Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

LlamaRAG Assist: A Retrieval-Augmented Generation Framework for Academic Regulation Assistance

Abhijith VA
Dept. of Computer Science & Engineering
Toc H Institute of Science & Technology
Kerala, India

Adithyan KP
Dept. of Computer Science & Engineering
Toc H Institute of Science & Technology
Kerala, India

Akash Thomas
Dept. of Computer Science & Engineering
Toc H Institute of Science & Technology
Kerala, India

Amiliya Thankam Abraham
Dept. of Computer Science & Engineering
Toc H Institute of Science & Technology
Kerala, India

Ms. Anju Kuriakose
Assistant Professor, Dept. of CSE
Toc H Institute of Science &
Technology Kerala, India

***Abstract*—Academic institutions generate a significant volume of policy documents including curriculum regulations, attendance rules, grading policies, and examination guidelines. These documents are typically distributed as lengthy PDF files that are difficult for students to navigate manually. Traditional keyword-based search tools are limited because they rely on exact term matching and cannot interpret the semantic meaning of natural language queries. Meanwhile, standalone large language models often generate hallucinated responses when asked domain-specific questions.**

This paper presents LlamaRAG Assist, an intelligent academic advisory system based on a Retrieval-Augmented Generation (RAG) architecture. The proposed system converts institutional documents into semantic vector embeddings and stores them in a vector database. When users submit queries, the system retrieves relevant document fragments and uses them as contextual input to a locally hosted LLaMA-based language model to generate accurate responses.

The architecture integrates document ingestion, semantic text chunking, embedding generation, and similarity-based vector retrieval using ChromaDB. Experimental evaluation demonstrates that the system significantly improves answer accuracy and reduces hallucination compared to standalone language models. The results highlight the potential of RAG-based systems for improving accessibility and usability of academic regulations.

***Index Terms*—Retrieval-Augmented Generation, Large Language Models, Semantic Search, Vector Databases, Academic Chatbots**

I. Introduction

Universities produce a multitude of academic and administrative documents, encompassing curriculum frameworks, course guidelines, examination regulations, and institutional policies. These documents are typically disseminated via university portals in PDF format. Despite their function as authoritative sources of information, the extraction of specific information from these documents can prove to be arduous.

Students often require access to these documents to ascertain rules pertaining to attendance eligibility, credit prerequisites for graduation, grading policies, or examination regulations. Nevertheless, the identification of this information frequently necessitates the manual perusal of hundreds of pages of documentation.

Conventional search engines depend on keyword-based matching methodologies. While keyword searches are effective when the user is familiar with the precise terminology employed in the document, they become ineffective when the query utilizes alternative phrasing. For instance, the query ”minimum attendance required for exams” may not correspond with a section that includes the phrase ”attendance eligibility criteria.”

Recent innovations in Large Language Models (LLMs) have facilitated the development of conversational question-answering systems that possess the capacity to comprehend natural language inquiries. However, independent LLMs may produce hallucinated responses when confronted with domain-specific questions, particularly in instances where they are devoid of access to the original documents.

Retrieval-Augmented Generation (RAG) mitigates this limitation by amalgamating document retrieval with language

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

model reasoning. Rather than relying exclusively on internal model knowledge, the system retrieves pertinent document segments from a knowledge base and supplies them as contextual information to the language model.

This study presents **LlamaRAG Assist**, a conversational academic assistant that facilitates natural language engagement with institutional regulations while ensuring that responses are firmly anchored in official documents.

A. Inspiration

There is a discrepancy between the quantity of material available and its accessibility due to academic institutions' increasing reliance on digital papers. Despite the fact that papers are readily available, many users find it difficult to extract valuable insights from them. In lengthy papers, students frequently struggle to uncover precise regulations, which can cause confusion and misunderstandings.

A clever system that not only retrieves pertinent data but also displays it in an understandable and user-friendly manner has been developed as a result of this circumstance. The suggested method uses Retrieval-Augmented Generation to bridge the gap between active knowledge engagement and static document collections. The objective is to make standard document retrieval more conversational and intuitive.

II. Related Work

Enhancing the dependability and factual foundation of Large Language Models (LLMs) has been the focus of recent advances in natural language processing. Retrieval-Augmented Generation (RAG), which improves response quality by incorporating external knowledge sources throughout the generation process, is one of the most successful strategies in this direction.

A thorough analysis of RAG methods and their development can be found in Gao et al. [1]. The study demonstrates how retrieval mechanisms anchor language models in actual facts, enabling them to produce more accurate and context-aware responses. The authors also go over various RAG topologies, from straightforward retrieval pipelines to sophisticated modular systems. They do point out that the quality of the information retrieved has a significant impact on RAG's efficacy.

A governance-focused RAG framework for enterprise systems is presented by Kesuma [2]. To guarantee dependable and transparent results, this method integrates elements like auditability, traceability, and compliance validation. Although the model greatly increases trustworthiness, it also adds more complexity to the system and lengthens response times.

Mihajlović [3] investigates a multimodal extension of RAG that enhances semantic understanding by utilizing both textual and visual data. The system generates richer and more contextually appropriate replies by merging embeddings from several data sources. Despite its benefits, the method adds more implementation difficulties and demands more processing power.

The use of RAG in public service chatbot systems is examined by Pujiono et al. [4]. Their research shows that combining retrieval processes with LLMs greatly increases response accuracy and lowers hallucinations. Nevertheless, when using sophisticated models, the study also identifies issues with scalability and higher operating costs.

A hybrid chatbot system that can handle both document-based and structured data sources is proposed by Haque [5]. The technology enables users to query various datasets using natural language by integrating RAG with a Text-to-SQL methodology. This enhances usability, but in complicated situations, the system's performance is susceptible to query interpretation and may yield inaccurate results.

Kiran [6] introduces a hybrid retrieval strategy that blends conventional keyword-based techniques with semantic vector search. By striking a balance between precise phrase matching and contextual comprehension, this method enhances retrieval performance. Nevertheless, using many retrieval techniques raises architectural complexity and might affect system effectiveness.

III. System Architecture

The architecture of LlamaRAG Assist consists of five primary components:

1) Document Ingestion
2) Text Chunking
3) Embedding Generation
4) Vector Retrieval
5) Response Generation

A. High-Level Architecture

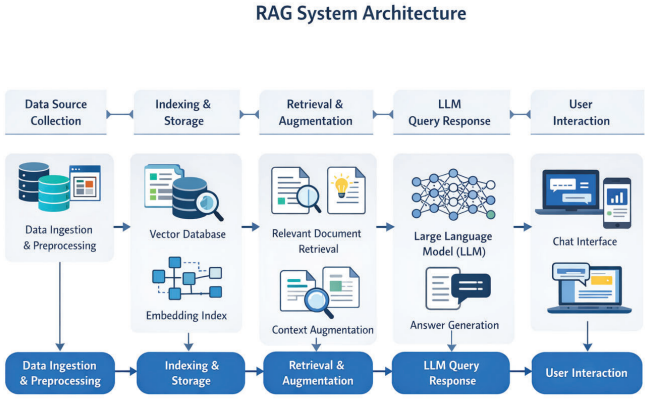

Fig. 1. Overall architecture of the LlamaRAG Assist system showing document ingestion, embedding generation, vector storage, and response generation.

The overarching architecture of the proposed system elucidates the methodology by which documents will be analyzed and corresponding responses will be generated. The system initiates the process by collecting documents from the institution, encompassing academic regulations and policy frameworks. These documents are subsequently processed and converted into a structured textual format.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

A sentence transformer model decomposes the processed text into discrete segments, generating vector embeddings from these fragments. A vector database is employed to archive these embeddings, facilitating their rapid retrieval based on similarity metrics.

When a user submits a query, it is converted into an embedding and matched against stored vectors. The most relevant document segments are retrieved and passed as contextual input to the language model. The language model then generates a response grounded in the retrieved information.

This architecture ensures that the generated responses are both context-aware and factually accurate, while maintaining scalability and modularity.

B. Low-Level Architecture

The low-level architecture provides a detailed view of internal system components and their interactions. The system is divided into multiple functional modules, each responsible for a specific task in the pipeline. **Document Ingestion Module:**

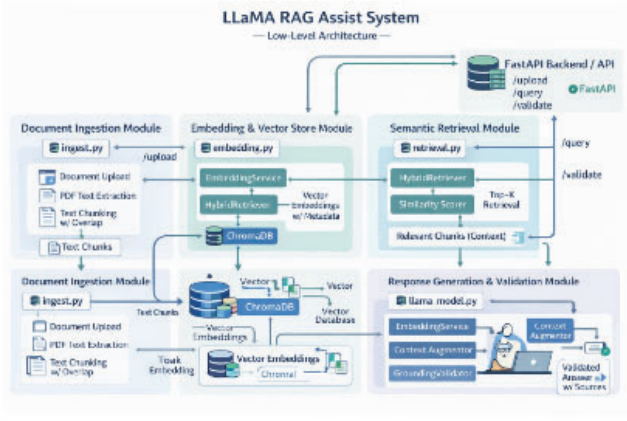

Fig. 2. Low-Level Architecture showing internal modules such as ingestion, chunking, embedding, retrieval, and response generation.

This module handles document upload and parsing. It extracts raw text from PDF files and performs preprocessing such as cleaning and normalization.

Chunking Module: The extracted text is divided into smaller overlapping segments to improve retrieval efficiency and preserve contextual continuity.

Embedding Module: Each text chunk is converted into a numerical vector representation using a sentence embedding model. These embeddings capture semantic meaning.

Vector Database Module: The embeddings are stored in a vector database (ChromaDB), which supports fast similarity search operations.

Retrieval Module: When a query is received, it is embedded and compared against stored vectors using cosine similarity. The top-k most relevant chunks are retrieved.

LLM Response Module: The retrieved chunks are passed as context to a locally hosted LLaMA model, which generates the final response.

User Interface Module: The system provides an interactive interface that allows users to submit queries and receive responses in real time.

The modular design ensures flexibility, scalability, and ease of maintenance while enabling efficient integration of additional components.

C. Design Considerations

Several design decisions were made to ensure the effectiveness of the system. First, chunk-based processing was adopted to improve retrieval precision and reduce computational overhead. Second, semantic embeddings were used instead of keyword matching to better capture the meaning of user queries.

Additionally, the system prioritizes retrieved context over model-generated knowledge to minimize hallucinations. A modular architecture was chosen to allow future enhancements such as hybrid retrieval, multimodal data integration, and real-time updates.

These considerations collectively contribute to building a robust and scalable academic assistance system. The architecture begins with document ingestion where institutional documents are collected and converted into structured text. These documents are then segmented into smaller chunks and converted into embeddings. The embeddings are stored in a vector database that enables efficient similarity search.

When a user submits a query, the system converts the query into an embedding and retrieves the most relevant document segments. These retrieved segments are then used as contextual input to the language model to generate the final response. In addition to the core components, the system incorporates a feedback-driven improvement mechanism that continuously enhances retrieval quality. User interactions are monitored to identify cases where the retrieved context is insufficient or ambiguous. This feedback is used to refine chunking strategies and optimize retrieval parameters.

Furthermore, the modular design of the architecture ensures scalability and flexibility. New document sources can be integrated without significant changes to the existing pipeline, making the system adaptable to evolving institutional requirements.

IV. Methodology

A. Document Processing

Academic regulation PDFs are converted into machine-readable text using automated document parsing tools.

The extracted text is cleaned by removing formatting artifacts, page numbers, and metadata before further processing.

B. Text Chunking

Large documents are divided into smaller overlapping segments to improve retrieval efficiency.

- Chunk Size: 1000 characters
- Chunk Overlap: 200 characters

Overlapping segments help preserve contextual relationships between sentences.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

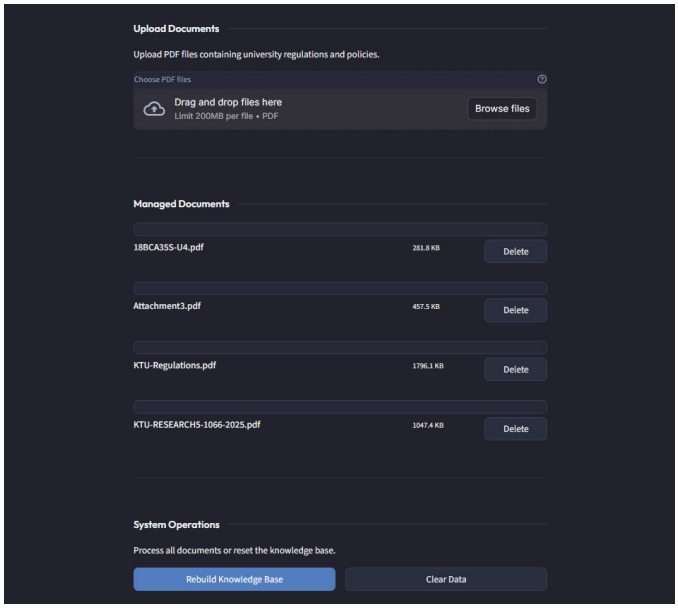

Fig. 3. Document ingestion and preprocessing pipeline used to convert academic regulation PDFs into structured text.

C. Embedding Generation

Each document chunk is converted into a vector embedding using a Sentence Transformer model. These embeddings capture semantic meaning and enable similarity-based retrieval.

D. Semantic Retrieval

Similarity search is performed using cosine similarity:

$$sim(q, d) = \frac{q \cdot d}{|q||d|}$$

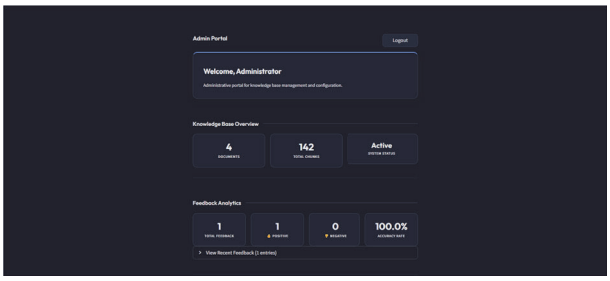

Fig. 4. RAG pipeline illustrating query embedding, document retrieval, and response generation.

E. Context Filtering and Ranking

To make retrieval better, the system uses filtering and ranking based on how similar and relevant the context is. A step to re-rank the segments puts the most important ones at the top and gets rid of content that is not very useful or is already there.

Redundancy filtering keeps the input short, and positional relevance helps bring attention to important parts, like definitions or rules. Threshold-based filtering also gets rid of results that aren't very reliable to cut down on noise.

Lastly, the language model can give correct and context-aware answers because the chosen parts are put into a structured prompt.

V. RAG Pipeline Algorithm

Algorithm 1 Enhanced Retrieval-Augmented Question Answering Pipeline

Load academic documents
Extract raw textual content from documents
Clean and preprocess text (remove noise, formatting, metadata)
Split documents into overlapping chunks
Generate embeddings for each chunk
Store embeddings in vector database
Receive user query q
Preprocess query (normalize, remove stopwords)
Convert query into embedding q_e
for each document embedding d_i **do**
    Compute cosine similarity:

$$sim(q_e, d_i) = \frac{q_e \cdot d_i}{||q_e|| \cdot ||d_i||}$$

end for
Retrieve top-k relevant chunks based on similarity
Remove duplicate or highly overlapping chunks
for each retrieved chunk c_i **do**
    Compute semantic similarity score S_s
    Compute keyword relevance score S_k
    Compute contextual coherence score S_c
    Combine scores:

$$Score(c_i) = \alpha S_s + \beta S_k + \gamma S_c$$

end for
Re-rank chunks based on final scores
Select top-n high-quality context chunks
Construct contextual prompt using selected chunks
Pass prompt to LLM
Generate response R
Validate response against retrieved context
if unsupported or hallucinated content detected **then**
    Return fallback or clarification response
else
    Return final validated response
end if
Log user query and response for feedback analysis
Update system parameters if necessary

VI. Experimental Evaluation

The system was evaluated using a dataset of 50 student queries related to academic regulations. These queries were designed to represent common questions asked by students regarding attendance policies, grading systems, and course requirements.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

TABLE I
PERFORMANCE COMPARISON

Metric	Keyword Search	Raw LLM	LlamaRAG Assist
Accuracy	62%	70%	**94%**
Hallucination Rate	0%	30%	**4%**
Response Time	1.2s	2.5s	4.5s

A. Error Analysis

Although the system achieves high accuracy, certain limitations were observed during evaluation. Errors primarily occurred in cases where the query was ambiguous or when relevant information was distributed across multiple document sections.

In some instances, retrieval returned partially relevant chunks, leading to incomplete answers. These cases highlight the importance of improving chunking strategies and incorporating advanced retrieval techniques such as hybrid search and query expansion.

B. User Experience Evaluation

A qualitative evaluation was also conducted to assess user satisfaction. Users reported that the system significantly reduced the time required to locate information compared to manual document search. The conversational interface was found to be intuitive, especially for users unfamiliar with technical terminology.

However, users suggested improvements in response formatting and the inclusion of direct citations from source documents, which can further enhance trust and usability.

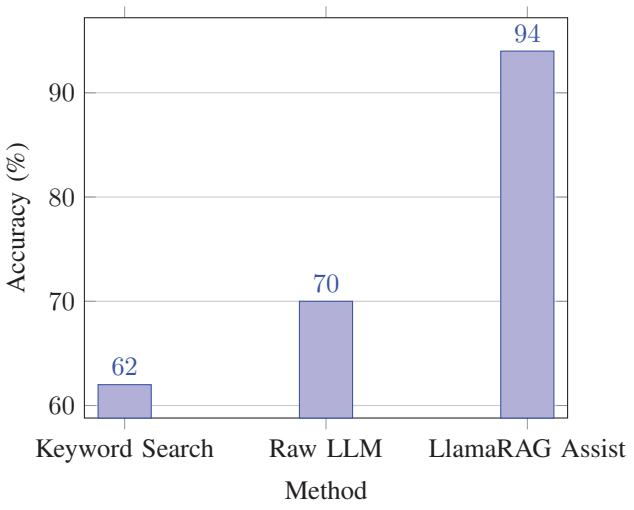

Fig. 5. Accuracy comparison of different approaches for academic regulation queries.

VII. DISCUSSION

The experimental results clearly demonstrate the effectiveness of integrating retrieval mechanisms with language models. By grounding responses in actual document content, the system minimizes hallucinations and improves reliability.

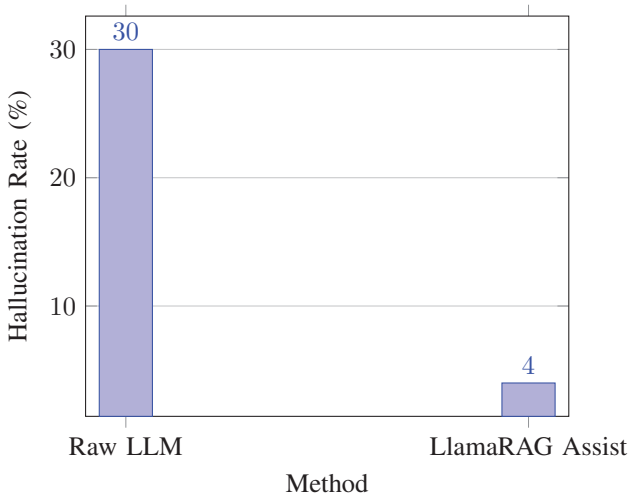

Fig. 6. Reduction in hallucination rate achieved using RAG-based retrieval.

One key observation is that retrieval quality plays a crucial role in overall system performance. Even a highly capable language model cannot compensate for poor or irrelevant context. Therefore, optimizing document preprocessing and retrieval strategies is essential.

The system also highlights the importance of balancing accuracy and response time. While the RAG-based approach introduces additional latency compared to traditional search systems, the improvement in answer quality justifies the trade-off.

Overall, the proposed system provides a practical solution for enhancing access to academic regulations, offering both efficiency and reliability.

VIII. CONCLUSION

This paper presented LlamaRAG Assist, a Retrieval-Augmented Generation system designed to improve access to academic regulations through a conversational interface. By integrating semantic retrieval with language model reasoning, the system enables users to obtain accurate and context-aware responses from institutional documents.

The experimental results demonstrate significant improvements in accuracy and a substantial reduction in hallucinated responses compared to traditional approaches. The system effectively addresses the limitations of keyword-based search and standalone language models by grounding responses in verified document content.

Future work will focus on enhancing retrieval performance through hybrid search techniques, incorporating multimodal data sources, and improving user interaction through better response formatting and citation support. The proposed system can be extended to other domains where efficient document-based knowledge access is required.

REFERENCES

[1] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv*, 2024.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

[2] C. Kesuma, "Retrieval Augment Generation (RAG) Governance Architecture for Enterprise Information Systems," *International Journal of Informatics, Economics, Management and Science*, vol. 4, no. 2, pp. 133–142, 2025.
[3] M. Mihajlović, "Multimodal Retrieval-Augmented Generation in Knowledge Systems: A Framework for Enhanced Semantic Search and Response Accuracy," SINTEZA 2025: International Scientific Conference on Information Technology, Computer Science and Data Science, 2025.
[4] I. Pujiono, I. M. Agtyaputra, and Y. Ruldeviyani, "Implementing Retrieval-Augmented Generation and Vector Databases for Chatbots in Public Services Agencies Context," JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer), pp. 216–223, 2024.
[5] I. C. R. Haque, "Implementation of Retrieval-Augmented Generation (RAG) and Large Language Models (LLM) for a Document and Tabular-Based Chatbot System," Journal of Electronics Technology Exploration, 2025.
[6] S. Kiran, "Hybrid Retrieval-Augmented Generation (RAG) Systems with Embedding Vector Databases," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 11, no. 2, pp. 2694–2702, Mar. 2025, doi: 10.32628/CSEIT25112702.
[7] L. Cunha, "Revolutionizing Operational Accounting with Low-Code Platforms: The Impact of n8n," International Journal of Scientific Research in Engineering and Management, 2025.
[8] Adiel Tuyishime, Francesco Basciani, Amleto Di Salle, Javier Luis C´anovas Izquierdo, and Ludovico Iovino, STREAMLINING WORKFLOW AUTOMATION WITH A MODEL- BASED ASSISTANT, 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) ,2024
[9] S.-V. Oprea and A. Bâra, "Development of a Retrieval-Augmented Generation (RAG) Chatbot," Ovidius University Annals, Economic Sciences Series, 2025.
[10] O. Omolayo and O. Babatope, "Comparative Evaluation of Vector Embedding Frameworks for Scalable Semantic Retrieval in PDF-Based RAG Systems," International Journal of Research Publication and Reviews, vol. 6, pp. 12506–12511, 2025.