Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Review Article

A Review on Retrieval-Augmented Generation: Architectures, Research Challenges, and Emerging Frontiers

Pratik Sharma [1] **and Saishab Bhattarai** [2,*]

[1] Department of Mathematics, Kathmandu University, Kavre 45200, Nepal; e-mail : ps2224578@gmail.com
[2] Department of Information Technology, Phoenix College of Management, Lincoln University College, Kota Bharu 4730,1 Malaysia; e-mail : saishab.bhattarai.75@gmail.com
* Corresponding Author : Saishab Bhattarai

Abstract: Retrieval-Augmented Generation (RAG) enhances the capabilities of Large Language Models (LLMs) by integrating external knowledge retrieval into the generation pipeline, enabling responses that are grounded, adaptive, and up to date. While RAG can improve factual accuracy compared to models relying solely on pre-trained data, its effectiveness in practice depends strongly on the quality, relevance, and interpretability of retrieved context, and does not eliminate hallucinations entirely. Recent architectures such as Fusion-in-Decoder, Atlas, and ColBERT-RAG demonstrate measurable gains in retrieval precision, scalability, and cross-domain generalization. However, persistent challenges remain, including retrieval noise that can override model reasoning, hallucinations that persist even with high-quality evidence, latency constraints that hinder real-time deployment, and fragile domain adaptation. Moreover, although partial metrics and task-specific benchmarks exist, the absence of a unified evaluation framework for retrieval–generation grounding and robustness complicates fair comparison and reproducible progress. Rather than offering an exhaustive survey, this review provides a focused analytical perspective on retrieval design and architectural evolution in RAG systems. It consolidates representative architectures while critically examining structural limitations related to retrieval–generation coupling as a design choice, context over-reliance, and privacy-preserving computation. Building on these insights, the paper outlines future research directions, including structured knowledge integration via GraphRAG, modular agent-based orchestration in Agentic RAG, improved retrieval filtering, and unified evaluation methodologies. As RAG architectures continue to evolve rapidly in a pre-standardization phase, a more analytically grounded understanding of their design trade-offs is essential for advancing trustworthy and domain-adaptive language systems.

Keywords: Context-Aware Generation; Domain-Specific Reasoning; Hallucination Mitigation; Human-in-the-Loop AI; Information Retrieval; Knowledge Grounding; Large Language Models; Retrieval-Augmented Generation.

Received: November, 20th 2025
Revised: December, 16th 2025
Accepted: December, 21st 2025
Published: January, 2nd 2026
Curr. Ver.: January, 2nd 2026

1. Introduction

Large Language Models (LLMs) are artificial intelligence models trained on massive datasets that excel at understanding and generating human language. They can complete sentences, answer questions, translate text, and even write code or stories. However, these models acquire all of their knowledge from the data available during training and are unable to access new or updated information once training has concluded. As a result, LLMs suffer from several persistent limitations. Their knowledge becomes outdated over time, they frequently hallucinate by producing fabricated or unsupported information, and they struggle to recall rare or domain specific facts that are poorly represented in training corpora [1]–[3]. These issues significantly limit their reliability in high stakes domains such as medicine, law, and business, where accuracy, timeliness, and traceability are essential.

To address these shortcomings, researchers have proposed augmenting LLMs with external knowledge sources that can be retrieved dynamically at inference time. This approach,

known as Retrieval-Augmented Generation (RAG), retrieves relevant documents in response to a user query and integrates them into the generation process [4]. By grounding responses in retrieved evidence, RAG has been shown to improve factual correctness and reduce certain forms of parametric hallucination, while enabling access to updated knowledge without retraining large models [4]–[6]. Consequently, RAG has emerged as a promising paradigm for knowledge-intensive tasks such as open-domain question answering and fact verification. Nevertheless, current RAG implementations still exhibit important limitations. Retrieval modules may surface noisy or partially relevant documents, generators may underutilize or misinterpret retrieved evidence, retrieval–generation alignment remains inconsistent, and latency often increases as retrieval quality improves. Moreover, RAG pipelines frequently transfer poorly across domains, requiring costly re-indexing or fine-tuning to maintain performance.

Despite substantial progress, research on RAG continues to reveal significant evidence gaps. While several task-specific metrics and partial benchmarks exist, there is currently no widely accepted unified evaluation framework for assessing grounding quality, hallucination behavior, and retrieval effectiveness, which complicates fair comparison across architectures. The interaction between retrieval and generation also remains insufficiently understood, particularly with respect to when models rely on parametric memory versus retrieved evidence. Privacy and security concerns are underexplored, especially in domains involving sensitive or proprietary data. In addition, most existing RAG studies focus on short-text contexts, with limited investigation into long-context reasoning or multimodal retrieval.

These unresolved challenges have motivated growing interest in emerging paradigms such as GraphRAG, which incorporates structured knowledge to support multi hop reasoning, and Agentic RAG, which introduces modular retrievers, planners, and generators to enable more adaptive workflows. Future research directions increasingly emphasize real time and freshness aware retrieval, multimodal RAG systems that integrate text with images and tabular data, privacy preserving retrieval pipelines, and unified evaluation methodologies for assessing factuality, relevance, robustness, and efficiency. Strengthening these dimensions is essential for advancing RAG toward trustworthy, domain adaptive, and explainable AI systems capable of supporting high stakes applications.

While Retrieval Augmented Generation has attracted significant attention, existing review studies largely adopt a broad survey perspective, emphasizing application coverage, benchmark comparisons, or end to end pipeline descriptions. Many of these surveys catalog retrieval methods and generative models independently, without critically examining how retrieval design choices interact with architectural structure to influence system behavior. As a result, key questions remain insufficiently analyzed, including how retrieval noise propagates through different architectures, how tradeoffs between latency and interpretability emerge, and why hallucinations persist even when high quality contextual evidence is available [3], [7], [8].

This review addresses these limitations by explicitly focusing on the coupling between retrieval mechanisms and architectural evolution in RAG systems. Rather than treating retrievers and generators as loosely connected components, we analyze them as interdependent design elements whose interactions jointly determine grounding fidelity, scalability, and system robustness. The primary contribution of this paper is a critical synthesis of retrieval strategies and architectural progression in RAG, highlighting recurring tradeoffs, structural bottlenecks, and emerging design principles that remain underexplored in prior surveys. By consolidating insights across representative RAG architectures, this review aims to provide a clearer analytical foundation for future research and system design. Accordingly, this paper does not aim to provide a comprehensive survey of all RAG techniques, but instead focuses on analytically examining how retrieval design and architectural choices jointly shape system behavior. As many recent advances in modular, graph-based, and agentic RAG remain in a pre-standardization phase, preprint literature is deliberately included to capture ongoing architectural experimentation that has not yet been consolidated into peer-reviewed surveys. By narrowing the scope in this manner, the review prioritizes conceptual clarity and critical insight over exhaustive coverage.

A RAG system operates through an interaction between retrieval, context augmentation, and generation, which may be implemented with varying degrees of coupling depending on architectural design choices. The retrieval component is responsible for identifying relevant

documents or passages from an external corpus in response to a user query, and its effectiveness directly shapes downstream performance. Retrieval errors, such as noisy, redundant, or partially relevant passages, can propagate through the pipeline and weaken factual grounding. Retrieved evidence is then integrated with the user query during the augmentation stage to construct a richer contextual input. While this process enables stronger grounding, poorly structured or excessively long augmented contexts may overwhelm the generator and amplify retrieval noise, motivating the development of adaptive strategies such as context compression, relevance scoring, and structured knowledge integration. Finally, the generation component produces responses conditioned on the augmented input using a large language model. Despite access to external evidence, generators may still ignore retrieved content or rely excessively on parametric memory, leading to hallucinations even when correct information is available. These interdependent stages highlight why retrieval design and architectural integration, whether tightly or loosely coupled, are central to the reliability, scalability, and grounding fidelity of RAG systems..

This paper is organized as follows. Section 2 provides an overview of retrieval mechanisms and architectural components in RAG systems, using a unified conceptual framework to analyze their interactions. Section 3 examines the evolution of RAG architectures, highlighting key design shifts and tradeoffs across representative models. Section 4 discusses persistent technical, architectural, and ethical challenges that limit real world deployment. Finally, Section 5 summarizes key findings, identifies remaining evidence gaps, and outlines future research directions for advancing retrieval augmented generation.

2. Core Components of Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG) systems are built upon the interaction between retrieval mechanisms and generative models, where external knowledge is dynamically incorporated to support grounded and reliable language generation. Rather than functioning as a simple linear pipeline, RAG architectures involve an interaction between retrieval quality, context construction, and generative reasoning, whose degree of coupling varies depending on architectural design choices. These components jointly shape system performance, influencing factual grounding, interpretability, scalability, and latency. Accordingly, this section examines the core components of RAG with an emphasis on how different retrieval strategies and generator designs affect these trade-offs.

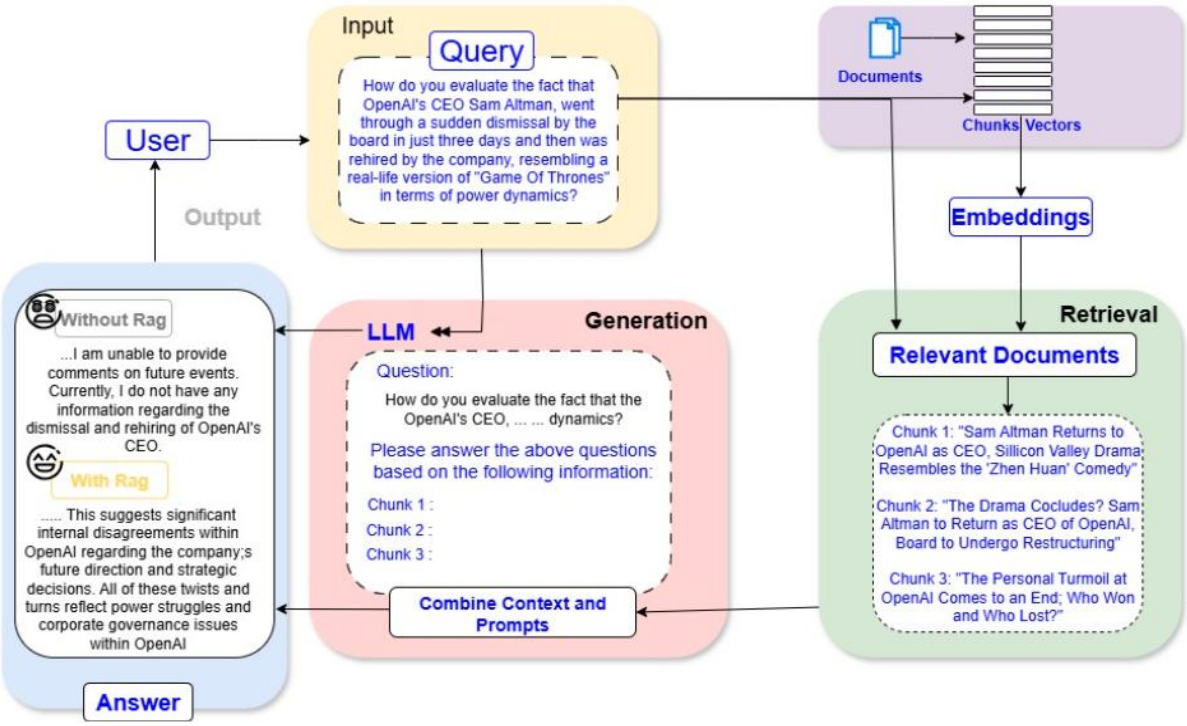

Figure 1. Retrieval augmented generation architecture

Figure 1 illustrates the overall architecture of RAG, highlighting the interaction between user queries, document retrieval, context augmentation, and generation. The figure serves as a conceptual anchor for this section by emphasizing that retrieval quality and chunk selection directly shape the reasoning behavior of the generator, rather than acting as a passive preprocessing step.

In RAG systems, the retriever plays a pivotal role by selecting relevant information from large document collections. Retrieval quality is often the primary determinant of downstream accuracy, as errors introduced at this stage can propagate through the generation process. To address diverse query types and domain requirements, a range of retrieval techniques has been developed, spanning traditional sparse methods, dense neural retrievers, and hybrid approaches that combine both paradigms.

2.1. Retriever in RAG

Sparse retrieval methods represent queries and documents as high dimensional vectors based on term frequency and lexical overlap. These methods are computationally efficient and highly interpretable, making them suitable for scenarios where speed and transparency are critical. However, their reliance on exact term matching limits their ability to capture semantic similarity, paraphrasing, and contextual meaning. Despite these limitations, sparse retrieval remains a foundational component of many RAG pipelines due to its low computational cost and strong performance on structured or well-defined queries.

Term Frequency Inverse Document Frequency (TF-IDF) quantifies the importance of a term by comparing its frequency within a document to its frequency across the entire corpus. This approach effectively emphasizes informative terms while down weighting common vocabulary, making it well suited for basic relevance scoring. Its primary limitation lies in its dependence on exact word matching, which reduces effectiveness for semantically rich or paraphrased queries. Formally, TF-IDF is defined as:

$$\mathrm{TF}(t,d) = \frac{\text{Number of occurrences of term } t \text{ in document } d}{\text{Total number of terms in document } d} \tag{1}$$

$$\mathrm{IDF}(t,D) = \log\left(\frac{\text{Total number of documents in collection } D}{\text{Number of documents containing term } t}\right) \tag{2}$$

where t denotes a term, d denotes a document, and D denotes the document corpus.

Best Matching 25 (BM25) extends TF IDF by incorporating term frequency saturation, document length normalization, and refined inverse document frequency weighting. These enhancements improve ranking robustness and make BM25 a strong baseline in modern search engines and retrieval systems [9]–[11]. Nevertheless, BM25 remains a purely lexical method and cannot capture deeper semantic relationships.

While sparse retrieval is efficient, it suffers from the vocabulary mismatch problem, where relevant documents may not share explicit lexical overlap with the query. Dense retrieval methods, such as Dense Passage Retriever, address this limitation by mapping queries and documents into continuous vector spaces using neural encoders [12]. Dense retrievers employ dual encoder architectures that independently embed queries and documents, enabling efficient approximate nearest neighbor search over large collections. By capturing semantic similarity rather than surface level term overlap, dense retrieval performs particularly well in open domain question answering. However, these methods incur higher computational cost, require large embedding indices, and are sensitive to domain shift.

To balance efficiency and semantic fidelity, hybrid retrieval approaches combine sparse and dense signals. Sparse methods provide precise keyword matching, while dense retrievers contribute semantic understanding through neural embeddings [13]. Hybrid systems often achieve higher recall and precision, especially in complex or mixed query scenarios such as open domain question answering and enterprise search [14]. Nevertheless, this improved performance comes at the cost of increased system complexity, higher latency due to re ranking, and more challenging parameter tuning.

Table 1 highlights the fundamental tradeoffs inherent in retrieval design for RAG systems. Sparse methods emphasize efficiency and interpretability but struggle with semantic generalization. Dense retrieval improves semantic fidelity and recall but introduces higher computational cost and reduced transparency. Hybrid approaches attempt to reconcile these

competing objectives by combining lexical precision with semantic matching, though at the expense of increased complexity and latency. These tradeoffs demonstrate that retrieval design is not merely an optimization choice but a structural decision that shapes deployment feasibility, scalability, and trustworthiness.

Table 1. Comparison of sparse, dense, and hybrid retrieval methods for RAG.

Method	Benefits	Costs	Limitations	When it Succeeds	When it Fails	Refs
TF IDF	Simple, interpretable, fast; effective for keyword retrieval	Very low computational cost; minimal memory	No semantic understanding; vocabulary mismatch; sensitive to paraphrasing	Exact keyword matches; structured technical documents	Semantic or paraphrased queries; cross domain questions	[11], [14]
BM25	Improved ranking; term frequency and length normalization; strong baseline	Low cost; efficient on large corpora	Lexical only; limited contextual understanding	High precision keyword tasks; static datasets	Semantic or context heavy queries	[10], [11]
Dense Retrieval (DPR)	Captures semantic similarity; robust to paraphrasing; high recall	High training cost; large embedding index	Sensitive to domain shift; expensive updates	Semantic search; open domain QA	Low resource domains; real time systems	[4], [12]
Hybrid Retrieval	High recall; balances precision and semantics	Highest computational cost; increased latency	Complex tuning; scalability challenges	Production RAG; mixed query types	Real time settings; constrained hardware	[13], [15], [16]

A common hybrid strategy is score fusion, where relevance scores from sparse and dense retrievers are combined using weighted aggregation. This approach was adopted in Facebook's RAG system, which employed BM25 alongside DPR to ensure both lexical coverage and semantic relevance [4]. Another strategy is cascade retrieval, where a lightweight sparse retriever first selects candidate documents, which are then re ranked by a more expensive dense model such as a BERT based cross encoder [17]. Recent methods such as SPLADE and uniCOIL further narrow the gap between sparse and dense retrieval by learning sparse representations using neural models, enabling improved expressiveness without abandoning inverted index structures [18]. Empirical results on benchmarks such as MS MARCO demonstrate that hybrid systems often outperform purely sparse or dense approaches, although computational overhead and tuning complexity remain open challenges [15], [16].

2.2. Generator in RAG

In Retrieval Augmented Generation, the generator is responsible for producing coherent and contextually relevant responses conditioned on retrieved evidence. At this stage, the system combines the original query with selected documents to form an augmented prompt, which is processed by a large language model to generate the final output. Depending on task requirements, the generator may supplement retrieved evidence with parametric knowledge or restrict generation strictly to the provided context [6].

Most generators used in RAG are based on the transformer architecture, which has become the dominant paradigm in natural language processing due to its ability to model long range dependencies and contextual relationships. Prominent models such as GPT, LLaMA, and Gemini are built upon transformer designs and have demonstrated strong performance across a wide range of tasks.

Text to Text Transfer Transformer (T5) [18] adopts an encoder decoder architecture that frames all natural language processing tasks as text-to-text transformations. Its unified formulation supports multi task learning and has achieved strong performance on benchmarks such as GLUE, SQuAD, and SuperGLUE, making it well suited for retrieval augmented generation scenarios involving diverse tasks.

Bidirectional and Auto Regressive Transformer (BART) [19] combines bidirectional encoding with autoregressive decoding. By training on corrupted text reconstruction, BART is particularly effective at handling noisy or partially relevant retrieved contexts, which are common in real world RAG systems. This capability improves factual consistency when integrating multiple retrieved documents.

Large Language Model Meta AI (LLaMA) [20] is an open-source decoder only transformer trained on publicly available data. Despite its smaller parameter size compared to proprietary models, LLaMA demonstrates strong efficiency and competitive performance, making it attractive for retrieval augmented settings where computational resources are constrained.

Generative Pre trained Transformer (GPT) [21] models are decoder only transformers trained on massive text corpora using self-supervised learning. Through successive architectural refinements and scaling strategies, GPT models have become widely adopted in RAG systems across domains such as healthcare, finance, and customer support, where their strong generative capabilities complement retrieval-based grounding.

3. The Evolution of Retrieval Augmented Generation Architectures

Retrieval-Augmented Generation (RAG) was originally introduced to address the limitations of purely parametric language models by combining retrieval-based grounding with the generative capabilities of LLMs. The original RAG framework consists of two closely integrated components: a retriever that selects the top-k relevant documents from an external knowledge source, and a generator that conditions on both the user query and the retrieved documents to produce an answer. This integration enabled language models to access external knowledge dynamically, improving factual accuracy and flexibility in knowledge-intensive tasks.

The original RAG architecture proposed by Meta AI combined dense retrieval with end-to-end language model training to support knowledge-grounded generation [4]. Two decoding strategies were introduced. RAG Sequence selects a single retrieved document for generation, while RAG Token marginalizes over multiple retrieved documents at the token level, allowing more fine-grained use of external evidence. Although this design represented a major step forward, early RAG models suffered from weak document fusion and competition among retrieved passages, which often limited their effectiveness as the number of retrieved documents increased.

Subsequent architectures sought to address these limitations through improved document fusion and retrieval-generation alignment. Fusion-in-Decoder (FiD) encodes each retrieved document independently before combining them in the decoder, reducing interference between passages and improving answer accuracy [22]. However, this improvement comes at the cost of quadratic memory and computational complexity as the number of retrieved documents grows. To mitigate this limitation, FiD-Light was later proposed as an efficiency-oriented extension that compresses encoder representations before decoding, substantially reducing inference latency while preserving competitive effectiveness across knowledge-intensive tasks [23]. This line of work highlights that document fusion quality and efficiency can be jointly optimized through architectural refinements rather than increasing retrieval depth alone. REALM further advanced retrieval-augmented learning by jointly training the retriever and generator using masked language modeling and contrastive objectives, enabling learned retrieval within the language modeling process itself [6]. While effective, REALM introduced training instability and substantial computational overhead. RETRO marked another important shift by treating retrieval as an external memory that can be accessed during both training and inference. By retrieving similar text chunks and incorporating them through decoder cross-attention, RETRO enables long-context generation with relatively smaller models, albeit at the expense of requiring extremely large retrieval corpora [24].

More recent RAG architectures emphasize modularity and scalability. Atlas demonstrates that strong performance can be achieved by pairing a frozen T5-based generator with a well-trained retriever, particularly in few-shot and multi-task learning scenarios [25]. This design reduces training complexity but limits adaptability at the generator level. ColBERT-based RAG systems further decouple retrieval from generation by focusing on fine-grained token-level interaction within the retriever, enabling highly efficient and scalable retrieval pipelines that can be flexibly integrated into modular RAG systems [17]. More recently, ColBERTv2 introduces lightweight late interaction through residual compression and denoised supervision, substantially reducing storage overhead while preserving or improving retrieval quality across in-domain and out-of-domain benchmarks. This advancement addresses one of the main practical limitations of earlier ColBERT models, namely their high indexing cost, and makes token-level retrieval more viable for large-scale and real-world RAG deployments

[26]. Together, these models illustrate a shift away from monolithic end-to-end pipelines toward more modular and retrieval-centric architectures that are better suited for real-world deployment across diverse domains such as law, medicine, and education.

To provide a structured view of this progression, the evolution of RAG architectures can be broadly categorized into three stages: Naive RAG, Advanced RAG, and Modular RAG. Each stage reflects increasing architectural sophistication, improved retrieval optimization, and greater adaptability to task-specific requirements. Figure 2 illustrates the high-level evolution of RAG architectures from early naive designs to advanced and modular systems, highlighting the progressive shift toward scalability, flexibility, and tighter retrieval-generation integration.

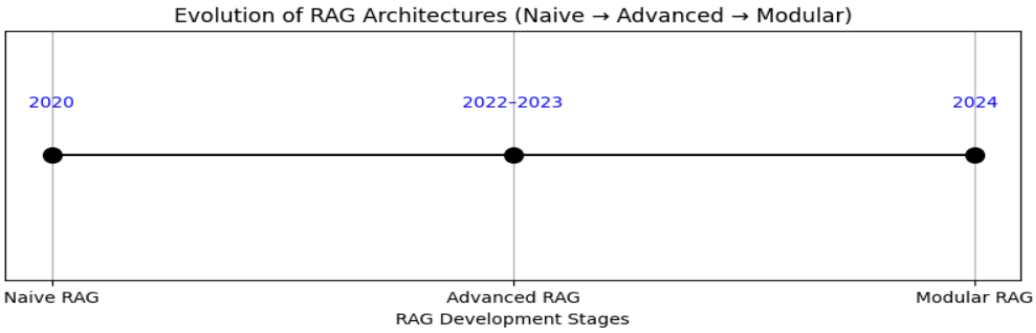

Figure 2. Evolution of RAG architectures

Table 2. Unified comparative analysis of major RAG architectures.

Architecture	Retrieval Type	Fusion Strategy	Generator	End-to-End Trainable	Scalability	Key Strengths	Key Limitations	Evolutionary Contribution	Ref
RAG	Dense	In-decoder	Yes	Yes	Medium	End-to-end learning; simple integration of retrieval and generation	Weak document fusion; competition among retrieved passages	Establishes the baseline RAG framework integrating retrieval with generation	[4]
FiD	Dense	Encoder-side (late fusion)	Yes	Partially	Medium	High answer accuracy through independent document encoding	Quadratic memory and computational cost	Improves RAG by reducing document interference during generation	[22], [23]
REALM	Learned dense	Encoder-side	Yes	Yes	Medium	Jointly trained retriever and generator	Training instability; high computational overhead	Introduces fully learned retrieval into language model training	[6]
RETRO	Dense, chunk-based	Decoder cross-attention	Yes	Partially	High	Enables long-context generation with smaller models	Requires extremely large retrieval corpora	Scales language modeling by treating retrieval as external memory	[24]
Atlas	Dense	Multi-document fusion	Yes	Yes (retriever-focused)	High	Strong few-shot and multi-task performance	Frozen generator limits adaptability	Promotes modular retriever–generator training paradigms	[25]
ColBERT	Dense (token-level)	Late interaction (retriever-side)	No	Retriever-only	Very High	Fine-grained semantic matching; efficient large-scale retrieval; ANN-friendly indexing	High storage cost; no native generation component	Enables retrieval-centric and modular RAG pipelines through token-level late interaction	[17], [26]

The architectures selected for comparison in this review, namely RAG, FiD, REALM, RETRO, Atlas, and ColBERT, represent key milestones in this evolution. Each model introduces a distinct architectural response to limitations observed in its predecessors, particularly in terms of document fusion, retrieval-generation alignment, and scalability. Collectively,

these architectures capture the dominant evolutionary trends in RAG research, progressing from early end-to-end pipelines toward more modular and retrieval-centric designs.

Table 2 summarizes the key characteristics, strengths, and limitations of representative RAG architectures. Early models tightly coupled retrieval and generation within a single pipeline, enabling end-to-end learning but suffering from weak document fusion and limited scalability. Architectures such as FiD and REALM improved retrieval utilization through better fusion and joint training but introduced higher computational cost and training complexity. RETRO demonstrated that retrieval can serve as an external memory to support long-context generation, shifting the scalability bottleneck from model size to retrieval infrastructure. More recent designs such as Atlas and ColBERT prioritize modularity by decoupling retrieval from generation, enabling flexible component replacement, improved scalability, and domain-specific customization.

This architectural evolution reflects a broader paradigm shift in how retrieval is conceptualized within language generation systems. Early RAG models treated retrieval largely as static context injection, assuming that better documents alone would yield better outputs. In contrast, advanced and modular architectures view retrieval as a dynamic, task-dependent process that must be tightly integrated with generative reasoning. This shift enables adaptive retrieval strategies, improved control over grounding behavior, and greater robustness under domain shift. As a result, modular RAG architectures represent a critical step toward scalable, adaptable, and trustworthy knowledge-grounded language systems.

3.1. Applications of Retrieval-Augmented Generation

Architectural advancements in RAG directly shape how these systems perform across different application domains. As RAG architectures evolve toward more modular and retrieval-aware designs, they become increasingly capable of handling domain-specific constraints, heterogeneous knowledge structures, and real-world deployment challenges. This subsection examines four representative application areas in which architectural design choices play a critical role in determining performance, reliability, and scalability.

3.1.1. Open-Domain Question Answering

RAG has demonstrated strong performance in open-domain question answering (ODQA), where users can pose questions spanning a wide range of topics without predefined knowledge boundaries. Traditional LLMs often hallucinate when relevant information is absent from their parametric memory, whereas RAG mitigates this issue by retrieving supporting documents prior to generation [7].

The effectiveness of RAG in ODQA largely stems from its combined use of semantic retrieval methods, such as Dense Passage Retriever (DPR), and lexical retrieval techniques, such as BM25. This hybrid retrieval strategy balances recall and precision, enabling more reliable evidence grounding. Benchmarks including Natural Questions and TriviaQA consistently show that retrieval-augmented approaches improve factual consistency compared to purely generative models [27].

Despite these gains, ODQA systems remain vulnerable to retrieval brittleness. Retrieved results often include partially relevant or overlapping passages, which can cause the generator to synthesize ambiguous or redundant answers. As datasets grow in scale and complexity, improving retrieval ranking, context filtering, and passage selection becomes increasingly important. Hierarchical retrieval and multi-hop reasoning strategies have been proposed as promising directions to further strengthen ODQA performance in RAG systems [28].

3.1.2. Customer Support and Virtual Assistance

Customer support and virtual assistance systems demand timely, context-aware, and accurate responses, often based on rapidly evolving product documentation or policy guidelines. RAG enables virtual assistants to dynamically retrieve relevant information from updated sources, reducing reliance on static, rule-based systems [29]. This capability allows organizations to scale customer support operations without sacrificing response accuracy.

However, domain drift remains a significant challenge. When documentation changes frequently, retrieval pipelines must be continuously re-indexed and validated to prevent outdated or inconsistent responses [30]. Without rigorous version control, RAG systems may retrieve obsolete or contradictory documents, leading to user confusion and dissatisfaction. Integrating freshness-aware retrieval mechanisms and metadata-informed ranking strategies can help prioritize authoritative and up-to-date sources in customer support settings.

3.1.3. Content Generation

RAG enhances content generation by grounding generated outputs in external, verifiable sources, making it particularly valuable for domains that require factual accuracy, such as journalism, academic writing, and report summarization. Rather than relying solely on parametric memory, which may be outdated or incomplete, RAG retrieves contextually relevant evidence before generation, reducing hallucinations and improving credibility [18]. Architectures such as FiD and Atlas further enable the integration of long-form contexts and multiple documents, supporting detailed outputs such as blogs, whitepapers, and scientific summaries with traceable sources [27].

By anchoring generation in retrieved evidence, RAG helps bridge the gap between creative generation and factual grounding. However, over-reliance on retrieval can constrain creativity, leading to overly factual or repetitive outputs. Balancing factual grounding with creative flexibility remains an open design challenge. Moreover, even with retrieval support, generators may still produce citations or references not explicitly present in the retrieved context, due to uncontrolled blending of retrieved content with parametric knowledge. Developing grounding score metrics that quantify how much of the generated output is directly supported by retrieved evidence could improve reliability in content-heavy applications.

3.1.4. Knowledge-Intensive Tasks

RAG is increasingly applied to knowledge-intensive tasks such as legal document analysis, biomedical information retrieval, financial report summarization, and educational tutoring. These domains require high precision and up-to-date knowledge, which RAG systems can provide through structured retrieval pipelines [31].

For instance, in biomedical applications, RAG can retrieve recent research articles to support diagnostic reasoning or literature review. In legal practice, RAG assists with case summarization and precedent retrieval but must satisfy strict accuracy and risk constraints. In financial analysis, RAG can synthesize information from market reports and regulatory documents, improving transparency and interpretability in decision making. Modular RAG architectures further enable domain-specific tuning without full model retraining, improving both performance and deployment efficiency [32].

Nevertheless, performance in these domains strongly depends on the quality and structure of the underlying knowledge base. Specialized corpora often include tables, PDFs, and semi-structured documents that complicate retrieval. Multimodal RAG systems capable of reasoning over text, tables, images, and graphs are therefore essential for high-stakes, knowledge-intensive applications [15].

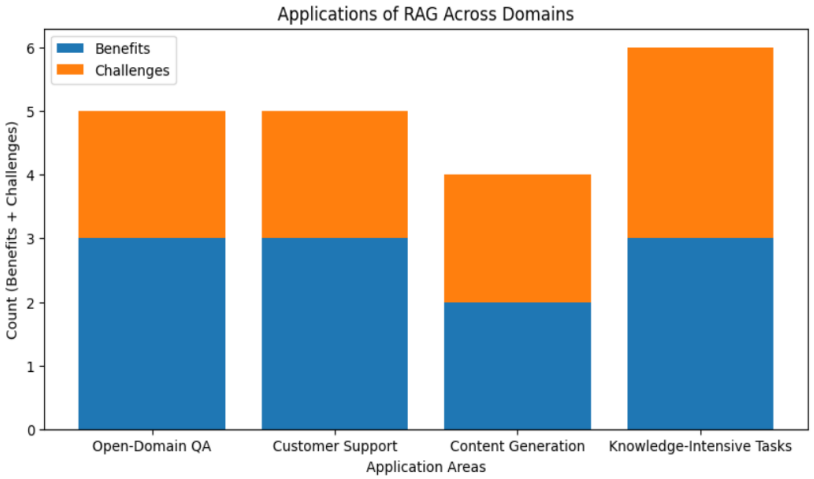

Figure 3. Applications of RAG across different domains.

Figure 3 provides a qualitative synthesis of the discussion in Sections 3.1.1–3.1.4, illustrating the balance between benefits and challenges associated with RAG across application domains. The values shown do not represent quantitative measurements but reflect a conceptual aggregation of observed trends. As domain complexity increases, RAG systems offer

greater benefits through grounding and adaptability, while simultaneously facing amplified challenges related to retrieval reliability, latency, and retrieval–generation alignment.

While architectural advancements have expanded the applicability of RAG, they also expose new limitations related to retrieval robustness, model alignment, and deployment constraints. The following section systematically examines these challenges, highlighting how technical, model-level, and ethical factors continue to shape the practical effectiveness of RAG systems.

4. Challenges of Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) systems face a range of challenges that directly affect the quality, reliability, and practical usability of their outputs. These challenges can be broadly categorized into three groups: technical issues related to retrieval and context construction, model-level alignment problems between retrieval and generation, and ethical as well as operational risks associated with real-world deployment.

4.1. Technical Challenges in Retrieval and Context Construction

One of the primary technical challenges in RAG systems lies in retrieval quality, as inaccurate or poorly matched retrieval often leads to hallucinations and incomplete responses. The effectiveness of RAG is closely tied to the relevance, granularity, and structural properties of the retrieved information. Noisy retrieval results, semi-structured documents such as tables and fragmented PDFs, and the difficulty of balancing coarse-grained versus fine-grained chunking all complicate the retrieval process and frequently degrade response quality [6].

Redundancy and document confusion represent another major technical challenge. Retrieved documents often contain overlapping or repetitive content, introducing noise and increasing computational overhead without necessarily improving response accuracy [29]. Such redundancy can overwhelm the generator, leading to verbose, repetitive, or incoherent outputs. The presence of inconsistent or conflicting retrieved sources further complicates the generation process and reduces factual consistency [4].

Latency and computational overhead also pose significant obstacles. Deep retrieval pipelines, cross-encoder re-ranking, vector similarity search, and the construction of large augmented prompts all contribute to increased processing time. As a result, RAG systems must navigate a difficult trade-off between retrieval accuracy and response speed, limiting their suitability for real-time or highly interactive applications [33].

Domain adaptation remains an additional unresolved technical limitation. RAG models trained on one domain often exhibit degraded performance when applied to another, due to differences in terminology, reasoning patterns, and contextual assumptions. This fragility makes it challenging to deploy a single RAG architecture across diverse professional or academic domains without substantial retraining, re-indexing, or corpus restructuring[31], [32].

4.2. Model-Level and Retrieval–Generation Alignment Issues

Even when retrieval quality is high, RAG systems are not immune to hallucinations. Generators may still rely excessively on parametric memory or misinterpret retrieved passages, blending external evidence with internal priors in ways that produce unsupported or fabricated statements. This issue persists across application domains and is particularly critical in high-stakes settings such as healthcare, law, and finance, where even minor inaccuracies can have serious consequences [8], [34], [35].

A related model-level limitation concerns the alignment between the retriever and the generator. Prior studies have reported failure cases in which LLMs either ignore retrieved content altogether or improperly weight retrieval signals, instead defaulting to internal knowledge representations [34] [39]. Such misalignment reduces the effectiveness of retrieval augmentation and weakens factual grounding, underscoring fundamental gaps in how retrieval evidence is incorporated and prioritized during generation.

4.3. Ethical and Operational Risks

Beyond technical and model-level challenges, RAG systems introduce important ethical and operational risks. Retrieval mechanisms may expose sensitive, proprietary, or confidential information, particularly when operating over private or semi-private knowledge bases. In

addition, adversarial actors may inject poisoned or misleading content into retrieval corpora, thereby compromising system reliability and trustworthiness.

Bias in retrieved documents presents another significant concern. Retrieval from skewed or unbalanced corpora can amplify existing biases, resulting in similarly biased generated outputs. These risks highlight the need for stronger safeguards, including fairness-aware retrieval scoring, robust data curation practices, and privacy-preserving RAG architectures. Addressing these ethical and operational issues is essential for ensuring responsible and trustworthy deployment of RAG systems.

5. Conclusion and Evidence Gaps

5.1. Summary of Key Findings

This review has examined RAG with a specific focus on the interaction between retrieval design and architectural evolution. The analysis demonstrates that improvements in factual grounding, robustness, and practical applicability do not arise solely from higher retrieval accuracy, but rather from how retrieval mechanisms are integrated within generative architectures. The progression from naïve retrieval–generation pipelines to advanced and modular RAG frameworks reflects a broader shift toward tighter retrieval–generation coupling, improved scalability, and greater adaptability across application domains. Architectural innovations such as hybrid retrieval frameworks, Fusion-in-Decoder models, Atlas, ColBERT-RAG, and modular retriever–generator designs illustrate that architectural choices play a central role in mitigating hallucinations, managing latency, and supporting domain-aware reasoning in knowledge-intensive tasks.

5.2. Evidence Gaps

Despite these advancements, several evidence gaps continue to limit the reliability and effectiveness of RAG systems in real-world settings. Retrieval noise, hallucinations that may persist even in the presence of correct context, computational latency, and redundancy in retrieved documents remain major bottlenecks. Domain adaptation also poses a significant challenge, as models trained on one type of corpus often struggle to generalize to others due to domain-specific terminology, knowledge organization, and reasoning patterns. In addition, concerns related to privacy, security, and bias further complicate deployment, as retrieval mechanisms may expose sensitive information or amplify corpus-level biases. These limitations underscore the need for more robust evaluation frameworks that emphasize transparency, fairness, grounding validity, and robustness, particularly in high-stakes environments.

A critical factor underlying many of these evidence gaps is the lack of widely adopted evaluation frameworks specifically designed for RAG systems. While a range of task-specific benchmarks and partial metrics exist, most current evaluations focus on isolated measures such as answer accuracy or retrieval recall, paying limited attention to whether generated responses are truly grounded in retrieved evidence, how systems behave under noisy or imperfect retrieval, or how corpus-level biases affect outputs across domains. Without more integrated evaluation methodologies that assess grounding validity, robustness, and fairness in combination, it remains difficult to meaningfully compare RAG architectures or assess their reliability in real-world applications. Developing such evaluation frameworks is therefore essential to ensure that architectural advances translate into trustworthy and responsible RAG deployments.

5.3. Future Research Directions

Looking forward, several promising research directions aim to address the challenges identified in this review. Real-time retrieval pipelines capable of indexing continuously updated content, such as news, scientific literature, and policy documents, can help mitigate the problem of outdated knowledge [36], [37]. Multimodal RAG (MRAG) has emerged as another important direction, enabling retrieval and generation over images, video, code, and tabular data, thereby expanding applicability to domains such as diagnostics, surveillance, and scientific modeling [38], [39]. Privacy-preserving retrieval approaches, including federated retrieval, encrypted indexes, and controlled-access pipelines, will be critical for deployment in sensitive

domains [40]. Graph-augmented RAG (GraphRAG) represents an additional trajectory, introducing structured reasoning, multi-hop evidence tracing, and improved explainability through knowledge graphs [41], [42].

Another emerging paradigm is agentic RAG, in which modular components such as retrievers, planners, evaluators, and reasoning agents collaborate within a coordinated system. This shift toward agent-based architectures enables more flexible, interpretable, and scalable pipelines, particularly for enterprise and high-assurance applications that require private data handling and tool-augmented reasoning. As highlighted in this review, this transition reflects a broader movement away from viewing RAG as a single model and toward understanding it as an orchestrated ecosystem of interacting components[43].

In summary, while RAG offers substantial benefits in grounding large language model outputs and improving factual accuracy, persistent challenges related to retrieval reliability, retrieval–generation alignment, domain generalization, latency, bias, and security remain unresolved. By critically synthesizing retrieval design and architectural evolution, this review clarifies key limitations and research opportunities, providing a more analytically grounded understanding of RAG systems and actionable guidance for future research and development.

Author Contributions: Conceptualization: P.S. and S.B.; Methodology: P.S. and S.B.; Formal analysis: P.S. and S.B.; Investigation: P.S. and S.B.; Resources: P.S.; Data curation: P.S.; Writing—original draft preparation: P.S. and S.B.; Writing—review and editing: P.S. and S.B.; Visualization: P.S.; Supervision: S.B.; Project administration: S.B.; Funding acquisition: Not applicable. P.S. and S.B. contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," *arXiv*. Jul. 22, 2020. [Online]. Available: http://arxiv.org/abs/2005.14165

[2] A. Chowdhery *et al.*, "PaLM: Scaling Language Modeling with Pathways," *arXiv*. Oct. 05, 2022. [Online]. Available: http://arxiv.org/abs/2204.02311

[3] J. T. A. Andrews, D. Zhao, W. Thong, A. Modas, O. Papakyriakopoulos, and A. Xiang, "Ethical Considerations for Responsible Data Curation," *arXiv*. Dec. 10, 2023. [Online]. Available: http://arxiv.org/abs/2302.03629

[4] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *arXiv*. Apr. 12, 2021. [Online]. Available: http://arxiv.org/abs/2005.11401

[5] S. Gupta, "Retrieval-Augmented Generation and Hallucination in Large Language Models: A Scholarly Overview," *Sch. J. Eng. Technol.*, vol. 13, no. 05, pp. 328–330, May 2025, doi: 10.36347/sjet.2025.v13i05.003.

[6] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, Feb. 2020. [Online]. Available: http://arxiv.org/abs/2002.08909

[7] Tongji Knowledge Graph and Language Model Lab, "RAG-Survey," *GitHub repository*, 2024. https://github.com/Tongji-KGLLM/RAG-Survey?tab=readme-ov-filewhat-is-rag

[8] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," *arXiv*, Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.12837

[9] K. Sparck Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments: Part 1," *Inf. Process. Manag.*, vol. 36, no. 6, pp. 779–808, Nov. 2000, doi: 10.1016/S0306-4573(00)00015-7.

[10] S. E. Robertson, S. Walker, and M. Beaulieu, "Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive," in *Text Retrieval Conference*, 1998. doi: 10.6028/NIST.SP.500-242.okapi.

[11] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.

[12] N. Khan, "Retrieval Techniques - Sparse, Dense, and Hybrid Representations," *LinkedIn*, 2024. https://www.linkedin.com/pulse/retrieval-techniques-sparse-dense-hybrid-najeeb-khan-ph-d--wmtpc/

[13] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *arXiv*. Feb. 13, 2019. [Online]. Available: http://arxiv.org/abs/1902.04885

[14] T.-Y. Liu, "Learning to Rank for Information Retrieval," *Found. Trends® Inf. Retr.*, vol. 3, no. 3, pp. 225–331, Jun. 2009, doi: 10.1561/1500000016.

[15] T. Formal, B. Piwowarski, and S. Clinchant, "SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2021, pp. 2288–2292. doi: 10.1145/3404835.3463098.

[16] P. Bajaj *et al.*, "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," *ArXiv*. Oct. 31, 2018. [Online]. Available: http://arxiv.org/abs/1611.09268

[17] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2020, pp. 39–48. doi: 10.1145/3397271.3401075.

[18] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *arXiv*, Sep. 2023, [Online]. Available: http://arxiv.org/abs/1910.10683

[19] M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *arXiv*. Oct. 29, 2019. [Online]. Available: http://arxiv.org/abs/1910.13461

[20] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *ArXiv*. Feb. 27, 2023. [Online]. Available: http://arxiv.org/abs/2302.13971

[21] G. Yenduri *et al.*, "Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions," *arXiv*. May 21, 2023. [Online]. Available: http://arxiv.org/abs/2305.10435

[22] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," *ArXiv*. Feb. 03, 2021. [Online]. Available: http://arxiv.org/abs/2007.01282

[23] S. Hofstätter, J. Chen, K. Raman, and H. Zamani, "FiD-Light: Efficient and Effective Retrieval-Augmented Text Generation," *ArXiv*. Sep. 28, 2022. [Online]. Available: http://arxiv.org/abs/2209.14290

[24] S. Borgeaud *et al.*, "Improving language models by retrieving from trillions of tokens," *ArXiv*. Feb. 07, 2022. [Online]. Available: http://arxiv.org/abs/2112.04426

[25] G. Izacard *et al.*, "Atlas: Few-shot Learning with Retrieval Augmented Language Models," *ArXiv*. Nov. 16, 2022. [Online]. Available: http://arxiv.org/abs/2208.03299

[26] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia, "ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction," *ArXiv*. Jul. 10, 2022. [Online]. Available: http://arxiv.org/abs/2112.01488

[27] J. W. Rae *et al.*, "Scaling Language Models: Methods, Analysis & Insights from Training Gopher," *arXiv*. Jan. 21, 2022. [Online]. Available: http://arxiv.org/abs/2112.11446

[28] K. Kim and J.-Y. Lee, "RE-RAG: Improving Open-Domain QA Performance and Interpretability with Relevance Estimator in Retrieval-Augmented Generation," *arXiv*. Oct. 24, 2024. [Online]. Available: http://arxiv.org/abs/2406.05794

[29] E. Mgbeahuruike, C. Ikotun, A. Ojo, E. Oyerinde, O. Famodimu, and T. Olorunoje, "Hybrid AI-Human Architecture for Real-Time Customer Support: Leveraging Retrieval-Augmented Generation," *Asian J. Res. Comput. Sci.*, vol. 18, no. 7, pp. 88–105, Jul. 2025, doi: 10.9734/ajrcos/2025/v18i7722.

[30] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," *arXiv*. Oct. 17, 2023. [Online]. Available: http://arxiv.org/abs/2310.11511

[31] M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880. doi: 10.18653/v1/2020.acl-main.703.

[32] J. Lin and X. Ma, "A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques," *arXiv*. Jun. 28, 2021. [Online]. Available: http://arxiv.org/abs/2106.14807

[33] J. Gu, "A Research of Challenges and Solutions in Retrieval Augmented Generation (RAG) Systems," *Highlights Sci. Eng. Technol.*, vol. 124, pp. 132–138, Feb. 2025, doi: 10.54097/364hex16.

[34] S. Hofstätter, M. Zlabinger, and A. Hanbury, "Interpretable & Time-Budget-Constrained Contextualization for Re-Ranking," *arXiv*, Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.01854

[35] O. Ayala and P. Bechard, "Reducing hallucination in structured outputs via Retrieval-Augmented Generation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, 2024, pp. 228–238. doi: 10.18653/v1/2024.naacl-industry.19.

[36] J. Larson and S. Truitt, "GraphRAG: Unlocking LLM discovery on narrative private data," *Microsoft*, 2024. https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/

[37] Y. Zhu, "From Static to Dynamic: A Streaming RAG Approach to Real-time Knowledge Base," *arXiv*, Jul. 2025, [Online]. Available: http://arxiv.org/abs/2508.05662

[38] L. Mei, S. Mo, Z. Yang, and C. Chen, "A Survey of Multimodal Retrieval-Augmented Generation," *arXiv*. Mar. 26, 2025. [Online]. Available: http://arxiv.org/abs/2504.08748

[39] W. Chen, H. Hu, X. Chen, P. Verga, and W. W. Cohen, "MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text," *arXiv*. Oct. 20, 2022. [Online]. Available: http://arxiv.org/abs/2210.02928

[40] Q. Mao *et al.*, "Privacy-Preserving Federated Embedding Learning for Localized Retrieval-Augmented Generation," *arXiv*. Apr. 27, 2025. [Online]. Available: http://arxiv.org/abs/2504.19101

[41] T. Ni *et al.*, "StepChain GraphRAG: Reasoning Over Knowledge Graphs for Multi-Hop Question Answering," *arXiv*. Oct. 03, 2025. [Online]. Available: http://arxiv.org/abs/2510.02827

[42] M. A. Shavaki, P. Omrani, R. Toosi, and M. A. Akhaee, "Knowledge Graph Based Retrieval-Augmented Generation for Multi-Hop Question Answering Enhancement," in *2024 15th International Conference on Information and Knowledge Technology (IKT)*, Dec. 2024, pp. 78–84. doi: 10.1109/IKT65497.2024.10892619.

[43] S. Satish, "RAG is dead: why enterprises are shifting to agent-based AI architectures," *techradar.com*, 2025. https://www.techradar.com/pro/rag-is-dead-why-enterprises-are-shifting-to-agent-based-ai-architectures