Advances in Retrieval-Augmented Generation (RAG) and Related Frameworks

Sumedha Arya[1], Nirmal Gaud[2]

[1]New Delhi, India
[2]ThinkAI - A Machine Learning Community, Bhopal, India

Abstract

Retrieval-Augmented Generation (RAG) has reshaped natural language processing by integrating external databases for knowledge retrieval and performing sequence-to-sequence generation. It improves the accuracy and relevance of responses in knowledge-intensive tasks. This review explores recent advances in RAG, focusing on novel frameworks, industry applications, and associated challenges. We examine innovations such as Mixture-Embedding and Confident RAG, multimodal and knowledge graph-based systems, and domain-specific applications in education, nutrition, and space industries. Additionally, we analyze RAG's execution flow, evaluation metrics, and enterprise implementations, highlighting its ability to access proprietary data securely. Technical, system-level, and ethical challenges, including retrieval quality, latency, privacy, and bias, are discussed alongside potential solutions. This review underscores RAG's potential to revolutionize AI applications while identifying critical areas for future research.

Keywords: RAG, Mixture-Embedding, Confident RAG, Knowledge Graph, Multimodal AI

1. Introduction

With rapid advancements in AI, the world is transforming toward new technologies. Large Language Models (LLMs) are a major breakthrough, but their knowledge is limited to their training data, which can lead to confident but incorrect answers, known as hallucinations. RAG addresses this by incorporating up-to-date facts during response generation, reducing errors and ensuring more accurate answers. For instance, when a question is posed, the RAG system searches a large database for relevant information and uses it to generate a more accurate response. This makes RAG a smarter and more reliable approach to answer generation.

Given a query x, the retriever R picks a small set of documents Z = {z1, z2, ..., zK} from a large collection C. The generator then creates a response y based on x and the selected documents Z, using a probability distribution:

P(y | x) = ∑(i=1 to K) P_ret(z_i | x) P_gen(y | x, z_i),

where P_ret(z_i | x) is the retriever's probability of selecting document z_i, and P_gen(y | x, z_i) is the generator's conditional probability. The retriever employs dense passage retrieval (DPR), computing similarity via:

sim(x, d) = E_q(x)^T E_p(d).

This approach reduces hallucinations in large language models by providing outputs in retrieved evidence, enabling dynamic knowledge updates without retraining. The following sections summarize recent advances in RAG and related frameworks.

2. Execution Flow of a RAG System

A typical RAG system processes a query through the following steps:

- Query Encoding: The retriever's query encoder E_q performs encoding of the input query x into a vector v_q, capturing its semantic meaning in a dense embedding space.
- Document Retrieval: Using v_q, the system searches the index of documents (corpus C) by computing similarity scores sim(x, d) = E_q(x)^T E_p(d) for documents d, retrieving the top K documents Z = {z1, ..., zK} with the highest scores, preferred as most relevant.
- Context Preparation: The text of the retrieved documents Z is accessed from the knowledge database, providing K passages (for example, Wikipedia paragraphs) as context. Depending on the fusion strategy, these passages are concatenated or handled separately.
- Answer Generation: The query x and the context Z are fed into the generator. In early fusion, all or a subset of Z is input to the encoder. In late fusion, the generator considers each z_i individually, producing an output sequence y by attending to relevant parts of Z. The probability is computed as in the equation above, often using sampling or beam search to select the best y.
- Fusion and Output: Multiple candidate outputs (if generated per document) are aggregated, typically selecting the most likely sequence y. The system returns y as the response, optionally including source passages for provenance.

This flow integrates retrieval and generation efficiently, reducing the problem space from millions of documents to a few relevant ones. With vector databases, such as FAISS, query encoding and retrieval take milliseconds, enabling real-time applications with high accuracy.

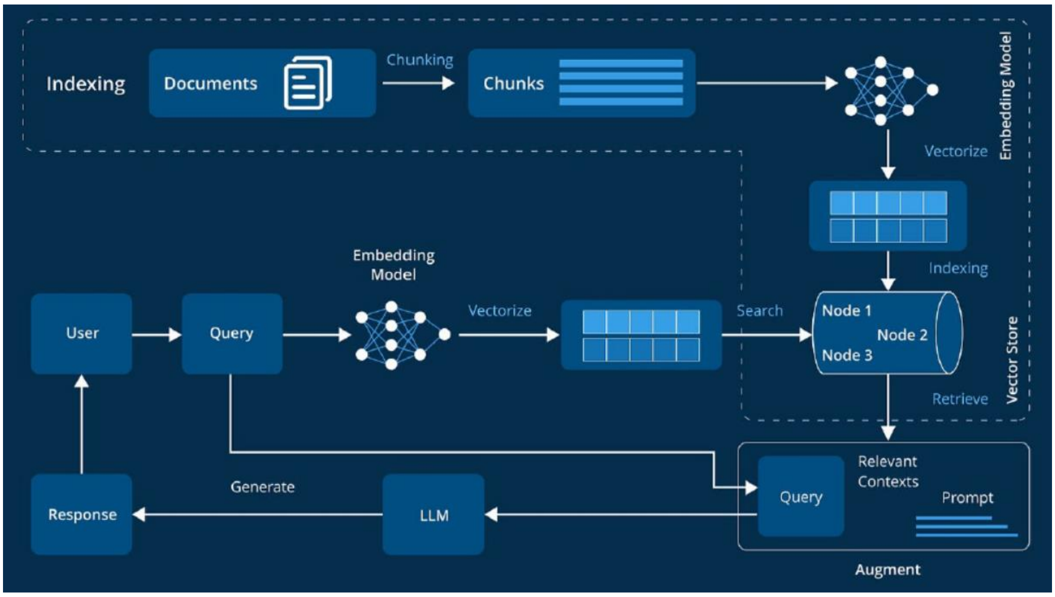

Figure 1: RAG Architecture [1]

3. Literature Review

Retrieval-Augmented Generation (RAG) improves Large Language Models by adding it with external databases during inference, providing access to updated information without retraining [1]. The effectiveness of RAG depends significantly on the choice of embedding models. Although prior studies focused on improving individual embeddings through fine-tuning for domain-specific tasks, limited research explored combining multiple embeddings. To address this, Chen et al. (2025) introduced two approaches: Mixture-Embedding RAG, which aggregates results from multiple embeddings but failed to outperform standard RAG, and Confident RAG, which generates multiple outputs using different embeddings and selects the most confident one, achieving an improvement of approximately 10% over basic LLMs and 5% over vanilla RAG [2].

Understanding relationships between documents is crucial for organizing large databases. Traditional methods rely on manual labeling or fixed relationship categories, limiting scalability and flexibility, particularly for complex or cross-domain links. Iwamoto et al. (2025) propose EDR-MQ, a framework that uses query marginalization to identify document co-occurrence patterns across diverse user queries. At its core, MC-RAG (Multiply Conditioned Retrieval-Augmented Generation) conditions document retrieval on previously retrieved content, enabling the discovery of meaningful clusters, evidence chains, and cross-domain connections without labeled data. This approach enhances document relationship discovery for information retrieval and academic search systems [3].

The adoption of Large Language Models in educational settings has been limited to prototypes or synthetic evaluations. To address real-world needs, Nguyen-Duc et al. (2025) introduce MARAUS, a Multi-Agent and Retrieval-Augmented Generation system for university admission counseling in Vietnam. Combining hybrid retrieval, multi-agent orchestration, and LLM-based generation, MARAUS handles more than 6,000 real user queries with 92% accuracy, reduces hallucination rates from 15% to 1.45%, and maintains low latency and cost, outperforming traditional RAG and LLM-only systems in practical settings [4].

Current AI-based assessment systems for diet often overlook non-visual factors. Such factors include ingredient substitutions and personal dietary needs. The Swiss Food Knowledge Graph (SwissFKG), introduced by Rahman (2025), unifies recipes, nutritional data, allergen information, and national dietary guidelines into a structured graph. A Graph-Retrieval Augmented Generation (Graph-RAG) system leverages SwissFKG to provide personalized nutrition recommendations based on queries in natural language, achieving up to 80% accuracy. This work lays the foundation for context-aware, culturally sensitive dietary tools [5].

Traditional Retrieval-Augmented Generation struggles with complex, multi-step inference. Li et al. (2025) categorize recent advances into three stages: Reasoning-Enhanced RAG, RAG-Enhanced Reasoning, and Synergized RAG-Reasoning systems. Synergized frameworks, which tightly integrate retrieval and reasoning, improve factual accuracy, logical consistency, and adaptability. The survey also highlights open challenges and future directions for developing human-centric, trustworthy, and multimodally adaptive RAG-reasoning systems [6].

Knowledge Graph-based Retrieval-Augmented Generation (KG-RAG) systems reduce hallucinations by incorporating structured data but introduce new security risks. Zhao et al. (2025) present the first in-depth study on knowledge poisoning attacks in KG-RAG, proposing a stealthy attack scenario in which adversaries inject triple perturbation into the knowledge graph to create misleading inference chains. These corrupted chains increase the likelihood of retrieving malicious content. The study calls for enhanced LLM robustness, hybrid defenses that combine general and KG-specific models, and expanded attack scenarios that involve deletion and modification [7].

Multimodal Large Language Models in medical Retrieval-Augmented Generation tasks face challenges like insufficient or over-retrieval, leading to factual inconsistencies. Wang et al. (2025) introduce MIRA, a framework that improves factual accuracy and contextual reliability. MIRA features a Rethinking and Rearrangement (RtRa) module to calibrate and filter retrieved content and a multimodal RAG pipeline integrating image embeddings, structured medical knowledge, and query rewriting. Visual attention maps confirm an improved focus on critical input segments, advancing trustworthy multimodal AI for clinical applications [8].

Requirements engineering in the space industry demands precision and compliance with strict standards, posing challenges for smaller organizations processing unstructured documents. Arora et al. (2025) propose a modular framework powered by AI that uses Retrieval-Augmented Generation to preprocess, classify, and retrieve significant information from mission briefs and technical standards, synthesizing draft requirements with LLMs. This approach streamlines RE workflows and democratizes access to safety-critical space missions [9].

4. Applying RAG to Proprietary Datasets in Real-World Scenarios

4.1.Current Approaches

Organizations are now using RAG based applications with their own internal data through secure systems. This helps them to keep important data within their company network using on-premise or cloud-hosted databases. A local model converts the user query into vector form to find the top-k most relevant parts of their private documents. These chunks are appended to the query as context for an LLM, such as GPT-4, Grok, Gemini, or self-hosted models. This separation reduces the risk of data leakage in model parameters, although the sensitivity of the output remains a concern. Access control layers filter documents based on user permissions, while secure channels or homomorphic encryption are explored to blind retrieval, albeit with performance overhead. Fine-tuning LLMs on domain data risks memorizing private text, whereas RAG supports data freshness by updating the knowledge store without retraining.

4.2.Industry Implementation Examples

- PGA Tour: Used RAG to give better answers about golf events and player stats by using their own data. This improved user engagement with efficiency and accuracy.
- Bayer: Applied RAG to help farmers get accurate farming information quickly, helping them manage crops better and increase productivity.
- Rocket Companies: Used RAG to speed up mortgage processing by reducing time and make the customer experience smoother with their own financial data.
- Shorenstein Properties: Used RAG to automatically tag and organize files, making it easier to find relevant information to work more efficiently.

- Cohere: Advanced RAG with source citation to reduce hallucinations and enhance transparency, integrating external texts like company documents.
- NVIDIA: Incorporated RAG into enterprise AI solutions, connecting LLMs with proprietary customer data and research for accurate, relevant responses.
- IBM: Employed RAG for domain-specific applications, using proprietary customer data and research to improve response accuracy and relevance.

These examples demonstrate the ability of the RAG to enhance information retrieval, decision-making, and efficiency across industries, though challenges such as data privacy and computational overhead persist.

4.3.New Patterns in RAG Adoption Across Industries

- Accuracy vs. Latency: Models like FiD and WebGPT prioritize accuracy but increase latency due to multiple document processing, while NVIDIA's RAG targets sub-second latency.
- Scaling Beyond Wikipedia: Industry RAG systems use both private company data and live web information, keeping them updated, while academic models usually work with data like Wikipedia.
- Reducing Hallucination: WebGPT and IBM Watsonx RAG show sources of their answers to help ensure that the information is correct.
- Model Size vs. Retrieval Efficiency: Smaller models with retrieval, such as Atlas and NVIDIA RAG, outperform larger standalone models, reducing scaling needs.
- Enterprise Adaptation: Industry RAG prioritizes reliability, database integration, and privacy compliance over academic benchmark performance.

5. Conclusion

Retrieval-Augmented Generation represents a significant advancement in natural language processing, combining information retrieval with generative modeling to produce accurate, contextually relevant responses. This paper reviews key innovations, including novel embedding strategies, multimodal and graph-based frameworks, and real-world applications in education, nutrition, and space industries. Marketplace implementations highlight RAG's ability to securely leverage proprietary data, improving decision-making and operational efficiency in sectors like finance, agriculture, and customer support. However, challenges such as retrieval quality, latency, data privacy, and bias remain. Future research should focus on improving retrieval precision through advanced encoders and query reformulation, reducing latency with efficient indexing and caching, and addressing ethical concerns like bias and misinformation through diverse data curation and robust verification mechanisms. By overcoming these challenges, RAG has the potential to become a cornerstone of trustworthy, scalable, and adaptive AI systems for knowledge-intensive applications.

References

1. Oche A.J., Folashade A.G., Ghosal T., Biswas A., A Systematic Review of Key Retrieval-Augmented Generation (RAG) Systems: Progress, Gaps, and Future Directions, arXiv preprint arXiv:2507.18910v1, July 2025.
2. Chen S., Zhao Z., Chen J., Each to Their Own: Exploring the Optimal Embedding in RAG, arXiv preprint arXiv:2507.17442, July 2025.
3. Iwamoto Y., Tsunoda K., Kaneiwa K., Extracting Document Relations from Search Corpus by Marginalizing over User Queries, arXiv preprint arXiv:2507.10726, July 2025.
4. Nguyen-Duc A., Manh C.V., Tran B.A., Ngo V.P., Chi L.L., Nguyen A.Q., An Empirical Study of Multi-Agent RAG for Real-World University Admissions Counseling, arXiv preprint arXiv:2507.11272, July 2025.
5. Rahman L.A., Introducing the Swiss Food Knowledge Graph: AI for Context-Aware Nutrition Recommendation, arXiv preprint arXiv:2507.10156, July 2025.
6. Li Y., Zhang X., Wang Y., Liu Z., Chen Q., Towards Agentic RAG with Deep Reasoning: A Survey of RAG-Reasoning Systems in LLMs, arXiv preprint arXiv:2507.09477, July 2025.
7. Zhao T., Li J., Zhang W., Sun Y., Wang Q., RAG Safety: Exploring Knowledge Poisoning Attacks to Retrieval-Augmented Generation, arXiv preprint arXiv:2507.08862, July 2025.
8. Wang J., Ashraf T., Han Z., Laaksonen J., Anwer R.M., MIRA: A Novel Framework for Fusing Modalities in Medical RAG, arXiv preprint arXiv:2507.07902, July 2025.
9. Arora C., Wang F., Tantithamthavorn C., Aleti A., Kenyon S., From Domain Documents to Requirements: Retrieval-Augmented Generation in the Space Industry, arXiv preprint arXiv:2507.07689, July 2025.