024.10691338.