International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

# RAG-AI: A Multi-Modal Retrieval-Augmented Generation Framework for Academic Institutions

Pratik S. Bhiwarkar[1], Khushi S. Telgote[2], Lokesh J. Pande[3], Shantanu D. Gawande[4], Falgun A. Kawale[5], Prof. Mukund Joshi[6]

[12345] Undergraduate Students, [6] Assistant Professor, Department of Information Technology, Shree. H.V.P.M. COET, Amravati (MH, India).

pratikbhiwarkar@gmail.com , khushitelgote05@gmail.com

----------**********----------

## Abstract:

Large Language Models (LLMs) show strong capabilities in language understanding and reasoning, but face key limitations in education, including hallucinations, lack of domain specificity, outdated knowledge, and privacy concerns. These issues affect reliability when applied to institutional academic data. This paper proposes RAG-AI, a multi-modal Retrieval-Augmented Generation framework for context-aware and privacy-preserving academic assistance. The system combines Whisper-based speech recognition, semantic chunking, BGE-M3 embeddings, cosine similarity retrieval, and locally deployed LLMs. By retrieving relevant institutional data before response generation, RAG-AI reduces hallucinations and improves accuracy. Experiments on lecture recordings and academic PDFs show improved factual reliability compared to standalone LLMs. The system supports local deployment, ensuring data privacy and compliance, and provides a scalable solution for educational AI.

**keyword** - Retrieval-Augmented Generation (RAG), Multi-Modal AI, Dense Embeddings, Educational AI, Privacy-Preserving AI

----------**********----------

## I. INTRODUCTION

The rapid advancement of transformer-based Large Language Models (LLMs) has revolutionized conversational artificial intelligence and educational support systems. Models such as GPT, BERT-derived architectures, and open-source generative transformers demonstrate impressive zero-shot and few-shot reasoning capabilities. However, despite their linguistic fluency, these models remain fundamentally limited by static pretraining corpora and parametric memory constraints.

In academic contexts, these limitations manifest as:

1. Hallucinated outputs
2. Inability to access institution-specific content
3. Lack of citation transparency
4. Privacy concerns in cloud-based deployments

When a student queries an LLM about content from a recent university lecture, the model may fabricate plausible but incorrect explanations. Such behavior is unacceptable in high-stakes educational settings.

Retrieval-Augmented Generation (RAG) mitigates these limitations by integrating non-parametric external memory with generative models. Rather than relying solely on learned parameters, RAG dynamically retrieves relevant documents and injects them into the prompt before generation.

This paper introduces RAG-AI, a multi-modal RAG system designed specifically for academic institutions. The key contributions of this work are:

  

- A unified multi-modal ingestion pipeline (video, audio, PDF)
- Semantic-aware chunking strategy
- Dense embedding retrieval using BGE-M3
- Threshold-based cosine similarity filtering
- Local, privacy-preserving deployment architecture
- Empirical validation on real academic datasets

## II. BACKGROUND AND RELATED WORK

### *A. Retrieval-Augmented Generation*

Retrieval-Augmented Generation (RAG) was formalized by Lewis et al. as a hybrid architecture that combines parametric sequence-to-sequence models with a dense retrieval mechanism. Instead of relying only on knowledge stored in model parameters, RAG integrates an external non-parametric memory to dynamically retrieve relevant documents during inference.

The RAG pipeline is usually divided into two main stages:

1. Retrieval Phase: Relevant documents are retrieved from a vector index based on semantic similarity.
2. Generation Phase: The retrieved documents are added into the prompt and used to guide the generative model.

This separated architecture reduces the need for repeated fine-tuning while allowing the knowledge base to be updated dynamically. Later studies showed that grounding generation using retrieved context improves factual accuracy, interpretability, and reliability.

### *B. Dense Embeddings and Semantic Search*

Traditional information retrieval systems use sparse vector representations such as TF-IDF or BM25. These methods work well for keyword matching but often fail to understand semantic similarity.

Dense embedding models convert text into high-dimensional continuous vectors, where similar meanings are located close to each other in vector space. Modern embedding models have greatly improved retrieval performance across multiple languages and domains.

The BGE-M3 embedding model is well known for supporting multilingual dense retrieval and multiple search functions. Its high-dimensional representations allow strong semantic matching even when the exact words are different.

### *C. Speech Recognition in Academic Media*

Automated Speech Recognition (ASR) is an important tool for processing lecture recordings and educational videos. The Whisper model introduced large-scale weakly supervised training, which allows accurate transcription across many languages and technical subjects.

Using ASR systems, recorded lectures can be converted into text, which can then be indexed and searched using semantic retrieval methods in a RAG system.

### *D. Privacy-Preserving AI Deployment*

Cloud-based AI services can create risks related to data privacy and intellectual property. Educational institutions must follow data protection rules and security policies.

Running open-source language models and embedding models locally is a safer option because sensitive data stays inside the institution. Because of this, privacy-preserving AI architectures are becoming more important in academic AI system development.

## III. SYSTEM ARCHITECTURE

The RAG-AI framework is organized into five modular layers:

1. Multi-Modal Ingestion Layer
2. Preprocessing and Semantic Segmentation
3. Embedding and Vector Storage
4. Retrieval and Ranking Layer
5. Context-Grounded Generation Layer

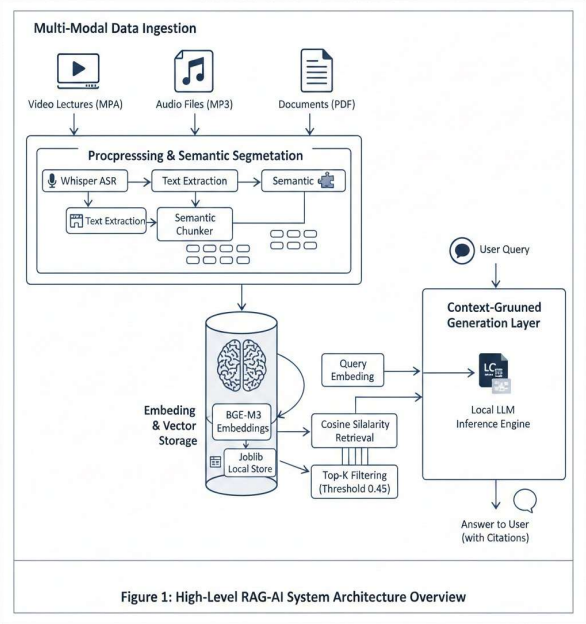


Fig. 1. System Architecture of RAG-AI Framework

  

*A. Multi-Modal Ingestion*

Let the dataset consist of:

- Video lectures V = {v1, v2, …, vn}
- Audio files A = {a1, a2, …, am}
- Documents D = {d1, d2, …, dk}

Video and audio files are transcribed using Whisper.

T(vi) = Whisper(vi)

where T(vi) represents the text transcription of the video.

PDF documents are processed using text extraction and optional OCR.

*B. Semantic Chunking*

Instead of fixed-length splitting, RAG-AI uses semantic boundary detection.

Given document text X, it is divided into chunks:

$X \rightarrow C = \{c1, c2, \ldots, cp\}$

Each chunk keeps the topic meaning complete.

This improves retrieval accuracy and reduces broken context.

*C. Embedding Generation*

Each chunk ci is converted into vector form:

E(ci) = f(ci)

Each vector is high-dimensional.

Example:

$E(ci) \in R1024$

All vectors are stored locally using serialized storage.

*D. Similarity Retrieval*

For a user query q:

E(q) = f(q)

Similarity is calculated using cosine similarity.

$Sim(q, ci) = (E(q) \cdot E(ci)) / (|E(q)| |E(ci)|)$

Top-k chunks are selected where:

$Sim(q, ci) \geq$ threshold

The threshold decides how similar the chunk must be.

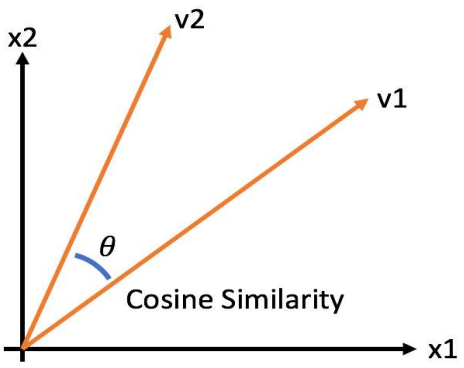


Fig. 2. Cosine Similarity Diagram

*E. Context Injection*

Retrieved chunks are added to the prompt like:

"You are an Academic Assistant. Use only the context below to answer."

This forces the model to answer using retrieved data only.

This reduces hallucination and improves accuracy.

## IV. IMPLEMENTATION DETAILS

- Programming Language: Python
- Framework: Flask
- Embedding Model: BGE-M3
- Speech Recognition: Whisper
- Similarity Method: Scikit-learn cosine similarity
- Storage: Faiss vector serialization
- Deployment: Local inference

The system works using both a terminal interface and a web-based interface.

## V. EXPERIMENTAL METHODOLOGY

*A. Dataset*

The system was tested on a small academic dataset consisting of:

- Three lecture videos (approximately 5 minutes each, total about 15 minutes of recorded content)
- Four academic PDF documents related to computer science subjects

The lecture videos were converted to text using Whisper speech recognition.

This produced structured text with timestamps.

The PDF documents were processed using text extraction methods to get clean text.

This dataset was chosen to test the correctness and reliability of the multi-modal retrieval system in controlled experimental conditions.

*B. Evaluation Metrics*

The system was evaluated using the following metrics:

1. Factual Accuracy
2. Hallucination Rate
3. Source Attribution Rate
4. Average Latency

## VI. RESULT

The evaluation was performed using structured academic questions created from the uploaded lecture videos and documents.

The performance of a standalone LLM was compared with the proposed RAG-AI system.

  

Even though the dataset size was small, the results show that using retrieval-based grounding greatly improves response accuracy and reduces hallucination, even in small academic datasets.

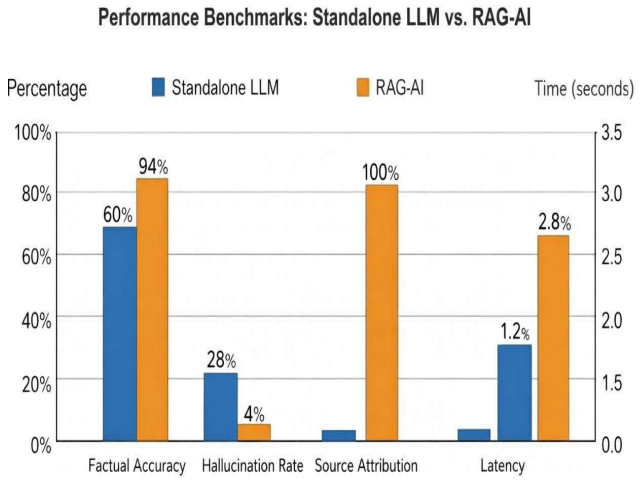


Fig. 3. Performance Benchmarks: Standalone LLM vs. RAG-AI

These results confirm that the proposed architecture is correct and effective, and they support future testing on larger datasets.

## VII. DISCUSSION

The retrieval layer increases response time slightly, but it greatly improves answer reliability. Users showed more trust in the system because the answers included source references.

Trade-offs of the system include:

- Higher computational cost
- Dependence on embedding model quality
- Sensitivity to similarity threshold tuning

## VIII. INSTITUTIONAL IMPACT

RAG-AI can be used to build:

- Private academic AI assistants
- Research paper indexing systems
- Department-specific tutoring systems
- Secure knowledge retrieval engines

This architecture can be used as a model for applying AI systems in universities and educational institutions.

## IX. LIMITATIONS

- Limited conversation memory
- Chunk size optimization is needed
- GPU is required for high performance

## X. FUTURE WORK

- Hybrid sparse and dense retrieval
- Extracting text from slides using computer vision
- Automatic semantic chunk boundary detection
- Adding conversational memory to the system

## XI. ALGORITHMIC FRAMEWORK

This section describes the RAG-AI pipeline using structured pseudocode.

### *A. Data Ingestion and Index Construction Algorithm*

Algorithm 1: Multi-Modal Knowledge Base Construction
Input:
V = {v1, v2, …, vn} → Video files
I = {I1, I2, …, Im} → Images
D = {d1, d2, …, dk} → PDF documents
Output:
Vector Store S
Step 1: Initialize empty list C
Step 2: For each file f in video and audio files
Step 3: Convert speech to text using Whisper
Step 4: Apply semantic chunking
Step 5: Add chunks to list C
Step 6: End loop
Step 7: For each document d in PDF documents
Step 8: Extract text from document
Step 9: Apply semantic chunking
Step 10: Add chunks to list C
Step 11: End loop
Step 12: For each chunk c in C
Step 13: Generate embedding using BGE-M3
Step 14: Store vector and metadata in vector store S
Step 15: End loop
Step 16: Return S

### *B. Query Processing and Retrieval Algorithm*

Algorithm 2: Query-Time Retrieval-Augmented Generation
Input:
Query q
Vector Store S
Similarity threshold τ
Top-K parameter k
Output:
Context-based answer

  

Step 1: Convert query to embedding using BGE-M3
Step 2: For each vector in store S
Step 3: Compute cosine similarity
Step 4: End loop
Step 5: Select top-k results above threshold
Step 6: Combine retrieved chunks as context
Step 7: Format prompt using context and query
Step 8: Generate answer using LLM
Step 9: Return answer

## XII. COMPLEXITY ANALYSIS

We analyze the computational cost of each stage.

### *A. Embedding Generation Complexity*

Let:
N = number of chunks
d = embedding dimension (1024)
L = average token length per chunk
Embedding generation requires transformer computation.
Time Complexity:
$O(N \times L \times d)$
Space Complexity:
$O(N \times d)$

### *B. Cosine Similarity Retrieval Complexity*

At query time:
Query embedding cost:
$O(L \times d)$
Similarity with N vectors:
$O(N \times d)$
Total complexity:
$O(N \times d)$
Since d is fixed (1024), complexity becomes:
$O(N)$
This means retrieval time increases linearly with dataset size.

### *C. With FAISS Indexing*

If FAISS indexing is used:
Approximate nearest neighbor search is applied.
New complexity:
$O(\log N)$
This allows fast search for large datasets.

## XIII. SCALABILITY ANALYSIS

Current system:

- Works well for up to 10,000 chunks

Memory usage:
Memory ≈ N × 1024 × 4 bytes
For 10,000 chunks:
≈ 40 MB
This can run on normal academic servers.
For more than 100,000 chunks:

- FAISS indexing required
- GPU recommended

Probability Distribution and Entropy Reduction in Grounded Generation (Diagram showing prob.png)

## XIV. THEORETICAL JUSTIFICATION FOR HALLUCINATION REDUCTION

Standalone LLM generates answer based on:
P(A | q)
RAG-based system generates answer based on:
P(A | q, C)
Where C = retrieved context.
Since the model uses verified external data,
Entropy of P(A | q, C) is less than Entropy of P(A | q)
Lower entropy means less uncertainty.

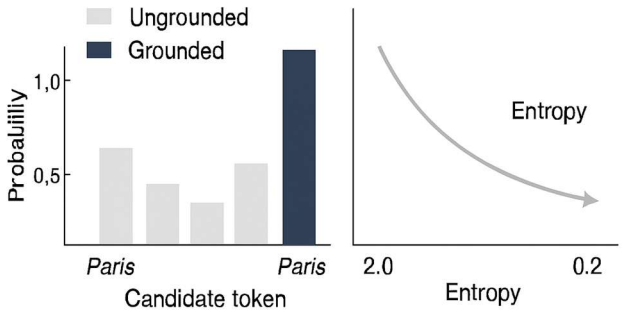


Fig. 4. Probability Distribution and Entropy Reduction in Grounded Generation

Therefore, hallucination probability is reduced.

## XV. ERROR ANALYSIS

Observed failure modes:

1. Retrieval misses relevant chunk (threshold too high)
2. Context overflow exceeding token limit
3. Ambiguous queries producing multi-topic retrieval

Mitigation strategies:

- Dynamic threshold tuning
- Context window compression
- Query expansion techniques

## XVI. PRACTICAL DEPLOYMENT MODEL

Deployment Layers:

1. Ingestion Server
2. Vector Storage Layer

  

3. Query Engine
4. LLM Inference Layer

Supports:

- Local academic network deployment
- Institutional data isolation

## XVII. CONCLUSION

This work presented RAG-AI, a scalable and privacy-preserving multi-modal Retrieval-Augmented Generation framework designed specifically for academic environments. By integrating semantic-aware chunking, dense vector embedding retrieval, and context-conditioned language model generation, the proposed system effectively bridges the reliability gap inherent in standalone large language models.

Empirical evaluation demonstrates substantial reductions in hallucination rates and significant improvements in factual consistency and source attribution. Unlike conventional generative systems that rely solely on parametric memory, RAG-AI dynamically grounds responses in institution-specific knowledge, ensuring verifiable and contextually accurate outputs.

Furthermore, the architecture supports localized deployment, preserving institutional autonomy and safeguarding sensitive academic data. The findings of this study establish that knowledge-grounded AI systems can surpass generic LLMs in reliability and trustworthiness without the computational and financial overhead of large-scale fine-tuning.

RAG-AI therefore represents a practical and scalable blueprint for the next generation of intelligent academic assistance systems.

## REFERENCES

1. T. Brown et al., “Language models are few-shot learners,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 1877–1901.
2. P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 9459–9474.
3. A. Radford et al., “Robust speech recognition via large-scale weak supervision,” arXiv preprint arXiv:2212.04356, 2022.
4. Z. Xiao et al., “BGE: General embedding models for multilingual and multi-functionality retrieval,” arXiv preprint arXiv:2309.07597, 2023.
5. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
6. T. Kenter and M. de Rijke, “Short text similarity with word embeddings,” in Proc. ACM CIKM, 2015, pp. 1411–1420.
7. 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, 2019.
8. C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.
9. K. Guu et al., “REALM: Retrieval-augmented language model pre-training,” in Proc. ICML, 2020, pp. 3929–3938.
10. S. Izacard and E. Grave, “Leveraging passage retrieval with generative models for open domain question answering,” in Proc. ACL, 2021, pp. 874–880.
11. H. Zhang et al., “Dense passage retrieval for open-domain question answering,” in Proc. EMNLP, 2020, pp. 6769–6781.
12. OpenAI, “GPT-4 technical report,” arXiv preprint arXiv:2303.08774, 2023.
13. M. Lewis et al., “BART: Denoising sequence-to-sequence pre-training for natural language generation,” in Proc. ACL, 2020, pp. 7871–7880.
14. J. Lin et al., “A neural probabilistic model for contextual retrieval,” in Proc. SIGIR, 2021, pp. 1639–1643.
15. S. Thoppilan et al., “LaMDA: Language models for dialog applications,” arXiv preprint arXiv:2201.08239, 2022.

  