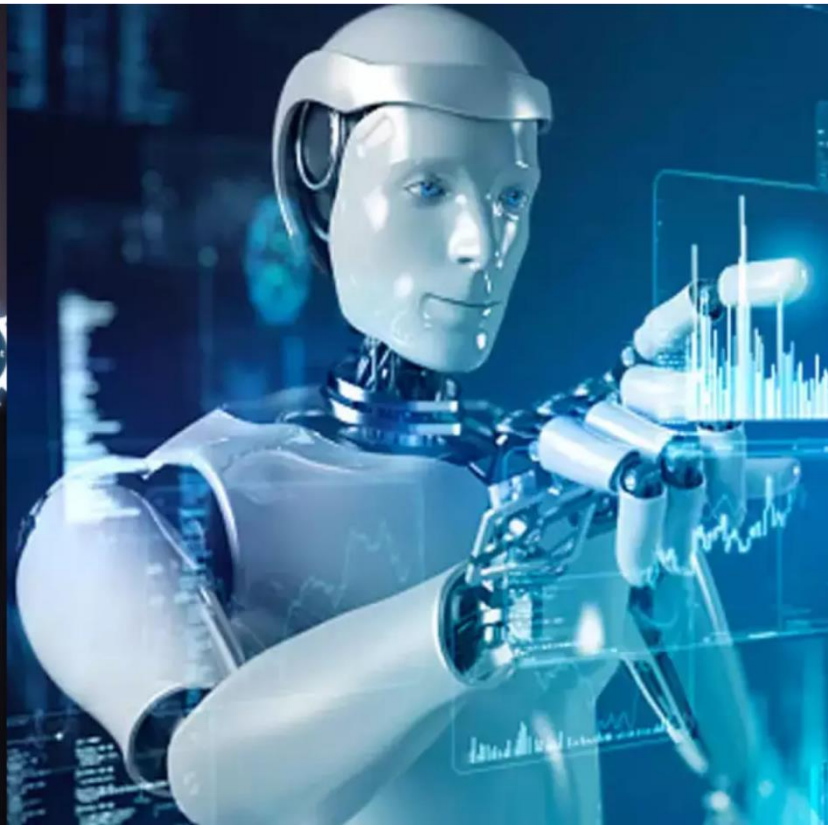
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|

PRAG

Mr. Bergi Veeresh Gowda,Vikas Gadagin, Srinivas N, Praveen Kumar B

Associate Professor, Department of CSE, Rao Bahadur Y Mahabaleshwarappa Engineering College, Karnataka, India

UG Student, Department of CSE, Rao Bahadur Y Mahabaleshwarappa Engineering College, Karnataka, India

UG Student, Department of CSE, Rao Bahadur Y Mahabaleshwarappa Engineering College, Karnataka, India

UG Student, Department of CSE, Rao Bahadur Y Mahabaleshwarappa Engineering College, Karnataka, India

ABSTRACT: Retrieval-Augmented Generation (RAG) enhances natural language processing by integrating retrieval-based and generative models. This approach addresses traditional generative models' limitations in handling real-time and domain-specific queries by dynamically accessing external data sources. Our study implements RAG within a Django-based framework, enabling real-time processing of user-uploaded PDF documents. This ensures adaptability and scalability across various domains. We discuss the system architecture, implementation challenges, and potential improvements, highlighting RAG's capability to enhance generative models with external knowledge for more accurate and context-aware responses

KEYWORDS: Retrieval-Augmented Generation, RAG, Natural Language Processing, Generative Models, Information Retrieval, Django Framework, Real-Time Query Processing.

I. INTRODUCTION

The LLaMA RAG model (LLaMA stands for Large Language Model Meta AI) is a cutting-edge variant of the Retrieval-Augmented Generation (RAG) approach, specifically designed to enhance the performance of language models by incorporating external retrieval mechanisms. LLaMA RAG integrates Meta's LLaMA language models with external knowledge sources, enabling them to dynamically fetch relevant information during the text generation process. This hybrid system combines the generative power of LLaMA with an external retrieval step, where the model queries databases or documents in real time to gather up-to-date and detailed information. The retrieved data is then used to generate more accurate, contextually rich, and factually sound responses. This makes LLaMA RAG models highly effective for tasks such as question answering, summarization, and complex reasoning, where access to specific and current knowledge is essential. By merging large-scale language models with information retrieval, LLaMA RAG aims to overcome the knowledge cutoff limitations of traditional language models and provide more precise and contextually relevant outputs.

II. LITERATURE SURVEY

Literature Survey

2.1 **Retrieval-Augmented Generation (RAG):** The RAG framework was introduced by Facebook AI, where the model retrieves relevant information from a large corpus during the text generation process.

- This enables the model to handle tasks like question answering and summarization more effectively than conventional language models, which are limited by their fixed training data.
- The RAG approach has shown improved accuracy and relevance in various NLP applications, making it a promising solution for dynamic content generation.

2.2 **LLaMA Models:** Meta's LLaMA (Large Language Model Meta AI) models are designed to be more efficient and effective than previous models like GPT-3, particularly for applications requiring a balance between accuracy and computational efficiency.

- LLaMA models are trained on large, diverse datasets, providing strong language generation capabilities.
- However, like other pre-trained models, they face limitations when dealing with real-time or specialized knowledge that was not included during training.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|

2.3 **Combining LLaMA with RAG:** Integrating LLaMA with a retrieval mechanism, the LLaMA RAG model allows the model to dynamically access external knowledge sources during the text generation process.

- This combination addresses the challenge of knowledge cutoff and improves the ability of the model to answer questions and generate text based on the most recent information.
- Research has demonstrated that the RAG approach boosts performance in tasks like fact-checking and answering domain-specific queries, where real-time access to information is crucial.

2.4 **Applications and Challenges:** LLaMA RAG models have been applied to a variety of tasks, including question answering, summarization, and information retrieval.

2.5 **Recent Developments:** Recent advancements in retrieval-augmented models, including improvements in retrieval efficiency, scalability, and fine-tuning techniques, have further enhanced the potential of LLaMA RAG models.

- Techniques like neural ranking models and knowledge graph integration are being explored to improve the retrieval step, ensuring more accurate and domain-specific results.
- Research continues to focus on improving the seamless integration between retrieval and generation to ensure smooth, coherent output while optimizing resource consumption.

.

III. METHODOLOGY

1 SYSTEM FUNCTIONAL SPECIFICATION

1.1 Functions Performed

- The system facilitates real-time retrieval-augmented generation (RAG) by integrating LLaMA models with external document repositories. It processes user queries, retrieves relevant documents, and generates contextually appropriate responses. The main functions include:
- User query processing
- Document retrieval and indexing
- Response generation using LLaMA RAG
- Dynamic adaptation to new knowledge sources

1.2 System Database/File Structure Preview

- The system employs a vector database to store document embeddings and metadata for efficient retrieval. It utilizes:
- FAISS for similarity search
- ChromaDB for indexing structured data
- sqlite for storing user interactions and query logs

1.3 User Input / Output Specification

- **Input:** User queries entered via a Django-based interface
- **Output:** AI-generated responses with referenced document citations
- **Format:** Text-based responses with interactive UI components

1.4 External and Internal Limitations and Restrictions

- **External:** Requires internet access for dynamic retrieval
- **Internal:** Model accuracy depends on the quality of indexed documents

2. SYSTEM DESIGN

2.1 System Architecture

- The architecture consists of:
- Frontend: Django web application

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

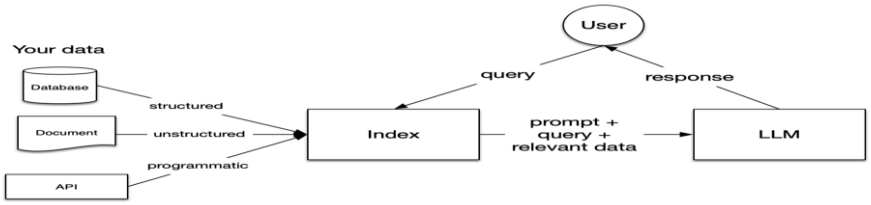
Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|

- Backend: LLaMA RAG model integrated with FAISS



Storage: Vector database for document embeddings

2.2 System Data Flow Diagrams

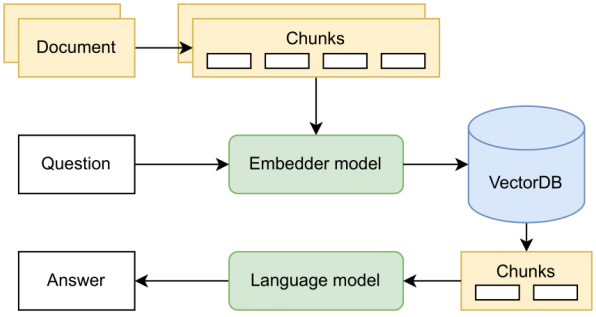
- The system follows a structured pipeline:
- User submits a query
- System retrieves relevant documents
- LLaMA model processes and generates responses
- Response is displayed with citations

2.3 Description of System Operation

- The system dynamically retrieves and integrates external knowledge sources to generate real-time responses.

2.4 Structured Chart and Module Description

- Query Processing Module
- Document Retrieval Module
- Response Generation Module



2.5 Use Case Diagrams

- Use case diagrams illustrate interactions between users and system components.

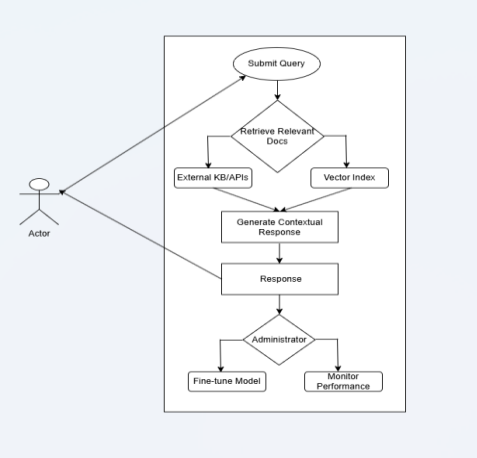
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

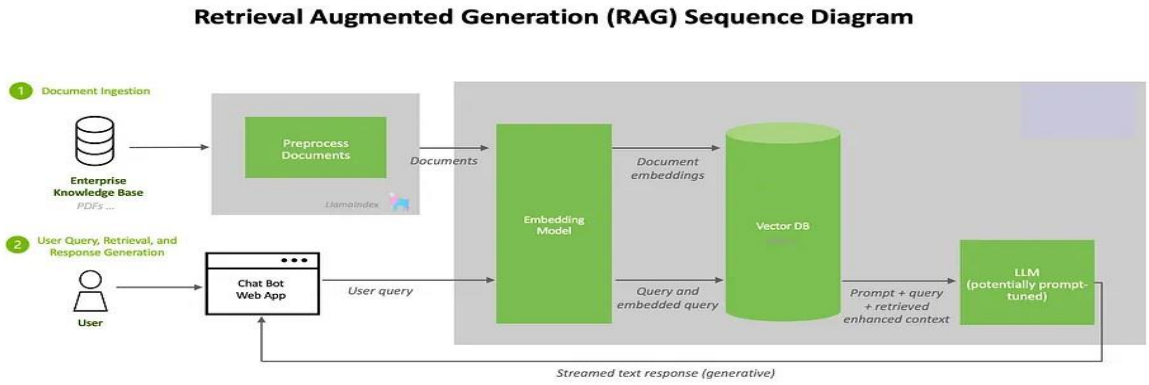
Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|



2.6 Sequence Diagrams

- Sequence diagrams represent the flow of data during query processing.



2.7 ER Diagram (if applicable)

- Entity-Relationship (ER) diagrams illustrate database relationships.

2.8 Algorithm Specification (Flowchart/Pseudocode)

- Flowchart representation of:

1. Query input
2. Document retrieval
3. Response generation
4. Displaying results

2.9 Equipment Configuration

- Server: Cloud-based hosting
- Model Deployment: GPU-accelerated instance

2.10 Implementation Languages

- Python (Django, LangChain, FAISS)
- JavaScript (Frontend UI)

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|

IV. SYSTEM VERIFICATION

System Verification Plan

Verification Aspect	Input	Expected Output	Verification Method
Data Ingestion	Unstructured text (PDFs, articles, etc.)	Successfully stored and indexed in a vector database	Check storage logs and ensure vector embedding is created
Query Processing	User query: "Explain block chain technology"	Query is tokenized and converted into vector embeddings	Debugging retrieval pipeline logs
Document Retrieval	Query embeddings	Retrieves top-k relevant documents	Ensure retrieved documents have a relevance score > 80%
Response Generation	Retrieved documents + Query	A concise, coherent response is generated	Compare generated response with expected content
Citations & References	Retrieved documents	Response includes references to sources	Verify if output cites relevant documents
User Feedback Handling	Feedback: "Response is incorrect"	Model improves response accuracy over time	Track model adaptation and user ratings
Performance Metrics	Multiple test queries	Response time <2s, Accuracy >90%	Run benchmark tests

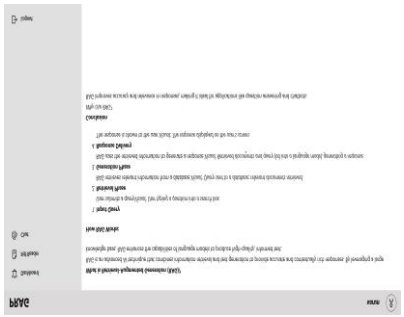
V. EXPERIMENTAL RESULTS

USER SCREEN SHOTS

LOGIN PAGE: SIGN IN

SIGN UP

DASHBOARD:



Chat: In the chat section, 'Students' can upload a PDF, ask questions, and receive answers directly from the document or external sources like Google. They can also save the entire conversation by clicking the "Save" button for future reference.

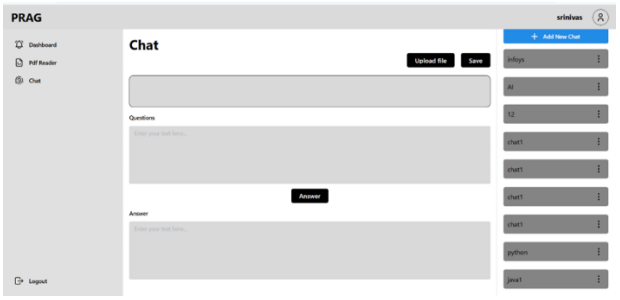
International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

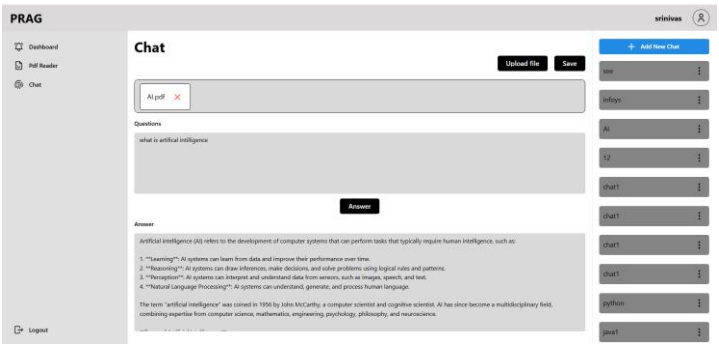
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|

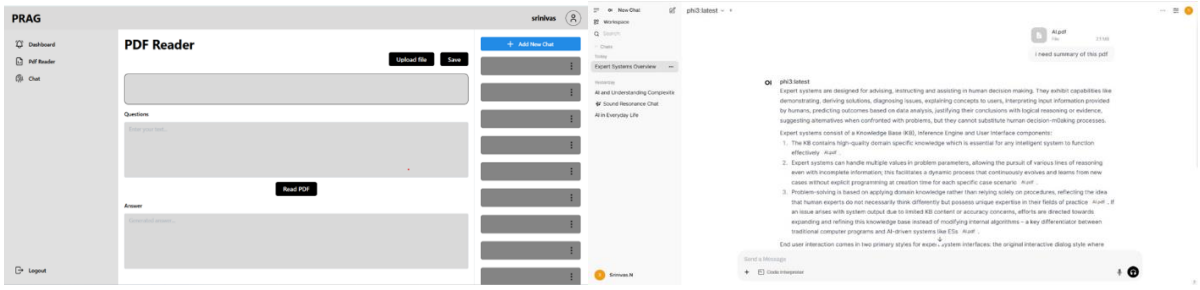


Prompt with output



Pdf Reader

In the **LLaMA RAG model**, a **PDF reader** can be integrated as part of the **information retrieval module**. It allows the system to read and extract relevant content from PDF documents to enhance the knowledge base used for generating responses.



VI. CONCLUSION

The implementation of a Retrieval-Augmented Generation (RAG) system using Django has significantly enhanced how information is accessed and utilized. By integrating advanced retrieval mechanisms with generative AI models, this system delivers precise and contextually relevant responses, reducing the need for extensive manual searches. Django's scalable framework ensures seamless data management, making the system highly effective for knowledge-based applications.

A key advantage of this approach is improved retrieval efficiency and content generation. The system rapidly fetches relevant documents, which are then processed by AI models to generate meaningful responses. This minimizes response time and enhances accuracy, making it ideal for real-time insights. Additionally, Django's ORM and API capabilities streamline data handling, eliminating inefficiencies in traditional workflows.

Scalability and security are central to the system's design, with Django's built-in authentication, encryption, and validation mechanisms ensuring robust data protection. Furthermore, analytical tools embedded within the system

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403349|

provide valuable insights into user interactions and content effectiveness, enabling continuous optimization of retrieval and generation processes.

Overall, the integration of RAG with Django demonstrates the powerful synergy between AI-driven content generation and efficient data retrieval. As organizations continue adopting AI-powered solutions, this system sets a benchmark for enhancing digital interactions, making information retrieval more intuitive, efficient, and scalable.

REFERENCES

1. Building a RAG Model with Open-Source LLM LLaMA 2 and Vector Store FAISS. Available at: Matillion Blog
2. RAG with LLaMA Using Ollama: A Deep Dive into Retrieval-Augmented Generation. Available at: Medium
3. RAG Using LLaMA 2, Langchain, and ChromaDB. Available at: Kaggle
4. Complete Step-by-Step Guide to Create a RAG LLaMA System. Available at: ChatBees AI
5. How to Use Retrieval-Augmented Generation (RAG) with LLaMA 2. Available at: A-Sphere
6. Llama Index Documentation: Build RAG with In-Line Citations. Available at: Llama Index Docs
7. GitHub Repositories: LLaMA2RAG. Available at:
    a. Repository 1
    b. Repository 2

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details