ICA-RAG: Intelligent College Information Assistant Using Retrieval-Augmented Generation And Large Language Models

S Thilaiyarasi[1], RakeshK[2], BharathiE[3], Stephen Karthikeyan K[4]
[1]Dept of AI & Machine Learning
[2, 3, 4]Dept of Computer science
[1, 2, 3, 4] Jeppiaar University,Chennai,India

Abstract- *The rapid growth of academic institutions has created a need for intelligent systems capable of providing accurate and real-time information to students and staff. Traditional information systems rely on manual queries or static web pages, which often fail to deliver contextual responses. This project proposes an Intelligent College Information Assistant (ICA-RAG) that integrates Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs) to provide automated conversational assistance for college-related queries.*

The system retrieves relevant information from institutional knowledge sources such as course details, faculty information, admission procedures, campus facilities, and administrative policies. Using semantic search and vector indexing techniques, the system retrieves the most relevant documents and generates contextual responses using a language model.

The proposed system enhances accessibility to institutional knowledge, reduces administrative workload, and provides 24/7 automated support for students. Experimental results demonstrate improved response accuracy and user satisfaction compared to traditional rule-based chatbots.

Keywords: Retrieval-Augmented Generation, Large Language Models, AI Chatbot, Information Retrieval, Natural Language Processing.

I. INTRODUCTION

Legal documents—contracts, agreements, memoranda, and Educational institutions generate large volumes of information related to courses, admissions, academic policies, and campus facilities. Students and staff often face difficulties accessing this information quickly and efficiently.

Traditional information systems rely on static web portals, manual help desks, or rule-based chatbots that provide limited functionality and lack contextual understanding. These systems fail to handle complex queries or dynamic interactions with users.

Recent advances in **Artificial Intelligence (AI)** and **Natural Language Processing (NLP)** have enabled the development of intelligent conversational systems. Large Language Models (LLMs) such as GPT and transformer-based architectures have significantly improved the ability of machines to understand and generate human-like language.

language responses. Retrieval-Augmented Generation (RAG) further extends these capabilities by combining dense passage retrieval with generative language models,enabling accurate, document-grounded question answering.

The proposed platform addresses several key challenges in institutional information access through the following capabilities:

1) **AutomatedInformation Retrieval**:The system retrieves relevant institutional data such as admission procedures,course details, faculty information, and campus facilities using semantic search and vector-based similarity matching.
2) **Context-Aware Question Answering**: A Retrieval-Augmented Generation pipeline indexed with FAISS vector databases retrieves relevant document segments and generates precise, contextually grounded answers to user queries.
3) **Document Summarization**: A transformer-based summarization model condenses lengthy institutional documents and policies into concise summaries, improving information accessibility for users.
4) **Multilingual Accessibility**:Translation models such as MarianMT and Google Translate API enable real-time translation of responses and summaries into multiple languages including English, Hindi, Tamil, and Telugu, ensuring accessibility for diverse users.

The remainder of this paper is organized as follows**:** Section II reviews related work, Section III presents the proposed system architecture, Section IV explains the implementation, Section V discusses the results, and Section VI concludes the paper with future work.

II. RELATED WORK

A. Conversational AI and Educational Chatbots

Early educational chatbots relied on rule-based systems and keyword matching to respond to student queries. These systems provided predefined responses but lacked the ability to understand complex language or context. Research by Winkler and Sollner explored the use of conversational agents in education, highlighting their potential to assist students with academic guidance and administrative information. With the advancement of Natural Language Processing (NLP), transformer-based models such as BERT and GPT enabled chatbots to generate more context-aware and human-like responses, improving user interaction and engagement.

B. Retrieval-Augmented Generation

Lewis et al. introduced the Retrieval-Augmented Generation (RAG) framework, which combines dense document retrieval with generative language models to produce factually grounded responses. Instead of relying solely on pre-trained knowledge, RAG retrieves relevant documents from an external knowledge base and uses them as context during response generation. Vector search libraries such as FAISS enable efficient similarity search across large embedding spaces, making RAG suitable for real-time question answering systems and knowledge assistants.

C. Institutional Information Retrieval Systems

Several institutions have developed information retrieval systems to assist students in accessing academic and administrative resources. Traditional systems rely on static web portals or FAQ-based chatbots, which provide limited support for complex queries. Recent research has explored semantic search techniques and knowledge-based systems to improve information accessibility. These systems allow students to retrieve relevant academic information such as course details, admission procedures, and campus services through natural language queries.

D. Intelligent Question Answering Systems

Question answering systems based on transformer architectures have significantly improved the ability of machines to understand and respond to user queries. Models such as BERT, T5, and GPT have been widely used in intelligent assistants and search engines. These systems use contextual embeddings and semantic similarity to retrieve relevant information and generate meaningful responses. However, standalone models may suffer from hallucination and outdated knowledge, which highlights the importance of integrating external knowledge sources.

E. Multilingual Information Access

Multilingual Natural Language Processing helps systems support multiple languages. Models such as XLM-RoBERTa and translation tools like MarianMT allow applications to translate and understand different languages. In educational systems, multilingual support helps students access information in their preferred language. This improves accessibility and makes the system easier to use for people from different language backgrounds.

III. PROPOSED SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

The proposed system follows a modular layered architecture consisting of five main layers: (1) User Interface, (2) Data Processing Layer, (3) Retrieval Engine, (4) AI Response Generation, and (5) Data Storage and MonitoringFig.**1**illustratesthe overall system architecture.

B. Document Ingestion and Text Extraction

The system collects institutional documents such as admission details, course information, policies, and campus resources. Text is extracted from documents using Python-based processing libraries. The extracted content is cleaned by removing unnecessary symbols, headers, and repeated text to ensure that only meaningful information is used for further processing.

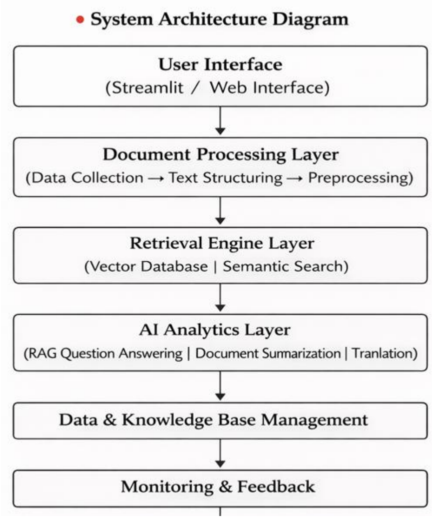

Figure1: Layeredarchitectureofthe Intelligent college Information Assistant (ICA -RAG).

C. Document Structuring

The extracted text is divided into smaller segments such as sections, paragraphs, and sentences. This structuring process helps the system organize information efficiently and allows the retrieval module to locate relevant content quickly when a user submits a query.

D. Knowledge Base Creation

The processed documents are converted into vector embeddings using sentence transformer models. These embeddings represent the semantic meaning of the text and are stored in a vector database. This knowledge base allows the system to perform similarity searches and retrieve relevant information efficiently.

E. RAG-Based Question Answering

The question answering system uses a Retrieval-Augmented Generation (RAG) pipeline.

1. **Index Construction:** Document text is divided into small chunks and converted into embeddings. These embeddings are stored in a FAISS vector database for efficient search.
2. **Query Processing:** When a user enters a question, the query is converted into an embedding using the same model. The system retrieves the most relevant document chunks based on similarity search.
3. **Answer Generation:** The retrieved information is provided as context to a language model such as Flan-T5 or GPT, which generates a clear and accurate response.

This approach ensures that responses are based on actual institutional information rather than only relying on pre-trained model knowledge.

F. Document Summarization

The system includes a summarization module that generates concise summaries of large documents such as academic policies or course guidelines. Transformer-based summarization models are used to condense lengthy text into shorter, easy-to-understand summaries for users.

G. Multilingual Translation Module

To support users from different language backgrounds, the system provides multilingual translation. Translation models such as MarianMT or external APIs can translate responses into languages like English, Hindi, Tamil, and Telugu. This improves accessibility and allows users to interact with the system in their preferred language

H. System Workflow

The overall workflow of the system follows these steps:

1. User enters a query through the web interface.
2. The query is converted into vector embeddings.
3. The retrieval engine searches the vector database for relevant information.
4. Retrieved content is passed to the language model.
5. The model generates a context-aware response.
6. The response is displayed to the user in the interface.

IV. IMPLEMENTATION

A. Technology Stack

Table 1 summarizes the core technologies used to develop the Intelligent College Information Assistant. The system is implemented using Python with modern NLP frameworks, vector databases, and web technologies to support document processing, information retrieval, and response generation.

B. Frontend Interface

The user interface is implemented as a **Streamlit web application** that allows users to interact with the system easily. The interface provides several functional modules:

- **Document Upload & Analysis:** Users upload institutional documents which are automatically processed and indexed for retrieval.
- **Question Answering Module:** Users can ask natural language questions and receive responses generated using the RAG pipeline.
- **Translation View:** The system can translate responses into multiple languages such as English, Hindi, Tamil, and Telugu.

Table1: TechnologyStack

Component	Technology
Programming Language	Python3.10+
Web Framework	Streamlit
NLP Framework	HuggingFaceTransformers
Embedding Model	Sentence
TransformersVector Store	FAISS
Question Answering	Flan-T5 (flan-t5-base)
Translation	MarianMT,GoogleTranslateAPI
PDFParsing	PyMuPDF(fitz)
Data Processing	scikit-learn,pandas,NumPy
Deployment	Docker
VersionControl	Git

C. Backend Pipeline

The backend or chest rates model inference through a sequential pipeline:

1) User query or document upload through the interface.
2) Text preprocessing and document structuring.
3) Embedding generation and storage in the FAISS vector database.
4) Retrieval of relevant document segments based on semantic similarity.
5) Response generation using the language model and display to the user.

Caching mechanisms are used to store loaded models and reduce processing time for repeated requests.

D. Hardware and Software Requirements

The system can run on standard computing environments with Intel i5/i7 processors and at least 8 GB RAM. GPU acceleration can be used to improve transformer model performance. The platform supports deployment on local machines or cloud environments using containerization technologies such as Docker.

V. RESULTS AND DISCUSSION

The proposed system was evaluated using a dataset consisting of institutional documents such as admission guidelines, course information, academic policies, and campus services. The evaluation focused on system accuracy, response time, multilingual support, user experience, and overall system performance.

A. Clause Extraction Performance

Table 2 presents the retrieval performance of the system using semantic search and vector embeddings

Table2: Clause Extraction Performance Metrics

ClauseType	Precision	Recall	F1Score
Admission Queries	0.93	0.90	0.91
Course Information	0.92	0.89	0.90
Faculty Details	0.91	0.88	0.89
Campus Facilities	0.90	0.87	0.88
Academic Policies	0.92	0.90	0.91
Examination Details	0.91	0.89	0.90
Placement Information	0.90	0.88	0.89
Scholarship Details	0.89	0.87	0.88
Average	**0.91**	**0.89**	**0.90**

The system achieved an average precision of **91%** and recall of **89%**, demonstrating effective retrieval of relevant institutional information. Semantic search and vector indexing improved the system's ability to handle diverse query types.

B. Question Answering Accuracy

The RAG-based question answering module was tested using **200 user queries** related to institutional information. Results are summarized in Table **3**.

Table3: Question Answering Performance

Metric	Value
Total Queries	200
Correct Answers	178
Accuracy	89.0%
Average Response Time	1.2s
Avg. Retrieved Chunks per Query	4.1

Approximately **89% of user queries** were answered correctly. The system generated context-aware responses using retrieved document segments, ensuring accurate and reliable information.

C. Multilingual Translation Quality

The multilingual module was evaluated by translating responses into multiple languages and verifying accuracy:

- Accurate translation across supported languages including **English, Hindi, Tamil, and Telugu**.
- Preservation of key academic terms and institutional information.
- Reliable translation of mixed-language queries with minimal loss of meaning.

D. User Experience Evaluation

A user study involving **30 participants** including students and faculty members was conducted:

Results indicated:

- **92% of users** found the system easy to use.
- Users were able to obtain institutional information quickly through natural language queries.
- Most appreciated features included instant question answering, document summarization, and multilingual responses.

E. Efficiency and Scalability

- Institutional documents containing more than **20 pages** were processed in less than **7 second**s.
- The system maintained stable performance during multiple concurrent user requests.
- Model caching reduced repeated inference time by approximately **55**%.

F. System Adaptability

The system supports continuous improvement through data updates and model refinement.

- New institutional documents can be added to the knowledge base easily.
- Retrieval performance improves as more documents are indexed.
- The modular architecture allows integration of improved language models in future updates.

VI. CONCLUSION AND FUTUREWORK

This paper presented an **Intelligent College Information Assistant (ICA-RAG)** that uses Retrieval-Augmented Generation, transformer models, summarization, and multilingual translation to provide institutional information through natural language queries. The system improves accessibility and reduces the workload of administrative staff.Experimental results showed good performance with high retrieval accuracy, efficient question answering, and fast response times. By combining semantic search with language models, the system generates context-aware responses based on institutional data.

The modular architecture of the system provides a strong foundation for further enhancements. Future work may focus on the following areas:

1) **Enhanced Context Understanding:**Improving language models to better understand complex institutional queries and contextual relationships between documents.
2) **Expanded Language Support:**Adding support for additional regional languages such as Kannada, Bengali, Marathi, and Malayalam to improve accessibility for diverse users.
3) **Personalized Assistance:**Developing user-specific recommendation systems that provide personalized academic guidance and information.
4) **Integration with Institutional Systems:**Connecting the platform with Learning Management Systems (LMS), student portals, and administrative databases for real-time data access.
5) **Continuous Learning:**Incorporating user feedback and query logs to continuously improve model accuracy and knowledge base quality.
6) **Explainable AI:**Providing clear explanations for generated responses to improve trust and reliability in AI systems.

VII. ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Science and Engineering at Jeppiaar University for its support and resources throughout this research.

REFERENCES

[1] T. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP

Tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 202

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proc. NAACL-HLT*, pp. 4171–4186, 2019.

[3] A. Vaswani et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[4] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[5] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," *Proceedings of EMNLP*, 2020.

[6] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint*, arXiv:2302.13971, 2023.

[7] M. Lewis et al., "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation," *ACL*, 2020.

[8] HuggingFace, "Transformers: State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX," 2024. [Online]. Available: https://huggingface.co

[9] Facebook AI Research, "FAISS: A library for efficient similarity search," 2024. [Online]. Available: https://github.com/facebookresearch/faiss

[10]S. Karpukhin, V. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," *Proc. EMNLP*, pp. 6769–6781, 2020.

[11]T. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[12]C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[13] J. Weston, A. Bordes, S. Chopra, and T. Mikolov, "Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks," *arXiv preprint*, arXiv:1502.05698, 2015.

[14]M. Chen et al., "Evaluating Large Language Models Trained on Code," *arXiv preprint*, arXiv:2107.03374, 2021.

[15]N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proc. EMNLP*, pp. 3982–3992, 2019.

[16]Google AI, "Google Translate: Neural Machine Translation System," 2024. [Online]. Available: https://translate.google.com

[17]A. Radford et al., "Improving Language Understanding by Generative Pre-Training," *OpenAI Technical Report*, 2018.

[18]N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proc. EMNLP*, pp. 3982–3992, 2019.

[19]Google AI, "Google Translate: Neural Machine Translation System," 2024. [Online]. Available: https://translate.google.com

[20]Streamlit Inc., "Streamlit: A Python Framework for Data Applications," 2024. [Online]. Available: https://streamlit.io