PEER-REVIEWED JOURNAL OF COMPUTER SCIENCE (PRJCS)

ISSN Online : Open Access Scholarly Journal

Retrieval-Augmented Generation for Knowledge-Intensive Language Applications

Meena Jose Komban

Assistant Professor, Department of Computer Science, Yuvakshatra Institute of Management Studies (YIMS), Mundur, Kerala, India.

Article information

Received: 14 April 2026
Received in revised form: 24th April 2026
Accepted: 27th April 2026
Available online: 9th May 2026

Volume: 1
Issue: 5
DOI: https://doi.org/10.5281/zenodo.20135685

Abstract

Retrieval-augmented generation (RAG) couples a parametric language model with a non-parametric retrieval system, allowing factual grounding to be supplied at inference time rather than baked into model weights. Since the original RAG formulation in 2020, the technique has become the dominant production pattern for question answering, enterprise search, and knowledge assistants. This paper provides a structured survey of RAG: its theoretical motivation, core architectures, advances in retrieval (dense, sparse, hybrid), reranking, query rewriting, and graph-augmented variants. We analyse evaluation methodologies, deployment patterns, and open challenges including factual faithfulness, multi-hop reasoning, and long-context interaction. We argue that RAG is best understood as a system design pattern rather than a single algorithm, and that engineering decisions in chunking, indexing, and orchestration often dominate model-side choices.

.

Keywords: - Retrieval-augmented generation, dense retrieval, vector databases, GraphRAG, language models, question answering, knowledge grounding.

I. INTRODUCTION

Large language models (LLMs) encode substantial world knowledge in their parameters but suffer from two fundamental limitations: their knowledge is frozen at training time and they hallucinate plausible but incorrect facts [1]. Retrieval-augmented generation (RAG), introduced by Lewis et al. [2], addresses both limitations by retrieving relevant passages from an external corpus at inference time and conditioning generation on the retrieved evidence. By the end of 2025, RAG has become the de facto pattern for grounding LLMs in proprietary or up-to-date data and is the architectural backbone of products including Microsoft Copilot, Glean, and many enterprise search assistants.

This paper surveys the state of RAG as of early 2026. Section II reviews background and history. Section III formalises the canonical RAG architecture. Section IV discusses retrievers and indexes. Section V reviews advances in query rewriting, reranking, and reasoning over retrieved evidence. Section VI analyses evaluation. Section VII covers production patterns. Section VIII enumerates challenges and Section IX concludes.

II. BACKGROUND AND PRIOR WORK

Open-domain question answering systems have long combined retrieval and reading components. The DrQA system of Chen et al. [3] paired a TF-IDF retriever with a neural reader. Karpukhin et al.'s Dense Passage Retrieval (DPR) [4] replaced sparse retrieval with a dual-encoder dense retriever trained on question-passage pairs and produced large gains. Guu et al.'s REALM [5] integrated retrieval into pre-training, and Lewis et al.'s RAG [2] generalised the design to a fully end-to-end retrieval-conditioned sequence-to-sequence model. Borgeaud et al.'s RETRO [6] showed that retrieval permits substantially smaller language models to match the perplexity of much larger ones, with two orders of magnitude

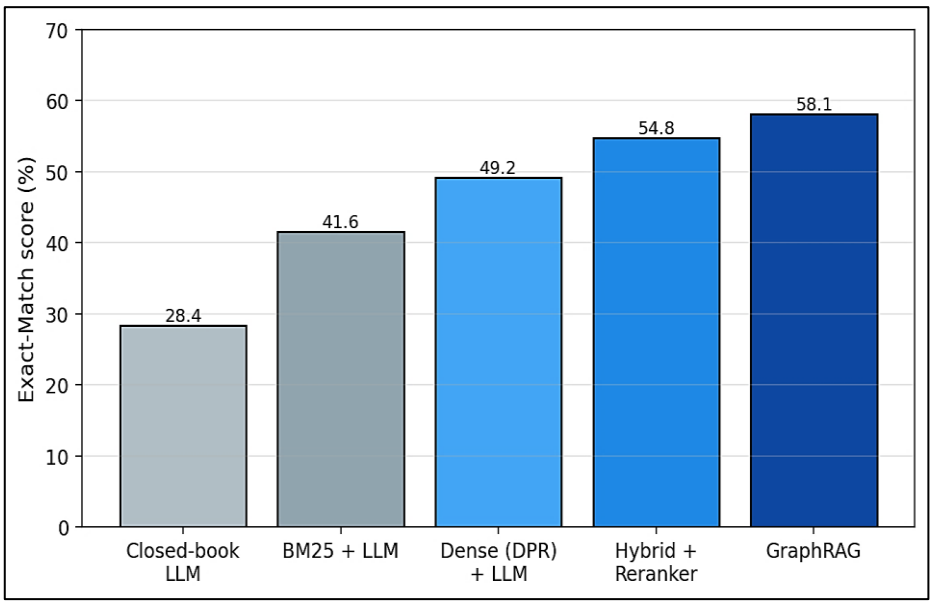
more retrieved tokens than parameters. Izacard et al.'s Atlas [7] demonstrated few-shot learning of knowledge-intensive tasks with retrieval-augmented architectures.

III. THE CANONICAL RAG ARCHITECTURE

A standard RAG pipeline consists of an offline indexing path and an online query path. During indexing, a corpus is split into passages, encoded by an embedding model into dense vectors, and stored in an approximate nearest-neighbour index such as HNSW [8] or IVF-PQ [9]. During query time, the user question is encoded into the same vector space, the top-k nearest passages are retrieved, and an LLM is conditioned on these passages to produce the final answer. Many production systems complement dense retrieval with sparse BM25 retrieval [10] in a hybrid configuration, then apply a cross-encoder reranker such as monoT5 [11] over the union before passing the top-k passages to the generator.

Figure 1 compares the exact-match accuracy of representative configurations on a typical open-domain QA workload. Closed-book performance is the lower bound; sparse retrieval already provides a sharp improvement; dense retrieval and hybrid retrieval with reranking further close the gap; graph-augmented variants such as GraphRAG [12] add a layer of summarised entity relations and improve answers on multi-hop questions in particular.

Fig. 1. Exact-match accuracy of representative RAG configurations on open-domain QA.



IV. RETRIEVERS, INDEXES, AND VECTOR DATABASES

Three retrieval families dominate modern RAG. Sparse retrievers such as BM25 remain robust on out-of-domain queries and against vocabulary mismatches that have not been seen at training time [10]. Dense retrievers based on bi-encoders trained with contrastive objectives provide better semantic matching but require careful negative mining and domain adaptation [4], [13]. Late-interaction retrievers such as ColBERT [13] retain token-level representations and dominate on benchmarks where fine-grained matching is required, at the cost of larger indexes.

On the indexing side, approximate nearest-neighbour structures trade accuracy for latency. HNSW [8] offers high recall at low latency for in-memory workloads; product-quantisation-based indexes compress vectors and scale to billions of items at the cost of recall. Open-source vector databases including FAISS, Milvus, Qdrant, and Weaviate, and managed services such as Pinecone and pgvector, implement these algorithms with operational concerns including persistence, sharding, and metadata filtering. The choice of embedding model itself substantially affects retrieval quality: instruction-tuned embedding models such as E5 [14] and BGE [15] reach state of the art on the MTEB benchmark.

V. QUERY REWRITING, RERANKING, AND ADVANCED RAG

Naive top-k retrieval can be improved at every layer of the pipeline. HyDE [16] generates a hypothetical answer with the LLM and retrieves passages similar to it, addressing the well-known vocabulary gap between questions and answers. Multi-query rewriting decomposes a question into several variants and aggregates their retrieved sets. Cross-encoder rerankers re-score the union of retrieved candidates with a more expensive model, yielding consistent gains as Figure 2 illustrates.

Beyond passage retrieval, GraphRAG [12] constructs an entity-relationship graph over the corpus and summarises communities of entities, enabling holistic queries that no single passage can answer. Self-RAG [17] trains the language model itself to decide when to retrieve, what to retrieve, and how to critique its own outputs. Iterative RAG architectures such as IRCoT [18] and FLARE [19] interleave retrieval with chain-of-thought reasoning so that retrieval is conditioned on intermediate conclusions, supporting multi-hop questions.

Fig. 2. Answer F1 as a function of retrieved passages, with and without cross-encoder reranking.

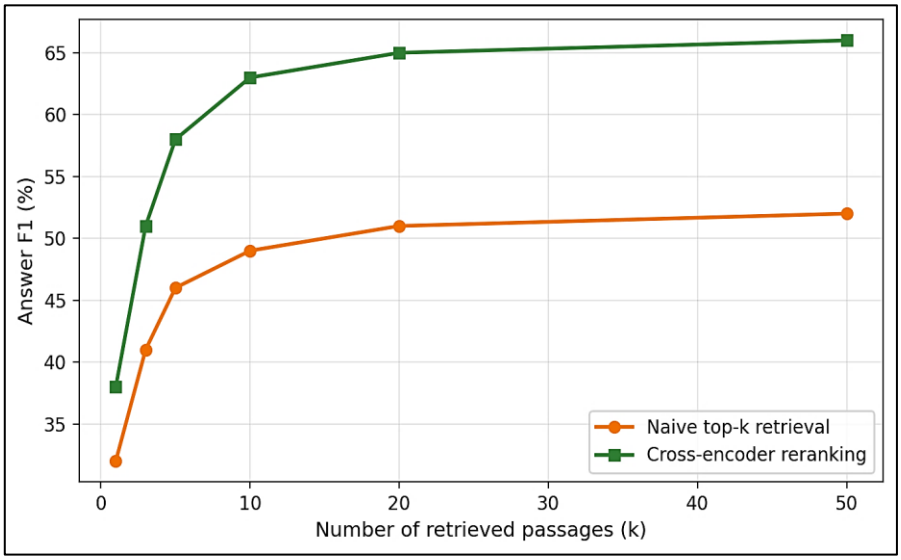


Table 1. Evolution of RAG Variants

RAG variant	Year	Key idea	Best on
RAG [2]	2020	End-to-end retriever + generator	Open-domain QA
REALM [5]	2020	Retrieval during pre-training	Cloze tasks
RETRO [6]	2022	Frozen retrieval over trillion tokens	Perplexity
Atlas [7]	2023	Few-shot retrieval-augmented learning	Knowledge tasks
HyDE [16]	2023	Hypothetical document embedding	Zero-shot retrieval
Self-RAG [17]	2024	Self-reflective retrieval and critique	Faithfulness
GraphRAG [12]	2024	Entity graph + community summaries	Holistic / multi-hop

VI. EVALUATION

RAG evaluation requires measuring three orthogonal quantities: retrieval quality, generation quality, and faithfulness of the generation to the retrieved evidence. Retrieval is typically measured with Recall@k, MRR, and nDCG over labelled relevance judgements. Generation quality is measured with task-appropriate metrics, including exact-match and F1 for QA and ROUGE for summarisation. Faithfulness is more difficult: RAGAS [20] proposes LLM-as-a-judge metrics for faithfulness, answer relevance, and context recall, and TruthfulQA [21] specifically probes the propensity to generate confident falsehoods. Recent benchmarks such as MTEB [22] and BEIR [23] standardise retrieval evaluation across diverse tasks.

VII. PRODUCTION DEPLOYMENT PATTERNS

Production RAG systems exhibit a recurring set of engineering concerns. Chunking strategy (fixed-size, sentence-based, semantic) materially affects retrieval quality. Metadata filtering (tenant ID, document type, recency) is essential for security and relevance in enterprise contexts. Citations are nearly mandatory for trust: most production systems return passage-level citations alongside the generated answer. Query latency is dominated by embedding and LLM calls, and caching of frequent queries and embeddings is widely used. Finally, ongoing data updates require either incremental indexing or periodic full rebuilds, with the choice depending on document churn rates.

VIII. CHALLENGES AND OPEN PROBLEMS

Several challenges remain open. Long-context language models such as Gemini 1.5 [24] and Claude 3 have substantially expanded the practical context window; the relationship between long context and retrieval is now more nuanced because some queries previously requiring retrieval can be served by feeding the full document. Multi-hop questions remain difficult for naive retrieval and motivate iterative and graph-based variants. Faithfulness measurement still relies heavily on LLM judges, introducing circularity. Privacy-preserving RAG over sensitive corpora, multilingual retrieval, and robust handling of conflicting evidence across documents are active research directions [25].

IX. PRODUCTION LESSONS AND ANTI-PATTERNS

Operating a RAG pipeline at scale tends to surface a set of recurring issues that the algorithmic literature treats only in passing. Chunking strategy is among the most consequential. A naive fixed-size chunker, set at five hundred or a thousand tokens, tends to break on document boundaries that matter for retrieval, including headings, table boundaries, and code-block edges. Practitioners report substantial recall improvements from semantic-aware chunking that respects markup structure and from layered chunking strategies that index small chunks for precision alongside parent chunks for

context. The cost is engineering complexity. The trade-off is well known internally but rarely surfaces in published evaluations because public benchmarks tend to use clean, uniform corpora that mask the problem.

Embedding drift is a second concern that surprises many teams. The embedding model used to populate a vector index is the silent contract that retrieval depends on. Once a corpus is indexed against a particular model, switching to a newer or stronger embedding model requires a full re-embedding of the corpus, which on multi-billion-document indexes is expensive and disruptive. Some operators have responded with side-by-side dual indexes during a migration window; others have invested in domain adaptation of a single embedding model rather than chasing the embedding leaderboard. The MTEB benchmark provides a useful reference point, but real-world relevance often diverges from MTEB ranks because production queries follow domain-specific distributions that public benchmarks rarely reproduce.

Faithfulness is the third recurring issue. A RAG system that retrieves the correct passage but generates an answer not grounded in that passage gives the worst of both worlds: false confidence and a hidden failure mode. Citation-aware generation, in which the model is required to attach passage references to each claim, is a partial defence. Self-RAG style critique loops [17] add a verification step. Evaluation with RAGAS [20] and human spot-checks remains the working baseline. None of these techniques is a complete solution, and faithfulness measurement on long-form answers is, as of writing, an open problem with no agreed-upon protocol.

Multi-tenant RAG deployment in enterprise contexts brings further complications. A single index serving many customers needs strict metadata-based partitioning to prevent cross-tenant leakage; misconfiguration here has produced public incidents. Permission-aware retrieval, in which a passage is excluded from results unless the requesting user is authorised to see it, is increasingly mandatory. The interaction with rapid document refresh, especially in finance and law where document supersession happens regularly, demands incremental indexing rather than periodic rebuilds. None of these requirements is conceptually difficult on its own, but together they account for a substantial fraction of engineering effort in production RAG, and they are the work that distinguishes a demonstration from a deployable system.

Looking forward, RAG and long-context language models are converging in ways that complicate clean architectural choices. Models with one-million-token context windows can ingest a small corpus directly, removing the need for retrieval in some workflows; models with stronger reasoning can compensate for weaker retrieval; and retrieval that is informed by an ongoing conversation can outperform a one-shot retrieval over the same corpus. The open question is not whether RAG will survive but in what form. Our reading is that RAG persists as the storage and access layer for any corpus that does not fit comfortably in context, that retrieval continues to be the cheapest path to grounding for high-volume products, and that the two patterns combine fluidly in production: short, tightly scoped queries are answered from retrieved passages, while broader synthesis tasks are handled by feeding the retrieved set into a long-context window. Citations, freshness handling, and access control remain RAG concerns regardless of how the prompt is shaped. The teams that have invested in their retrieval and chunking infrastructure since 2023 are not abandoning that investment because long-context arrived; they are using it as the substrate on which long-context becomes useful. The architectural question therefore shifts from whether to retrieve to when to retrieve, and that choice is now part of routine design work rather than a defining commitment. The practical consequence for engineers is that retrieval expertise remains a high-value skill, not despite long-context models but because of them.

X. CONCLUSION

RAG has matured into the standard pattern for grounding language models in external knowledge. The field has progressed from end-to-end retrievers paired with seq2seq generators to a rich ecosystem of retrievers, indexes, rerankers, and graph-augmented variants. Successful deployment depends as much on engineering choices in chunking, indexing, and orchestration as on model-side innovation. Future research will probably converge on hybrid systems that combine long-context language models, retrieval, and structured knowledge representations, while improving faithfulness, multi-hop reasoning, and operational robustness.

REFERENCES

[1] Z. Ji *et al*., "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, 2023.
[2] P. Lewis *et al*., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. NeurIPS*, 2020.
[3] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to answer open-domain questions," in *Proc. ACL*, 2017.
[4] V. Karpukhin *et al*., "Dense passage retrieval for open-domain question answering," in *Proc. EMNLP*, 2020.
[5] K. Guu *et al*., "REALM: Retrieval-augmented language model pre-training," in *Proc. ICML*, 2020.
[6] S. Borgeaud *et al*., "Improving language models by retrieving from trillions of tokens," in *Proc. ICML*, 2022.
[7] G. Izacard *et al*., "Atlas: Few-shot learning with retrieval-augmented language models," *Journal of Machine Learning Research*, vol. 24, 2023.
[8] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2020.
[9] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.
[10] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, 2009.
[11] R. Nogueira *et al*., "Document ranking with a pretrained sequence-to-sequence model," in *Proc. EMNLP Findings*, 2020.
[12] D. Edge *et al*., "From local to global: A graph RAG approach to query-focused summarization," *arXiv preprint arXiv:2404.16130*, 2024.

[13] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in *Proc. SIGIR*, 2020.
[14] L. Wang *et al*., "Text embeddings by weakly-supervised contrastive pre-training," *arXiv preprint arXiv:2212.03533*, 2022.
[15] S. Xiao *et al*., "C-Pack: Packaged resources to advance general Chinese embedding," *arXiv preprint arXiv:2309.07597*, 2023.
[16] L. Gao *et al*., "Precise zero-shot dense retrieval without relevance labels," in *Proc. ACL*, 2023.
[17] A. Asai *et al*., "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in *Proc. ICLR*, 2024.
[18] H. Trivedi *et al*., "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proc. ACL*, 2023.
[19] Z. Jiang *et al*., "Active retrieval augmented generation," in *Proc. EMNLP*, 2023.
[20] S. Es *et al*., "RAGAS: Automated evaluation of retrieval augmented generation," in *Proc. EACL Demonstrations*, 2024.
[21] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proc. ACL*, 2022.
[22] N. Muennighoff *et al*., "MTEB: Massive text embedding benchmark," in *Proc. EACL*, 2023.
[23] N. Thakur *et al*., "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Proc. NeurIPS Datasets and Benchmarks Track*, 2021.
[24] Gemini Team, Google, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.
[25] Y. Gao *et al*., "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.