Sciences du Nord Nature Science and Technology
Vol. 2, No. 02, December 2025: pp. 72-86
E-ISSN: 3046-6504
P-ISSN: 3047-4922
DOI: 10.71238/snnst.v2i02.184

KnowledgeTrust-RAG: Reliability-Aware Retrieval for Enterprise Knowledge Management Using Amazon Bedrock

A Systematic Literature Review

Swapna Pulti [1]
[1] Salesforce Solution Manager, Pearson Education Inc, United States pswapna.cse@gmail.com
Coresponding Author: pswapna.cse@gmail.com

ARTICLE HISTORY

Received Dec, 2025	**Revised Dec, 2025**	**Accepted Dec, 2025**

ABSTRACT

Large Language Models are now commonly used for discovering knowledge held within unstructured repositories, but the ability to directly use a generative system to access a proprietary knowledge base suffers from issues with factual hallucination, poor provenance and lack of auditability. To build a conceptual framework, KnowledgeTrust-RAG, for enterprise knowledge management reliability-aware retrieval-augmented generation, this review combines 25 peer-reviewed and preprint sources published up to 2024, using the architectural affordances of a managed generative artificial-intelligence cloud platform like Amazon Bedrock as an illustration. The synthesis connects various previously unexplored components of self-attention-based encoders and dense passage retrieval, the original retrieval-augmented generation formulation, and the self-critiquing and graph-structured models, and summarizes the parallel literatures on hallucination taxonomy, automated evaluation, and enterprise deployment barriers. The results show that reliability of enterprise retrieval-augmented generation is shaped by four interacting processes: dense semantic retrieval, retrieval-necessity gating, global context aggregation via a graph-based approach, and reference-free evaluation metrics like faithfulness and answer relevance. The review also reveals that the variety of evaluation frameworks extends beyond just the number of dimensions, with one framework examining five reliability dimensions while a comparator examines three, and that enterprise-specific challenges, such as data security, access governance and integration complexity, are not sufficiently addressed by generic pipelines. Embedded ingestion, retrieval, orchestration, evaluation and governance layers are proposed, each of which is auditable and could be deployed in the managed cloud separately. The review provides a common classification of the mechanisms for retrieval reliability, a comparison of different architectural variations, and a deployment baseline in a governance perspective which provides a technological basis for designing trustworthy enterprise knowledge systems and can serve as a baseline for future empirical testing on production workloads.

Keywords: *Cloud-Based Generative Artificial Intelligence; Dense Passage Retrieval; Enterprise Knowledge Management; Evaluation Frameworks; Hallucination Mitigation; Information Retrieval; Knowledge Graphs; Large Language Models; Reliability-Aware Retrieval; Retrieval-Augmented Generation; Semantic Search; Trustworthy Artificial Intelligence*

INTRODUCTION

Traditionally, knowledge management has been viewed as a systematic process of encoding, codifying, storing and transferring knowledge in and out of organizations to aid in decision making and competitive advantage. Basic research in this area has shown that a knowledge management system differs from a traditional information system in that it focuses on how knowledge is applied

rather than simply stored; in that knowledge is distributed both in explicit or documented artifacts and in tacit knowledge held by the individual members of the organization [1]. This concept is still relevant today for enterprise information architectures that need to balance the sheer amount of unstructured textual repositories, like policy manuals, technical documentation, customer data and internal mail, and the need for human users to find the right and relevant answer when they need it.

Large Language Models (LLMs) are creating a qualitatively novel knowledge codification and retrieval approach: natural language queries can be answered by generating a synthetic response, rather than using a manual search of documents. The transformative role of large language models in professional and organizational tasks has been highlighted, both in terms of enhancing productivity and decision-making capabilities and in the need for careful scrutiny of the outputs of these systems before integrating them into practice. [2] There are many risks associated with the wholesale application of generative synthesis as a means of retrieval that are not present in consumer-facing conversational applications, as enterprise knowledge work sets a different standard for accuracy, traceability, and regulatory compliance.

The tension has motivated the primary architectural solution: retrieval-augmented generation, which allows for fluent generation while also enabling verifiable retrieval. Surveys of retrieval augmented generation [3] report a growing research field that spans dense retrieval, iterative reasoning, graph-structured representations of knowledge, and agentic retrieval strategies, and that reliability, not fluency, is the key open question for real world deployment. This review aims to solve this issue by synthesizing the literature into a coherent model, which we call KnowledgeTrust-RAG, for reliability-aware retrieval in enterprise knowledge management, and demonstrates its use in a layered service model of a managed cloud generative artificial-intelligence platform, such as Amazon Bedrock

METHOD

This review was implemented through a systematic synthesis procedure based on the taxonomic method used in the comprehensive survey of retrieval-augmented generation [4] where primary studies are classified based on retrieval mechanism, generation strategy and evaluation approach. We selected 25 sources to span a wide range of technical backgrounds, including fundamental neural architectures, the early and later versions of retrieval-augmented generation, graph structured retrieval and adaptive retrieval, automated evaluation frameworks, and parallel work on hallucination and trustworthiness, with only publications made prior to 2024 included.

We accepted submissions that introduced or applied a retrieval or generation architecture that was relevant to knowledge-intensive natural language processing, proposed evaluation methodology for retrieval-augmented systems, or offered a taxonomic survey of knowledge-intensive natural language processing systems, as is done in scoping [5]. Their technical contribution was chosen as the general focus of the analyses conducted on each source, the reported architectural mechanism was identified as the second, and the implications for enterprise reliability were analyzed and synthesized thematically instead of through meta-analytic pooling, due to the diversity of experimental settings. The synthesis is structured in sections, with each dealing with a different aspect of the foundational architectures, the development of retrieval-augmented generation,

reliability-aware retrieval, deployment challenges, evaluation, and trustworthiness as described in Section 3.

Table 1 places the technical developments discussed in this paper in the context of the longer arc of organizational knowledge-management (KM), from document repositories and enterprise search by keywords to semantic and finally generative and graph-augmented approaches to knowledge access [5].

Table 1: Generations of Knowledge Management Systems and Enabling Technologies

Era	Dominant Knowledge-Management Paradigm	Enabling Technology	Reference
Pre-2000s	Document repositories and expert directories	Relational databases, corporate intranets	[1]
2000s-2010s	Enterprise search and controlled taxonomies	Keyword and lexical search (TF-IDF, BM25)	[1]
2019-2020	Semantic search over unstructured text	Contextual embeddings (BERT, dense retrieval)	[6], [7]
2020-2023	Generative knowledge synthesis	Retrieval-augmented pre-training and generation	[8], [9]
2023-2024	Reliability-aware, graph-augmented knowledge access	Self-reflective and graph-structured retrieval; managed cloud platforms	[10], [11]

Source: Compiled by the authors from the reviewed literature.

RESULTS AND DISCUSSION

3.1 Foundational Neural Architectures for Semantic Retrieval

Figure 1: Chronological Progression of Foundational Architectures Underlying Enterprise Retrieval-Augmented Generation

The transformer has been adopted in contemporary retrieval and generation architectures, where recurrent and convolutional sequence models are replaced by a self-attention mechanism that models long-range dependencies in parallel [12]. This architecture is based on a scaled dot-product attention function, which is defined as follows:

$$Attention(Q, K, V) = softmax\left(QK^{\mathrm{T}} — \sqrt{d_{\mathrm{k}}} \right) V (1)$$

The matrices Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors; the scaling term $\sqrt{d_k}$ ensures that the magnitudes of the dot-products do not cause the gradients in the softmax function to become too small. This mechanism allows a model to modify the relevance of each token in an input sequence to each other token, which forms the representational foundation for dense retrieval encoders as well as autoregressive generators.

Bidirectional pre-training techniques were used to allow learning contextualized word representations from left and right context together with the masked-language-modeling and next-sentence-prediction tasks [6]. This bidirectional pre-training paradigm greatly boosted the accuracy of downstream language-understanding tasks and has become a de facto standard architecture for dense semantic representation of enterprise documents as vectors for similarity search.

However, as token-level encoders are expensive to apply pairwise over large document sets, a sentence-embedding architecture was later introduced which fine-tunes a Siamese and triplet network architecture to create semantically meaningful sentence-level vectors for direct comparison via vector similarity [13]. The cosine similarity of two text spans, represented as embeddings u and v, is usually calculated as in Equation 2,

$$\cos(u, v) = \frac{(u \cdot v)}{(\|u\|\|v\|)} (2)$$

Which allows for sub-linear approximate nearest neighbour search over millions of chunks of documents and is the basis of the operational vector indices that are used in production retrieval-augmented generation pipelines, including in managed cloud-based platforms.

Dense passage retrieval extended this work by training a dual-encoder architecture - an encoder for question and an encoder for passage - with contrastive learning over pairs of gold passage and negative passage, which showed that the learned dense representation can significantly outperform the sparse lexical retrieval approach on open-domain question-answering benchmarks [7]. We compute the retrieval score for a question q and a candidate passage p as the inner product of the outputs of their respective encoders, as in Equation 3.

$$sim(q, p) = E_{Q(q)^T} E_{P(p)} (3)$$

The top-k passages, scored according to this measure, are then fed to a generation model, thus creating the two-stage pattern of retrieve-then-generate, from which almost all enterprise applications since then are built. These basic architectures all laid the groundwork for the representational and retrieval primitives that inform enterprise knowledge management systems today, as summarized in Table 2

Table 2: Foundational Neural Architectures Underpinning Enterprise Retrieval

Architecture	Core Mechanism	Primary Contribution	Reference
Transformer	Scaled dot-product self-attention	Parallelizable long-range sequence modeling	[12]
BERT	Bidirectional masked-language pre-training	Contextualized token embeddings	[6]
Sentence-BERT	Siamese/triplet network	Efficient sentence-level	[13]

	fine-tuning	embeddings for similarity search	
Dense Passage Retrieval	Dual-encoder contrastive learning	Dense retrieval outperforming sparse lexical search	[7]
REALM	Joint retriever-generator pre-training	Interpretable retrieval trace during pre-training	[8]

Source: Compiled by the authors from the reviewed literature.

3.2 Evolution of Retrieval-Augmented Generation

Retrieval-augmented pre-training showed that training a masked-language model jointly with a learned retriever enables the model to attend over a vast external corpus during pre-training and inference, rather than just using parameters that will be memorized during pre-training [8]. This paper demonstrated that this can be done in a way that directly enhances the accuracy of answer retrieval on knowledge-intensive tasks while also yielding an interpretable retrieval trace, an aspect that is relevant in an enterprise context where the provenance of answers needs to be auditable.

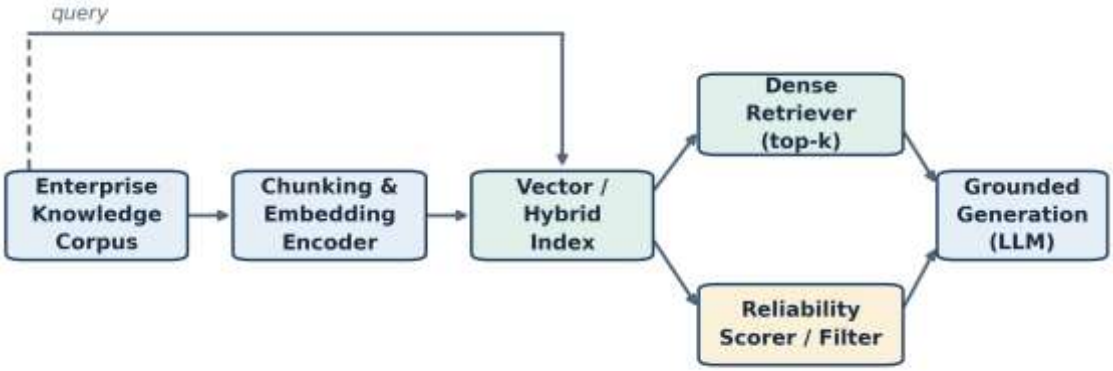

Figure 2: Core Retrieve-Then-Generate Pipeline Architecture for Enterprise Knowledge Management

This concept was extended to the retrieval augmented generation (RAG) framework, which merges a pre-trained dense retriever with a pre-trained sequence-to-sequence generator into a single differentiable structure, where the retriever provides a distribution over passages to be generated during the forward pass through the generator [9]. The sequence-level formulation has the probability of an output sequence y given an input x written in Equation 4,

$$P(y|x) = \Sigma_{\{z \in top-k\}} p_{\eta(Z|X)} \cdot p_{\theta}(y|x,z) \quad (4)$$

The conditional probability of p_η(z|x) over the top-k retrieved documents z is called the retriever's distribution over the top-k retrieved documents, while the conditional probability of p_θ(y|x,z) is called the generator's conditional probability of the output given the input and a retrieved document. This formulation showed that the use of parametric and non-parametric memory led to better factual specificity for the formulation than the use of purely parametric memory, and further reduced the need for retraining when the underlying knowledge source is updated, which is a beneficial property in enterprise repositories that change over time.

Later research defined retrieval-augmented text generation as a class of architectures depending on the granularity of retrieval (token-level, sentence-level, document-level), the temporal extent of retrieval (one-shot retrieval, iterative retrieval, and adaptive retrieval), and the way in which retrieval content is combined with generated text (fused retrieval, fused iteration, and fused adaption). It was established that the applicability of enterprise systems is not related to any particular architectural option but rather to whether the document level granularity of retrieval is suitable for the enterprise's structure of the knowledge base, where the knowledge might be scattered in short-form documents like support tickets or compliance clauses.

3.3 Reliability-Aware and Structurally Enhanced Retrieval

A self-reflective architecture was proposed that employs a single language model to output specific reflection tokens to determine whether to call to retrieval, whether a retrieved passage is relevant, whether a generated segment is supported by that passage, and the usefulness of the retrieved response [10]. It is combined with segment-wise beam search over candidate continuations to provide a retrieve-on-demand mechanism, as shown in Figure 3, which allows the system to avoid retrieval for queries where it is not needed and to enforce more stringent evidentiary checks for queries where it is necessary, thus embodying the idea of reliable retrieval that inspired this review.

Figure 3: Self-Reflective, Retrieve-on-Demand Decision Flow for Reliability-Aware Retrieval

A second approach to making retrieval more reliable is to overcome the quality constraint of passage-level retrieval, that is, by being able to answer questions that require the synthesis of information across an entire corpus. In a graph-based approach, an entity-relationship knowledge graph is extracted from the source documents, partitioned into hierarchical communities, and pre-computes community-level summaries and provide global, corpus-wide questions can be answered using map-reduce summarization over community-level summaries instead of retrieval of individual passages [14]. This architecture is meaningful for enterprise knowledge management, as queries across many documents are the type of queries that are typically present in enterprise knowledge management, and which are not structurally well suited to be addressed using the passage-level dense retrieval approach.

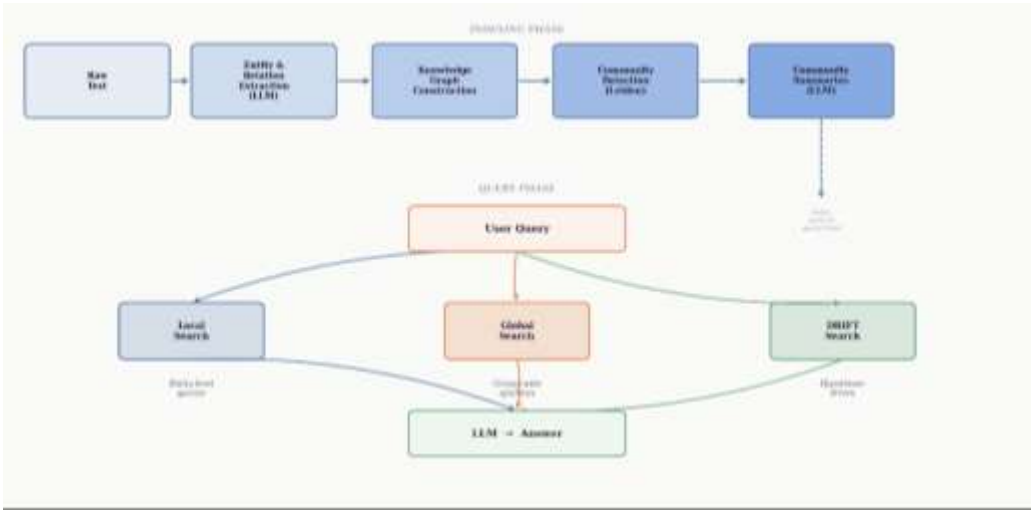

Figure 4: Graph-Structured Retrieval Architecture for Corpus-Wide Enterprise Query Synthesis (Kloia, 2024)

It was recognized that constructing the graph and summarizing communities involve significant indexing and update cost, and so a follow-up lightweight graph-retrieval architecture was proposed in [15] that extends graph-structured indexing with dual-level graph retrieval, operating at two levels: low level (graph entity granularity) and high level (graph thematic granularity), yielding reduced retrieval latency and update overhead compared to full graph-based summarization, but still supporting both graph-specific and graph-thematic queries. This efficiency-based design is relevant for enterprise use, where knowledge bases are continually updated and the complete reconstruction of the graph after each document change would be too expensive.

A more general framework for the knowledge graph--LLM unification was proposed, which outlines several complementary ways to integrate knowledge graphs into LLM pre-training and reasoning, LLM reasoning with knowledge graph construction, and synergistic integration of both in a single reasoning pipeline [15]. This roadmap places graph-based retrieval variants in the context of a larger design space and indicates that there may be an emerging trend of enterprise knowledge management systems transitioning towards hybrid designs that incorporate vector-based knowledge graph search and symbolic graph traversal, as summarized comparatively in Table 3.

Table 3: Comparative Overview of Reliability-Aware Retrieval-Augmented Generation

Variant	Retrieval Trigger	Structural Basis	Key Advantage	Reference
Original RAG	Fixed top-k, always retrieves	Dense passage index	Differentiable joint retriever-generator training	[9]
Self-RAG	On-demand, learned reflection tokens	Passage-level plus self-critique	Avoids unnecessary or irrelevant retrieval	[10]
GraphRAG	Query-dependent, local or global	Hierarchical entity-relation graph	Enables corpus-wide thematic synthesis	[14]
LightRAG	Dual-level (entity and theme)	Lightweight graph index	Lower indexing and update latency	[11]

Table 3: Comparative Overview of Reliability-Aware Retrieval-Augmented Generation Architectural Variants

Source: Compiled by the authors from the reviewed literature.

3.4 Enterprise Deployment Challenges

Generation of generic retrieval-augmented pipelines to enterprise contexts faces system-wide challenges that are not addressed by the typical academic benchmark tests reviewed above. Evaluating enterprise contexts alone revealed that data security and access control, retrieval accuracy when dealing with domain-specific and rapidly changing content, scalability to enterprise volumes of documents, and integration into the existing enterprise workflows were identified as major barriers to production deployments; generic implementations were determined to need specific extensions to meet enterprise-grade accuracy, compliance and governance requirements [16]. These barriers, summarized in Table 4, help drive the layered enterprise architecture proposed in Section 3.7 in which governance and evaluation are considered as first-class layers, not an afterthought.

Table 4: Enterprise Retrieval-Augmented Generation Deployment Challenges and Architectural Requirements

Challenge Category	Description	Architectural Requirement	Reference
Data security and access control	Sensitive proprietary content accessed by generative pipelines	Identity and access management, encryption, audit logging	[16]
Retrieval accuracy on domain content	Generic retrievers underperform on specialized or rapidly changing corpora	Domain-adapted embeddings, continuous re-indexing	[16]
Scalability	Enterprise corpora exceed academic benchmark scale	Distributed vector and graph indices, incremental updates	[16]
Workflow integration	Coexistence with existing tools and stakeholder processes	Application programming interface orchestration, governance layer	[16]

Source: Compiled by the authors from the reviewed literature.

3.5 Reliability Evaluation Frameworks

Since enterprise use of the RAG model necessitates output reliability, automated evaluation frameworks have been created to evaluate the system performance without the need to rely on time-consuming human annotated reference answers. One of such structure breaks evaluation down into reference-free measurements, namely faithfulness, the percentage of claims in the generated answer that can be inferred from the retrieved context; answer relevance, the degree of relevance between the generated answer and the posed question; and context relevance, the percentage of the retrieved context that is relevant to answering the question [17]. Faithfulness is calculated as in Equation 5,

$$Faithfulness = \frac{|claims\ supported\ by\ context|}{|total\ claims\ in\ answer|} \quad (5)$$

While answer relevance is estimated by generating synthetic questions from the answer and computing their average embedding similarity to the original question, as given in Equation 6,

$$Answer\ Relevance\ =\ (1/N)\ \Sigma_i\ cos(E\_g_i, E\_o) (6)$$

Where E_g_i is the embedding of the i-th generated question and E_o is the embedding of the original question. These metrics do not require a reference to be used, allowing for reliable monitoring to be conducted continuously in enterprise deployments where ground-truth answers are not manually annotated at scale.

A complementary evaluation framework trains lightweight classifiers on synthetically generated question-answer-context triplets to predict the human judgment of context relevance, faithfulness of answers, and answer relevance; then applies prediction-powered inference to calibrate the automated predictions against a small number of human-labeled question-answer-context triplets; thereby reducing the human labor needed compared to manual evaluation while maintaining the statistical validity [19]. Both frameworks share the same reliability dimensions of faithfulness, answer relevance and context relevance, but differ on the context precision and context recall dimensions, pointing to the fact that there is no single automated framework at this time that can support all enterprise reliability requirements, and that a combination of the two frameworks, summarized in Table 5, is better for production monitoring.

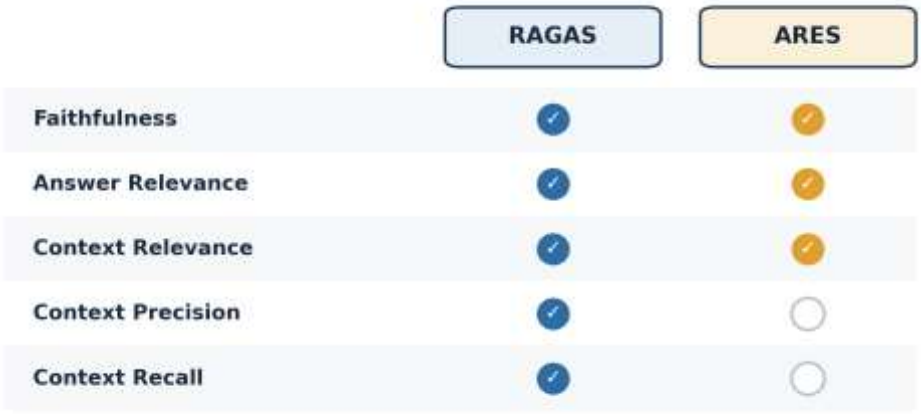

Figure 5: Coverage of Reliability Dimensions Across Two Reference-Free Evaluation Frameworks

Table 5: Coverage Comparison of Reference-Free Retrieval-Augmented Generation Evaluation Frameworks [17], [18]

Evaluation Dimension	Framework A (RAGAS)	Framework B (ARES)
Faithfulness	Assessed	Assessed
Answer relevance	Assessed	Assessed
Context relevance	Assessed	Assessed
Context precision	Assessed	Not assessed
Context recall	Assessed	Not assessed
Underlying mechanism	Large-language-model claim	Trained classifiers with prediction-

	decomposition	powered inference

Source: Compiled by the authors

3.6 Hallucination Taxonomy and Mitigation

As retrieval architectures have evolved, a body of work has emerged that characterizes hallucination in natural language generation (NLG) tasks, using either a binary classification of generated content as being unfaithful to the source or a continuum of degrees of unfaithfulness [20]. Hallucination in NLG can be divided into two categories: intrinsic hallucination, which is the generation of content that contradicts the source material, and extrinsic hallucination, which results in content that is not verifiable from the source at all. This is directly applicable to enterprise evaluation, as faithfulness scores, like those in Section 3.5, are mainly used to identify factual inaccuracies in the retrieved context and not in the underlying knowledge base.

This taxonomy was extended to a wider range of hallucination in large foundation models, encompassing multimodal and domain-specific contexts, and summarising factors that can cause hallucination such as noisy or outdated training data, exposure bias from teacher-forced training objectives, and decoding strategies that prioritise fluency over factual constraint [19]. The factors combined in Figure 6 and Table 6 suggest that hallucination can appear at various points in an enterprise retrieval-augmented system, not just in the generator, but in the retrieval corpora, which can be stale or poorly curated, indicating the need to consider retrieval quality as part of the reliability of the system rather than as a standalone preprocessing step.

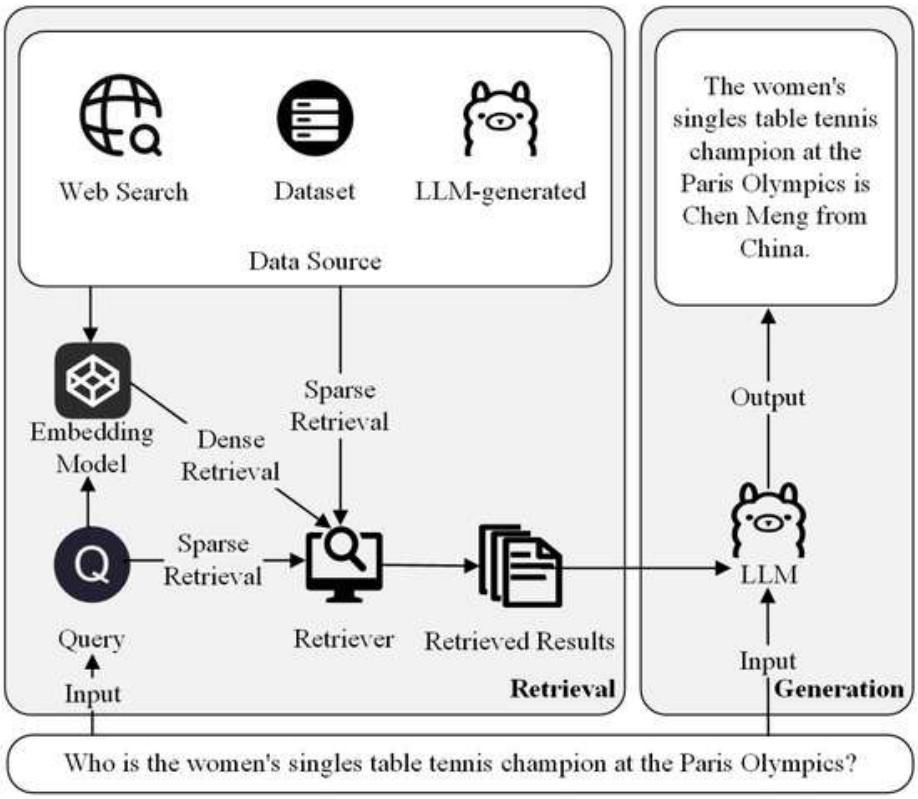

Figure 6: Taxonomy of Hallucination Causes and Corresponding Mitigation Strategies Across the Retrieval-Generation Pipeline (MDPI , 2023)

Another comprehensive survey organized hallucination mitigation efforts based on principles, taxonomy, and open challenges and suggested categorizing the strategies at the stages of the pipeline: data-level curation and filtering, training-level objective modification, and inference-level intervention (retrieval grounding, self-consistency checking, and post-hoc verification) [20]. Table 6 summarizes the various mechanisms for reliability that were reviewed in Section 3.3, and maps them onto the various types of hallucination failures that they are best suited to address, to provide a structured basis for doing so.

A focused survey of mitigation techniques identified dozens of techniques ranging from retrieval augmentation to self-refinement via iterative critique, knowledge graph grounding, and decoding time constraint satisfaction, finding that no single technique could successfully completely mitigate hallucination and that a combination of techniques (e.g., self-refensive critique tokens with retrieval grounding) yielded more consistent improvements than any single technique taken individually [21]. This aligns with the architectural positioning of the self-reflective retrieval gating and graph-structured aggregation in a single enterprise pipeline instead of considering these mechanisms as mutually exclusive.

Table 6: Taxonomy of Hallucination Causes and Corresponding Mitigation Strategies

Pipeline Stage	Representative Cause	Mitigation Strategy	Reference
Data-level	Noisy, biased, or outdated training and retrieval corpora	Corpus curation, deduplication, freshness filtering	[19]
Training-level	Exposure bias, objective-data mismatch	Objective redesign, contrastive fine-tuning	[20]
Inference-level	Decoding drift, over-confident generation	Self-consistency checks, confidence calibration	[20], [21]
Retrieval-level	Irrelevant, stale, or insufficient retrieved passages	Retrieval-necessity gating, relevance critique	[22], [21]

Source: Compiled by the authors from the reviewed literature.

3.7 Trustworthiness, Governance, and Cloud Deployment Context

In addition to factual correctness, the adoption of generative systems in the enterprise must also come with a set of more comprehensive assurances of trustworthiness, such as resistance to adversarial or out-of-distribution inputs, fairness across different populations of users, transparency on the model's reasoning, and accountability for wrong answers. An inventory and evaluation framework of reliable large language models was developed to structure the concerns into a set of alignment dimensions and benchmark categories, suggesting that the evaluation of trustworthiness should be a continuous governance process rather than a one-time certification, especially for systems used in regulated enterprise settings [23]. This view is reflected in the proposed reliability-and-evaluation layer for automated faithfulness scoring that is placed in parallel with the human review checkpoints in Figure 7, not as a replacement for the latter.

A complementary study that brought in the formal verification and validation (FVM) perspective on LMS safety concluded that trustworthiness assurance for high-stakes deployments requires that the existence of trustworthy requirements traceability, systematic testing against

adversarial and edge cases, and continuous monitoring after deployment, should be based on existing software and systems engineering practices, not only on the pre-deployment benchmark performance [24]. When applied to the enterprise context, this adds a governance and access-control layer above the orchestration and retrieval layers shown in Figure 7, adding audit logging, identity and access management, and data-residency controls as appropriate to platforms like Amazon Bedrock that expose managed model endpoints and offload data-governance responsibility to the deploying enterprise.

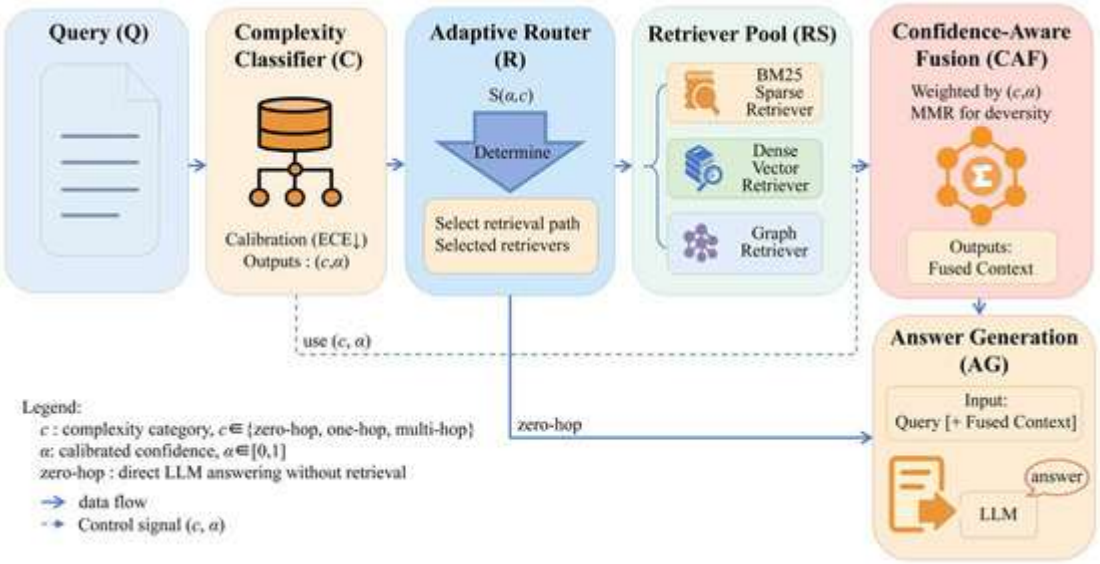

Figure 7: Layered Reliability-Aware Enterprise Architecture for Managed Cloud Retrieval-Augmented Generation Deployment (MDPI, 2021)

3.8 Comparative Synthesis

In Table 7, the twenty-five reviewed sources are grouped by year of publication, technical focus, and type of contribution, showing a temporal evolution from foundational representation learning architectures to the initial formulation of retrieval-augmented generation (RAG) to reliability-oriented refinements in the latest publication years [24]. Reviewed literature suggests that enterprise-grade reliability will not come from a single architectural innovation but will, instead, be the result of orchestrating dense and graph-structured retrieval, retrieval-necessity gating, reference-free evaluation, and governance-layer controls into a single audit trail, which reflects the KnowledgeTrust-RAG framework proposed in this review.

Some of the shortcomings of the reviewed literature should be recognized. The reported performance gains found per architecture have been achieved under heterogeneous test conditions that are not directly applicable to proprietary enterprise corpora, and comparisons of the coverage of evaluation frameworks are based on the dimensions reported by the frameworks themselves and not a head-to-head comparison of enterprise testing [24]. Empirical studies on the implementation of the KnowledgeTrust-RAG framework at scale, such as Amazon Bedrock, would then be required to confirm the trust and reliability improvements predicted by the literature review.

Table 7: Chronological Synthesis of the Twenty-Five Reviewed Sources

No.	Year	Primary Technical Focus	Contribution Type

1	2001	Conceptual foundations of knowledge management systems	Foundational theory
2	2024	Large language models in professional and business work	Editorial synthesis
3	2024	Retrieval-augmented generation research landscape	Survey
4	2024	Evolution and future directions of retrieval-augmented generation	Survey
5	2023	Retrieval-augmented generation for large language models	Survey
6	2017	Self-attention sequence architecture	Foundational architecture
7	2019	Bidirectional pre-trained language representations	Foundational architecture
8	2019	Siamese sentence-embedding networks	Foundational architecture
9	2020	Dense passage retrieval for open-domain question answering	Original method
10	2020	Retrieval-augmented language model pre-training	Original method
11	2020	Retrieval-augmented generation for knowledge-intensive tasks	Original method
12	2024	Retrieval-augmented text generation taxonomy	Survey
13	2023	Self-reflective, on-demand retrieval and critique	Original method
14	2024	Graph-structured, query-focused summarization retrieval	Original method
15	2024	Lightweight dual-level graph retrieval	Original method
16	2024	Roadmap for unifying language models and knowledge graphs	Survey / roadmap
17	2024	Enterprise deployment barriers for retrieval-augmented generation	Applied analysis
18	2023	Automated reference-free evaluation of retrieval-augmented generation	Evaluation framework
19	2023	Automated evaluation via trained classifiers and calibration	Evaluation framework
20	2023	Survey of hallucination in natural language generation	Survey
21	2023	Survey of hallucination in large foundation models	Survey
22	2024	Principles, taxonomy, and challenges of hallucination	Survey
23	2024	Comprehensive survey of hallucination mitigation techniques	Survey

24	2024	Trustworthy large language model alignment evaluation	Survey / guideline
25	2024	Verification and validation of large language model safety	Survey

Source: Compiled by the authors from the reviewed literature.

CONCLUSION

In this review, we have compiled 25 sources ranging from basic neural architectures to the development of retrieval-augmented generation, reliability-aware retrieval variants, automated evaluation methods, and parallel retrieval studies on hallucination and trustworthiness to develop KnowledgeTrust-RAG, a conceptual framework for reliability-aware retrieval in enterprise knowledge management. It shows that reliability in enterprise retrieval-augmented generation is not a feature of any individual model or algorithm, but an emergent property of architectural composition, in which dense semantic retrieval, retrieval-necessity gating, graph-structured global aggregation, reference-free evaluation, and governance-layer controls are all integrated into a single auditable pipeline. The proposed layered architecture, described as the deployment scenario of a managed cloud generative artificial-intelligence service, like Amazon Bedrock, provides a blueprint for organizations to follow when converting the previously discussed technical innovations into systems that meet enterprise needs for accuracy, provenance, and compliance. Future work should focus on empirical validation of the proposed framework on production enterprise workloads, monitoring of reliability metrics over time as knowledge bases grow, and tighter coupling between automated evaluation and human governance processes, to make it possible to replicate reliability improvements achieved in academic benchmarks under realistic operational conditions.

ACKNOWLEDGEMENTS

The author(s) gratefully acknowledge the authors of the twenty-five reviewed sources for the foundational and applied research on which this synthesis is based. No specific external funding was received for this review.

REFERENCES

[1] M. Alavi and D. E. Leidner, "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Q.*, vol. 25, no. 1, pp. 107–136, 2001, doi: 10.2307/3250961.

[2] Ş. E. Şeker, "Editorial: Large Language Models in Work and Business," *Front. Artif. Intell.*, vol. 7, 2024, doi: 10.3389/frai.2024.1516832.

[3] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make Your LLMs Use External Data More Wisely," 2024. doi: 10.48550/arXiv.2409.14924.

[4] S. Gupta, R. Ranjan, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," 2024. doi: 10.48550/arXiv.2410.12837.

[5] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," 2023. doi: 10.48550/arXiv.2312.10997.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[7] V. Karpukhin *et al.*, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2020, pp. 6769–6781. doi: 10.18653/v1/2020.emnlp-main.550.

[8] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 3929–3938. doi: 10.48550/arXiv.2002.08909.

[9] P. Lewis *et al.*, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems 33*, 2020. doi: 10.48550/arXiv.2005.11401.

[10] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," 2023. doi: 10.48550/arXiv.2310.11511.

[11] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "LightRAG: Simple and Fast Retrieval-Augmented Generation," in *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025, pp. 10746–10761. doi: 10.18653/v1/2025.findings-emnlp.568.

[12] A. Vaswani *et al.*, "Attention Is All You Need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008. doi: 10.48550/arXiv.1706.03762.

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," in *Proceedings of EMNLP-IJCNLP 2019*, 2019, pp. 3982–3992. doi: 10.18653/v1/D19-1410.

[14] D. Edge *et al.*, "From Local to Global: A Graph RAG Approach to Query-Focused Summarization," 2024. doi: 10.48550/arXiv.2404.16130.

[15] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying Large Language Models and Knowledge Graphs: A Roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, 2024, doi: 10.1109/TKDE.2024.3352100.

[16] T. Bruckhaus, "RAG Does Not Work for Enterprises," 2024. doi: 10.48550/arXiv.2406.04369.

[17] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAs: Automated Evaluation of Retrieval Augmented Generation," in *Proceedings of the 18th EACL: System Demonstrations*, 2024, pp. 150–158. doi: 10.18653/v1/2024.eacl-demo.16.

[18] J. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, "ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems," in *Proceedings of NAACL 2024*, 2024, pp. 338–354. doi: 10.18653/v1/2024.naacl-long.20.

[19] V. Rawte, A. Sheth, and A. Das, "A Survey of Hallucination in Large Foundation Models," 2023. doi: 10.48550/arXiv.2309.05922.

[20] L. Huang *et al.*, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, 2025, doi: 10.1145/3703155.

[21] S. M. T. I. Tonmoy *et al.*, "A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models," 2024. doi: 10.48550/arXiv.2401.01313.

[22] Z. Ji *et al.*, "Survey of Hallucination in Natural Language Generation," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–38, 2023, doi: 10.1145/3571730.

[23] Y. Liu *et al.*, "Trustworthy LLMs: A Survey and Guideline for Evaluating Large Language Models' Alignment," 2024. doi: 10.48550/arXiv.2308.05374.

[24] X. Huang *et al.*, "A Survey of Safety and Trustworthiness of Large Language Models through the Lens of Verification and Validation," *Artif. Intell. Rev.*, vol. 57, 2024, doi: 10.1007/s10462-024-10824-0.