International Journal on Computational Modelling Applications
Vol. 03, Iss. 01, S. No. 02, pp. 15--28, March 2026
ISSN (Online): 3048-8516

Received: 02 Feb 2026, Accepted: 01 March 2026, Published: 20 April 2026
Digital Object Identifier: https://doi.org/10.63503/j.ijcma.2026.235

Review Article

A Survey on Graph-Based Retrieval-Augmented Generation: Architectures, Methods, and Applications

Tanay Chowdhury

Data Science Lead – Gen AI Center of Innovation, Amazon Web Services

Seattle, USA

tanayz@outlook.com

*Corresponding author: Tanay Chowdhury, tanayz@outlook.com

ABSTRACT

Retrieval-Augmented Generation (RAG) stands out as a promising paradigm that can be applied to improve large language models by serving text generation with external sources of knowledge. Nevertheless, traditional RAG models are mostly based on flat text corpora and similarity-based retrieval and thus are not able to realize intricate relations, multi-hop dependencies, and structured semantics. The survey provides a high-level introduction to graph-based Retrieval-Augmented Generation (Graph-RAG), which is an emerging methodology adding systematic knowledge representations to the retrieval and generation process. Graph-RAG supports relational knowledge reasoning, contextual cohesion and explicit dependency modeling, complementary to text-only retrieval by organizing knowledge into graphs e.g., knowledge graphs, citation networks, and semantic graphs. The survey discusses the principles underlying RAG, gives a description of the weaknesses of the vector-based and token-based retrieval frameworks, and explains how the graph-aware architecture overcomes these weaknesses. Graph-RAG methods are systematically classified based on the architectural design options, retrieval schemes, learning methods, and reasoning methods, such as graph neural networks, hybrid graph-vector retrieval, and multi-hop inference. The main areas of application including healthcare, finance, and scientific question answering are discussed, and their areas are improved in factual grounding, interpretability, and robustness. Lastly, there are open issues that are concerned with scalability, graph dynamic updates, interpretability, and evaluation.

Keywords: *Retrieval-Augmented Generation, Graph-RAG, Knowledge Graphs, Hybrid Retrieval, Explainable AI.*

1. Introduction

Retrieval-Augmented Generation (RAG) has become a potent paradigm of augmenting large language models (LLMs) with external sources of knowledge to ground generation. In the traditional RAG models, models usually make use of flat text corpora and similarity-based retrieval, in which the relevant documents are brought in through dense or sparse representations and are then injected into the generation mechanism [1]. Although useful in most open-domain tasks, these methods typically have difficulty in eliciting the more complex relationship, multi-hop dependencies and structured semantics of real-world knowledge. The weaknesses of unstructured retrieval, especially the problem of factual inconsistency, the depth and breadth of reasoning, and contextual coherence, have become more visible as LLMs are used in high-stakes contexts, and they follow the same pattern of increasing weakness with high stakes.

Graph-based retrieval has been given a lot of focus to solve these issues as a modification of traditional RAG pipelines. Graph-based RAG makes it possible to have a structured reasoning regarding entities, relations, and contextual links, by modelling knowledge as graphs, including knowledge graphs, citation networks, document graphs, or semantic graphs [2]. The graphs offer a clear way of capturing dependencies, allow multi-hop traversal, and retain global information in the course of retrieval. Graph-aware retrieval can be used together with generative models to achieve higher relevance of the evidence retrieved, as well as improve interpretability and controllability of the generation process. This turn is indicative of a larger tendency to integrate structured knowledge, in which symbolic representations are added to the probabilistic model of neural reasoning.

More recent innovations have experimented with a wide range of design options in graph-based RAG models such as graph construction algorithms, graph neural networks (GNNs) to learn representations, hybrid dense-symbolic retrieval, and iterative logic. Subgraph extraction, path-based reasoning and query-sensitive graph traversal techniques enable systems to select evidence dynamically to match underlying reasoning needs of a query [3]. Moreover, graph-enhanced prompting and planning-based generation have been shown to perform better on duplicability-related tasks, long-context reasoning, and domain-specific reasoning. Regardless of these optimistic trends, the design space of graph-based RAG is still not unified, and there are different assumptions of the structure of graphs, factors of scalability, and the level of integration with LLMs.

In this regard, this survey gives a detailed and systematic overview of graph-based Retrieval-Augmented Generation with respect to architectures, methods, and applications. To classify available techniques systematically according to the type of graphs, retrieval operations, and generation methods with emphasizing on their advantages and disadvantages. Also, to review some of the most important application domains such as question answering and recommendation systems to the scientific literature analysis and cybersecurity, in which graph-based RAG has demonstrated practical advantages. The attempt to synthesize the latest findings and define areas of unresolved issues, including scalability, dynamic graph updates, and the standardization of evaluation. This survey should provide a conceptual background as well as practical input to the field of researchers and practitioners developing the next-generation RAG systems.

The paper is structured to provide a comprehensive overview of RAG and its graph-based extensions. Section 2 discusses the fundamentals of RAG, while Section 3 explores architectures for Graph-Based RAG. Section 4 covers methods and learning strategies employed in these systems. Section 5 examines applications, challenges, and future directions. Section 6 provides a literature assessment of current developments, while Section 7 concludes the paper and suggests areas for further research.

2. Fundamentals of Retrieval-Augmented Generation

The concept of RAG is a paradigm that extends the abilities of LLMs to use external knowledge sources when generating. Instead of using model weights in pre-trained forms only, RAG systems combine a retrieval element that retrieves relevant passages/documents in a large knowledge base and makes them available as context when generating answers [4]. Such hybrid method allows models to produce more relevant and informative and context sensitive writing, particularly when it comes to knowledge-intensive tasks that demand up-to-date or domain-specific information. The essence of the RAG was presented, and its principles were formally established in the earlier publications, including the RAG of Knowledge-Intensive NLP Tasks, which introduced the framework of how the retrieval-based strategies could be integrated with generative models to enhance the performance of the system in several tasks, including question answering and summarization.

RAG systems generally have several retrieval techniques that are used to retrieve relevant information each with its unique nature and standard application scenarios. Such techniques are summarized in Table I, which emphasizes the differences in the methods and use.

Table 1. Retrieval Techniques Used in RAG Systems

Technique	Description	Key Characteristics	Typical Use Case
Nearest Neighbour Search	Retrieves semantically similar documents using dense vector embeddings	Embedding-based, semantic similarity, cosine or dot-product matching	Context-aware retrieval in open-domain QA
Token-Based Retrieval	Matches query tokens directly with document tokens	Keyword-based, exact or partial token matching	Precise term-based document lookup
Approximate Nearest Neighbor (ANN)	Accelerates semantic search by approximating nearest neighbors	Scalable, low-latency, graph- or tree-based indexing	Large-scale document retrieval
Retrieval-to-Generation Pipeline	Feeds retrieved documents to the generator as context	Improves factual grounding and relevance	Knowledge-intensive text generation

2.1 Vector-Indexed Approaches in RAG

Data Storage: Convert and store text data as vector embeddings.

Pre-processing: Standard NLP pre-processing (tokenization, stop word removal, etc.).

Vector Search: It is possible to use HNSW to do an efficient approximate closest neighbor search in a high-dimensional vector space.

Obtain Associated Data: Retrieve the relevant text associated with the closest embedding.

Send to LLM: Pass the retrieved text and user query to an LLM for response generation, see the process in Fig. 1.

The most popular method of creating a RAG application is this one. Within the given method, it saves the information that the user desires the LLM (Large Language Model) to read and respond to into the form of vector embeddings. They subsequently get the appropriate text in the database with the help of vector-indexed search.

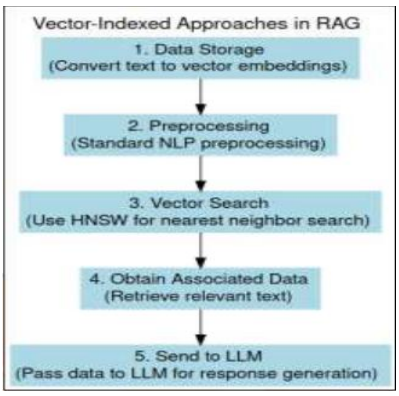

Fig. 1. Vector-Indexed Approaches in RAG.

2.2 Limitations of Vector-Based and Text-Only Retrieval

Dense embedding-based methods of retrieval, typically based on cosine similarity, have been the foundation of contemporary RAG systems [5]. Although they are good at homologizing semantic similarity, they usually prove to be shallow in matching, with a retrieved document being semantically adjacent, but factually irrelevant. Embedding-based retrieval is also disadvantaged in handling complex queries that require structured reasoning, multi-hop inference or explicit relational constraints because the rich symbolic relationships are likely to be reduced to latent spaces that are hard to interpret or manipulate. Consequently, important contextual requirements, including entity hierarchies, time associations and causal relationship are often ignored resulting in partial or inaccurate retrieval results.

The complementary limitations of text-only methods of retrieval such as the keyword-based and token-matching methods are equally important [6]. They are very sensitive to semantic similarity even though offering greater precision in finding exact matches, but are very sensitive to vocabulary mismatch, synonymy and query formulations, so do not recall semantically similar material well. In addition, text-only methods are incapable of representing explicit relationships across documents, hence do not support problems that need to reason across related knowledge, such as reason over multiple sources or domain specific expert systems. These constraints reveal a core weakness of classical retrieval models, which drives the incorporation of graph-based models that explicitly represent entities, relationships, and structural constraints to facilitate stronger, explainable, and reasoning-aware retrieval in future-generation RAG models.

3. Architectures for Graph-Based Retrieval-Augmented Generation

Graph RAG has its roots in the traditional RAG paradigm, where the external knowledge bases complement the inference of LLM by an act of retrieval [7]. But rather than considering external knowledge as a flat, unstructured text (in the form of discrete document chunks), Graph RAG structures this knowledge as graphs, i.e. networks of nodes (entities, concepts or passages) linked together by edges, showing semantic, relational or hierarchical relationships. This company has a number of benefits:

- **Some Relational Structures:** Expressive Relational Structure Relationships between many nodes expressed with multiple hops and n-ary relationships that cannot be described with isolated facts.
- **Contextual Cohesion Graph:** traversal-based retrieval offers a natural way to synthesize distributed and dependent evidence [8], avoiding context fragmentation, and facilitating deep chain-of-thought reasoning.
- **Multimodal and Domain-Specific Data:** Graph RAG frameworks are able to encode knowledge graphs, citation graphs, molecular graphs, and attributed social or tabular graphs, and provide rich domain adaptation.
- **Fewer Hallucinations:** MLM responses are grounded, which enhances the factuality and interpretability of the responses, especially in high-stakes problems.
- Graph RAG architectures consist of several steps, starting with graph construction, through to retrieval and generation, each having particular methodological variants and technologies (see Table 2).

TABLE 2. GRAPH-RAG SYSTEM ARCHITECTURE AND METHODOLOGICAL VARIANTS

Stage	Typical Methods/Technologies	Distinguishing Features
Graph Construction/Indexing	Entity/relation extraction; GNN encoding;	Heterogeneous graphs (e.g., triple-based, attributed, hierarchical, hypergraph);

	manual/LLM-driven chunking	Mu9lti-level, domain-specific, and knowledge fusion approaches
Graph-Guided Retrieval	BFS/DFS, Personalized PageRank, random walks	Retrieval of nodes, subgraphs, paths, or hyperedges relevant to the query;
	Beam search, subgraph extraction, community	supports multi-hop and dependency-aware traversal
	detection, KG traversal	
Graph-Enhanced Generation	Linearization (template, path-based),	Entity/path-based prompting, reasoning chain concatenation,
	graph summarization, evidence chain	context-aware summarization, and hybrid graph-textual context infusion
Training and Optimization	Supervised (contrastive, margin loss, etc.),	Process-constrained RL (e.g., PRA/CAF), LLM-guided retriever alignment,
	RL (policy optimization), LLM feedback	curriculum/phase-dependent reward schedules, self-distilled supervision

3.1 Comparison Between RAG and GraphRAG Architectures

The architectural variations between the normal RAG and Graph-Based Retrieval-Augmented Generation (Graph RAG). The traditional RAG pipeline includes three steps of processing: first, unstructured corpora are divided into chunks and indexed in a vector database with the help of embedding-based representations [9].

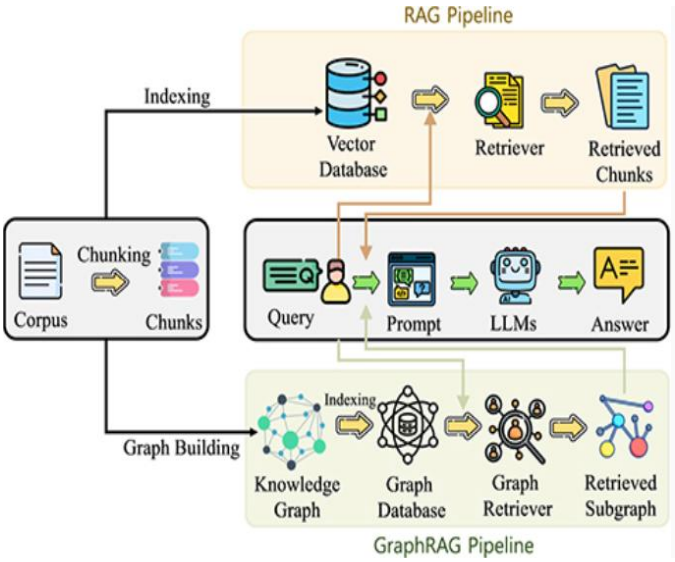

Fig. 2. Comparisons Between RAG and GraphRAG.

The user inputs a query, and the LLM generates replies by retrieving the most semantically related chunks using a vector similarity search, concatenating them with the prompt, and sending them on to the next step (Fig. 2, top pipeline). Although useful in surface-level semantic matching, in this way, retrieved content is viewed as a text fragment and therefore restricts relational reasoning and multi-hop inference.

In comparison, the Graph RAG pipeline either adds or substitutes indexing with vectors only with explicit graph building (Fig. 2, bottom pipeline). Graph RAG does not only use chunk embeddings and constructs a knowledge graph that consists of entities or concepts as nodes and their semantic, hierarchical, or relational relationships as edges [10]. The query during retrieval will activate a graph-aware retriever which is able to extract not only individual nodes but also coherent subgraphs or

relational paths that are pertinent to the query context (Fig. 2). These recalled subgraphs maintain structure-dependent relationships among entities, allowing more contextual grounding generation.

The other important difference is the very way of retrieving. Graph RAG supports multi-hop traversal, dependency-aware exploration and subgraph extraction but Standard RAG only supports one step of similarity matching retrieval. Fig. 2 illustrates that the graph retriever allows contextual expansion to similarity based on nearest-neighbor similarity by following relational edges to enable the LLM to retrieve interconnected evidence instead of fragmented pieces of evidence. This architecture specifically encourages complex reasoning tasks which involve the comprehension between multiple facts in terms of relationships.

Lastly, Graph RAG enhances the strength of generations by basing the results of LLM on the structured and interpretable evidence [11]. This enables the LLM to reason on explicit entity relationships by incorporating retrieved subgraphs into the prompt to minimize hallucinations and factual consistency over text-only retrieval in conventional RAG (Fig. 2). Consequently, Graph RAG has been shown to be more appropriate to knowledge-intensive and high-stakes applications where explain ability and relational accuracy are paramount as is the case in Table 3.

Table 3. Summary Comparison (Derived from Fig. 2)

Aspect	RAG	GraphRAG
Knowledge Representation	Unstructured text chunks	Structured knowledge graphs
Indexing	Vector database	Graph database (nodes + edges)
Retrieval Strategy	Semantic similarity search	Graph traversal and subgraph extraction
Reasoning Capability	Limited, single-hop	Multi-hop, relational reasoning
Context Quality	Fragmented chunks	Coherent, interconnected evidence
Hallucination Control	Moderate	Reduced through structured grounding

4. Methods and Learning Strategies in Graph-Based RAG

Graph-RAG systems are based on special learning strategies that can effectively use structured knowledge graphs in retrieval and generation. Graph-RAG in contrast to conventional RAG models, that mostly rely on similarity of vectors, has graph-informed learning behavior, in which relational dependencies, structural context, and multi-hop interactions amongst entities are learnt. These techniques permit better retrieval of things, logical thinking, and better grounding in facts as shown in Table IV [12]. The fundamental methods and learning strategies that form the basis of Graph-RAG systems are addressed in this section, in the case of graph neural networks, hybrid retrieval mechanisms, and reasoning-biased inference strategies.

4.1 Graph Neural Networks for Contextual Retrieval

GNNs play a pivotal role in Graph-RAG by learning expressive representations of nodes, edges, and subgraphs. By propagating information across neighboring nodes, GNNs capture both semantic features and topological structure, enabling context-aware retrieval beyond isolated entity matching. During training, node embeddings are updated through message-passing mechanisms that aggregate

information from connected entities, allowing the model to encode relational dependencies and structural importance.

Graph Neural Networks (GNNs) are central to Graph-RAG since they are able to learn expressive representations of nodes, edges and subgraphs. GNNs learn both semantic features and topological structure by transmitting information to neighboring nodes, allowing the context-sensitive retrieval of entities beyond single entity matching [13]. In the course of training, message-passing processes update node embeddings with information that other related entities provide, enabling the model to encode relational dependencies and structural significance.

4.2 Hybrid Graph–Vector Retrieval Techniques

Graph-RAG pipelines can be used to index the graph elements of a graph database with GNN-based embeddings [14]. The relevant nodes are discriminated and extended by graph traversal by utilizing learned representations at query time. It is a method that can favor dependency-conscious retrieval, meaning that the objects are not only retrieved according to the relevance of the objects themselves, but also depending on their relationship closeness and the contribution to the overall picture. Consequently, the GNN-based retrieval boosts the contextual coherence and facilitates downstream reasoning.

Most Graph-RAG systems use hybrid graph and vector retrieval methods in order to trade-off the generalization of semantics and structural accuracy [15]. Typically, in these methods, initial semantic matching of the query and candidate entities or passages is done using dense vector representations, and then the results are refined using graph-based retrieval based on explicit relational connections. This two-step retrieval scheme fuses the scalability of a vector search and the interpretability, as well as ability to reason, of graph traversal.

4.3 Reasoning, Path Selection, and Multi-Hop Inference

Hybrid retrieval pipelines tend to rank the candidates based on the product of embedding similarity scores and graph-based measures like centrality of the node, relevance of the path or strength of the connections [16]. A combination of the two representations diminishes the drawback of exclusively vector-based retrieval including fragmentation of context, but retains efficiency. Hybrid methods are especially efficient in large-scale or noisy knowledge environment, where the semantic similarity can possibly recall incomplete or weakly related evidence.

The path selection strategies can be heuristic, learning, or reinforcement learning goal-oriented with the goal of rewarding informative and short reasoning chains. Multi-hop inference enables the model to follow many relational steps, which is used to support complex queries, which need indirect evidence or dependency resolution. Graph-RAG can make the generation process more interpretable, less prone to hallucinations, and better at solving knowledge-intensive tasks by incorporating structured ways to follow reasoning into the generation process.

Table 4. Methods and Learning Strategies in Graph-Based Rag

Method Category	Techniques	Purpose in Graph-RAG	Key Advantages
Graph Neural Networks	GCN, GAT, message passing, node embedding learning	Context-aware node and subgraph representation	Captures relational structure and semantic dependencies
Hybrid Retrieval	Vector similarity + graph traversal	Joint semantic and structural retrieval	Improves recall, coherence, and retrieval precision

Path Selection	Heuristic search, learned path ranking	Identification of reasoning chains	Enables interpretable and multi-hop reasoning
Multi-Hop Inference	Graph traversal, subgraph extraction	Evidence aggregation across relations	Supports complex and dependency-aware queries
Learning Strategies	Supervised, contrastive, reinforcement learning	Optimization of retrieval and reasoning	Aligns retrieval behavior with generation objectives

5. Applications, Challenges, and Future Directions

Graph-RAG is an important extension of the original RAG, as it explicitly represents external knowledge in the form of graphs as opposed to text chunks [17]. With this structural representation, large language models can not only look in the relevant facts but also the connections and dependencies between them, which leads to the generation of more context-aware, explainable, and logically consistent. As such, Graph-RAG has found application in areas where it is critical to have multi-hop reasoning, traceability and factual reliability.

Simultaneously, the combination of graph reasoning and large language models poses new system-level and methodological problems. Scalability concerns, dynamic knowledge updates, and interpretability are a growing concern as Graph-RAG systems are used in the real-world and large-scale setting. In this section, the main domains of application (in Table V) are discussed, and then the major challenges and future research perspectives which determine the future trend of Graph-RAG are presented.

Table 5. Representative Applications of Graph-RAG

Domain	**Graph Knowledge Used**	**Typical Tasks**	**Key Advantages**
Healthcare	Ontologies, clinical entities	Medical QA, decision support	Explainability, reliability
Finance	Transaction and entity graphs	Fraud detection, compliance QA	Transparency, traceable reasoning
Scientific QA	Citation and concept graphs	Literature synthesis, hypothesis QA	Evidence-backed generation

5.1 Domain-Specific Applications

Graph-RAG has been especially useful in domain-specific applications where knowledge is relational, hierarchical, and constantly changing in nature. Through graph-based retrieval, these systems can use large language models to operate on interrelated entities instead of doing so determined by superficial semantic similarity.

Medical knowledge in healthcare is inherently organized in the form of relationships between symptom, diagnosis, treatment, and outcomes [18]. Graph-RAG systems combine biomedical ontologies, clinical guidelines and patient entities into knowledge graphs, thus making reliable multi hop reasoning and explainable medical question answering available. This systemic retrieval makes a major decrease in hallucinations and enhances clinical reliability.

Graphs can be conveniently used in the financial field where complex interactions among transactions, accounts, institutions and regulations can be modelled [19]. Graph-RAG enables the logic of transaction flows and ownership structure, which applies to fraud detection, risk analysis, compliance with regulations, and answering financial queries. Based on relational evidence, these systems enhance transparency and accountability.

The other important application area is scientific question answering. Scientific knowledge is disseminated through publications, datasets and citation networks and relational retrieval has become essential [20]. Graph-RAG models map citation graphs and relationship between concepts, allowing large language models to compose evidence on multiple studies and produce answers with a structured rationale.

Key Application Highlights

- Enables multi-hop reasoning across interconnected entities
- Improves factual grounding and explain ability
- Supports high-stakes decision-making domains
- Reduces hallucinations compared to text-only RAG

5.2 Scalability, Dynamic Graph Updates, and Interpretability

Graph-RAG presents significant computational and architectural challenges, even though it has its benefits. With larger and more intricate knowledge graphs, it becomes even harder to find useful indexing, traversal and embedding computation [21]. Graphs with large scale can have a considerable effect on retrieval latency, so scalability is a primary issue to practical implementation.

The other significant difficulty is as a result of the dynamic nature of real-world knowledge. Healthcare and finance domains are not static, and the graph structure, embeddings, and retrieval indexes need to be updated constantly. The ability to keep consistency and freshness without retraining costs is an unsolved research problem. Along with an important asset of Graph-RAG, interpretability poses challenges, as can be seen in Table VI). Even though graph-based retrieval offers clear reasoning paths, the ability to make this path intelligible and understandable to users is not an easy task. The important thing is that the mechanisms of explanation must be effective to enable user trust especially where there is the need to have safety-critical applications.

Major System Challenges

- High computational cost for large-scale graph traversal
- Difficulty in maintaining up-to-date dynamic graphs
- Limited user-friendly explanation of reasoning paths.

Table 6. Core Challenges and Research Directions

Challenge	Description	Active Research Directions
Scalability	High traversal and embedding cost	Approximate search, distributed graphs
Dynamic Updates	Continuous knowledge evolution	Incremental and streaming graph learning
Interpretability	Complex reasoning paths	Visualization and explanation layers

5.3 Open Research Challenges and Emerging Trends

In the future, there are still a number of open challenges that may influence the creation of Graph-RAG systems. One of the basic problems is how to make graph-based reasoning consistent with the internal generation processes of large language models. Recalled graph structures have to be adequately incorporated in the process of decoding so as to achieve faithful and consistent use.

The new directions are toward multimodal Graph-RAG, in which a textual, visual, numerical, and temporal information are presented in the same graph types. Also more adaptive and task-aware reasoning is being brought by learning-based retrieval strategies, including the graph-traversal reinforcement learning. Lastly, the unavailability of the standard benchmarks and evaluation protocols is a restrictive factor. As Graph-RAG is evolving, a stronger focus is being put on durability, justification, and moral aspects, making Graph-RAG an essential building block of credible and understandable AI systems.

Emerging Trends and Future Directions

- The correspondence between graph reasoning and LLM decoding.
- Multimodal and time knowledge graphs.
- Learning-based adaptive retrieval strategies.
- Unified performance standards and measures.

6. Literature Review

In this section, a brief review of recent studies on Retrieval- Augmented Generation (RAG) is given with an emphasis on approaches that incorporate external knowledge into LLMs. The major contributions and methodology of the preceding work are presented in Table 7.

Fan et al. (2024) offering extra knowledge would help RAG to support existing generative AI to create quality outputs. LLMs have recently been a game-changer for language creation and comprehension, however they still have limitations including hallucinations and outdated internal information. RA-LLMs were created to supplement the LLMs' internal knowledge with external, trustworthy sources in order to produce higher-quality content, due to RAG's ability to provide up-to-date, relevant auxiliary information [22]. Zhao et al. (2024) analyze current attempts to apply RAG techniques to AIGC situations. Instead, they group RAG foundations based on the way the retriever enhances the generator and extract the basic abstractions of the augmentation methodologies of different retrievers and generators. This integrated thinking cuts across all the RAG situations, shedding light on the developments and critical technologies that assist with the future possible developments. they also outline other improvement techniques to RAG, which made the RAG systems easy to engineer and implement. Next in another perspective, they look at the practical uses of RAG in various modalities and tasks, as a good source of reference to researchers and practitioners [23].

Sha et al. (2023) propose a new model of RAKG, resolving the above problems with two major novelties. On the one hand, they create a better subgraph of the knowledge graph by forcing the model to reason along the corrected knowledge route using the density matrix. Second, they use a GCN to merge and update two representations, one for questions and one for graph entities, and then they integrate them using a bidirectional attention technique. They used two well-known benchmark datasets, Common Sense QA and Open Book QA, to evaluate the effectiveness of their approach. The case study reveals that while answering questions, the enhanced subgraph provides reasoning on the rectified route of knowledge [17]. Gao et al. (2023) present an overview of the origins and evolution of recommender systems as well as graph neural networks. However, current studies in the field of recommender systems may be categorized according to four factors: stage, scenario, aim, and

application. The available methods for graph neural networks are separated into two categories: spectral and spatial models. They then describe the reasons for incorporating the theory of GNNs into recommender systems, which include the enhanced supervision signal, high-order connectivity, and data structural properties. The problems with propagation/aggregation, model optimization, and computational efficiency in graph creation are then carefully examined [24].

Opdahl et al. (2022) provide different origin, and have a different form; they reach a different speed, and in a different quantity. One known method of unifying such heterogeneous information is semantic knowledge graphs (KGs). As a result, it will grow in significance within the news industry and is in line with the goals of news distributors and producers. This article provides a concise overview of semantic knowledge graphs by reviewing its research in the news business and discussing its uses in news production, distribution, and consumption. The goal is to present a comprehensive overview of the issue, explore what it is, and identify opportunities and requirements for future research and progress [25]. Makarov et al. (2021) intends to explain the essence of graph embeddings and to give a number of taxonomies of their description. First, they deal with the methodology, classifying graph embedding models into three groups based on matrix factorization, random-walks, and DL. They then explain the effect of various kinds of networks on the capacity of the models to integrate a combination of structural and attributed information in a single embedding. Following this, they perform an in-depth evaluation of graph embedding's uses in graph ML, specifically in areas such as node classification, link prediction, clustering, visualization, compression, and a set of algorithms suitable for graph classification, similarity, and alignment [5].

Table 7. Comparative Analysis of RAG, Knowledge Graphs, and Graph-Based Learning Approaches

Authors & Year	**Core Focus**	**Primary Technique / Model**	**Knowledge Source / Structure**	**Key Contribution**	**Application Domain**	**Limitations / Challenges**
Fan et al., 2024	Enhancing LLM generation quality	Retrieval-Augmented Generation (RAG)	External authoritative knowledge bases	Demonstrates how external knowledge mitigates hallucinations and outdated knowledge in LLMs	Generative AI, NLP	Retrieval quality and dependency on external data freshness
Zhao et al., 2024	Systematic survey of RAG in AIGC	Unified RAG framework and taxonomy	Multiple retrievers and generators	Provides a comprehensive classification of RAG foundations, enhancements, and multimodal applications	AIGC, Multimodal AI	Engineering complexity and lack of standardized benchmarks
Sha et al., 2023	Knowledge-guided reasoning for QA	Retrieval-Augmented Knowledge Graph (RAKG) with GCN	Knowledge Graph subgraphs	Introduces corrected knowledge-path reasoning using density matrix and bidirectional attention	Question Answering (QA)	Computational overhead and graph construction dependency

Gao et al., 2023	GNN-based recommender systems	Graph Neural Networks (spectral & spatial)	User–item interaction graphs	Analyzes challenges and motivations for integrating GNNs into recommender systems	Recommendation Systems	Scalability, graph construction, and optimization issues
Opdahl et al., 2022	Semantic integration of heterogeneous data	Semantic Knowledge Graphs	News-related structured and unstructured data	Reviews KG usage across news production, distribution, and consumption	News & Media Industry	Maintenance cost and semantic alignment complexity
Makarov et al. (2021)	Graph representation learning	Graph Embedding Models (MF, RW, DL)	Attributed and structural graphs	Provides taxonomies and applications of graph embeddings across ML tasks	General Graph ML	Trade-off between expressiveness and computational efficiency

7. Conclusion and Future Work

Graph-based Retrieval-Augmented Generation (Graph-RAG) builds upon standard RAG with the difference that external knowledge is represented as structured graphs that allows relational reasoning, multi-hop inference and better contextual coherence. Going beyond flat text retrieval, Graph-RAG has addressed challenges of context fragmentation, shallow semantic matching and hallucinations, and improved interpretability and factual grounding. The architectures and learning strategies reviewed exhibit obvious advantages in the knowledge intensive fields such as: healthcare, finance, and scientific question answering, where explicit modeling of entities and relationships are critical. However, there are a number of challenges that are still not solved. A significant challenge is scalability because large, heterogeneous graphs are expensive to retrieve and compute, and dynamic knowledge updates need mechanisms capable of ensuring consistency without retraining required by them. The transparency in graph-based retrieval is enhanced; however, it is still difficult to explain complicated reasoning paths effectively to end users. It is believed that future research will involve integration of graph reasoning and language model decoding further, adaptive and learning-based graph traversal policies and multimodal and temporal information incorporated into unified graph structure. It will be extremely important to have standard benchmarks, assessment structures so as to provide a systematic comparison and to have deployable, robust, and trustworthy Graph-RAG systems.

Conflict of Interest

The authors declare no potential conflict of interest in this publication.

References

[1] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, “GRAG: Graph retrieval-augmented generation,” arXiv preprint arXiv:2405.16506, 2024.

[2] M. Arslan, S. Munawar, and C. Cruz, “Business insights using RAG–LLMs: A review and case study,” J. Decis. Syst., pp. 1–30, Oct. 2024, doi: 10.1080/12460125.2024.2410040.

[3] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 2, pp. 494–514, Feb. 2021, doi: 10.1109/TNNLS.2021.3070843.

[4] Y. Liu, X. Zhang, Y. Li, J. Zhou, X. Li, and G. Zhao, "Graph-based facial affect analysis: A review," IEEE Trans. Affect. Comput., vol. 14, no. 4, pp. 2657–2677, Oct. 2023, doi: 10.1109/TAFFC.2022.3215918.

[5] Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj, "Survey on graph embeddings and their applications to machine learning problems on graphs," PeerJ Comput. Sci., vol. 7, p. e357, Feb. 2021, doi: 10.7717/peerj-cs.357.

[6] L. Zhong, J. Wu, Q. Li, H. Peng, and X. Wu, "A comprehensive survey on automatic knowledge graph construction," ACM Comput. Surv., vol. 56, no. 4, pp. 1–62, Apr. 2023, doi: 10.1145/3618295.

[7] L. Waikhom and R. Patgiri, "Graph neural networks: Methods, applications, and opportunities," arXiv preprint arXiv:2108.10733, 2021.

[8] L. Wang, C. Sun, C. Zhang, W. Nie, and K. Huang, "Application of knowledge graph in software engineering field: A systematic literature review," Inf. Softw. Technol., vol. 164, p. 107327, Dec. 2023, doi: 10.1016/j.infsof.2023.107327.

[9] V. Hassija et al., "Interpreting black-box models: A review on explainable artificial intelligence," Cognit. Comput., vol. 16, no. 1, pp. 45–74, Jan. 2024, doi: 10.1007/s12559-023-10179-8.

[10] M. S. Wajid, H. Terashima-Marin, P. Najafirad, and M. A. Wajid, "Deep learning and knowledge graph for image/video captioning: A review of datasets, evaluation metrics, and methods," Eng. Rep., vol. 6, no. 1, Jan. 2024, doi: 10.1002/eng2.12785.

[11] M. J. Buehler, "Generative retrieval-augmented ontologic graph and multiagent strategies for interpretive large language model-based materials design," ACS Eng. Au, vol. 4, no. 2, pp. 241–277, Apr. 2024, doi: 10.1021/acsengineeringau.3c00058.

[12] L. Opdahl et al., "Semantic knowledge graphs for the news: A review," ACM Comput. Surv., vol. 55, no. 7, pp. 1–38, Jul. 2023, doi: 10.1145/3543508.

[13] X. Zhang, Y. Zhou, and J. Luo, "Deep learning for processing and analysis of remote sensing big data: A technical review," Big Earth Data, vol. 6, no. 4, pp. 527–560, Oct. 2022, doi: 10.1080/20964471.2021.1964879.

[14] V. Pal and S. K. Chintagunta, "Transformer-based graph neural networks for real-time fraud detection in blockchain networks," Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 3, pp. 1401–1411, Jul. 2023, doi: 10.48175/IJARSCT-11978Y.

[15] V. T. Hoang et al., "Graph representation learning and its applications: A survey," Sensors, vol. 23, no. 8, p. 4168, Apr. 2023, doi: 10.3390/s23084168.

[16] L. Alzubaidi et al., "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," J. Big Data, vol. 8, no. 1, p. 53, 2021, doi: 10.1186/s40537-021-00444-8.

[17] Y. Sha et al., "Retrieval-augmented knowledge graph reasoning for commonsense question answering," Mathematics, 2023, doi: 10.3390/math11153269.

[18] G. Sarraf, "Privacy preserving blockchain for healthcare: Addressing security challenges through decentralized architecture," Tech. Int. J. Eng. Res., vol. 10, no. 11, pp. 111–117, 2023, doi: 10.56975/tijer.v10i11.159993.

[19] E. Watson, T. Viana, and S. Zhang, "Augmented behavioral annotation tools, with application to multimodal datasets and models: A systematic review," AI, vol. 4, no. 1, pp. 128–171, Jan. 2023, doi: 10.3390/ai4010007.

[20] P. Chandrashekar, "A survey of tools, techniques, and best practices: CI/CD integration in DevOps workflows," Int. J. Adv. Res. Sci. Commun. Technol., vol. 3, no. 3, pp. 1366–1376, Jul. 2023, doi: 10.48175/IJARSCT-11978V.

[21] S. Garg, “Predictive analytics and auto remediation using artificial intelligence and machine learning in cloud computing operations,” Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci., vol. 7, no. 2, 2019.

[22] W. Fan et al., “A survey on RAG meeting LLMs: Towards retrieval-augmented large language models,” in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2024, pp. 6491–6501, doi: 10.1145/3637528.3671470.

[23] P. Zhao et al., “Retrieval-augmented generation for AI-generated content: A survey,” arXiv preprint, Jun. 2024.

[24] C. Gao et al., “A survey of graph neural networks for recommender systems: Challenges, methods, and directions,” ACM Trans. Recomm. Syst., vol. 1, no. 1, pp. 1–51, Mar. 2023, doi: 10.1145/3568022.

[25] L. Opdahl et al., “Semantic knowledge graphs for the news: A review,” ACM Comput. Surv., vol. 55, no. 7, pp. 1–38, Jul. 2022, doi: 10.1145/3543508.