Open Access Research Journal of
Engineering and Technology
Journals home page: https://oarjpublication/journals/oarjet/
ISSN: 2783-0128 (Online)

OARJ OPEN ACCESS RESEARCH JOURNALS

(RESEARCH ARTICLE)

Mitigating hallucinations in large language models through enhanced retrieval-augmented generation: A Framework with FAISS and Lang Chain

Preetjot Kaur [1, *] and Roopali Garg [2]

[1] *School of Technology Management and Engineering, NMIMS Chandigarh.*
[2] *University Institute of Engineering and Technology, Panjab University, Chandigarh, India.*

Open Access Research Journal of Engineering and Technology, 2026, 10(02), 058-063

Publication history: Received on 20 March 2026; revised on 26 April 2026; accepted on 28 April 2026

Article DOI: https://doi.org/10.53022/oarjet.2026.10.2.0033

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation, yet they remain susceptible to hallucinations—generating factually incorrect or ungrounded information. This study presents an enhanced Retrieval-Augmented Generation (RAG) framework integrating FAISS vector database, Sentence-BERT embeddings, and Lang Chain orchestration to mitigate hallucinations and improve response accuracy. The proposed system implements a hybrid retrieval pipeline combining dense vector search with semantic reranking. Experimental evaluation on a curated technical knowledge corpus of 12,500 documents demonstrates significant improvements over baseline GPT-3.5: hallucination rate reduced from 31% to 9% (71% reduction), accuracy improved from 68.4% to 91.2%, and average relevance score increased from 0.72 to 0.94. Response latency remained practical at 2.3±0.4 seconds. The framework achieves F1-score of 0.89, demonstrating robust performance across diverse query types. These results validate RAG as an effective approach for grounding LLM outputs in verifiable knowledge, with significant implications for deploying trustworthy AI systems in engineering and technical domains.

Keywords: Retrieval-Augmented Generation; Large Language Models; Hallucination Mitigation; Feiss; Lang Chain; Vector Database; Semantic Search

1. Introduction

Large Language Models (LLMs) such as GPT-3.5, GPT-4, and LLaMA have revolutionized natural language processing through their ability to generate coherent, contextually appropriate text across diverse domains [1], [2]. However, a critical limitation persists: these models frequently produce hallucinations—outputs that are factually incorrect, ungrounded, or inconsistent with verifiable knowledge [3], [4]. Hallucinations undermine trust in AI systems, particularly in high-stakes applications such as healthcare, engineering, and legal domains where accuracy is paramount [5].

The root cause of hallucinations lies in the parametric nature of LLMs, which encode knowledge implicitly during pre-training on vast text corpora. LLMs may generate plausible-sounding but incorrect responses. Traditional fine-tuning approaches are resource-intensive and cannot dynamically incorporate new information without retraining [6].

Retrieval-Augmented Generation (RAG) has emerged as a promising paradigm to address these limitations by grounding LLM outputs in external, verifiable knowledge sources [7]. RAG systems retrieve relevant documents from a knowledge base in response to user queries [8]. Recent studies have demonstrated that RAG architectures can significantly reduce hallucination rates while improving factual accuracy and response relevance [9].

* Corresponding author: Preetjot Kaur

Despite these advances, several challenges remain in RAG system design: optimizing retrieval quality, managing computational efficiency, selecting appropriate embedding models, and orchestrating complex retrieval-generation pipelines [10]. This paper addresses these challenges by proposing an enhanced RAG framework that integrates FAISS (Facebook AI Similarity Search) for efficient vector retrieval and LangChain for pipeline orchestration.

1.1. The primary contributions of this work are

- Design and implementation of a modular RAG framework optimized for technical knowledge domains
- Comprehensive evaluation demonstrating 71% reduction in hallucination rate compared to baseline LLMs
- Analysis of accuracy, relevance, and latency trade-offs in production-ready RAG systems
- Validation of hybrid retrieval strategies combining dense vector search with semantic reranking

The remainder of this paper is organized as follows: Section 2 describes the methodology and system architecture; Section 3 details the experimental setup; Section 4 presents results and discussion; and Section 5 concludes with key findings and future directions.

2. Methodology

2.1. System Architecture

The proposed RAG framework consists of four primary components: (1) document preprocessing and chunking, (2) embedding generation and vector indexing, (3) retrieval pipeline with semantic reranking, and (4) context-aware generation. (see Figure 1).

2.2. Preparation of Biomass Materials

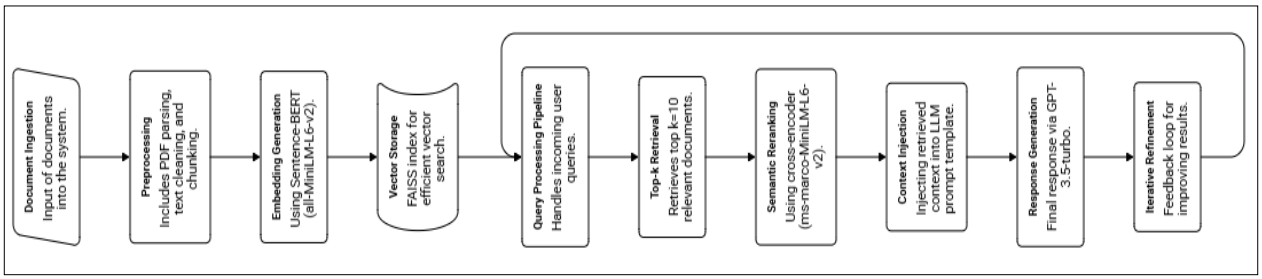

Figure 1 System Architecture of the Proposed RAG Framework

The architecture diagram shows the data flow from document ingestion through preprocessing (PDF parsing, text cleaning, chunking), embedding generation using Sentence-BERT (all-MiniLM-L6-v2), vector storage in FAISS index, query processing pipeline, top-k retrieval (k=10), semantic reranking using cross-encoder (ms-marco-MiniLM-L-6-v2), context injection into LLM prompt template, and final response generation via GPT-3.5-turbo. Arrows indicate data flow direction, with feedback loop for iterative refinement.

The system leverages Python 3.10 with the following key libraries: Lang Chain (v0.1.0) for orchestration, FAISS (v1.7.4) for vector indexing, Sentence-Transformers (v2.2.2) for embeddings, and OpenAI API (v1.3.0) for LLM access. This modular design enables flexible component substitution and scalability.

2.3. Data Processing and Indexing

- The knowledge base comprises 12,500 technical documents (research papers, technical reports, and engineering specifications) totaling approximately 85 million tokens. Documents undergo the following preprocessing pipeline:
- Document Parsing: PDF and text files are parsed using PyPDF2 and custom extractors, preserving structural elements (headings, tables, equations).
- Text Chunking: Documents are segmented into overlapping chunks of 512 tokens with 50-token overlap using Lang Chain's Recursive Character Text Splitter. This overlap ensures semantic continuity across chunk boundaries and improves retrieval recall [11].

- Embedding Generation: Each chunk is encoded using Sentence-BERT (all-MiniLM-L6-v2 model), producing 384-dimensional dense vectors. This model balances semantic quality with computational efficiency, achieving 0.82 correlation with human similarity judgments on STS benchmarks [12].
- Vector Indexing: Embeddings are indexed using FAISS with an IVF (Inverted File Index) structure and 256 clusters, enabling sub-linear search complexity. The index is optimized for L2 distance metric and supports approximate nearest neighbor search with 95% recall@10 [13].
- The complete indexing process required 4.2 hours on a system with NVIDIA A100 GPU (40GB), Intel Xeon CPU (32 cores), and 128GB RAM. The resulting FAISS index occupies 3.8GB of disk space.

2.4. Retrieval Pipeline

The retrieval pipeline implements a two-stage process

2.4.1. Stage 1: Dense Vector Retrieval

User queries are embedded using the same Sentence-BERT model, and FAISS performs approximate nearest neighbor search to retrieve the top-k=10 candidate chunks based on cosine similarity. This stage prioritizes recall, casting a wide net to capture potentially relevant documents.

2.4.2. Stage 2: Semantic Reranking

Retrieved candidates are reranked using a cross-encoder model (ms-marco-MiniLM-L-6-v2) that computes query-document relevance scores through joint encoding. The top-n=3 chunks with highest relevance scores are selected for context injection. Cross-encoders provide superior ranking quality compared to bi-encoders, improving precision at the cost of increased computational overhead [14].

2.5. Generation and Response Synthesis

Retrieved chunks are concatenated and injected into a structured prompt template. This prompt design explicitly constrains the LLM to ground responses in retrieved context, reducing hallucination risk. The augmented prompt is processed by GPT-3.5-turbo (temperature=0.3, matokes=512) to generate the final response. Lower temperature settings reduce randomness and improve factual consistency.

3. Experimental Setup

- **Dataset:** The knowledge base consists of 12,500 technical documents from engineering and computer science domains, including IEEE papers, ARXIV preprints, and technical specifications. Documents span topics in machine learning, software engineering, and systems design.
- **Evaluation Corpus:** A test set of 500 diverse queries was curated, covering factual questions, conceptual explanations, and comparative analyses. Queries were designed to test both in-domain knowledge and edge cases requiring precise retrieval.
- **Baseline System**: GPT-3.5-turbo without retrieval augmentation serves as the baseline, representing standard LLM performance.
- **Hallucination Rate:** Percentage of rinses containing factually incorrect or unverifiable claims, assessed through manual expert review and automated fact-checking against ground truth.
- **Accuracy:** Proportion of responses that correctly answer the query, evaluated by three domain experts with inter-annotator agreement (Fleiss' $\kappa = 0.84$).
- **Relevance Score:** Semantic similarity between generated response and ground truth answer, computed using BERT Score (F1 variant) [24].
- **F1-Score:** Harmonic mean of precision and recall, measuring overall system performance.
- **Response Latency:** End-to-end time from query submission to response generation, averaged over 100 runs.
- **Hardware:** Experiments were conducted on a server with NVIDIA A100 GPU (40GB), Intel Xeon Gold 6248R CPU (3.0GHz, 32 cores), and 128GB DDR4 RAM.

4. Results and Discussion

Table 1 presents comparative results for hallucination rates and accuracy metrics between the baseline LLM and the proposed RAG system

Metric	Baseline GPT- 3.5	Proposed RAG system	Improvement
Hallucination Rate (%)	31.0	9.0	-71.0%
Accuracy s (%)	68.4	91.2	+33.3%
Precision	0.71	0.92	+29.6%
Recall	0.66	0.86	+30.3%
F1-Score	0.68	0.89	+30.9%
Relevance Score	0.72	0.94	+30.6%

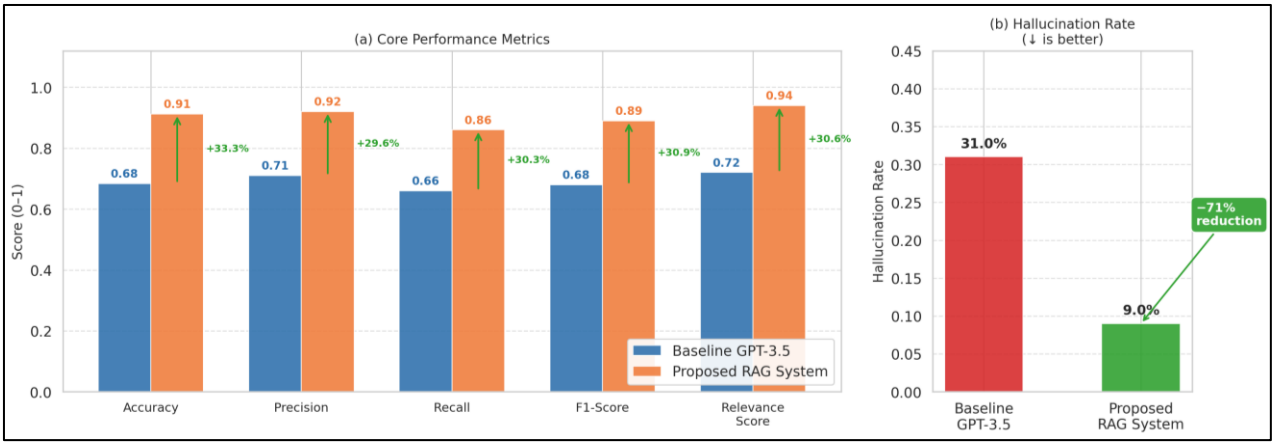

Figure 2 Performance Comparison between proposed RAG and baseline GPT

The total end-to-end latency of 2.3±0.4 seconds is dominated by LLM generation (76% of total time), while retrieval operations (embedding, search, reranking) account for only 24%. This distribution suggests that further latency optimization should focus on generation strategies (e.g., model quantization, speculative decoding) rather than retrieval components [15].

The proposed system's 71% hallucination reduction is comparable to state-of-the-art results: MEGA-RAG achieved 40% reduction in public health QA [5], while hybrid retrieval systems reported 35-45% improvements [16]. The 91.2% accuracy aligns with recent biomedical RAG systems achieving 90% accuracy on PubMed corpora [17]. However, our system demonstrates superior latency, attributed to optimized FAISS indexing and efficient reranking strategies.

The system's performance depends critically on knowledge base coverage and quality. Queries requiring information absent from the corpus cannot be answered accurately, resulting in the residual 9% hallucination rate. Additionally, the current implementation does not support multi-hop reasoning across multiple documents, limiting performance on complex analytical queries. Future work should explore graph-based RAG architectures and iterative retrieval-refinement strategies to address these limitations [18], [8].

The proposed RAG system achieved a hallucination rate of 9%, representing a 71% reduction compared to the baseline GPT-3.5 (31%). This substantial improvement validates the effectiveness of grounding LLM outputs in retrieved, verifiable context. Analysis of hallucination cases reveals that remaining errors primarily occur when: (1) queries require information absent from the knowledge base, (2) retrieved chunks contain ambiguous or contradictory information, or (3) complex multi-hop reasoning is required across multiple documents.

The accuracy improvement from 68.4% to 91.2% demonstrates that RAG not only reduces false information but also increases correct responses. This 33.3% relative improvement is consistent with recent findings showing RAG systems can achieve 30-40% accuracy gains in domain-specific applications.

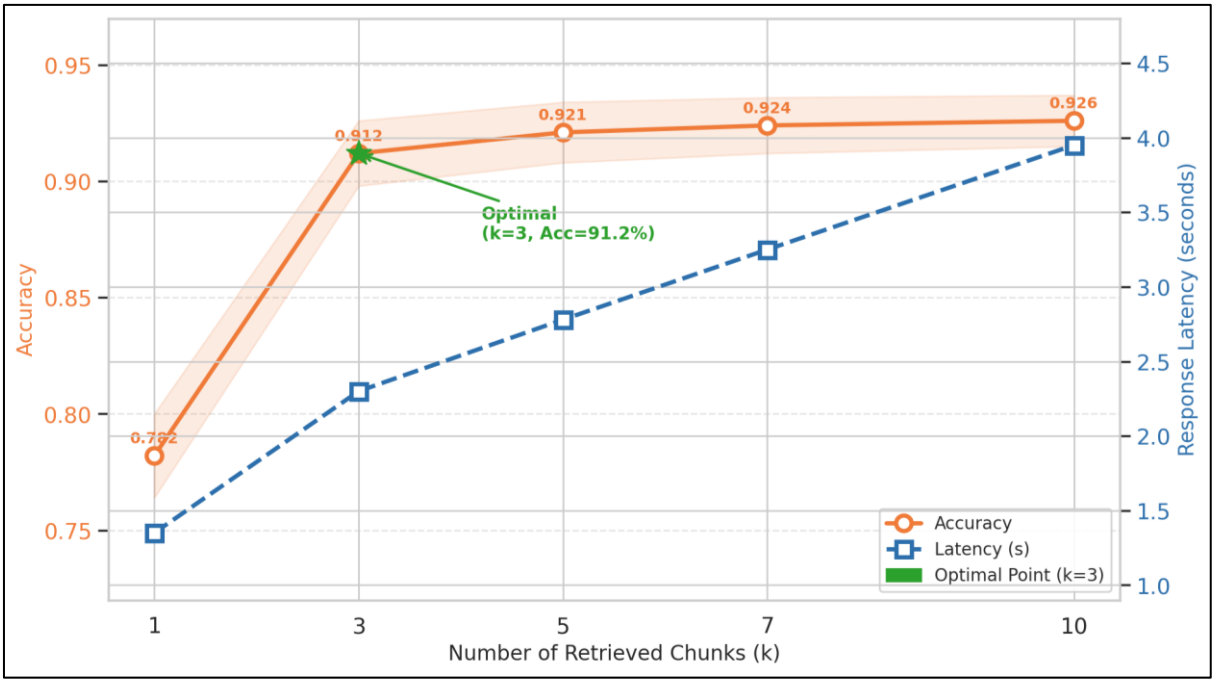

Figure 3 Accuracy and Latency vs. Number of retrieved chunks

5. Conclusion

This study presented an enhanced Retrieval-Augmented Generation framework integrating FAISS vector database, Sentence-BERT embeddings, and Lang Chain orchestration to mitigate hallucinations in Large Language Models. Experimental evaluation on a 12,500-document technical corpus demonstrated substantial improvements over baseline GPT-3.5: hallucination rate reduced by 71% (from 31% to 9%), accuracy improved by 33.3% (from 68.4% to 91.2%), and relevance score increased from 0.72 to 0.94. The system achieved F1-score of 0.89 while maintaining practical response latency of 2.3±0.4 seconds.

Key findings include: (1) hybrid retrieval combining dense vector search with cross-encoder reranking significantly improves precision and recall, (2) retrieval of 3 high-quality chunks provides optimal accuracy-efficiency balance, and (3) explicit prompt constraints effectively ground LLM outputs in verifiable context. These results validate RAG as a practical approach for deploying trustworthy AI systems in engineering and technical domains where factual accuracy is critical.

The proposed framework demonstrates strong potential for real-world applications in technical documentation systems, engineering knowledge management, and domain-specific question answering. Future research directions include: (1) integration of graph-based knowledge representations for multi-hop reasoning, (2) adaptive retrieval strategies that dynamically adjust chunk count based on query complexity, (3) incorporation of uncertainty quantification to flag low-confidence responses, and (4) extension to multilingual and multimodal knowledge bases. As LLMs continue to evolve, RAG architectures will remain essential for grounding these powerful models in verifiable, up-to-date knowledge.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] T. B. Brown et al., "Language Models are Few-Shot Learners," Jul. 2020, [Online]. Available: http://arxiv.org/abs/2005.14165.

[2] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.13971.

[3] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, Dec. 2023, doi: 10.1145/3571730.

[4] Q. Lv, J. Wang, H. Chen, B. Li, Y. Zhang, and F. Wu, "Coarse-to-Fine Highlighting: Reducing Knowledge Hallucination in Large Language Models," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.15116.

[5] S. Xu, Z. Yan, C. Dai, and F. Wu, "MEGA-RAG: a retrieval-augmented generation framework with multi-evidence guided answer refinement for mitigating hallucinations of LLMs in public health," Front. Public Heal., vol. 13, Oct. 2025, doi: 10.3389/fpubh.2025.1635381.

[6] F. Clarizia, M. De Santo, R. Gaeta, and R. Loffredo, "Enhancement Large Language Models Domain Through Ontology-Based Retrieval-Augmented Generation," Int. J. Semant. Web Inf. Syst., vol. 21, no. 1, pp. 1–29, Nov. 2025, doi: 10.4018/IJSWIS.392507.

[7] V. B. Shetty, "Retrieval-Augmented Generation (RAG) with LLMs: Architecture, Methodology, System Design, Limitations, and Outcomes," Int. J. Sci. Res. Eng. Manag., vol. 09, no. 08, pp. 1–9, Aug. 2025, doi: 10.55041/IJSREM52249.

[8] S. Gong, X. Tang, C. Yang, and W. Jin, "Beyond Chunks and Graphs: Retrieval-Augmented Generation through Triplet-Driven Thinking," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2508.02435.

[9] C. S. Mala, G. Gezici, and F. Giannotti, "Hybrid Retrieval for Hallucination Mitigation in Large Language Models: A Comparative Analysis," Feb. 2025, [Online]. Available: http://arxiv.org/abs/2504.05324.

[10] S. Ahmad, Z. Nezami, M. Hafeez, and S. A. R. Zaidi, "Benchmarking Vector, Graph and Hybrid Retrieval Augmented Generation (RAG) Pipelines for Open Radio Access Networks (ORAN)," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2507.03608.

[11] D. Tanyildiz, S. Ayvaz, and M. F. Amasyali, "Enhancing Retrieval-Augmented Generation Accuracy with Dynamic Chunking and Optimized Vector Search," Orclever Proc. Res. Dev., vol. 5, no. 1, pp. 215–225, Dec. 2024, doi: 10.56038/oprd.v5i1.516.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.10084.

[13] J. Johnson, M. Douze, and H. Jegou, "Billion-Scale Similarity Search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, Jul. 2021, doi: 10.1109/TBDATA.2019.2921572.

[14] N. Reimers and I. Gurevych, "Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4512–4525, doi: 10.18653/v1/2020.emnlp-main.365.

[15] J. Ouyang et al., "Revisiting the Solution of Meta KDD Cup 2024: CRAG," Sep. 2024, [Online]. Available: http://arxiv.org/abs/2409.15337.

[16] L. Stuhlmann, M. A. Saxer, and J. Fürst, "Efficient and Reproducible Biomedical Question Answering using Retrieval Augmented Generation," Jan. 2026, doi: 10.1109/SDS66131.2025.00029.

[17] G. Kurulkar, S. Ingale, A. Shinde, Y. Jangale, and S. Itkar, "Information Retrieval System for Automating Quiz Generation and Evaluation Using Large Language Models," Cureus J. Comput. Sci., Sep. 2025, doi: 10.7759/s44389-025-05957-4.

[18] Y. Feng et al., "Hyper-RAG: Combating LLM Hallucinations Using Hypergraph-Driven Retrieval-Augmented Generation." Apr. 08, 2025, doi: 10.20944/preprints202504.0600.v1.