A Comprehensive Systematic Review of Retrieval-Augmented Generation (RAG): Developments, Limitations, and Future Pathways

Anvar[1], Sreeji K.B.[2]

1(Department of Computer Applications, Nehru College of Engineering and Research Centre, Pampady, India
Email: anvarkangadiyil@gmail.com)

2(Department of Computer Applications, Nehru College of Engineering and Research Centre, Pampady, India
Email: sreejirithu2018@gmail.com)

Abstract:

Widespread deployment of large language models (LLMs) across knowledge-intensive industries has brought their core architectural weakness into sharp focus: a fixed internal knowledge state that cannot reflect post-training developments and that carries a persistent risk of generating plausible yet factually unsupported content. Retrieval-Augmented Generation (RAG) offers a compelling remedy by coupling the generative capacity of LLMs with a dynamically queryable external knowledge store, thereby decoupling reasoning from memorisation. This work conducts a structured systematic review of RAG research spanning the period 2020 through 2026, charting its progression from rudimentary retrieve-then-read configurations toward sophisticated pipelines that incorporate modular retrieval components and autonomous agent-driven reasoning. Core technical mechanisms are analysed in depth, covering bi-encoder and late-interaction retrieval models, multi-passage fusion strategies, and the complementary roles of lexical and semantic search. Quantitative evidence drawn from widely adopted open-domain benchmarks confirms that retrieval-augmented systems consistently surpass purely parametric baselines on factual question-answering tasks. The review further examines how self-critique loops and structured knowledge graphs are being employed to reduce model hallucinations at scale. Concluding observations chart priority research directions in multimodal retrieval, temporal knowledge decay, and privacy-safe retrieval, positioning RAG as the foundational knowledge infrastructure for next-generation trustworthy AI deployments.

Keywords — Retrieval-Augmented Generation, Large Language Models, Dense Passage Retrieval, Knowledge Grounding, Agentic AI Systems, Semantic Vector Search

I. INTRODUCTION

Advances in transformer architecture over the past decade have yielded language models of extraordinary scale, capable of engaging in nuanced dialogue, synthesising multi-document summaries, and solving complex reasoning tasks with little to no task-specific training. Yet as these systems move from controlled research benchmarks into high-stakes production environments — clinical decision support, legal research, engineering documentation — a sobering pattern has emerged: their outputs, while grammatically polished, are not always factually trustworthy.

The root cause is architectural. An LLM's knowledge is encoded in billions of learned weight parameters during pre-training on a fixed corpus snapshot. Once training concludes, that knowledge is sealed — the model cannot incorporate new information, correct outdated facts, or access domain-specific proprietary data without an expensive retraining cycle. Compounding this limitation, the statistical next-token prediction mechanism that drives generation can produce confident, fluent sentences about events that never occurred — a phenomenon widely referred to as hallucination [1]. In regulated engineering contexts, where a single erroneous material specification or safety parameter could have severe consequences, such unreliability is operationally unacceptable.

Retrieval-Augmented Generation addresses this gap by reformulating the LLM as a reasoning engine that operates over evidence rather than memory [2]. Rather than relying solely on frozen weights, a RAG system queries a live, updateable knowledge store at inference time, fetching relevant passages and conditioning the generator on retrieved facts. This separation of storage from reasoning yields a semi-parametric system that inherits the fluency and instruction-following capacity of large generative models while gaining the factual fidelity of a curated information retrieval backend [7].

The field has matured rapidly since its introduction. What began as a straightforward pipeline — retrieve a fixed number of passages, prepend them to the prompt, generate — has grown into a rich ecosystem of modular architectures featuring intelligent query rewriting, hybrid retrieval, learned re-ranking, iterative self-critique, and graph-based global reasoning [12]. Each layer of sophistication addresses a specific failure mode identified in prior generations, reflecting a disciplined, empirically driven engineering progression.

This paper traces that progression systematically, examining landmark contributions from foundational dense retrieval work in 2020 through to agentic RAG deployments observed in 2025 and 2026 [15]. Drawing upon analysis of more than ninety primary research sources, it provides engineers and researchers with a structured reference covering system architectures, performance benchmarks, evaluation frameworks, and an evidence-based research agenda for the years ahead.

II. LITERATURE REVIEW

The conceptual groundwork for RAG was established by Lewis et al. [1], whose NeurIPS 2020 paper demonstrated that augmenting a seq2seq language model with a differentiable dense retrieval index over Wikipedia produced measurable gains on multiple knowledge-intensive benchmarks. Crucially, the retrieved passages also served as interpretable evidence, offering a degree of answer provenance absent from purely parametric models. Shortly thereafter, Guu et al. [5] introduced REALM, which moved beyond post-hoc retrieval by jointly optimising the retriever and the language model during pre-training, rewarding the retriever whenever its selections improved the model's masked token prediction.

A major catalyst for practical adoption was the Dense Passage Retriever (DPR) of Karpukhin et al. [5], which established that a lightweight dual-encoder trained with contrastive objectives could substantially outperform the classical BM25 lexical baseline on question-answering retrieval. The ColBERT model [6] extended this by introducing late-interaction scoring — encoding queries and documents independently but computing relevance through a fine-grained token-level interaction at query time — achieving a favourable balance between the expressiveness of cross-encoders and the speed of bi-encoders.

On the generation side, Izacard and Grave [7] showed through their Fusion-in-Decoder (FiD) architecture that a T5-based model could effectively synthesise evidence from up to one hundred retrieved passages by encoding each passage independently and performing cross-attention fusion only in the decoder. This design circumvented context-window bottlenecks that had constrained earlier approaches. Borgeaud et al. [8] took a more radical integration strategy with RETRO, embedding chunked cross-attention layers at every transformer block to continuously condition generation on retrieved neighbours from a trillion-token index, enabling a comparatively small model to reach the perplexity of much larger parametric counterparts.

A qualitative shift occurred in 2023 with the introduction of self-aware retrieval frameworks. Self-RAG [10] trained a generative model to emit special reflection tokens that signal whether retrieval is warranted for a given span and whether a retrieved passage actually supports the generated claim. This made retrieval selective rather than unconditional, reducing latency and irrelevant context injection. Complementarily, IRCoT [11] demonstrated that interleaving chain-of-thought reasoning steps with retrieval calls — using each intermediate reasoning conclusion as a new query — enabled models to navigate multi-hop inference chains that outpaced what any single retrieved document could support.

By 2024 and into 2026, structural extensions dominated the literature. Microsoft Research's GraphRAG [16] identified a persistent weakness in vector-search-based RAG: while it excels at targeted fact lookup, it struggles to answer holistic, corpus-level questions such as identifying overarching themes across a large document collection. GraphRAG addressed this by constructing entity-relationship graphs during indexing, enabling community detection and thematic summarisation at query time. Concurrent agentic RAG research [13] explored delegating retrieval decisions to LLM-powered planning agents, introducing tool-use and multi-step task decomposition as first-class capabilities. Outstanding challenges around multimodal document understanding and conflicting evidence resolution continue to motivate active research.

III. METHODOLOGY

A production-grade RAG system is an orchestrated assembly of specialised components rather than a monolithic model. Its operation is best understood through three sequential phases — offline indexing, online retrieval, and context-conditioned generation — each of which contains multiple configurable engineering choices that materially affect overall system quality [7].

A. Indexing and Pre-Processing

Before any query can be served, the target knowledge corpus must be transformed into a representation that supports rapid semantic lookup. This offline pipeline begins with document ingestion: source materials ranging from technical PDFs to internal wikis are extracted, normalised, and stripped of formatting artefacts that would degrade embedding quality [11].

1) Chunking and Segmentation: Extracted text is partitioned into fixed-length or semantically coherent segments. Window size is a design trade-off: overly short windows lose surrounding context and yield underspecified embedding vectors, while excessively long windows dilute the topical signal and trigger the known degradation in LLM attention at mid-sequence positions [11]. Sliding-window overlap and recursive hierarchical chunking are common mitigations [36].

2) Semantic Embedding: Each text segment is encoded by a transformer-based dense encoder into a fixed-dimension float vector that captures distributional semantics. For specialist domains — particularly engineering subdisciplines with dense technical vocabulary — off-the-shelf general encoders may map domain jargon unreliably; continued pre-training or fine-tuning on in-domain corpora is therefore recommended [7].

3) Vector Indexing and Storage: Encoded vectors are persisted in a dedicated vector store such as FAISS, Milvus, or Pinecone. Efficient approximate nearest-neighbour (ANN) search is typically achieved through Hierarchical Navigable Small World (HNSW) indexing, which enables sub-linear query times across corpora of many millions of documents [34].

B. The Retrieval Phase

At inference time, the retrieval phase is responsible for identifying the knowledge segments most likely to contain evidence relevant to the user query. Three sub-components govern retrieval quality [10].

1) Hybrid Search: Pure dense retrieval can underperform on queries containing rare identifiers, model numbers, or domain-specific codes, where lexical overlap with stored text is highly diagnostic. Hybrid retrieval pipelines execute both a dense ANN search and a sparse keyword search (typically BM25) in parallel, then merge the ranked lists via Reciprocal Rank Fusion (RRF) [7]. The combined ranking benefits from the semantic generalisation of dense retrieval and the exact-match precision of sparse retrieval.

2) Query Transformation: User queries are often ambiguous, underspecified, or phrased in a vocabulary that diverges from document language. Multi-Query reformulation addresses this by prompting an LLM to generate several semantically distinct rephrasings of the original query, broadening recall [7]. Hypothetical Document Embedding (HyDE) takes an alternative approach: a generative model first produces a hypothetical answer document, and the embedding of that hypothetical document — rather than the query itself — is used to drive the nearest-neighbour search.

3) Re-ranking: Bi-encoder retrieval optimises for computational throughput rather than fine-grained relevance scoring. A subsequent cross-encoder re-ranker — which processes the (query, passage) pair jointly — applies a more discriminative relevance model to the top-k candidates before they enter the generator prompt. This two-stage architecture achieves high recall at the retrieval stage and high precision at the generation stage without prohibitive inference cost [36].

C. The Generation and Fusion Stage

The generator receives a structured prompt assembled from the user query together with the re-ranked retrieved segments. Its objective is to synthesise a response that is grounded in — and attributable to — the supplied evidence [10].

1) Context Integration: In modular RAG variants, retrieved passages are not merely concatenated; architectures such as Fusion-in-Decoder process each passage through the encoder independently and allow the decoder to attend across all passage representations simultaneously [10]. This permits effective utilisation of far more evidence than a standard prompt-window concatenation approach would allow.

2) Self-Reflection and Verification: To guard against the generator hallucinating beyond its retrieved evidence, iterative self-reflection loops have been proposed. After producing a draft answer, a secondary verification pass — either a dedicated critic model or the same LLM with an evaluation prompt — checks whether every factual claim is explicitly supported by a retrieved passage [12]. Unsupported claims trigger targeted re-retrieval before a revised response is produced [39].

IV. RESULTS AND DISCUSSION

Assessing RAG system quality requires a dual lens: task-level performance metrics that situate the system within the broader NLP literature, and retrieval-specific evaluation dimensions that diagnose the faithfulness and relevance of generated content. This section examines both perspectives, drawing on published benchmark results and the emerging RAG Triad evaluation framework.

A. Benchmark Data Analysis

Open-domain question-answering benchmarks — Natural Questions (NQ), TriviaQA, and HotpotQA — have served as the primary common ground for comparing RAG architectures over successive years [25]. Table I collates exact-match (EM) and F1 scores reported across key systems from the literature, illustrating the trajectory of improvement from 2020 through 2025.

TABLE I
PERFORMANCE COMPARISON OF RAG ARCHITECTURES

Architecture	Dataset	Metric	Score
Naive RAG (2020)	NQ	EM	44.5%
REALM (2020)	NQ	EM	40.4%
FiD (T5-Large)	NQ	EM	51.4%
Self-RAG (2023)	NQ	EM	53.7%
MA-RAG (2025)	NQ	EM	59.5%
FiD-KD (2021)	TriviaQA	EM	72.5%
Self-RAG (2023)	TriviaQA	EM	78.4%
MA-RAG (2025)	TriviaQA	EM	87.2%
IRCoT (2023)	HotpotQA	F1	55.2%
MA-RAG (2025)	HotpotQA	F1	52.1%

Several patterns emerge from Table I. First, multi-agent coordination introduced in MA-RAG [42] delivers the largest single generational leap on NQ — a gain of roughly fifteen points over the 2020 Naive RAG baseline — by decomposing complex queries into independently retrievable sub-questions before answer synthesis. Second, self-reflective token mechanisms (Self-RAG) provide consistent gains across both single-hop and multi-hop benchmarks without requiring architectural changes to the underlying generator. Third, HotpotQA F1 scores remain comparatively lower across all systems, confirming that cross-document multi-hop reasoning continues to represent the outstanding unsolved challenge in the field.

B. The RAG Triad of Evaluation Metrics

Standard NLP surface metrics such as BLEU and ROUGE measure n-gram overlap between system output and a reference answer, but are insensitive to whether the generated content is actually grounded in the retrieved evidence. For safety-critical applications, a hallucinated answer with high ROUGE overlap is worse than no answer at all. The RAG Triad [43] was developed specifically to address this evaluative gap, as shown in Table II.

TABLE II
THE RAG TRIAD: EVALUATION DIMENSIONS

Metric	Definition
Context Relevance	Does the retrieved context genuinely address the query intent?
Groundedness	Are all claims in the response derivable from retrieved passages?
Answer Relevance	Does the final response satisfy the original information need?

Automated evaluation frameworks including RAGAS [12] and TruLens [45] implement the RAG Triad using a secondary LLM-as-judge, scoring each dimension without requiring human-labelled gold answers. A Groundedness score of 1.0 certifies that the response contains no claims beyond what the retrieved passages explicitly support — a practical proxy for hallucination absence within the scope of the knowledge base. Groundedness is particularly valuable during regression testing after index updates, as it surfaces retrieval gaps that degrade factual fidelity before they reach end users.

C. Engineering Implications and Limitations

Despite steady benchmark progress, several engineering challenges warrant careful consideration before RAG deployment at scale. Attention degradation at mid-sequence positions means that injecting large numbers of retrieved passages does not guarantee proportional quality gains — positioning of the most critical evidence within the prompt remains an open research question [38]. Multi-hop tasks continue to expose limitations in the retriever's ability to follow logical dependency chains across documents, as reflected in the relatively modest HotpotQA scores in Table I [35].

From an infrastructure standpoint, RAG shifts the computational bottleneck from model size to retrieval latency and index maintenance overhead. As corpus sizes grow and update frequencies increase, ensuring index freshness and query-time performance becomes a non-trivial systems engineering challenge [25]. Emerging test-time compute allocation strategies, which dynamically budget reasoning steps according to query complexity, represent a promising path toward efficient RAG at enterprise scale [13].

V. CONCLUSION

Retrieval-Augmented Generation has undergone a transformation from a conceptually attractive but architecturally simple prototype into a mature, industrially deployable paradigm for knowledge-grounded AI. By externalising the knowledge store and introducing structured retrieval between user intent and model generation, RAG directly addresses the two most consequential weaknesses of pre-trained LLMs: temporal knowledge decay and hallucination under uncertainty.

The review presented here documents a clear developmental arc: from single-stage retrieve-and-read to modular pipelines with hybrid retrieval, adaptive re-ranking, and self-reflective generation; and from static document indices to dynamic graph-structured knowledge bases navigated by autonomous reasoning agents. Benchmark trajectories confirm that each architectural refinement delivers measurable performance gains, particularly on complex multi-hop tasks that require synthesising evidence distributed across many documents. For the engineering practitioner, RAG's modular design is its greatest practical asset: domain knowledge can be updated, audited, and replaced independently of the generator, enabling compliance with data governance requirements that would be impractical to satisfy through model fine-tuning.

VI. FUTURE WORK

Analysis of the current literature surfaces four directions where RAG research is expected to deliver significant near-term advances.

A. Multimodal and Heterogeneous RAG

Virtually all mature RAG infrastructure remains text-centric, yet real engineering knowledge is embedded in circuit diagrams, chemical structure images, tabular datasheets, and CAD files. Architectures that can index, retrieve, and jointly reason over these heterogeneous modalities within a single coherent pipeline represent a critical frontier for industrial AI adoption [33].

B. Temporal Knowledge Management

Static indices age rapidly in fast-moving technical domains. Future RAG systems will require explicit temporal metadata on retrieved passages and the capacity to reason about information recency — favouring recent evidence when timeliness is critical and flagging potential contradictions between older archived documents and current standards [31].

C. Privacy-Preserving Retrieval

Enterprise RAG deployments frequently operate over commercially sensitive data whose processing is constrained by data-residency regulations and frameworks such as the EU AI Act. Federated retrieval architectures, homomorphic-encryption-compatible index structures, and differentially private embedding generation are active areas of investigation that will determine the viability of RAG in regulated sectors [31].

D. Graph-Agent Synergy

The convergence of knowledge-graph-structured retrieval and LLM planning agents is perhaps the most promising direction for achieving human-level corpus-scale sensemaking. An agent that can traverse entity-relationship graphs to build an intermediate conceptual map before issuing targeted vector-search queries is expected to surpass current retrieval architectures on thematic summarisation and cross-domain inference tasks [13].

ACKNOWLEDGMENT

The authors record their appreciation for the institutional support extended by the Department of Computer Applications at Nehru College of Engineering and Research Centre (NCERC), Pampady. Special acknowledgment is due to Ms. Sreeji K.B. for her scholarly mentorship, constructive feedback at every stage of manuscript preparation, and the consistent encouragement that was essential to bringing this work to completion.

REFERENCES

[1] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.

[2] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. W. Chang, "REALM: Retrieval-augmented language model pre-training," in Proc. ICML, 2020.

[3] J. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023.

[4] W. Yu et al., "Survey of retrieval-augmented generation: Architectures, techniques and applications," IEEE Trans. Knowl. Data Eng., 2024.

[5] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020, pp. 6769–6781.

[6] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in Proc. SIGIR, 2020, pp. 39–48.

[7] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. EACL, 2021, pp. 874–880.

[8] S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," in Proc. ICML, 2022.

[9] W. Shi et al., "REPLUG: Retrieval-augmented black-box language models," arXiv:2301.12652, 2023.

[10] A. Asai et al., "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in Proc. ICLR, 2024.

[11] H. Trivedi et al., "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in Proc. ACL, 2023, pp. 10014–10037.

[12] S. Es et al., "RAGAs: Automated evaluation of retrieval augmented generation," in Proc. EACL System Demonstrations, 2024, pp. 150–158.

[13] P. Sarthi et al., "RAPTOR: Recursive abstractive processing for tree-organized retrieval," in Proc. ICLR, 2024.

[14] B. Peng et al., "Graph retrieval-augmented generation: A survey," J. ACM, vol. 37, no. 4, Art. 111, Sep. 2024.

[15] S. Yao et al., "ReAct: Synergizing reasoning and acting in language models," in Proc. ICLR, 2023.

[16] B. Edge et al., "GraphRAG: A graph-based RAG approach for global sensemaking," Microsoft Research, 2024.

[17] Q. Zhao et al., "LongRAG: A dual-perspective retrieval-augmented generation paradigm," in Proc. EMNLP, 2024, pp. 22600–22632.

[18] Z. Jiang, X. Ma, and W. Chen, "LongRAG: Enhancing retrieval-augmented generation with long-context LLMs," arXiv:2406.15319, 2024.

[19] S. Wang et al., "InstructRetro: Instruction tuning post retrieval-augmented pretraining," arXiv:2310.07713, 2023.

[20] RAG 2.0: The 2025 guide to advanced retrieval-augmented generation. [Online].Available: https://vatsalshah.in/blog/the-best-2025-guide-to-rag