INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Retrieval-Augmented Generation for Trustworthy Enterprise LLM Assistants

Bini P B

Assistant Professor, Department of Computer Science, CCSIT Dr John Matthai Center, Thrissur, India

Article information

Received: 13th April 2026
Received in revised form: 16th May 2026
Accepted: 18th June 2026
Available online: 30th July 2026

Volume: 2
Issue: 3
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0033

Abstract

Large language models (LLMs) have changed enterprise knowledge work. Their value, however, is capped by three failures: they hallucinate, their parametric memory is frozen and grows stale, and they cannot read the proprietary data that holds most business answers. Retrieval-Augmented Generation (RAG) targets all three. It grounds generation in passages fetched at inference time from an external, continuously updatable corpus, so answers become verifiable and citation-backed without any model retraining. This paper presents a technical synthesis of RAG for trustworthy enterprise assistants. The end-to-end pipeline is described in full: document chunking, embedding, vector indexing, retrieval, cross-encoder re-ranking, and grounded generation with inline citations. Advanced variants are then surveyed, namely hybrid sparse-dense retrieval, Hypothetical Document Embeddings (HyDE), graph-based RAG, and agentic iterative retrieval. A RAGAS-style evaluation method quantifies faithfulness, answer relevance, and context precision and recall. On an illustrative enterprise question-answering scenario, an advanced configuration that combines hybrid retrieval with cross-encoder re-ranking lifts faithfulness from 0.71 to 0.91 and context precision from 0.62 to 0.84 over naive dense-only RAG. Agentic retrieval reaches 0.95 faithfulness, but pays for it in latency. Enterprise concerns, including document-level access control, data security, cost, and latency budgets, are treated as first-class design constraints. The reported metrics are illustrative. They characterise representative trade-offs rather than a specific deployed study.

Keywords:- Retrieval-Augmented Generation, Large Language Models, Hallucination, Dense Retrieval, Vector Database, Re-ranking, Faithfulness, Enterprise AI.

I. INTRODUCTION

Large language models (LLMs) such as the GPT and LLaMA families read and write text with remarkable fluency. They summarize, answer questions, and reason across long documents, and they now sit behind many enterprise assistants [1], [2]. Fluency is not accuracy. Three structural limits hold these models back in serious business use. First, they hallucinate, asserting confident statements that no source supports [3]. Second, their knowledge is baked into the weights at training time and goes stale as products, policies, and regulations change. Third, they cannot see the proprietary material, the internal wikis, support tickets, contracts, and engineering specifications, where most enterprise answers actually live.Retrieval-Augmented Generation (RAG), introduced by Lewis et al. [4], attacks all three limits at once. It pairs a parametric generator with a non-parametric, searchable knowledge store. At inference time the user query retrieves relevant passages from an external corpus, and those passages are handed to the LLM as grounding context. Knowledge lives outside the weights. It can therefore be refreshed continuously without retraining, kept private inside an organisational boundary, and attributed through

citations so users can check each claim [5]. Separating reasoning from knowledge is exactly what an enterprise assistant needs to be both current and accountable.

This paper gives a technical synthesis of RAG for trustworthy enterprise LLM assistants. The contributions are fourfold:

- A structured account of the end-to-end RAG pipeline and its design choices;
- A survey of advanced retrieval and generation variants, namely hybrid sparse-dense retrieval, HyDE, graph-based RAG, and agentic iterative RAG;
- A RAGAS-style evaluation method with illustrative results that characterise the faithfulness and relevance of each configuration; and
- An analysis of enterprise deployment concerns such as access control, security, cost, and latency.

All quantitative figures are illustrative. They convey representative trade-offs, not results from a specific production deployment.

II. BACKGROUND AND RELATED WORK

A. LLMs and the Hallucination Problem

Modern LLMs are built on the Transformer architecture [6], which uses self-attention to model long-range dependencies, and are trained on web-scale text by next-token prediction followed by instruction tuning and alignment from human feedback [7]. The weights hold a vast but lossy compression of that training data. Ask about facts that are rare, recent, or simply absent, and the model interpolates a plausible answer instead of abstaining. That is a hallucination. Ji et al. [3] survey the phenomenon across generation tasks and separate intrinsic hallucinations, which contradict the supplied source, from extrinsic ones, which cannot be checked against any source. For an assistant handling financial, legal, or medical content, neither is acceptable.

B. Retrieval and Knowledge-Grounded Generation

Augmenting models with retrieval predates instruction-tuned LLMs. REALM [8] trained a retriever and a masked language model jointly, so retrieval was optimised end to end for downstream accuracy. Dense Passage Retrieval (DPR) [9] then showed that a dual-encoder trained with contrastive learning beats sparse lexical baselines on open-domain question answering. Fusion-in-Decoder [10] demonstrated that a generative reader conditioned on many passages scales gracefully with the passage count. Lewis et al. [4] unified retrieval and generation under the RAG name, marginalising the generator over retrieved documents, and set the template that current systems extend.

C. The Modern RAG Landscape

Gao et al. [5] survey the field and organise it into naive, advanced, and modular paradigms, cataloguing the optimisation points across indexing, retrieval, and generation. The practical appeal is direct: RAG converts the open-ended goal of factual accuracy into a more tractable retrieval-and-grounding problem while leaving the underlying LLM frozen. That economics suits enterprises well. A general-purpose model can be wired to a private corpus instead of fine-tuning or pre-training a bespoke model [4], [5].

III. RAG PIPELINE ARCHITECTURE

A production RAG system has two paths. An offline indexing path ingests enterprise documents and builds a searchable index. An online query path answers user queries by retrieving and grounding. Fig. 1 shows both. The subsections below walk through each component.

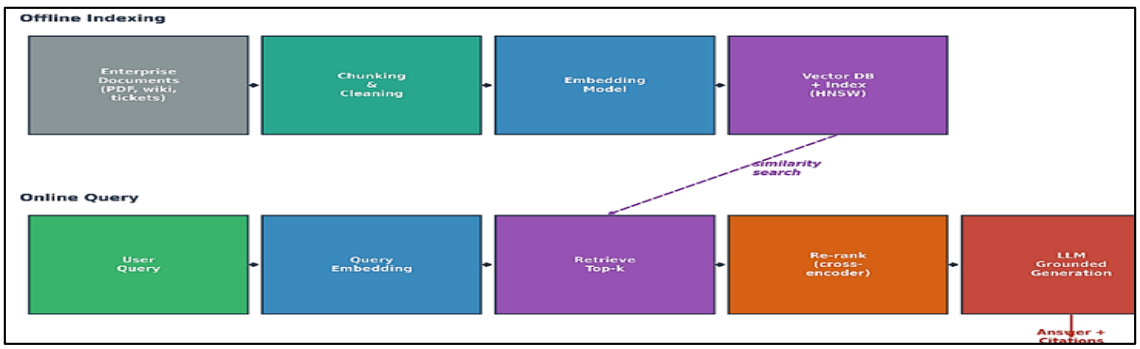

Fig. 1: End-to-end RAG pipeline showing the offline indexing path (chunking, embedding, vector indexing) and the online query path (retrieval, re-ranking, grounded generation with citations).

A. Chunking and Embedding

Source documents are split into chunks. A chunk must be small enough to fit the retriever and generator budget yet large enough to stay semantically coherent. Common strategies include fixed-size token windows with overlap, recursive splitting on document structure such as headings and paragraphs, and semantic chunking that groups sentences by embedding similarity. Each chunk is mapped to a dense vector by an embedding model, usually a sentence-level bi-encoder such as Sentence-BERT [11], which places similar text near itself in vector space. Chunk size, overlap, and metadata (source, section, access tags) materially affect retrieval quality and are key tuning parameters.

B. Vector Indexing and Retrieval

Chunk embeddings are stored in a vector database that supports approximate nearest-neighbour (ANN) search. FAISS [12] provides GPU-accelerated similarity search over billions of vectors, while graph-based indices such as HNSW [13] give logarithmic-time search at high recall. At query time the question is embedded with the same model, and the top-k most similar chunks are returned. Dense retrieval captures semantic similarity but can miss exact-match signals such as product codes or error identifiers. Sparse lexical retrieval such as BM25 [14] excels at those exact matches yet ignores synonymy. This complementarity is what motivates the hybrid retrieval of Section IV.

C. Re-ranking

First-stage retrieval optimises recall over a large corpus, so it returns some loosely relevant passages. A second-stage re-ranker raises precision by scoring each query-passage pair jointly. The bi-encoder embeds query and passage independently. A cross-encoder re-ranker based on BERT [15], [16] instead attends over the concatenated pair, yielding far more accurate relevance scores at higher compute cost. ColBERT [17] sits between the two through late interaction over token-level embeddings. Re-ranking the top 50 to 100 candidates down to the top three to five passages is one of the highest-impact moves for faithfulness.

D. Grounded Generation and Citation

The re-ranked passages are placed into a prompt template that tells the LLM to answer using only the supplied context and to attribute every claim to its source passage. Grounding the generator in retrieved evidence sharply cuts hallucination, and inline citations let users verify answers, which builds trust in the assistant [4], [5]. A good prompt also instructs the model to abstain, replying that the answer is not in the knowledge base, when retrieved context is insufficient. In an enterprise setting, an honest abstention beats a confident fabrication.

IV. ADVANCED RAG TECHNIQUES

Naive RAG, dense retrieval followed by single-shot generation, leaves accuracy on the table. Table 1 summarises advanced techniques that strengthen retrieval and reasoning, and Fig. 2 contrasts naive and advanced configurations across the core metrics.

A. Hybrid Sparse-Dense Retrieval

Hybrid retrieval fuses lexical BM25 [14] scores with dense bi-encoder [9] similarities, usually through reciprocal rank fusion or a weighted score combination. The fusion captures exact-match terms and semantic paraphrase together. In enterprise corpora thick with identifiers, acronyms, and code, it is among the most dependable improvements over dense-only retrieval [5].

B. Query Transformation and HyDE

User queries are often short and underspecified next to the documents that answer them. Hypothetical Document Embeddings (HyDE) [18] close that gap. The LLM drafts a hypothetical answer to the query, and that synthetic document is embedded to retrieve real passages with similar content. Related query-transformation methods rewrite, expand, or decompose the query before retrieval, shrinking the lexical and semantic distance between question and evidence.

C. Graph-based RAG

Flat passage retrieval struggles when an answer is scattered across many documents. GraphRAG [19] builds a knowledge graph of entities and relationships extracted from the corpus, then summarises communities within that graph to answer global, query-focused questions. Traversing explicit relationships supports multi-hop reasoning and corpus-level sensemaking that similarity search alone cannot deliver.

D. Agentic and Iterative RAG

Agentic RAG treats retrieval as an action the model can invoke repeatedly inside a reasoning loop. Self-RAG [20] trains the model to decide when to retrieve and to critique the relevance and support of retrieved passages using reflection tokens. IRCoT [21] interleaves retrieval with chain-of-thought reasoning, so each reasoning step issues a fresher, sharper query. FLARE [22] triggers retrieval whenever the model is about to generate low-confidence content. These iterative methods reach the highest faithfulness on complex multi-hop questions. The price is extra retrieval rounds and added latency.

Table 1. Comparison of RAG Variants

RAG Variant	Key Mechanism	Strength	Primary Cost
Naive (dense)	Bi-encoder top-k + single-shot generation	Simple, low latency	Misses exact match; weak precision
Hybrid	BM25 + dense fusion	Exact match + semantic recall	Fusion tuning
+ Re-ranking	Cross-encoder re-scoring	High context precision	Extra inference per passage
HyDE	LLM drafts hypothetical doc to embed	Bridges short-query gap	One extra LLM call
GraphRAG	Entity graph + community summaries	Multi-hop, global queries	Graph construction cost
Agentic / Iterative	Retrieve-reason-critique loop	Highest faithfulness	Multiple rounds, latency

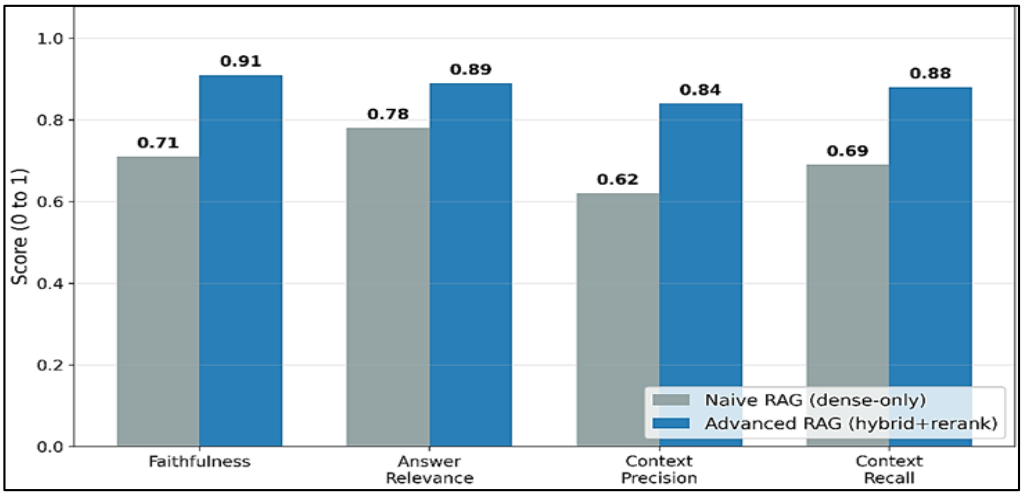

Fig. 2: Naive (dense-only) versus advanced (hybrid retrieval with cross-encoder re-ranking) RAG on RAGAS-style metrics (illustrative).

V. EVALUATION METHODOLOGY AND RESULTS

A. Metrics

A RAG system has to be judged on two axes, retrieval and generation. Following the RAGAS framework [23], four reference-light metrics are used. Faithfulness, or groundedness, is the fraction of claims in the answer that the retrieved context entails; it measures hallucination directly. Answer relevance captures how well the answer addresses the question. Context precision is the share of retrieved passages that are actually relevant, and context recall is the share of the information needed to answer that retrieval found. Each can be estimated with an LLM-as-judge, which avoids large hand-labelled reference sets [23].

B. Experimental Configurations

Six configurations are compared on an illustrative enterprise question-answering scenario over a mixed corpus of policy documents, support tickets, and technical manuals. They are an ungrounded LLM-only baseline, dense RAG, hybrid RAG, hybrid RAG with cross-encoder re-ranking, that configuration plus HyDE, and an agentic iterative variant. Retrieval uses an HNSW index [13]; re-ranking uses a cross-encoder [16]; faithfulness

and relevance are scored by an LLM judge [23]. The reported values are illustrative, chosen to reflect trends that recur in the literature rather than a single measured deployment.

Fig. 3 plots faithfulness and answer relevance across the six configurations. The ungrounded baseline is least faithful at 0.52, confirming that an unaided LLM often asserts unsupported claims. Dense retrieval lifts faithfulness to 0.71, hybrid retrieval to 0.79, and cross-encoder re-ranking to 0.91. The lesson is blunt: the precision of the supplied context, not merely its presence, governs groundedness. HyDE and agentic retrieval add further gains, to 0.93 and 0.95. Table 2 reports the full metric set.

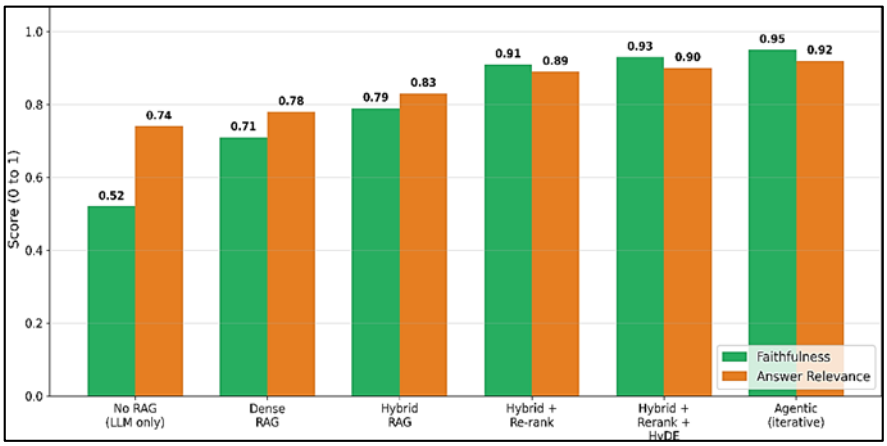

Fig. 3: Faithfulness and answer relevance across six RAG configurations (illustrative).

Table 2. RAGAS-style Evaluation Across Configurations (Illustrative)

Configuration	Faithfulness	Answer Relevance	Context Precision	Context Recall
LLM only (no RAG)	0.52	0.74	n/a	n/a
Dense RAG	0.71	0.78	0.62	0.69
Hybrid RAG	0.79	0.83	0.71	0.81
Hybrid + Re-rank	0.91	0.89	0.84	0.88
Hybrid + Rerank + HyDE	0.93	0.90	0.86	0.90
Agentic (iterative)	0.95	0.92	0.88	0.93

C. Retrieval Depth and Latency Trade-offs

Fig. 4(a) shows how precision and recall move with the number of retrieved passages k. Precision falls and recall climbs as k grows, and hybrid retrieval dominates dense-only retrieval at every depth. The generator context window and the re-ranker cost both scale with k. Practical systems therefore retrieve a wide candidate set, re-rank it, and present only the top few passages. Fig. 4(b) plots faithfulness against end-to-end latency. Re-ranking buys a large faithfulness gain for modest added latency. The agentic loop's extra faithfulness, by contrast, carries a steep latency premium. That tension is the central engineering trade-off an enterprise has to budget for.

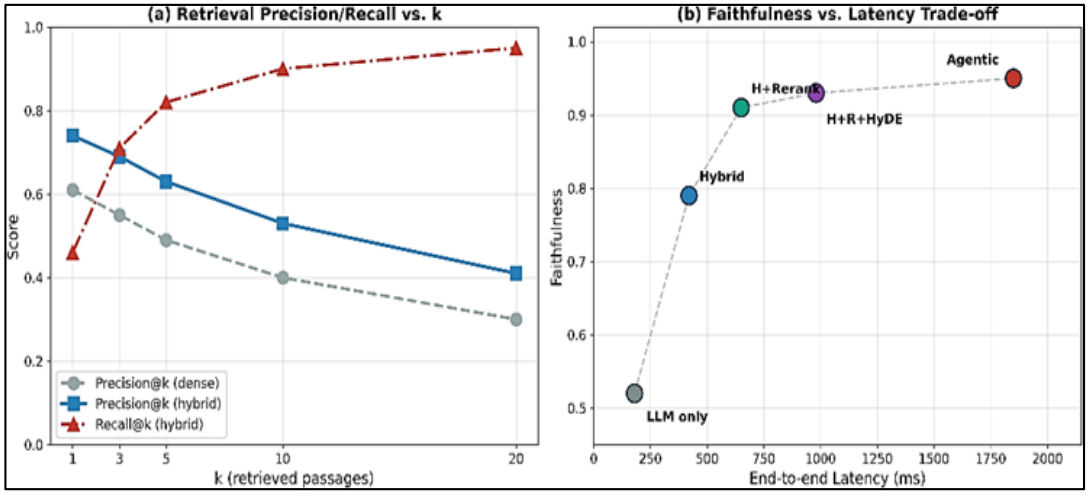

Fig. 4: (a) Retrieval precision and recall versus k for dense and hybrid retrieval; (b) faithfulness versus end-to-end latency across configurations (illustrative).

VI. ENTERPRISE DEPLOYMENT CHALLENGES

A. Access Control and Data Security

Enterprise corpora hold documents with mixed sensitivity and per-user access rights. A correct RAG system enforces those rights at retrieval time, filtering the candidate set to documents the requesting user is authorised to see before any passage reaches the LLM. Access tags stored as chunk metadata make such pre-retrieval filtering possible. Skip this step and a serious leakage channel opens, because the generator will faithfully surface whatever it is handed. Sensitive data must also be protected in transit and at rest, and personally identifiable information may need redaction during ingestion [5].

B. Cost and Latency

Every advanced component adds compute. Embedding and re-ranking cost per-query inference, HyDE and agentic loops add extra LLM calls, and a larger k inflates the generation context and the token bill. Fig. 4(b) makes the implication plain. Teams should pick the cheapest configuration that hits their faithfulness target, not chase accuracy without limit. Caching frequent queries and their retrieved contexts, distilling re-rankers, and right-sizing k are the usual levers for staying inside latency and cost budgets.

C. Maintenance and Governance

RAG knowledge lives outside the model, so the index has to stay synchronised with source systems as documents are created, updated, and deleted. Stale or deleted content that lingers in the index reintroduces the very staleness RAG was meant to cure. Continuous evaluation with the metrics of Section V, human review of low-confidence answers, and audit logs that record which passages grounded each response are all needed for governance and regulatory compliance [23].

VII. CONCLUSION

Retrieval-Augmented Generation has become the default architecture for trustworthy enterprise LLM assistants because it grounds generation in an external, updatable, access-controlled corpus and supports verifiable citations. This paper synthesised the end-to-end RAG pipeline, surveyed advanced variants (hybrid retrieval, HyDE, GraphRAG, and agentic iterative RAG), and described a RAGAS-style evaluation method. One finding stands out across the illustrative results. Context precision is the dominant lever for faithfulness: hybrid retrieval with cross-encoder re-ranking raises faithfulness from 0.71 to 0.91 over naive dense RAG, while agentic retrieval reaches 0.95 at a real latency cost [4], [5], [23].

The practical message for enterprises is simple. Retrieval quality, access control, and latency budgets are first-class design constraints, not afterthoughts, and the right configuration is the cheapest one that meets a target faithfulness threshold. Future directions include multimodal RAG over tables and figures, end-to-end optimisation of retrievers against generation quality, stronger automatic faithfulness verifiers, and tighter integration of structured knowledge graphs with dense retrieval for complex multi-hop enterprise reasoning [5], [19], [20].

REFERENCES

[1] T. B. Brown *et al*., "Language models are few-shot learners," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[2] H. Touvron *et al*., "LLaMA: Open and efficient foundation language models," *arXiv* preprint arXiv:2302.13971, 2023.

[3] Z. Ji *et al*., "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, Art. no. 248, pp. 1–38, Mar. 2023.

[4] P. Lewis *et al*., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 9459–9474, 2020.

[5] Y. Gao *et al*., "Retrieval-augmented generation for large language models: A survey," *arXiv* preprint arXiv:2312.10997, 2023.

A. Vaswani *et al*., "Attention Is All You Need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, pp. 5998–6008, 2017.

[6] L. Ouyang *et al*., "Training language models to follow instructions with human feedback," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, pp. 27730–27744, 2022.

[7] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-augmented language model pre-training," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 3929–3938.

[8] V. Karpukhin *et al*., "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 6769–6781.

[9] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open-domain question answering," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL)*, 2021, pp. 874–880.

[10] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[11] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[12] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, Apr. 2020.
[13] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT)*, 2019, pp. 4171–4186.
[15] R. Nogueira and K. Cho, "Passage re-ranking with BERT," *arXiv* preprint arXiv:1901.04085, 2019.
[16] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, 2020, pp. 39–48.
[17] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2023, pp. 1762–1777.
[18] D. Edge *et al*., "From local to global: A graph RAG approach to query-focused summarization," *arXiv* preprint arXiv:2404.16130, 2024.
A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.
[19] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2023, pp. 10014–10037.
[20] Z. Jiang *et al*., "Active retrieval augmented generation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2023, pp. 7969–7992.
[21] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAS: Automated evaluation of retrieval-augmented generation," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL): Syst. Demonstrations*, 2024, pp. 150–158.