BULGARIAN ACADEMY OF SCIENCES
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2025 • Vol. **83**, pp. 32-57
p-ISSN: 2738-7356; e-ISSN: 2738-7364
https://doi.org/10.7546/PECR.83.25.03

Retrieval-Augmented Generation (RAG): Advances and Challenges

Miroslava Dimitrova[1]

[1]*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria*
Emails: miroslava.dimitrova@iict.bas.bg, mddimitova@gmail.com

Abstract: The growing reliance on Large Language Models (LLMs) in knowledge-intensive tasks has led to the rapid adoption of Retrieval-Augmented Generation (RAG) as a strategy for improving factual grounding and domain adaptability. This review traces the evolution of RAG systems, from their roots in Information Retrieval (IR) and early Natural Language Processing (NLP) to current modular architectures that support dynamic reasoning and real-time knowledge integration. It categorizes key frameworks according to the specific challenges they target – such as retrieval precision, hallucination reduction, domain specialization, and interpretability – and analyzes how each addresses recurring failure modes in real-world applications. Through a comparative lens, the paper highlights both the fragmented nature of current solutions and the need for more unified, self-aware designs. Evaluation frameworks, including RAGAS, RGB, and PaSSER, are discussed in light of these gaps. Based on this analysis, the review outlines core directions for future research, emphasizing the importance of real-time retrieval validation, sentence-level attribution, failure correction mechanisms, and adaptable query rewriting. The findings suggest that RAG research is entering a phase of consolidation, where system reliability, transparency, and domain robustness will define progress more than generative fluency alone.

Keywords: *Artificial Intelligence, Information Retrieval, Large Language Models, Natural Language Processing, Retrieval-Augmented Generation*

1. Introduction

The landscape of software engineering and AI is experiencing unprecedented changes driven by the rapid progress in LLMs. These models, powered by deep learning architectures – particularly transformer-based neural networks – have

revolutionized various NLP tasks, including machine translation, content generation, and conversational systems. Their extraordinary ability to understand, generate, and contextualize human language has significantly impacted fields such as healthcare, finance, education, and technology.

Nevertheless, traditional LLMs struggle with static knowledge, high computational demands, and limited adaptability to evolving information. To overcome these limitations, RAG has appeared as a method that dynamically retrieves external knowledge before generating responses, ensuring greater contextual relevance.

RAG systems enhance the capabilities of traditional LLMs by softening the risks of knowledge hallucinations and inaccuracies, offering improved transparency and traceability of generated content. Although these systems have demonstrated remarkable potential, they also introduce novel complexities regarding retrieval precision, heterogeneous data integration, and computational overhead.

In addition to tracing key milestones that led to the rise of RAG, this review employs a comparative approach, examining the capabilities and limitations of representative RAG frameworks across diverse use cases. By highlighting influential architectures, exploring prevailing challenges, and evaluating existing approaches through practical lenses, the discussion provides a clearer picture of RAG's evolution and real-world performance. Furthermore, by outlining theoretical foundations and practical implications, this review aims to guide future research directions, fostering improvements that support more efficient, transparent, and reliable knowledge-driven applications.

2. Beginning: First steps

The foundations of RAG are deeply rooted in historical developments within IR and NLP. To better trace how these innovations converged, it is helpful to group the seminal work and technical breakthroughs into four categories: basic indexing and data organization, formal IR evaluation and early IR–NLP fusion, advanced semantic retrieval and NLP methods, large-scale IR–NLP integration and modern embedding-based developments. Table 1 summarizes these categories of innovation and their impact on the evolution of RAG systems.

The first set of advancements focused on fundamental methods for indexing and organizing textual data, thereby laying a foundation for large-scale retrieval. Memex [1], pioneered the notion of linking information through associative trails, effectively prefiguring modern hypertext navigation by allowing users to create and traverse context-rich connections among documents. Around the same time, statistical text analysis [2] advanced beyond manual indexing by applying word-frequency counts and distributional patterns to large textual corpora, systematically measuring how terms co-occur or spread across documents.

Table 1. Key Technological Milestones Contributing to the Development of RAG

Category	Key Advancements	Contribution to RAG Development
Basic Indexing and Data Organization	Memex, Statistical Text Analysis, KWIC Indexing	Laid the groundwork for efficient large-scale retrieval of information, a fundamental requirement for RAG systems to access and utilize external knowledge.
Formal IR Evaluation and Early IR–NLP Fusion	Cranfield Studies, BASEBALL System	Established methods for evaluating retrieval effectiveness and demonstrated early attempts to use natural language for querying, paving the way for more sophisticated query handling and evaluation in RAG.
Advanced Semantic Retrieval and NLP Methods	TF-IDF, Vector Space Models, Word2Vec	Enabled retrieval based on semantic similarity rather than just keywords, and provided methods for understanding the meaning of text, which are crucial for selecting relevant context to augment generation in RAG.
Large-Scale IR–NLP Integration and Modern Embeddings	IBM Watson, Transformers, Dense Passage Retrieval, GPT Series	Showcased the integration of retrieval and generation at scale, and provided the powerful embedding techniques and fluent generative models that are the core components of modern RAG systems.

This data-driven approach enabled more objective methods of categorizing and retrieving information, as it drew on quantifiable evidence rather than subjective classifications. Another significant innovation, Key Word in Context (KWIC) indexing [3], automated the extraction of localized snippets around each keyword, expediting relevance assessments with concise, context-rich summaries. Meanwhile, the WRU Searching Selector [4] devised by Kent and Rees employed rods or channels to physically filter and retrieve documents based on selected attributes, while Mooers's edge-notched cards [5] allowed users to "notch" card edges to indicate indexing categories and quickly isolate relevant items. By introducing systematic, semi-automated approaches to identifying pertinent data, these early developments laid a solid groundwork for the large-scale and integrated retrieval techniques now fundamental to retrieval-augmented generation.

A second cluster of advancements revolved around formalizing evaluation metrics and incorporating elementary language-processing techniques into IR. One notable example was the BASEBALL system [6], which used basic parsing rules and pattern matching to interpret domain-specific queries, thus illustrating how natural language input could successfully drive search and retrieval. Meanwhile, Cyril Cleverdon's Cranfield studies [7] solidified the importance of

standardized performance metrics – namely precision and recall – transforming retrieval from an ad hoc process into a more scientific discipline. By quantifying how effectively a system retrieved relevant documents while filtering out irrelevant ones, these metrics became indispensable benchmarks for evaluating IR approaches. Linking data retrieval with simplified natural language processing and systematically measuring retrieval quality ultimately laid essential groundwork for future architectures that merge retrieval with generative capabilities.

Third group of advancements revolved around deepening the semantic understanding of queries and documents, enabling more flexible and context-aware retrieval strategies. Between the 1970s and 1990s, IR advanced well beyond basic Boolean methods, placing renewed emphasis on how linguistic nuances affect both user queries and underlying content. Winograd's natural language understanding systems [8] introduced context-aware parsing, leveraging syntactic and situational cues—such as grammatical relationships and anaphoric references—to interpret queries more accurately. Around the same time, Spärck Jones's Term Frequency–Inverse Document Frequency (TF-IDF) algorithm [9] applied statistical weighting, giving terms that appear frequently in a single document but rarely throughout an entire corpus greater significance. This approach guided retrieval engines toward more discriminative keywords. In parallel, domain-specific frameworks like LUNAR [10] highlighted how rule-based NLP methods could employ explicit grammar rules and specialized lexicons to handle queries in specialized fields, such as geological data. Building on these insights, vector space models [11] placed both queries and documents in a high-dimensional space, where each dimension corresponded to a term or feature, thereby enabling similarity measures (e.g., cosine similarity) that surpassed simple keyword overlap. Although these methods led to more conversational interfaces by the late 1980s and 1990s, scaling them across diverse domains remained challenging, ultimately motivating the flexible retrieval-generation paradigms that define modern RAG systems.

Finally, fourth category dealt with large-scale systems that integrated or further refined these earlier ideas, leading to the emergence of prototypes that blended retrieval and generation in a manner akin to modern RAG systems. An early exemplar was IBM Watson [12], whose DeepQA pipeline decomposed complex queries, retrieved evidence from both structured and unstructured data, generated possible answers, and then ranked these answers based on confidence scores.

Fig. 1, follows a multi-step pipeline consisting of query decomposition, hypothesis generation, supporting evidence retrieval, and final answer ranking.

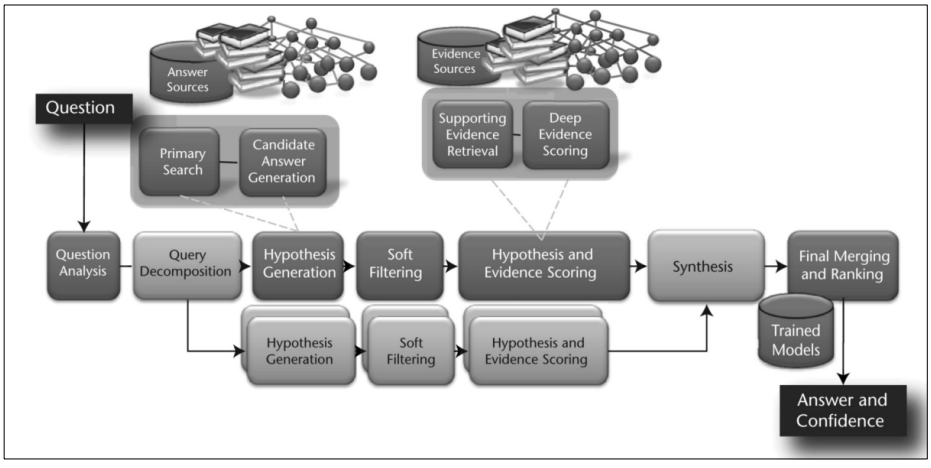

Fig. 1. DeepQA High-Level Architecture, illustrating Watson's pipeline for evidence retrieval and answer generation. Reprinted from [12]

Despite its successful performance on the quiz show Jeopardy!, Watson encountered scalability and real-time integration challenges, prompting further exploration into more modular retrieval-generation designs.

Subsequent advancements included Word2Vec [13], which produced continuous vector embeddings to represent semantic relationships between words. This technique moved beyond mere keyword overlap by evaluating conceptual similarities among terms. Building on such embedding methods, transformer architectures [14] introduced an attention-based mechanism for parallelizing the processing of sequences, allowing all tokens within a sentence to be analyzed in tandem. This not only sped up training compared to purely sequential models, but also enhanced the ability to capture intricate context and dependencies.

For retrieval tasks, Dense Passage Retrieval (DPR) [15] expanded on the embedding idea by placing queries and documents in a shared high-dimensional space, thereby matching textual content based on semantic affinity rather than strict keyword alignment. On the generative side, models like the GPT series [16] achieved unprecedented fluency in synthesizing text by leveraging large-scale transformer architectures trained on extensive text corpora. Specifically, these models learned language patterns from billions of tokens spanning diverse domains, so that each token in a sequence could be generated with an awareness of the surrounding context.

Taken together, the above categories illustrate the evolution of retrieval and language understanding techniques, from rigid keyword matching to semantically rich, context-aware systems. This progression laid the theoretical and technical foundation for RAG. Building directly on decades of IR and NLP innovation, RAG emerged as a practical response to the limitations of traditional language models – especially their inability to access up-to-date or verifiable information.

3. RAG

The RAG framework was **formally introduced in 2020** by Facebook AI Research (now Meta AI) in the foundational study Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [17]. It integrates information retrieval directly into the generative process, addressing key limitations of traditional LLMs, such as reliance on static training data, lack of source attribution, and a tendency to produce fluent but inaccurate outputs – commonly known as hallucinations.

Unlike conventional models that encode all knowledge within their parameters, RAG separates retrieval and generation. At inference time, it dynamically retrieves relevant external documents – typically from Wikipedia – which condition the response. This approach improves factual accuracy, enables transparency, and supports domain adaptation without retraining.

RAG consists of two main components: a retriever and a generator. The retriever uses DPR, a neural method that converts both queries and documents into high-dimensional vectors (dense embeddings) using transformer encoders based on the Bidirectional Encoder Representations from Transformers (BERT) architecture. BERT processes text bidirectionally—capturing context from both the left and right of each word—enabling rich semantic representations ideal for similarity-based retrieval.

Relevant documents are identified via Maximum Inner Product Search (MIPS), a technique that ranks documents by computing the inner product between their embeddings and the query vector. To maintain efficiency at scale, MIPS is typically implemented using approximate nearest neighbor (ANN) algorithms.

The generator component is based on the Bidirectional and Auto-Regressive Transformers (BART) architecture, which combines a bidirectional encoder and an auto-regressive decoder. This hybrid design captures deep contextual relationships while generating fluent responses token-by-token. BART operates within a sequence-to-sequence (seq2seq) framework—originally developed for machine translation—which transforms an input sequence (query + retrieved passages) into a coherent, evidence-based output. This design enables RAG to synthesize responses that are both contextually grounded and factually verifiable.

Fig. 2 provides a high-level overview of the RAG architecture, illustrating the integration of dense retrieval and generative language modeling within an end-to-end framework. The retriever module, denoted as P_{n^1}, encodes the input query into a dense vector representation and retrieves the most semantically relevant documents from an external knowledge base using MIPS. This retrieval process, grounded in vector similarity rather than keyword matching, facilitates context-aware document selection.

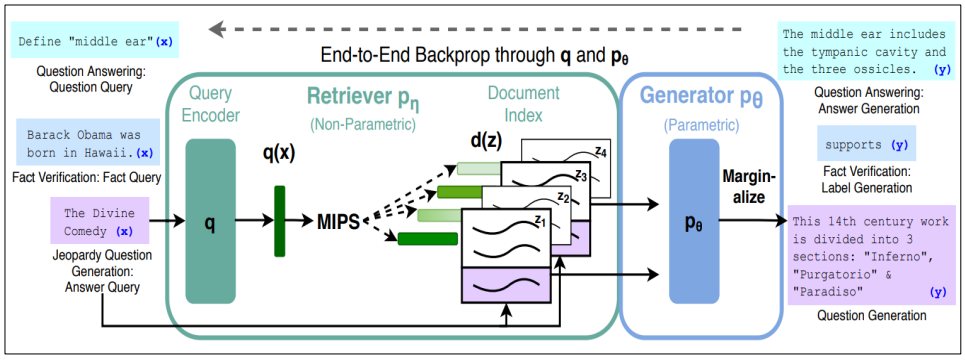

Fig. 2. Overview of the RAG Architecture, illustrating the integration of dense retrieval and generative language modeling. Reprinted from [17].

The retrieved documents are subsequently passed to the generator module, denoted as P_θ, which synthesizes a coherent and contextually grounded response by incorporating information from multiple sources. This multi-passage integration step reduces dependence on any single document, thereby improving factual consistency and resilience to retrieval noise. Additionally, the figure demonstrates RAG's flexibility across a range of downstream tasks, including question answering, fact verification, and open-ended generation, underscoring its capacity to produce well-grounded outputs in knowledge-intensive applications.

By embedding retrieval directly into the generation process, RAG helped reduce hallucinations and improved output traceability. Its use of external corpora also enabled seamless updates—new knowledge could be incorporated by refreshing the retrieval index, without retraining the model.

However, RAG's reliance on sources like Wikipedia also introduced risks of bias or misinformation. These concerns underscore the ongoing need for more selective and reliable retrieval strategies. Still, RAG laid the groundwork for a new generation of retrieval-aware language models.

4. Enhancing RAG: Key Innovations and Their Targets

Following its initial release, RAG underwent a series of substantial enhancements aimed at improving retrieval precision, generative accuracy, interpretability, and computational efficiency. As the field matured, research shifted from foundational development to more focused improvements, each addressing a specific limitation of the original framework. These improvements span a wide range of goals, from architectural efficiency to grounding factuality.

Table 2 presents a functional overview of the advancements, organized by the core challenges they were intended to solve.

Table 2. Functional Categorization of Recent RAG Advancements

Focus Area	Objective
Architectural Efficiency and Scalability	Reduce computational cost, improve inference speed, and support real-time applications
Data-Centric Optimization	Improve training quality through noise reduction, sampling, and data selection techniques
Knowledge Integration *(Structured and Unstructured Sources)*	Combine symbolic and neural retrieval to access diverse knowledge representations
Domain Adaptation and Specialization	Enable RAG systems to perform well in specialized domains or narrow knowledge fields
Iterative Retrieval and Self-Refinement	Introduce mechanisms for multi-step reasoning, response revision, or feedback-based retrieval
Multimodal Extension	Expand RAG to operate across multiple input modalities, such as text and images
Few-Shot and Low-Resource Enhancement	Improve generalization with limited training data through retrieval-enhanced few-shot learning
Factual Verification and Grounding	Reduce hallucinations and improve transparency by anchoring outputs in verifiable sources

4.1. Architectural Efficiency and Scalability

One of the earliest and most impactful enhancements to the RAG architecture was the introduction of FiD-Light [18], a computationally efficient variant of the Fusion-in-Decoder (FiD) model. In the original FiD architecture, multiple retrieved documents are concatenated and passed jointly through a decoder, enabling the model to synthesize responses that integrate information from diverse sources. While effective in terms of output quality, this design introduces considerable computational overhead, as the decoder must attend uniformly to all tokens across all retrieved passages.

FiD-Light addresses this bottleneck through two key innovations: selective attention mechanisms and source pointer strategies. The selective attention mechanism allows the model to dynamically filter and prioritize only the most relevant segments from the retrieved passages. This focused attention reduces unnecessary computation and minimizes the influence of irrelevant or noisy content, thereby improving efficiency without sacrificing performance. In parallel, the source pointer strategy tracks the provenance of information used during generation, preserving alignment between generated content and its evidence source. This enhances interpretability and helps mitigate hallucination, especially in fact-sensitive applications.

As illustrated in Fig. 3, FiD-Light demonstrated state-of-the-art performance on the Knowledge-Intensive Language Tasks (KILT) benchmark [19], significantly outperforming prior RAG models in retrieval precision, response fluency, and computational efficiency. These results marked a step toward making retrieval-augmented generation suitable for real-time and production-scale deployment.

	Model	Open Domain QA			Fact	Slot Filling		Dialog
		NQ	HotpotQA	TriviaQA	FEVER	T-REx	zsRE	WOW
		KILT-EM	*KILT-EM*	*KILT-EM*	*KILT-AC*	*KILT-AC*	*KILT-AC*	*KILT-F1*
Top Leaderboard Entries								
1	RAG (Petroni et al., 2021)	32.7	3.2	38.1	53.5	23.1	36.8	8.8
2	DPR + FiD (Piktus et al., 2021)	35.3	11.7	45.6	65.7	64.6	67.2	7.6
3	KGI (Glass et al., 2021)	36.4	–	42.9	64.4	69.1	72.3	11.8
4	Re2G (Glass et al., 2022)	43.6	–	57.9	78.5	75.8	–	12.9
5	Hindsight (Paranjape et al., 2021)	–	–	–	–	–	–	**13.4**
7	SEAL + FiD (Bevilacqua et al., 2022)	38.8	18.1	50.6	71.3	60.1	73.2	11.6
Ours								
8	FiD-Light^{SP} (T5-Base, $k = 64$)	45.6	25.6	57.6	80.6	76.0	81.1	11.9
9	FiD-Light^{SP} (T5-Large, $k = 32$)	49.9	28.2	61.4	82.1	**76.7**	**84.1**	12.2
10	FiD-Light^{SP} (T5-XL, $k = 8$)	**51.1**	**29.2**	**63.7**	**84.5**	76.3	84.0	13.1

Fig. 3. Performance evaluation of FiD-Light across the Knowledge-Intensive Language Tasks (KILT) benchmark, demonstrating improved accuracy and efficiency. Reprinted from [18].

To support more structured and context-aware retrieval, LightRAG [20] introduced a graph-based architecture that organizes knowledge into entity-relationship graphs, moving beyond flat document representations. This structure encodes semantic relationships between entities, enabling the retriever to preserve logical flow and improve coherence across documents. Unlike standard dense retrievers that rely solely on vector similarity, LightRAG uses graph topology to guide retrieval toward semantically connected content – especially effective in multi-hop reasoning tasks requiring evidence from multiple sources. Evaluations on different datasets showed notable gains in retrieval recall, answer accuracy, and factual grounding. Its design also supports dynamic knowledge expansion, making it suitable for real-time applications with frequently updated corpora.

Extending this line of innovation, Auto-RAG [21] introduces a self-optimizing architecture that autonomously adjusts its internal retrieval and generation components without requiring manual intervention. Traditional RAG implementations often depend on fixed retrieval-generation pipelines or require labor-intensive fine-tuning to adapt to new domains or data conditions. In contrast, Auto-RAG is designed to monitor its own performance during inference and make real-time adjustments to its operational strategy. This is achieved through self-supervised feedback loops, in which the system evaluates the quality of its generated outputs – based on internal consistency checks, retrieval relevance

scores, or proxy supervision signals – and uses this feedback to iteratively refine both retrieval selection and generative behavior.

This capacity for real-time self-adjustment makes Auto-RAG particularly valuable in high-stakes or constantly evolving environments, such as customer support systems, technical documentation assistants, and large-scale knowledge access platforms. In these contexts, models must handle unpredictable user input and rapidly changing information without the latency or overhead of human-guided retraining.

While architectural innovations such as FiD-Light, LightRAG, and Auto-RAG significantly enhanced the efficiency, scalability, and modularity of RAG systems, improvements in model architecture alone are not sufficient to ensure robust performance across diverse tasks and domains. As RAG systems rely heavily on large and often heterogeneous datasets, the quality of the training data becomes a critical determinant of retrieval relevance and generation accuracy. Consequently, the efforts have turned toward data-centric strategies aimed at minimizing noise, enhancing supervision signals, and improving generalization without the need for extensive manual tuning.

4.2. Data-Centric Optimization

Unlike architecture-driven solutions that focus on design improvements or parameter tuning, data-centric methods aim to enhance how models learn by selecting, filtering, and weighting training examples to reduce noise – especially in multi-task and low-resource scenarios where inconsistent data hampers generalization.

A key contribution here is Relevance Sampling [22], which addresses training quality by assigning confidence scores to examples based on model uncertainty, retrieval relevance, or internal consistency. Low-confidence examples are excluded, allowing the model to learn from cleaner, more informative data. This approach is task-agnostic, applicable across domains, and has been shown to improve generalization, reduce hallucination, and stabilize convergence in knowledge-intensive tasks.

Complementing this, Speculative RAG [23] introduces a two-stage generation strategy: an initial speculative draft is produced, then refined through retrieval-informed verification. The second phase re-checks evidence and corrects or strengthens claims, inspired by human reasoning. This process filters out unsupported content and improves factual grounding and fluency.

Together, these techniques reflect a shift toward self-aware learning, where models not only train on higher-quality data but also refine their outputs dynamically. They mark a broader trend in data responsibility, acknowledging that robust reasoning requires disciplined interaction with data – not just better architectures.

While these strategies improve how RAG systems learn, another emerging priority is broadening what they can learn from. Early RAG systems mostly used unstructured text (e.g., Wikipedia), but real-world applications increasingly demand access to heterogeneous sources – like knowledge graphs, relational databases, and metadata-enriched corpora. The following section explores hybrid retrieval methods that combine neural and symbolic techniques.

Evaluations of these hybrid strategies demonstrate improved retrieval accuracy and reduced hallucinations, particularly when structured sources such as knowledge graphs are integrated with unstructured text [24].

4.3. Knowledge Integration (Structured and Unstructured Sources)

A key challenge emerged as RAG systems evolved: integrating structured knowledge sources – such as knowledge graphs, relational databases, and ontologies – alongside traditional unstructured text. While free-text corpora like Wikipedia are common in early RAG setups, many applications demand information organized by entities, attributes, and relationships, which offer greater precision and verifiability.

GraphRAG [25] is a leading approach to structured knowledge integration in RAG systems. As shown in Figure 5, it converts unstructured documents into entity-relationship graphs using LLM-based entity recognition and relationship extraction during indexing. This organizes content into semantically coherent clusters, improving interpretability and retrieval precision.

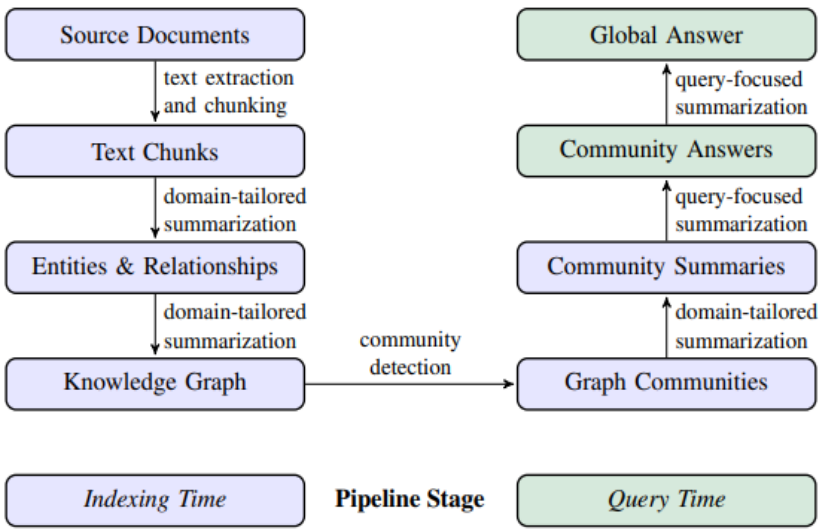

Fig. 5. GraphRAG pipeline illustrating the transformation from source documents into structured knowledge graphs, applying community detection and query-focused summarization. Reprinted from [25].

To refine structure, community detection algorithms segment the graph into subgraphs that capture domain-specific relationships. At inference time, these subgraphs act as targeted retrieval units, enabling query-focused summarization and domain-aware responses. By parallelizing retrieval and summarization during both indexing and inference, GraphRAG enhances efficiency and relevance.

A major advantage of this architecture lies in its separation of knowledge structuring from retrieval execution, allowing for greater modularity, scalability, and specialization. GraphRAG performs intermediate structuring and refinement, leading to more coherent, contextually grounded, and non-redundant outputs. This architectural distinction reflects the broader principles of Modular RAG, enabling more interpretable and adaptable systems – particularly useful in complex, evolving knowledge environments that demand high factual precision and contextual depth.

In addition to graph-based retrieval, researchers have developed hybrid approaches [26] that combine dense vector search with symbolic querying and structured data. These systems aim to blend the semantic flexibility of neural retrieval with the precision and traceability of structured queries. Common techniques include SPARQL-style queries – structured formats similar to the SPARQL protocol [27] used for querying knowledge graphs – along with metadata tagging, schema alignment, and post-retrieval reranking. These methods incorporate entity types, ontological links, and domain filters to improve retrieval relevance and reduce noise.

By fusing symbolic and neural retrieval, RAG systems can pull from both unstructured text and structured knowledge bases, producing more grounded and interpretable outputs. This hybridization is especially beneficial in domains like biomedicine, finance, and law, where information is complex and factual accuracy is paramount.

However, hybrid strategies alone may not suffice for domain-specialized tasks. Real-world applications also demand systems that adapt to specific terminologies, citation styles, and data structures. To meet these needs, recent RAG architectures now incorporate custom pipelines, modular design, and iterative refinement to better serve specialized contexts.

4.4. Domain Adaptation and Specialization

While general-purpose RAG systems perform well on open-domain benchmarks, they often struggle when applied to specialized domains such as biomedical research, legal analysis, or technical documentation.

One of the most impactful frameworks designed to support domain adaptation in RAG is RaLLe [28], a modular and extensible research environment that facilitates systematic evaluation and fine-tuning of RAG pipelines. RaLLe allows researchers to interchange retrieval components, without requiring

architectural overhaul, enabling efficient experimentation across tasks and domains. Its modular design supports custom ranking mechanisms, query expansion strategies, reranking models, and fusion techniques, making it well-suited for tailoring RAG systems to the nuanced requirements of specialized fields.

A key feature of RaLLe is its integrated benchmarking suite, offering preconfigured workflows for datasets like *Natural Questions*, *TriviaQA*, and *HotpotQA*. This enables direct comparison of retrieval-generation strategies. RaLLe also supports diverse knowledge sources – structured databases, unstructured corpora, and multimodal inputs – making it suitable for domains such as biomedicine, law, and finance. Its domain-specific configurations allow for fine-tuning of retrieval depth, query reformulation, and reranking logic, optimizing performance for multi-hop, long-form, and fact-sensitive tasks. With standardized metrics like retrieval recall, relevance precision, and factual grounding, RaLLe facilitates rigorous and reproducible evaluation. As an open-source framework, it also promotes collaborative development and real-world RAG deployment.

Similarly, PaperQA [29], is a domain-focused RAG system tailored for scientific literature synthesis. PaperQA ensures all answers are explicitly grounded in scientific references, which is crucial for citation accuracy and evidence traceability in academic settings. Its pipeline follows three steps: search via academic APIs and structured repositories, evidence gathering using Maximum Marginal Relevance (MMR) and vector retrieval, answer generation with citation-aligned synthesis.

This process delivers both fluency and verifiability, making PaperQA highly effective in research-intensive applications. As illustrated in Figure 6, PaperQA's architecture includes dynamic evidence refinement based on LLM-driven relevance scoring. Retrieved document chunks are scored and re-ranked iteratively, and only the most reliable are passed to the generation module. This approach reduces misinformation and reinforces citation accuracy.

Empirical results on biomedical benchmarks such as PubMedQA and LitQA confirm PaperQA's superior performance in handling domain-specific queries. Its success underscores the growing importance of targeted RAG pipelines in fields that demand factual rigor and domain expertise.

Beyond academia, similar domain-adaptive RAG systems are gaining traction in enterprise knowledge management. For example, legal and financial institutions increasingly rely on hybrid retrieval pipelines that combine structured documents – such as earnings reports or legal filings – with unstructured commentary, enabling more informed decision-making, compliance monitoring, and risk assessment.

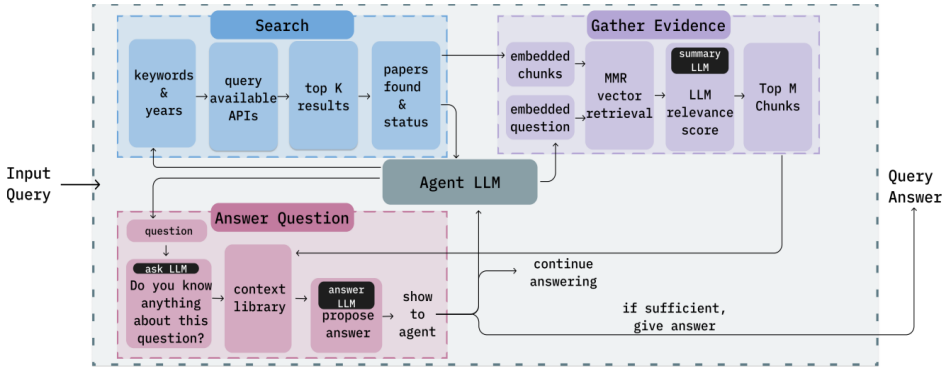

Fig. 6. Workflow diagram illustrating the PaperQA retrieval-augmented generation process tailored specifically for scientific literature. Reprinted from [28].

4.5. Iterative Retrieval and Self-Refinement

While domain-specific customization improves relevance and precision, many knowledge-intensive tasks require not just better retrieval – but retrieval that evolves during the generation process. Static, one-shot retrieval pipelines often fall short when handling multi-hop questions, ambiguous prompts, or incomplete evidence. To overcome these limitations, recent frameworks have introduced iterative and self-reflective mechanisms that allow RAG systems to re-evaluate, revise, and refine their outputs dynamically. These systems go beyond fixed pipelines by integrating retrieval directly into the inference loop, enabling stepwise reasoning, uncertainty detection, and response revision based on intermediate feedback. The following models illustrate how this class of retrieval-aware architectures improves coherence, factual grounding, and adaptability in complex question-answering scenarios.

One of the most intuitive implementations of this paradigm is Self-RAG [30], which incorporates a self-reflective retrieval loop into the generation process. Rather than relying solely on its initial output, the model critiques its own response and initiates additional retrieval when it identifies potential factual inconsistencies. This retrieval-based self-correction enables the system to iteratively refine its output, improving alignment with external evidence. Particularly in open-domain and zero-shot contexts, Self-RAG significantly reduces hallucination and enhances the reliability of generated content by embedding a mechanism of *self-assessment* within the RAG architecture.

While Self-RAG emphasizes introspective critique, FLARE [31] builds on this idea by introducing a more dynamic, confidence-driven mechanism that continuously reassesses the generation in real time – bridging reasoning with retrieval at every step. Instead of treating all generated content equally, FLARE identifies segments with low confidence scores – determined through internal uncertainty estimation – and triggers additional retrieval operations to supply

more relevant context. As illustrated in Figure 7, this feedback-driven mechanism issues new queries (e.g., q_2, q_3) whenever gaps in confidence are detected, ensuring that evidence acquisition is tightly coupled to the unfolding generation process.

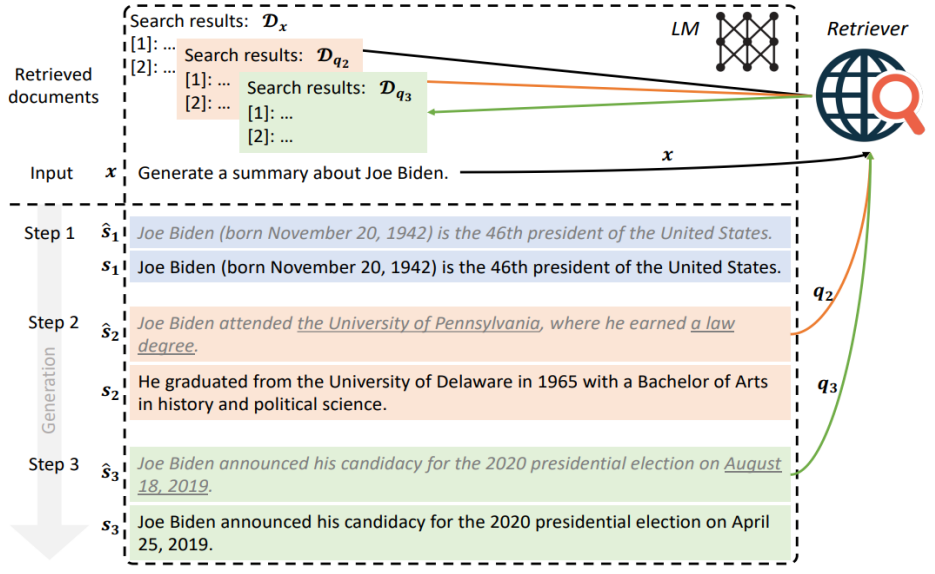

Fig. 7. An illustration of the FLARE iterative retrieval-generation process, highlighting its active retrieval and iterative refinement mechanism. Reprinted from [30].

FLARE's architecture is particularly effective for multi-hop reasoning, legal and financial analysis, and any task where contextual precision is paramount. By decoupling retrieval from a single pre-generation phase and embedding it within the generative loop, FLARE reduces hallucinations and improves answer consistency.

FLARE thus emphasizes retrieval as an active, iterative process driven by uncertainty. RAVEN [29] complements this by integrating retrieval even more deeply – embedding it within the attention mechanisms of the encoder-decoder itself, enabling simultaneous reasoning over both prompts and retrieved knowledge. The key innovation lies in its dual attention mechanism, which allows the encoder-decoder model to simultaneously attend to the original query prompt and the retrieved evidence. This fusion prevents context fragmentation and ensures that all parts of the output are informed by both task intent and external knowledge. RAVEN's iterative retrieval during inference enables multi-step reasoning, making it particularly well-suited for complex tasks that require maintaining dependencies across multiple retrieved facts.

Together, these models represent a shift toward adaptive and reflective retrieval paradigm, where the RAG system becomes an active participant in its

own reasoning cycle – constantly revisiting its evidence, questioning its outputs, and refining its answers in real time.

4.6. Multimodal Extension

While text-based retrieval-augmented systems have advanced considerably, real-world tasks also require integrating multiple modalities—such as images, charts, or diagrams—alongside textual data. This is particularly relevant in domains like education, technical documentation, and medical imaging, where visual information is central to understanding. To address this, recent frameworks have extended RAG architectures to support multimodal retrieval and generation, allowing models to synthesize knowledge from both textual and non-textual sources.

A notable contribution in this direction is MuRAG [31], which introduces a dual-retriever framework that processes and fuses textual and visual evidence. MuRAG includes two parallel retrieval pathways: a text retriever that searches through traditional corpora, and a vision retriever that locates relevant images or visual artifacts from associated databases. These retrieved inputs are then jointly embedded and passed to a multimodal generation module that produces a response grounded in both linguistic and visual evidence.

This multimodal integration significantly enhances the model's capacity to answer complex questions that depend on visual context, such as interpreting charts, annotating diagrams, or correlating textual claims with accompanying illustrations. By aligning representations across modalities, MuRAG enables more informed and complete responses than would be possible through unimodal systems. Furthermore, its design supports cross-modal reasoning, allowing the model to reconcile and synthesize information that may be partially encoded in each modality.

Empirical evaluations demonstrate that MuRAG improves performance on multimodal benchmarks by a substantial margin, particularly in tasks involving diagram understanding, text-image alignment, and cross-modal retrieval-based question answering. Its architecture represents a major step toward more general-purpose, perception-aware RAG systems capable of engaging with the full range of content encountered in complex, real-world applications.

4.7. Few-Shot and Low-Resource Enhancement

While LLMs demonstrate strong performance across a range of NLP tasks, they typically rely on extensive labeled datasets for fine-tuning. However, in many real-world scenarios—such as specialized scientific domains, under-resourced languages, or rapidly evolving topics – such data may be limited or entirely unavailable. These situations are referred to as low-resource settings, where

training data is insufficient to support conventional supervised learning approaches.

To address this challenge, researchers have turned to few-shot learning, where models are expected to perform tasks after seeing only a few examples (typically fewer than ten). A leading approach in this space is Atlas [32], a framework designed to enhance few-shot performance by avoiding task-specific fine-tuning. Instead, Atlas uses dense retrieval to dynamically gather relevant information from large external corpora. Built on the Fusion-in-Decoder (FiD) architecture, it combines multiple retrieved passages during generation to produce more informed and grounded responses.

By leveraging context-specific evidence instead of static memorization, Atlas enhances factual consistency and adaptability—even in constrained settings. As shown in Figure 8, it supports iterative refinement across tasks through retrieval-grounded outputs.

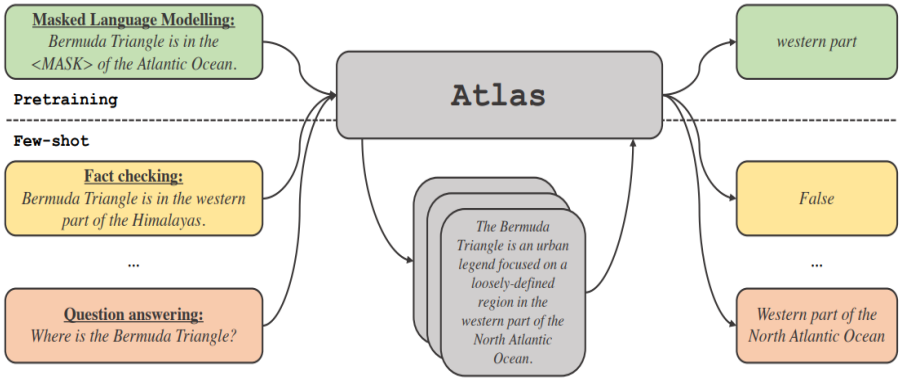

Fig. 8. Workflow diagram illustrating the Atlas framework's retrieval-augmented few-shot learning approach. Reprinted from [32].

Atlas operates across a wide range of NLP tasks – including masked language modeling, fact verification, and question answering – and excels in zero-shot or low-supervision contexts. During inference, Atlas issues retrieval queries for each prompt, processes the retrieved evidence in parallel, and fuses the content into a coherent output. This retrieval-enhanced generation pipeline acts as a proxy for missing training examples, allowing the model to perform tasks it has seen few or no direct examples of.

4.8. Factual Verification and Grounding

One persistent challenge for RAG systems remains factual consistency – ensuring that generated outputs are traceable to verifiable sources and free from hallucinated or unsupported claims. This issue becomes particularly critical in high-stakes domains such as scientific research, journalism, medicine, and law,

where output provenance and citation fidelity are essential. To address these challenges, recent innovations have focused on improving the alignment between generated content and external evidence, introducing mechanisms for explicit grounding, citation-level attribution, and post-generation verification.

A notable example is ReClaim [33] , a retrieval-augmented framework designed to enforce sentence-level attribution in generated responses. Traditionally RAG models cite entire documents or broad passages, ReClaim ensures that each factual statement in the output is explicitly linked to a supporting source sentence. This granularity enhances both transparency and trust, allowing users to verify specific claims directly. By constraining generation to content grounded in verifiable evidence, ReClaim significantly reduces hallucination and improves factual reliability. Empirical evaluations have shown that ReClaim outperforms baseline RAG models in source attribution accuracy and is particularly well suited for domains where traceability is non-negotiable.
ReClaim exemplifies a movement toward trustworthy and verifiable generation, where the credibility of outputs is enhanced not just through better retrieval, but through explicit, structured connections between claims and sources. These systems reinforce the need for evaluation frameworks that go beyond accuracy and fluency, incorporating measures of evidence alignment, citation fidelity, and user interpretability.

All these innovations reveal the growing sophistication and specialization of Retrieval-Augmented Generation. From architectural streamlining and data curation to domain-specific optimization and factual verification, RAG research has moved beyond foundational breakthroughs toward highly modular, task-aware, and adaptable architectures. However, the very complexity and diversity of these systems now demand standardized, transparent, and multi-dimensional evaluation frameworks. As RAG systems become increasingly embedded in sensitive, high-stakes domains, rigorous assessment of retrieval quality, response validity, and knowledge integration is not merely beneficial – it is essential.

5. Evaluating RAG

RAG systems evolve in complexity and scope and evaluating their performance requires more than simple accuracy metrics or surface-level comparisons. Effective assessment must capture not only how well a model retrieves and generates, but also how coherently it integrates external knowledge and grounds its outputs in verifiable sources.

One of the prominent contributions in this space is RAGAS [34], which standardizes and automates the evaluation of RAG models. RAGAS introduces a structured multi-dimensional evaluation framework to assess retrieval precision, generation fluency, and the integration of retrieved knowledge. It automates scoring mechanisms using neural models trained on annotated datasets, enabling

large-scale performance comparison across different tasks. Empirical results highlight common weaknesses in RAG models, including retrieval failures, improper knowledge integration, and hallucination risks. By offering a unified benchmark, RAGAS provides a foundation for optimizing retrieval-augmented architectures across diverse applications.

Complementing this, the Retrieval-Augmented Generation Benchmark (RGB) [35] evaluates RAG models under four conditions: noise robustness, negative rejection, information integration, and counterfactual robustness. Notably, findings reveal that models struggle most with negative rejection, often generating responses despite insufficient evidence. These insights highlight the need for improved fact-verification mechanisms.

In addition to general benchmarking frameworks, domain-specific evaluations are essential for assessing RAG performance in practical applications. The study Web Application for Retrieval-Augmented Generation: Implementation and Testing [36] presents PaSSER, a RAG-based web application designed as a fully functional implementation rather than a standalone benchmarking tool. Unlike traditional evaluations that assess pre-trained models in isolated test environments, PaSSER integrates the entire RAG pipeline—retrieval, generation, evaluation, and blockchain-based verification—offering a real-world framework for performance assessment. This distinction enables direct, reproducible testing of RAG models in dynamic settings, ensuring practical applicability across different domains.

A key feature of PaSSER is its modular architecture, which facilitates scalable retrieval and response generation while ensuring transparent evaluation through blockchain integration. The system is built using LangChain and ChromaDB for efficient retrieval, while its front-end leverages PrimeReact and WharfKit for interactive, real-time user engagement. Fig. 9 illustrates the PaSSER architecture, detailing its local LLM deployment, web-based interaction layer, and blockchain-backed data verification.

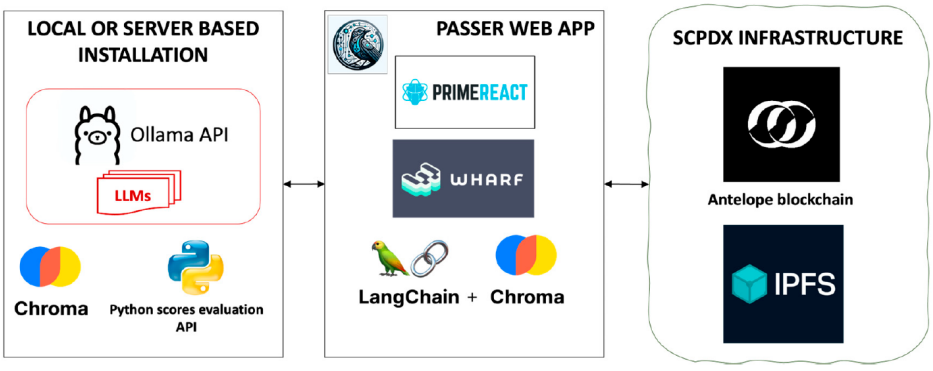

Fig. 9. Illustration of the PaSSER architecture. Reprinted from [36].

Building on this work, the Similarity Thresholds in RAG study [37] further explores the impact of similarity threshold optimization on retrieval performance and response quality. A systematic evaluation of three open-source LLMs (Mistral:7b, Llama2:7b, and Orca2:7b) investigates how fine-tuning similarity thresholds affects precision-recall trade-offs and composite model performance. As illustrated in Fig. 10, higher similarity thresholds improve factual accuracy by filtering out less relevant retrieved documents but risk excluding useful contextual information. Conversely, lower thresholds enhance recall, retrieving a broader set of passages at the expense of increased irrelevant data.

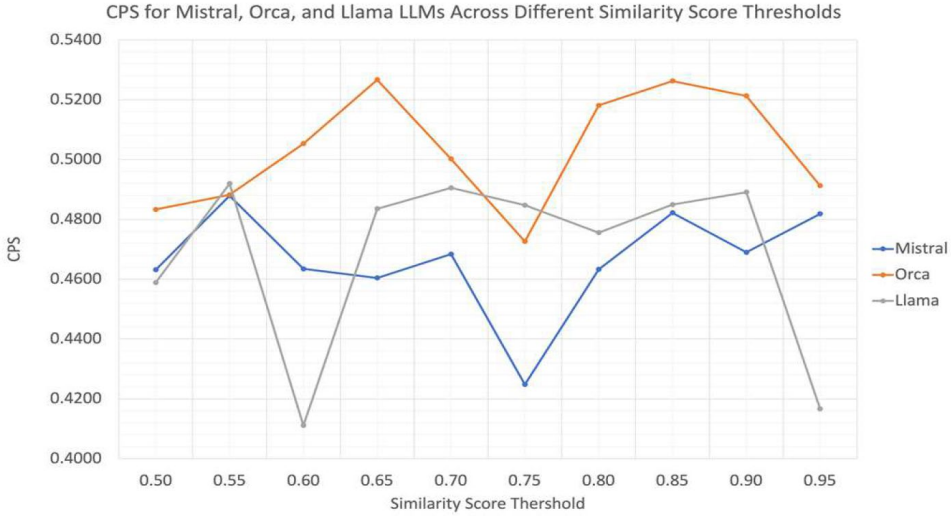

Fig. 10. Composite Performance Score (CPS) across different similarity thresholds for Mistral, Orca, and Llama LLMs, illustrating the impact of threshold tuning on retrieval precision, recall, and overall response quality in RAG systems. Reprinted from [37].

To quantify overall model effectiveness, the study introduces a Composite Performance Score (CPS), which aggregates multiple evaluation metrics. As illustrated in Fig. 10, the CPS varies across similarity thresholds, highlighting the trade-off between retrieval precision and recall. Higher similarity thresholds improve factual accuracy by filtering out less relevant retrieved documents but risk excluding useful contextual information. Conversely, lower thresholds enhance recall, retrieving a broader set of passages at the expense of increased irrelevant data.

To ensure the continued advancement of RAG systems, rigorous evaluation and benchmarking frameworks play a central role in identifying strengths, weaknesses, and areas for improvement. The development of standardized assessment methods, has provided valuable insights into retrieval accuracy, generative quality, and integration effectiveness, allowing researchers to refine models systematically. Additionally, domain-specific evaluations, highlight the

importance of computational infrastructure and transparency in real-world implementations.

While these evaluation frameworks have significantly enhanced the reliability of RAG models, persistent challenges remain in scalability, adaptability, and factual grounding. Addressing these limitations requires a forward-looking approach, integrating emerging methodologies and refining existing techniques to ensure the robustness of future RAG systems. The following section explores key directions for overcoming these challenges and improving the next generation of retrieval-augmented models.

6. Challenges

Despite the considerable progress in Retrieval-Augmented Generation (RAG), real-world deployments continue to expose a number of persistent challenges [38]. These include:

- **Missing content**, where relevant knowledge is absent from the corpus, leading to outdated or incomplete answers
- **Missed top-ranked documents**, when the retrieval mechanism fails to surface the most relevant evidence
- **Fragmented context**, caused by incoherent or contradictory sources that undermine generation quality
- **Poor content extraction**, especially in technical domains where key information is buried or ambiguously expressed
- **Inconsistent structuring**, where output formatting fails to meet task-specific needs
- **Incorrect specificity**, producing answers that are either too vague or overly detailed
- **Incomplete responses**, where outputs omit essential information, leaving queries only partially addressed

These failure points highlight the structural and operational fragilities that still affect many RAG systems. As RAG continues to permeate sensitive application areas like scientific research, healthcare, and legal reasoning, the demand for verifiable, context-aware, and scalable solutions is growing.

However, many of these challenges are already being mitigated by targeted innovations introduced in recent frameworks.

For example, Auto-RAG and GraphRAG address the issue of missing content by enabling dynamic corpus updates and modular knowledge structuring, ensuring the system remains aware of evolving information. Failures related to missed top-ranked documents are tackled by Relevance Sampling, which improves training signal quality, and FLARE, which iteratively reassesses and enhances retrieval during generation.

To prevent fragmented or contradictory responses – categorized as not in context—approaches like LightRAG and Self-RAG introduce semantic filtering and feedback-driven refinement to improve coherence. Similarly, not extracted issues are mitigated by frameworks such as RAVEN, FLARE, and PaperQA, which enhance the precision of information extraction through attention mechanisms and iterative evidence validation.

More pragmatic failure points – such as wrong format or incorrect specificity—are addressed by customizable pipelines like RaLLe, which allows developers to tailor output structure and granularity to specific domain requirements. Finally, issues of incomplete responses are mitigated by Speculative RAG, Self-RAG, and FLARE, all of which incorporate mechanisms for multi-pass reasoning and dynamic retrieval that help ensure comprehensive and well-grounded outputs. These developments demonstrate that while core challenges remain, the RAG research community has begun to proactively design mechanisms that directly address known vulnerabilities – paving the way for more resilient and context-aware systems.

Nevertheless, no single framework comprehensively resolves all of these challenges. Most existing systems are highly specialized, optimized for specific tasks or domains, and often incompatible with one another. This fragmentation has led to a patchwork of partial solutions rather than a cohesive architecture. As RAG systems are increasingly applied in high-stakes fields like healthcare, legal reasoning, scientific research, and enterprise knowledge management, the need for integrated, transparent, and self-aware architectures becomes urgent.

7. Future Directions

Although recent frameworks have begun to address specific RAG failure modes – such as missing content, incoherent context, or incomplete outputs – these solutions remain fragmented across different architectures. The current landscape is marked by specialized systems that are often incompatible, difficult to integrate, or narrowly focused on individual tasks or domains. To fully realize the potential of Retrieval-Augmented Generation, the next phase of research must emphasize convergence: developing unified, modular frameworks that integrate the most effective innovations into a cohesive and generalizable architecture.

Four strategic priorities should guide this effort:

- **Real-time retrieval validation**: Future systems must embed mechanisms to assess the accuracy, currency, and contextual relevance of retrieved content before it is used for generation. This includes timestamp-aware retrieval, domain-adaptive filters, and low-latency consistency checks that prevent outdated or incorrect evidence from influencing outputs.
- **Dynamic reranking and query rewriting**: As a response to uncertainty or failure detection during inference, systems should be able to adapt retrieval

strategies on the fly. This includes rephrasing queries, expanding retrieval depth, or reranking results based on the evolving generative context – enhancing flexibility and resilience in multi-hop reasoning or ambiguous tasks.

- **Sentence-level attribution mechanisms**: Especially in high-stakes domains, future RAG architectures must link each generated statement to a verifiable source at sentence granularity. This would improve transparency, traceability, and user trust – while also enabling downstream auditing, fact-checking, and human-in-the-loop validation.
- **Explicit failure detection and correction layers**: Runtime modules should be capable of detecting hallucinations, omissions, or irrelevance as they occur, and trigger corrective retrieval or regeneration steps. Such mechanisms may involve confidence scoring, evidence alignment checks, or pattern-based failure mode recognition – drawing inspiration from recent advances in reflective and speculative generation.

Ultimately, these directions signal a shift in RAG development from performance-centric metrics to trust-centric design. The goal is no longer simply to generate plausible responses, but to ensure that every output is verifiable, complete, and contextually appropriate. As RAG continues to expand into mission-critical domains, its future will depend not only on architectural sophistication but on its ability to consistently retrieve the right evidence – and explain why it matters.

8. Conclusion

This review has traced the technical evolution and architectural diversification of Retrieval-Augmented Generation (RAG) systems, emphasizing the functional innovations developed to enhance retrieval precision, domain adaptability, scalability, and factual consistency. Through a comparative analysis of major RAG frameworks, the paper demonstrated how specific solutions – such as iterative refinement, modular pipelines, and multimodal integration – have responded to known system vulnerabilities, including incomplete retrieval, hallucinated content, and weak evidence alignment.

This review offered a synthesized view of these advancements into thematic categories, highlighting how current frameworks address recurring failure points in real-world applications. However, the analysis also revealed a persistent fragmentation in the RAG landscape: while individual frameworks target isolated issues effectively, no single system offers a comprehensive solution capable of generalizing across domains, modalities, and dynamic data conditions.

In light of these findings, the findings suggest the need for a shift in RAG research from isolated innovation to architectural consolidation. Future work should prioritize the development of unified, self-aware systems that embed real-time retrieval validation, adaptive reranking and query rewriting, sentence-level

attribution, and integrated failure correction layers. These features are essential for ensuring not just the fluency of generated content, but its credibility, traceability, and reliability in high-stakes domains.

Ultimately, the next generation of RAG systems will be evaluated not by their capacity to generate responses, but by their ability to retrieve relevant knowledge, ground outputs in verifiable evidence, and transparently justify their claims.

References

[1] Bush, V.L As we may think. The Atlantic, Jul. (1945), https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/

[2] Weaver, W., Translation, Machine Translation of Languages - 14 essays, pp. 15–23, MIT Press, (1955).

[3] Luhn, H.P.: Key word-in-context index for technical literature (KWIC index), American Documentation, vol. 11(4), pp. 288–295, (1960), https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.5090110403.

[4] Kent, A., Rees, J.: Mechanized searching experiments using the WRU searching selector, American Documentation, vol. 9(4), pp. 277–303, (1958), https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.5090090404.

[5] Mooers, C. N.: Zatocoding applied to mechanical organization of knowledge, American Documentation, vol. 2(1), pp. 20–32, (1951), Available: https://onlinelibrary.wiley.com/doi/10.1002/asi.5090020107.

[6] Green, B. F., Wolf, A. K., Chomsky, C., Laughery, K., Baseball: An automatic question-answerer, in Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference on - IRE-AIEE-ACM '61, Los Angeles, California: p. 219, ACM Press, (1961), http://portal.acm.org/citation.cfm?doid=1460690.1460714.

[7] Cleverdon, C.W., Keen, M.: Aslib Cranfield research project – Factors determining the performance of indexing systems; Vol. 2, Test results, (1966), http://hdl.handle.net/1826/863.

[8] Winograd, T.: Understanding natural language, Cognitive Psychology, vol. 3(1), pp. 1–191, Jan. (1972), https://www.sciencedirect.com/science/article/pii/0010028572900023.

[9] Sparck Jones, K.: A statistical interpretation of term specificity and its application in Retrieval, Journal of Documentation, vol. 28(1), pp. 11–21, Jan. (1972), https://doi.org/10.1108/eb026526.

[10] Woods, W., Kaplan, R., Webber, B.: The Lunar Science Natural Language Information System: Final Report, Jan. (1972), Cambridge, Mass.: Bolt, Beranek and Newman, inc., (1972)

[11] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing, Commun. ACM, vol. 18(11), pp. 613–620, (1975), https://doi.org/10.1145/361219.361220.

[12] Ferrucci, D.A.: Introduction to 'This is Watson,' IBM Journal of Research and Development, vol. 56(3.4), p. 1:1-1:15, May 2012, https://ieeexplore.ieee.org/document/6177724.

[13] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space, In: Proc. of Workshop at ICLR, vol. 2013, Jan. (2013). arXiv:1301.3781, https://doi.org/10.48550/arXiv.1301.3781.

[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need, Aug. 02, (2023), arXiv:1706.03762, https://doi.org/10.48550/arXiv.1706.03762.

[15] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.: Dense passage retrieval for open-domain question answering, In: Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, pp. 6769–6781, (2020), https://aclanthology.org/2020.emnlp-main.550/.

[16] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners., OpenAI blog, (2019), https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[17] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks, Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Apr. 12, (2021), arXiv:2005.11401, https://doi.org/10.48550/arXiv.2005.11401.

[18] Hofstätter, S., Chen, J., Raman, K., Zamani, H.: FiD-Light: Efficient and Effective Retrieval-Augmented Text Generation, Sep. 28, (2022), arXiv:2209.14290, https://doi.org/10.48550/arXiv.2209.14290.

[19] KILT Benchmarking, https://ai.meta.com/tools/kilt, last accessed: Mar. 23, (2025).

[20] Guo, Z., Xia, L., Yu, Y., Ao, T., Huang, C., LightRAG: Simple and Fast Retrieval-Augmented Generation, Nov. 07, (2024), arXiv:2410.05779, https://doi.org/10.48550/arXiv.2410.05779.

[21] Yu, T., Zhang, S., Feng, Y.: Auto-RAG: Autonomous retrieval-augmented generation for large language models, Nov. 29, (2024), arXiv:2411.19443, https://doi.org/10.48550/arXiv.2411.19443.

[22] Hofstätter, S., Chen, J., Raman, K., & Zamani, H.: Multi-task retrieval-augmented text generation with relevance sampling, Jul. 07, (2022), arXiv:2207.03030 https://doi.org/10.48550/arXiv.2207.03030.

[23] Wang, Z., Wang, Z., Le, L., Zheng, H., Mishra, S., Perot, V., Zhang, Y., Mattapalli, A., Taly, A., Shang, J., Lee, C.-Y., Pfister, T.: Speculative RAG: Enhancing retrieval augmented generation through drafting, Jul. 11, 2024, arXiv:2407.08223. https://doi.org/10.48550/arXiv.2407.08223.

[24] Yu, W.: Retrieval-augmented generation across heterogeneous knowledge. In Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, D. Ippolito, L. H. Li, M. L. Pacheco, D. Chen, and N. Xue, Eds., Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, pp. 52–58, Jul. (2022), https://doi.org/10.18653/v1/2022.naacl-srw.7.

[25] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Larson, J.: From local to global: A graph RAG approach to query-focused summarization, Apr. 24, 2024, arXiv: arXiv:2404.16130. https://doi.org/10.48550/arXiv.2404.16130.

[26] Mandikal, P., Mooney, R.: Sparse meets dense: A hybrid approach to enhance scientific document retrieval, Jan. 08, 2024, arXiv: arXiv:2401.04055. https://doi.org/10.48550/arXiv.2401.04055.

[27] SPARQL 1.1 Query Language. https://www.w3.org/TR/sparql11-query/, Last accessed: Mar. 23, (2025).

[28] Hoshi, Y., Miyashita, D., Ng, Y., Tatsuno, K., Morioka, Y., Torii, O., Deguchi, J, RaLLe: A framework for developing and evaluating retrieval-augmented large language models, Oct. 16, 2023, arXiv: arXiv:2308.10633. https://doi.org/10.48550/arXiv.2308.10633.

[29] Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., Rodriques, S. G., White, A. D. PaperQA: Retrieval-augmented generative agent for scientific research, Dec. 14, (2023), arXiv: arXiv:2312.07559. https://doi.org/10.48550/arXiv.2312.07559.

[30] Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., Neubig, G.: Active retrieval augmented generation, Oct. 22, (2023), arXiv: arXiv:2305.06983. https://doi.org/10.48550/arXiv.2305.06983.

[31] Chen, W., Hu, H., Chen, X., Verga, P., Cohen, W.W.: MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text, Oct. 20, (2022), arXiv: arXiv:2210.02928. https://doi.org/10.48550/arXiv.2210.02928.

[32] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S., Grave, E.: Atlas: Few-shot learning with retrieval augmented language models, Nov. 16, (2022), arXiv: arXiv:2208.03299. https://doi.org/10.48550/arXiv.2208.03299.

[33] Xia, S., Wang, X., Liang, J., Zhang, Y., Zhou, W., Deng, J., Yu, F., Xiao, Y.: Ground every sentence: Improving retrieval-augmented LLMs with interleaved reference-claim generation, Jul. 01, (2024), arXiv: arXiv:2407.01796. https://doi.org/10.48550/arXiv.2407.01796.

[34] Es, S., James, J., Espinosa-Anke, L., Schockaert, S. RAGAS: Automated Evaluation of Retrieval Augmented Generation, Sep. 26, (2023), arXiv: arXiv:2309.15217. doi: 10.48550/arXiv.2309.15217.

[35] Chen, J., Lin, H., Han, X., Sun, L.: Benchmarking large language models in Retrieval-augmented generation, Dec. 20, 2023, arXiv: arXiv:2309.01431. https://doi.org/10.48550/arXiv.2309.01431.

[36] Radeva, I., Popchev, I., Doukovska, L., Dimitrova, M.: Web application for retrieval-augmented generation: Implementation and testing. Electronics, 13, 7, MDPI, 2024, https://doi.org/10.3390/electronics13071361.

[37] Radeva, I., Popchev, I., Dimitrova, M.: Similarity thresholds in retrieval-augmented generation. In: Proc. of the 12th IEEE International Conference on Intelligent Systems - IS'24, 29-31 August 2024, Varna, Bulgaria, IEEE Xplore, 2024, https://doi.org/10.1109/IS61756.2024.10705214.

[38] Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., Abdelrazek, M.: Seven failure points when engineering a retrieval augmented generation system, Jan. 11, 2024, arXiv:2401.05856, https://doi.org/10.48550/arXiv.2401.05856.