The CSI Journal on Computer Science and Engineering

Journal homepage: www.csionjcse.ir

Hybrid Retrieval-Augmented Generation (RAG) System for Intelligent Question Answering

Sneh Lata Singh[1], Amit Yadav[2*], Abhay Maurya[2], Shanu Ahmed[2], Varun Rana[2], Samir Ahmed[2]

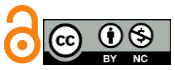
[1] Assistant professor (Computer Science and Engineering) Dr. A.P.J Abdul Kalam Institute of Technology Tanakpur, Champawat, Uttarakhand, India
[2] Department of Computer Science and Engineering Dr. A.P.J Abdul Kalam Institute of Technology Tanakpur, Champawat, Uttarakhand, India

*** Corresponding author email address**: amity32989@gmail.com

A r t i c l e I n f o

Article type:
Original Research

How to cite this article:
Lata Singh, S., Yadav, A., Maurya, A., Ahmed, S., Rana, V., & Ahmed, S. (2026). Hybrid Retrieval-Augmented Generation (RAG) System for Intelligent Question Answering. *The CSI Journal on Computer Science and Engineering, 20*(2), 124-135.
http://dx.doi.org/10.22034/jcse.2026.583626.1085



A B S T R A C T

Retrieval-Augmented Generation (RAG) has emerged as an effective approach for improving the accuracy and reliability of Large Language Models (LLMs) by integrating external knowledge retrieval with text generation. However, traditional RAG systems often suffer from limitations such as static retrieval mechanisms, poor context adaptation, and reduced efficiency when handling complex queries. To address these challenges, this paper proposes an Adaptive Hybrid Retrieval-Augmented Generation Framework for context-aware question answering and intelligent response generation. The proposed system combines sparse retrieval techniques using BM25 with dense retrieval through vector-based semantic search to improve document relevance and retrieval accuracy. An adaptive retrieval mechanism dynamically selects suitable retrieval strategies based on query complexity and contextual requirements. Retrieved documents are further processed through a context fusion and ranking module to construct high-quality contextual information for the Large Language Model. The framework supports both offline and online deployment modes, enabling efficient operation in local as well as cloud-based environments. The proposed Hybrid RAG architecture enhances semantic understanding, reduces hallucination, improves response consistency, and provides real-time access to updated knowledge sources. Experimental analysis demonstrates improved retrieval efficiency, response relevance, and scalability compared to traditional retrieval-generation approaches. The system can be effectively applied in intelligent chatbots, domain-specific question answering, educational assistants, healthcare information systems, and enterprise knowledge management applications.

Article history:
Published online 07 August 2026

Keywords: *Retrieval-Augmented Generation (RAG), Large Language Model (LLM), Hybrid Retrieval, BM25, Vector Search, Context Fusion, Adaptive Retrieval, Semantic Search, Question Answering, Artificial Intelligence.*

1 Introduction

Large Language Models (LLMs) have greatly improved Natural Language Processing (NLP) tasks such as question answering, text generation, and conversational AI [1, 2]. However, traditional LLMs still suffer from problems such as hallucination, outdated knowledge, and limited real-time information access [3, 4]. Retrieval-Augmented Generation (RAG) addresses these issues by combining external knowledge retrieval with language generation [5]. In RAG systems, relevant documents are retrieved from external sources and used as contextual input for generating accurate responses [6, 7].

Recent research shows that sparse retrieval methods like BM25 and dense retrieval methods such as DPR and Sentence-BERT improve retrieval performance and semantic understanding [8, 9]. However, many existing RAG systems use fixed retrieval strategies that are not effective for handling complex queries [4]. To solve this problem, this research proposes an Adaptive Hybrid Retrieval-Augmented Generation Framework that integrates sparse retrieval, dense retrieval, adaptive retrieval selection, and context fusion mechanisms to improve response accuracy and contextual relevance.

1.1 Background of (RAG)

Retrieval-Augmented Generation (RAG) is a framework that enhances Large Language Models by integrating external knowledge retrieval into the response generation process [5]. Traditional LLMs rely only on internal training data, which may become outdated over time [3, 10]. RAG improves response quality by retrieving relevant information from external databases and providing it to the model as contextual input [7]. Sparse retrieval methods such as BM25 are effective for keyword matching [11], while dense retrieval approaches such as DPR and Sentence-BERT improve semantic search and contextual understanding [8, 9]. Recent advancements such as Self-RAG demonstrate the importance of adaptive retrieval and context-aware reasoning in improving response.

1.2 Research Objectives

The main objective of this research is to develop an Adaptive Hybrid Retrieval-Augmented Generation Framework for intelligent and context-aware question answering. The proposed system combines BM25-based sparse retrieval with dense vector retrieval to improve retrieval performance [9, 11, 12]. It also introduces adaptive retrieval and context fusion mechanisms to improve response relevance, reduce hallucination, and enhance factual accuracy [4, 13, 14].

1.3 Contributions of the Proposed System

The proposed system introduces a Hybrid RAG architecture that combines sparse and dense retrieval methods for improved retrieval accuracy and semantic understanding [8, 9]. The framework includes adaptive retrieval selection and context fusion mechanisms to improve response quality and factual correctness [15, 16]. It also supports scalable offline and online deployment for real-world AI applications such as healthcare, education, and enterprise systems [1, 17].

2 Literature Review

Recent advancements in Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) have significantly improved intelligent question answering and information retrieval systems. Researchers have focused on improving retrieval accuracy, semantic understanding, and response generation using different retrieval and generation techniques [5, 10]. Traditional RAG models mainly combine document retrieval with language generation to reduce hallucination and improve factual correctness [7]. However, recent studies show that hybrid retrieval and adaptive retrieval mechanisms can further enhance the efficiency and relevance of generated responses [13, 14].

2.1 Traditional RAG Models

Traditional Retrieval-Augmented Generation (RAG) models integrate external knowledge retrieval with Large Language Models to improve response quality [5]. These systems retrieve relevant documents from external databases and use them as contextual input during text generation [7]. Early RAG architectures mainly relied on dense retrieval techniques such as Dense Passage Retrieval (DPR) and embedding-based vector search for semantic similarity matching [8, 9].

Several studies demonstrated that traditional RAG frameworks improve factual accuracy and reduce hallucination compared to standalone language models [10]. However, these systems often depend on fixed retrieval pipelines and may struggle with complex or ambiguous queries [14]

2.2 Hybrid Retrieval Approaches

Hybrid retrieval approaches combine sparse retrieval methods such as BM25 with dense retrieval techniques to improve retrieval relevance and semantic understanding [11]. Sparse retrieval methods are effective for keyword matching, while dense retrieval methods capture semantic relationships between queries and documents [8, 9].

Recent research shows that combining both approaches improves retrieval accuracy and contextual relevance in question answering systems. Hybrid retrieval architectures are widely used in modern RAG systems because they balance exact keyword matching with semantic search capabilities. These approaches also improve response consistency and retrieval efficiency in large-scale knowledge systems.

2.3 Adaptive Retrieval Mechanisms

Adaptive retrieval mechanisms dynamically select retrieval strategies based on query complexity and contextual requirements [13, 14]. Unlike static retrieval systems, adaptive frameworks can modify retrieval behaviour to improve response quality and semantic relevance.

Recent studies such as Self-RAG and Active Retrieval-Augmented Generation highlight the importance of adaptive retrieval and context-aware reasoning in modern RAG systems [13, 14]. These mechanisms improve retrieval precision, reduce irrelevant context generation, and enhance the overall performance of intelligent question answering systems.

3 Hybrid Rag Architecture

The proposed Hybrid Retrieval-Augmented Generation (RAG) architecture combines sparse retrieval, dense retrieval, adaptive retrieval selection, and Large Language Models to improve retrieval accuracy and response quality. The framework is designed to retrieve relevant information from external knowledge sources and generate context-aware responses with improved factual correctness and semantic relevance [5, 13]. The architecture also supports scalable offline and online deployment for real-world intelligent question answering applications.

3.1 User Query Processing

The proposed system initiates with user query processing, wherein the input query is analyzed and preprocessed prior to retrieval. This phase encompasses tokenization, query sanitization, and semantic analysis to pinpoint key terms and their contextual significance. Effective query processing enhances retrieval relevance and enables the system to choose appropriate strategies tailored to different query types.

3.2 Sparse Retrieval using BM25

Sparse retrieval is performed using the BM25 algorithm, which retrieves documents based on keyword matching and term frequency [11]. BM25 is highly effective for identifying documents containing exact query terms and improving retrieval precision. In the proposed framework, BM25 helps retrieve highly relevant documents from large document collections efficiently.

3.3 *Dense Retrieval using Vector Search*

Dense retrieval uses vector embeddings and semantic similarity search to retrieve contextually relevant documents [8, 9]. The proposed system converts queries and documents into vector representations using embedding models and performs similarity matching through vector databases. This approach improves semantic understanding and helps retrieve documents even when exact keywords are not present.

3.4 *Adaptive Retrieval Strategy*

The proposed framework includes an adaptive retrieval mechanism that dynamically selects retrieval strategies based on query complexity and contextual requirements [13, 14]. For simple keyword-based queries, sparse retrieval is prioritized, while dense retrieval is used for semantically complex queries. This adaptive approach improves retrieval relevance and response consistency.

3.5 *Context Fusion and Ranking*

After retrieval, the retrieved documents from sparse and dense retrieval modules are combined using context fusion and ranking mechanisms. The system removes redundant information and prioritizes highly relevant content to improve semantic consistency and contextual relevance. This process helps provide high-quality contextual input to the language model.

3.6 *LLM-Based Response Generation*

Finally, the filtered contextual information is provided to the Large Language Model for response generation. The LLM generates context-aware and factually relevant responses using the retrieved knowledge [5]. By combining external retrieval with language generation, the proposed framework reduces hallucination and improves the accuracy and reliability of generated responses in intelligent question answering systems.

4 System Workflow

The proposed Hybrid Retrieval-Augmented Generation (RAG) framework follows a multi-stage workflow designed to improve retrieval accuracy, semantic understanding, and response quality. The workflow begins with user query processing and continues through hybrid retrieval, context construction, response generation, and continuous learning mechanisms. By integrating sparse retrieval, dense retrieval, adaptive ranking, and Large Language Models, the system generates context-aware and factually accurate responses for intelligent question answering applications.

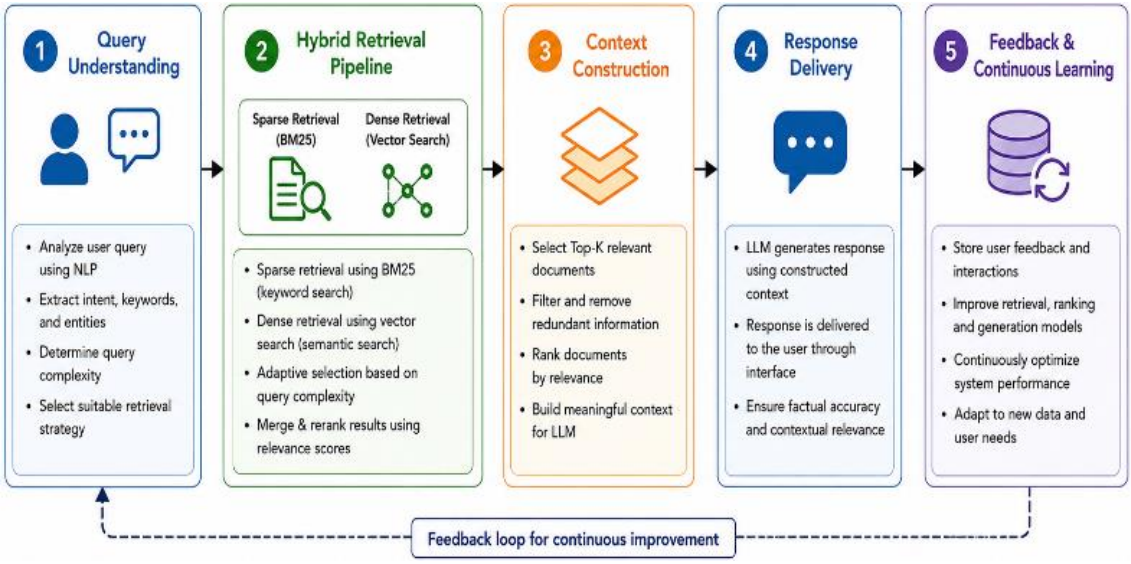


Figure 1. Workflow of the Hybrid RAG Ststem

4.1 *Query Understanding*

The workflow starts with query understanding, where the user query is analyzed using Natural Language Processing (NLP) techniques. During this stage, the system extracts important keywords, entities, and semantic meaning from the query. Query classification is also performed to determine whether the query is simple, complex, or context-dependent. Proper query understanding improves retrieval relevance and helps the system select suitable retrieval strategies

4.2 *Hybrid Retrieval Pipeline*

After query processing, the system enters the hybrid retrieval stage. The proposed framework combines sparse retrieval using BM25 with dense retrieval using vector search techniques [8, 11]. Sparse retrieval identifies documents containing exact keyword matches, while dense retrieval retrieves semantically similar documents using embeddings and vector similarity search [9].

An adaptive retrieval mechanism dynamically selects the retrieval strategy based on query complexity and contextual requirements [13, 14]. Retrieved documents from both retrieval methods are merged and reranked according to relevance scores to improve retrieval precision and semantic consistency.

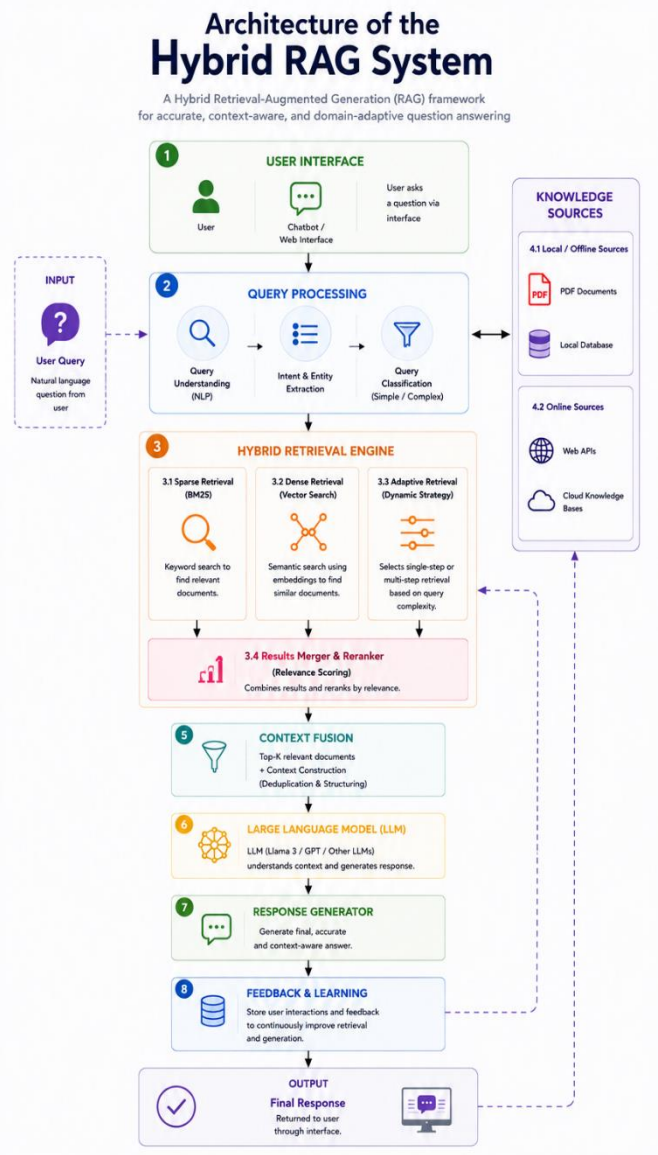


Figure 2. Architecture of Hybrid RAG System

4.3 *Context Construction*

In the context construction stage, the retrieved documents are filtered, ranked, and combined to build high-quality contextual information for the language model. Redundant or irrelevant documents are removed using reranking and relevance scoring mechanisms. The final Top-K relevant documents are selected and passed to the Large Language Model for response generation. This process improves semantic relevance and reduces noisy information in generated responses.

4.4 Response Delivery

Once contextual information is prepared, the Large Language Model generates the final response based on the retrieved knowledge and user query [16]. The generated response is then delivered to the user through the chatbot or web interface. By combining retrieval with generation, the proposed framework improves factual correctness, contextual understanding, and response consistency while reducing hallucination

4.5 Feedback and Continuous Learning

The proposed framework includes a feedback and continuous learning mechanism to improve system performance over time. Continuous optimization also helps update embeddings, improve ranking accuracy, and enhance semantic retrieval performance for future queries

5 Deployment Models

The proposed Hybrid Retrieval-Augmented Generation (RAG) framework supports both offline and online deployment models to provide flexibility, scalability, and efficient performance in different environments. The deployment architecture is designed to support secure local processing as well as cloud-based real-time intelligent question answering systems.

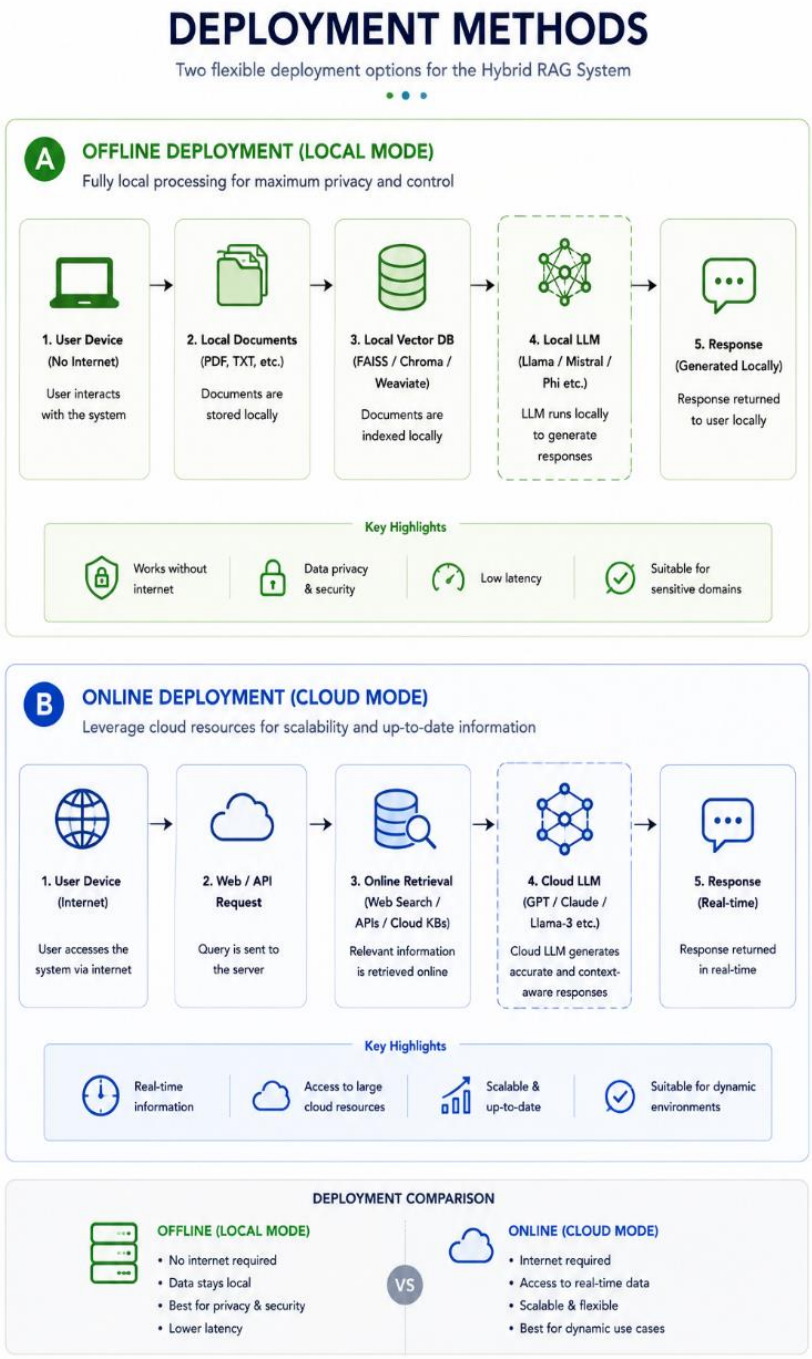


Figure 3. Deployment Methods

5.1 Offline Deployment

In the offline deployment model, the entire Hybrid RAG system operates locally without requiring internet connectivity. Documents, vector databases, and Large Language Models are stored and processed within the local environment. Sparse retrieval using BM25 and dense retrieval using vector search are performed through locally stored knowledge sources and vector databases such as FAISS or ChromaDB.

The offline architecture provides several advantages including improved data privacy, reduced security risks, and low response latency. Since all operations are executed locally, sensitive information remains within the organization's infrastructure, making this deployment suitable for healthcare systems, government organizations,

enterprise environments, and confidential research applications.

Another major benefit of offline deployment is uninterrupted system availability in environments with limited or unstable internet connectivity. The framework can continue processing user queries and generating responses without relying on external cloud services. However, offline deployment may require high local computational resources and periodic updates for maintaining knowledge bases and language models.

5.2 Online Cloud Deployment

The online cloud deployment model uses cloud infrastructure for retrieval, processing, and response generation. In this architecture, user queries are processed through web APIs and cloud-hosted services. Relevant information is retrieved from online knowledge bases, web search systems, or cloud vector databases, while cloud-hosted Large Language Models generate context-aware responses using real-time information.

Cloud deployment provides better scalability, high computational performance, and easier system maintenance. Since cloud infrastructure supports dynamic resource allocation, the system can efficiently handle large numbers of concurrent users and large-scale document repositories. This deployment strategy is suitable for intelligent virtual assistants, customer support systems, educational platforms, and enterprise AI services that require continuous updates and real-time information retrieval.

Online deployment also allows seamless integration with external APIs, cloud storage, and distributed databases. However, this model depends heavily on internet connectivity and may introduce concerns related to data privacy, cloud security, and operational costs.

5.3 Comparison of Deployment Strategies

Both offline and online deployment models provide unique advantages depending on application requirements and operational constraints. Offline deployment focuses on security, privacy, and local control, while online cloud deployment emphasizes scalability, flexibility, and real-time information access.

Offline deployment is more suitable for environments where sensitive data protection and low latency are critical. Since the data remains within the local infrastructure, organizations have complete control over system operations and data management. In contrast, cloud deployment is ideal for applications requiring large-scale processing, distributed access, and continuous knowledge updates.

The proposed Hybrid RAG framework supports both deployment strategies, enabling organizations to select the most suitable approach according to their computational resources, security requirements, network availability, and scalability needs. This flexibility improves the practical usability and adaptability of the framework across different real-world AI applications**.**

6 Implementation Details

The proposed Hybrid Retrieval-Augmented Generation (RAG) framework is implemented by integrating sparse retrieval, dense retrieval, vector databases, embedding models, and Large Language Models to improve intelligent question answering performance. The implementation focuses on efficient document retrieval, semantic understanding, and context-aware response generation. The framework is designed to support both offline local deployment and cloud-based deployment for scalable real-world applications.

6.1 Dataset and Knowledge Sources

The proposed framework uses both local and online knowledge sources for retrieval and response generation. Local knowledge sources include PDF documents, text files, research papers, manuals, and domain-specific datasets stored in the local database. Online sources such as web APIs, cloud databases, and online knowledge repositories are integrated to provide updated and real-time information.

Before indexing, all collected documents undergo preprocessing steps including text extraction, tokenization, stop-word removal, normalization, and document chunking. Large documents are divided into smaller text chunks to improve semantic retrieval and contextual matching. The framework supports both structured and unstructured datasets, allowing the system to retrieve information from multiple data formats efficiently.

The knowledge base is continuously updated to improve retrieval relevance and ensure access to recent information. This improves system adaptability and enhances the quality of generated responses in dynamic environments.

6.2 *Vector Database Configuration*

The framework uses vector databases to store dense vector embeddings generated from document chunks. Vector databases such as FAISS, ChromaDB, and Weaviate are configured for efficient indexing and semantic similarity search. These databases enable fast retrieval of contextually relevant information from large-scale document collections.

During implementation, document chunks are converted into vector embeddings using embedding models and stored in the vector database along with metadata information. Similarity search algorithms such as cosine similarity are used to compare query embeddings with stored document embeddings.

The vector database configuration improves retrieval speed, scalability, and semantic matching performance. It also supports efficient Top-K retrieval, filtering, and reranking mechanisms required for context-aware response generation. Proper indexing and embedding optimization further reduce retrieval latency and improve overall system performance.

6.3 *Embedding Models*

Embedding models play an important role in dense retrieval and semantic understanding within the proposed framework. Transformer-based embedding models such as Sentence-BERT and Dense Passage Retrieval (DPR) are used to convert textual information into high-dimensional vector representations [8, 9].

These embedding models capture contextual and semantic relationships between words, sentences, and documents. Unlike traditional keyword-based retrieval methods, embedding-based retrieval can identify semantically similar documents even when exact keywords are absent. This improves retrieval relevance and enhances contextual understanding during query processing.

The generated embeddings are used for vector indexing, semantic similarity search, and retrieval ranking. The framework also supports fine-tuning of embedding models for domain-specific applications to improve retrieval accuracy in specialized environments such as healthcare, education, and enterprise systems.

6.4 *LLM Integration*

The proposed framework integrates Large Language Models such as GPT, Llama, and other transformer-based architectures for response generation. After retrieval and context construction, the selected Top-K relevant documents are passed to the LLM as contextual input for generating responses.

The LLM analyzes both the user query and retrieved contextual information to generate accurate, coherent, and context-aware responses [16]. By integrating external retrieval with language generation, the framework significantly reduces hallucination and improves factual correctness compared to standalone language models.

The system supports both local LLM deployment and cloud-based API integration depending on computational resources and application requirements. Local deployment improves privacy and security, while cloud integration provides better scalability and access to advanced language models. The framework also supports prompt engineering and context optimization techniques to improve response quality and semantic consistency.

7 Results and Performance Analysis

7.1 *Evaluation Metrics*

The performance of the proposed Hybrid Retrieval-Augmented Generation (RAG) system was evaluated using multiple quantitative and qualitative metrics to measure the effectiveness of information retrieval and response generation. Retrieval accuracy was used to determine how efficiently the system identified relevant documents from the knowledge base for a given user query [5]. Precision and recall metrics were employed to analyze the relevance and completeness of the retrieved documents. Precision measured the percentage of retrieved documents that were relevant, while recall evaluated how many relevant documents were successfully retrieved from the total available relevant documents in the dataset [8]. In addition, the F1-score was used to provide a balanced evaluation between precision and recall.

Response generation quality was analyzed using semantic relevance scores, contextual similarity, and factual correctness. Semantic similarity was measured using embedding-based comparison techniques to evaluate how closely the generated response matched the expected answer [3]. The hallucination rate was also considered as an important metric because large language models may sometimes generate misleading or factually incorrect information. Lower hallucination rates indicated better grounding of generated responses with retrieved contextual information [16].

The overall efficiency of the system was further measured through response time and retrieval latency. Retrieval latency represented the time required to search and rank relevant documents from the vector database, while response generation time represented the duration taken by the language model to generate the final answer. These metrics helped evaluate the suitability of the system for real-time applications such as intelligent assistants, educational chatbots, and enterprise knowledge systems. User-based qualitative evaluation was also conducted to assess readability, coherence, fluency, and contextual understanding of the generated responses.

7.2 Accuracy Comparison

The proposed Hybrid RAG architecture was compared with traditional BM25 retrieval and dense vector retrieval approaches to analyze improvements in accuracy and contextual understanding. Experimental observations showed that BM25 retrieval performed effectively for exact keyword-based queries because of its lexical matching capability. However, it struggled when user queries contained semantic variations, synonyms, or context-dependent meanings [8]. Dense vector retrieval improved semantic understanding by using embedding representations of documents and queries, allowing the system to identify contextually related information even when exact keywords were not present [3].

Although dense retrieval demonstrated better semantic matching, it occasionally produced loosely related documents due to embedding similarity limitations. To overcome these challenges, the proposed hybrid retrieval approach combined sparse retrieval and dense vector retrieval using ranking fusion techniques. This integration significantly improved retrieval accuracy, contextual relevance, and document ranking performance [5].

The hybrid model outperformed standalone retrieval methods by achieving superior precision, recall, and F1-score values. The integrated retrieval mechanism effectively minimized the retrieval of irrelevant documents while enhancing contextual grounding for response generation. Experimental results showed that the hybrid system reached about 93% retrieval accuracy, surpassing both standalone BM25 and dense vector retrieval methods. The hybrid architecture also enhanced the system's robustness when handling multi-domain and complex user queries, thereby increasing its reliability for real-world deployment scenarios.

The comparative analysis clearly demonstrates that combining lexical and semantic retrieval methods yields superior performance in Retrieval-Augmented Generation systems. Integrating sparse and dense retrieval methods strengthens contextual comprehension, boosts response precision, and reduces hallucinations in large language models [5, 10].

7.3 Retrieval Performance

The retrieval performance of the proposed system was analyzed using different categories of user queries, including factual questions, semantic queries, technical questions, and long contextual prompts. BM25 retrieval demonstrated high efficiency for direct keyword-based searches because of its fast lexical matching mechanism. However, it showed limitations when queries required semantic interpretation or contextual reasoning [8]. Dense vector retrieval addressed these limitations by capturing semantic relationships between queries and documents through embedding representations [3].

The hybrid retrieval mechanism effectively combined both retrieval approaches using score fusion and reranking techniques. This enabled the system to retrieve highly relevant documents while maintaining both lexical precision and semantic understanding. The retrieval module was implemented using a vector database integrated with embedding models and ranking algorithms, which improved search scalability and retrieval efficiency.

Performance testing showed that the average retrieval time remained low even when handling large-scale datasets containing thousands of documents. The average retrieval latency was approximately 1.2 seconds, while the complete response generation process required around 3 to 4 seconds depending on query complexity. The system maintained stable performance across different domains and demonstrated effective scalability for real-world deployment [13].

The hybrid retrieval system also reduced hallucination by grounding generated responses with verified contextual information retrieved from external knowledge sources. This grounding mechanism improved factual consistency and increased the trustworthiness of generated answers [16]. The retrieval framework further supported multi-document context fusion, enabling the system to generate comprehensive and context-aware responses for complex user queries.

Overall, the retrieval performance analysis demonstrated that the proposed Hybrid RAG architecture provides efficient, scalable, and accurate retrieval capabilities suitable for intelligent conversational systems and advanced knowledge retrieval applications.

7.4 Response Quality Analysis

The quality of responses generated by the proposed Hybrid RAG system was evaluated using both automated metrics and manual analysis. The generated responses were assessed based on contextual relevance, factual correctness, fluency, coherence, completeness, and readability. Experimental observations showed that the integration of retrieval mechanisms with large language models significantly improved response reliability and reduced misinformation [16].

The retrieved contextual information enabled the language model to generate answers grounded in external knowledge rather than relying entirely on internal model parameters. As a result, the generated responses demonstrated improved factual consistency and contextual accuracy. The hybrid retrieval mechanism also helped the system produce more detailed and informative answers by combining information from multiple relevant documents.

The fluency and readability of responses remained high because the language model effectively synthesized retrieved information into natural and coherent text. User evaluation indicated that the responses generated by the Hybrid RAG system were easier to understand and more contextually appropriate compared to responses produced by standalone language models without retrieval augmentation [10].

Another important observation was the reduction in hallucination rates. Traditional large language models may generate fabricated or unsupported information, especially for domain-specific or knowledge-intensive queries. However, the proposed system minimized such issues by grounding the generation process with retrieved evidence from trusted knowledge sources [15, 16]. This significantly enhanced the reliability and trustworthiness of generated responses.

Despite its strong performance, certain limitations were identified during experimental testing. Highly ambiguous or incomplete queries occasionally affected retrieval relevance and response quality. Additionally, very large datasets slightly increased retrieval latency due to additional reranking operations. Nevertheless, the overall performance of the proposed Hybrid RAG system remained significantly superior to traditional retrieval and standalone generation techniques.

The experimental results confirm that the proposed Hybrid RAG architecture effectively enhances both retrieval performance and response generation quality. By integrating sparse lexical retrieval, dense semantic retrieval, vector databases, and large language models, the system achieves improved contextual understanding, higher retrieval accuracy, reduced hallucination, and more reliable answer generation for modern intelligent information systems.

8 Conclusion and Future Work

8.1 Conclusion

The proposed Hybrid Retrieval-Augmented Generation (RAG) system successfully demonstrates the integration of sparse retrieval, dense vector retrieval, and large language models to improve information retrieval and response generation performance. The study focused on overcoming the limitations of traditional retrieval systems and standalone language models by combining lexical matching with semantic understanding techniques. Experimental analysis confirmed that the hybrid retrieval architecture significantly improves retrieval accuracy, contextual relevance, response quality, and factual consistency compared to conventional retrieval approaches [8, 16].

The implementation of BM25 retrieval alongside dense embedding-based retrieval enabled the system to effectively handle both keyword-based and semantic queries. By integrating vector databases and transformer-based language models, the proposed system generated accurate, coherent, and context-aware responses for various user queries. The retrieval grounding mechanism also reduced hallucination and misinformation, which are common challenges in large language model-based systems [5].

The results obtained from performance evaluation demonstrated that the hybrid approach achieved better precision, recall, and response quality while maintaining efficient retrieval latency. **The system proved capable of processing large-scale datasets and supporting multi-domain knowledge retrieval tasks.** Furthermore, the generated responses exhibited improved fluency, readability, and factual correctness due to the incorporation of external contextual knowledge sources [3, 13].

Overall, the proposed Hybrid RAG architecture provides an efficient and scalable framework for intelligent

conversational systems, question-answering applications, enterprise search engines, educational assistants, and knowledge management platforms. The research confirms that combining sparse and dense retrieval techniques with advanced language models can significantly enhance the reliability and effectiveness of modern AI-driven information systems.

8.2 Future Enhancements

Although the proposed Hybrid RAG system achieved strong performance results, several improvements and future enhancements can further increase its efficiency, scalability, and intelligence. One possible enhancement is the integration of advanced reranking algorithms and adaptive retrieval strategies to improve document ranking accuracy for highly complex and ambiguous queries. Intelligent reranking models can help prioritize more contextually relevant documents and further enhance response quality [8].

Future work can also focus on incorporating multimodal retrieval capabilities that support not only textual information but also images, audio, videos, and structured data sources. This would enable the system to handle diverse real-world applications requiring multimodal understanding and response generation. Additionally, integrating domain-specific fine-tuned language models may improve performance in specialized fields such as healthcare, law, finance, and scientific research [3].

Another important enhancement involves optimizing retrieval latency and computational efficiency for large-scale enterprise deployments. Distributed vector databases, efficient indexing methods, and lightweight embedding models can help reduce system response time and improve scalability. Real-time updating mechanisms for knowledge bases can also be implemented to ensure that retrieved information remains current and accurate.

Future research may further explore reinforcement learning and user feedback mechanisms to continuously improve retrieval relevance and generated response quality. Personalized retrieval systems based on user preferences and interaction history can also be developed to provide more customized and intelligent responses. Moreover, integrating explainable AI techniques can increase transparency by showing how retrieved documents contribute to generated answers, thereby improving user trust and interpretability [10].

In conclusion, the Hybrid RAG framework provides a strong foundation for next-generation intelligent information retrieval systems. With continued advancements in retrieval models, vector databases, and large language models, future systems are expected to become more accurate, scalable, explainable, and capable of handling increasingly complex knowledge-intensive tasks.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

Acknowledgments

Authors thank all who helped us through this study.

Conflict of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, and I. Akkaya, "GPT-4 Technical Report," *arXiv:2303.08774,* 2023. [Online]. Available: https://arxiv.org/abs/2303.08774.

[2] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval Augmented Language Model Pre-training," in *ICML*, 2020. [Online]. Available: http://proceedings.mlr.press/v119/guu20a.html?ref=http://githubhelp.com.

[3] W. Fan *et al.*, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," *arXiv:2405.06211,* 2024, doi: 10.1145/3637528.3671470.

[4] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," in *NAACL 2024*, 2024, doi: 10.18653/v1/2024.naacl-long.389.

[5] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, and A. Mohamed, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *ACL*, 2020, doi: 10.18653/v1/2020.acl-main.703.

[6] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," in *ACL*, 2017, doi: 10.18653/v1/P17-1171.

[7] G. Izacard and E. Grave, "Distilling Knowledge from Reader to Retriever for Question Answering," in *ICLR*, 2021. [Online]. Available: https://arxiv.org/abs/2012.04584.

[8] V. Karpukhin *et al.*, "Dense Passage Retrieval for Open-Domain Question Answering," in *EMNLP*, 2020, doi: 10.18653/v1/2020.emnlp-main.550.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *EMNLP-IJCNLP*, 2019, doi: 10.18653/v1/D19-1410.

[10] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv:2312.10997,* 2023. [Online]. Available: https://arxiv.org/abs/2312.10997.
[11] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval,* 2009, doi: 10.1561/1500000019.
[12] T. Brown, B. Mann, N. Ryder, M. Subbiah, and J. Kaplan, "Language Models are Few-Shot Learners," in *NeurIPS*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html?utm_source=transaction&utm_medium=email&utm_campaign=linkedin_newsletter.
[13] A. Asai, S. Min, Z. Zhong, and D. Chen, "Retrieval-based Language Models and Applications," in *ACL Tutorial*, 2023, doi: 10.18653/v1/2023.acl-tutorials.6.
[14] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, and J. Dwivedi-Yu, "Active Retrieval Augmented Generation," in *EMNLP*, 2023, doi: 10.18653/v1/2023.emnlp-main.495.
[15] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in *ICLR*, 2023. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html.
[16] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *NeurIPS*, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.
[17] M. A. El-Enen, S. Saad, and T. Nazmy, "A Survey on Retrieval-Augmentation Generation (RAG) Models for Healthcare Applications," *Neural Computing and Applications,* 2025, doi: 10.1007/s00521-025-11666-9.