Proceedings of CONF-CDS 2026 Symposium: Machine Learning and Neural Network Applications in Engineering
DOI: 10.54254/2755-2721/2026.AST36546

Application of Knowledge Graph-Based GraphRAG in Intelligent Question Answering Systems

Fanhao Zhou

Faculty of Innovation Engineering, Macau University of Science and Technology, Macao, China
2902935801@qq.com

Abstract. Large language models (LLMs) are developing rapidly and have been widely applied in intelligent question answering, knowledge retrieval, education, healthcare, enterprise services, and other fields. However, LLMs still exhibit limitations in knowledge updating, understanding complex relationships, and tracing answer sources. This paper adopts a literature review approach to survey knowledge graph-enhanced GraphRAG. It analyzes the indexing, retrieval, and generation workflows, as well as key techniques including graph traversal, community summarization, hybrid retrieval, and graph-augmented generation, and discusses applications in medical question answering and enterprise knowledge base question answering. Existing studies indicate that GraphRAG can use entity relations and hierarchical summaries to mitigate the fragmented context, limited multi-hop reasoning, and insufficient global information coverage of conventional RAG. Nevertheless, challenges remain in dynamic graph maintenance, semantic alignment, cost, and governance for trustworthiness.

Keywords: Knowledge Graph, GraphRAG, Intelligent Question Answering, Retrieval-Augmented Generation

1. Introduction

With the widespread adoption of large language models (LLMs) in intelligent question answering, knowledge retrieval, and knowledge-intensive natural language processing tasks, improving answer accuracy, interpretability, and knowledge updatability has become a key concern in artificial intelligence research. Although traditional LLMs can store substantial factual knowledge in pretrained parameters, their ability to precisely access and manipulate external knowledge and trace sources remains limited. Lewis et al. proposed retrieval-augmented generation (RAG), which combines a parametric language model with an external non-parametric knowledge base and enables the model to retrieve relevant documents before generating an answer. This improves factuality and traceability in open-domain question answering and knowledge-intensive tasks [1]. However, conventional RAG typically relies on flat retrieval of text chunks and may still suffer from fragmented context and insufficient information coverage when handling complex entity relations, multi-hop reasoning, and global questions.

Edge et al. proposed GraphRAG, which constructs an entity knowledge graph from source documents and generates summaries for communities of closely related entities. This approach

DOI: 10.54254/2755-2721/2026.AST36546

better supports global questions about an entire corpus, including thematic synthesis, viewpoint aggregation, and complex information integration [2]. Guo et al. later introduced LightRAG, which combines graph structure with vector representations. Its dual-level retrieval mechanism supports both low-level entity-relation retrieval and high-level knowledge discovery, while incremental updates improve efficiency in dynamic knowledge environments [3]. These studies indicate that graph structure is becoming an important technical direction for improving RAG systems.

This paper presents a literature review of the application of GraphRAG in intelligent question answering systems, focusing on its basic architecture, key technical methods, application scenarios, and future challenges. By synthesizing existing research, this paper aims to summarize the advantages and limitations of GraphRAG relative to conventional RAG and to provide a reference for the design and optimization of future intelligent question answering systems.

2. Overview of GraphRAG

2.1. Basic concepts and overall architecture

A knowledge graph represents entities or concepts as nodes and their semantic relationships as edges. It can also record attributes, provenance, and temporal information, thereby forming a computable knowledge network [4]. The conventional RAG pipeline comprises query encoding, text retrieval, context assembly, and answer generation. It is simple to implement and flexible in response to document updates, but its retrieval units are typically independent text chunks [5]. The central change introduced by GraphRAG is the integration of graph structure into indexing and retrieval. The system therefore not only identifies semantically similar passages but also organizes evidence through entity neighborhoods, relation paths, community structures, and hierarchical summaries.

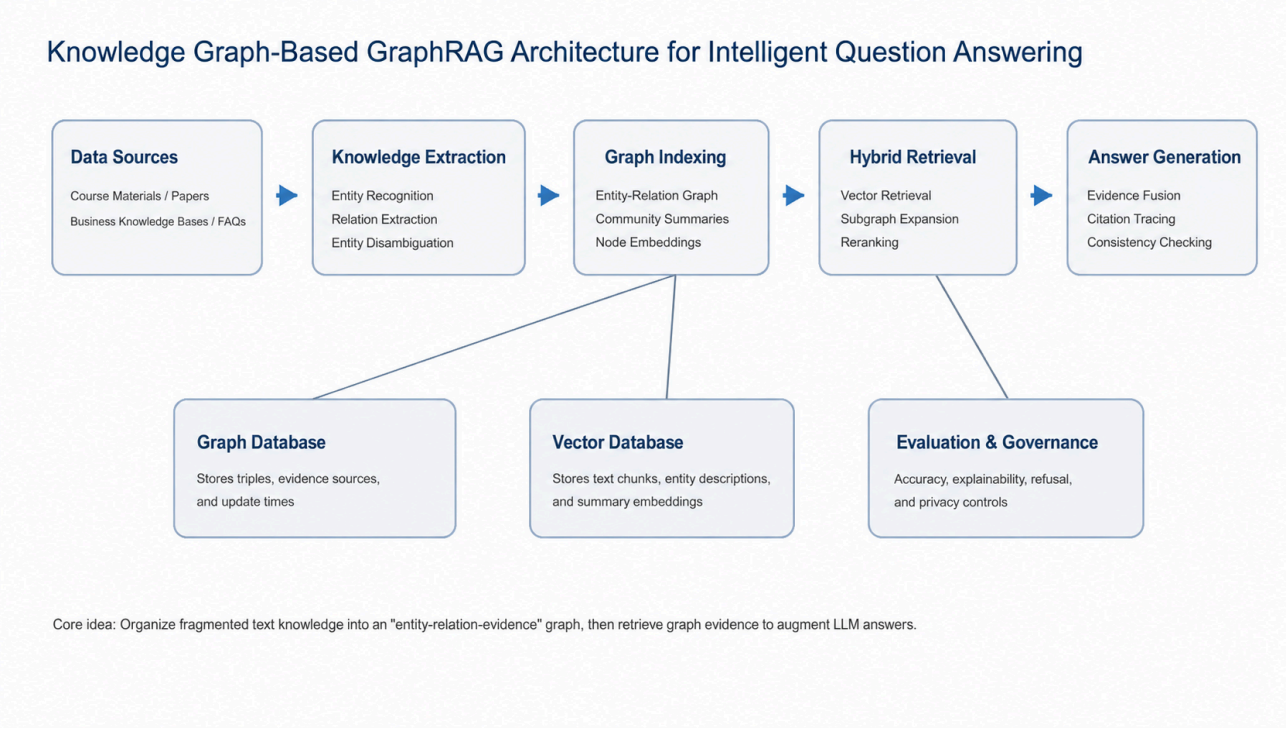

Figure 1. Knowledge graph-based GraphRAG architecture for iIntelligent question answering [2, 3]

DOI: 10.54254/2755-2721/2026.AST36546

GraphRAG refers to a class of graph-enhanced frameworks. Microsoft GraphRAG emphasizes community summaries, whereas LightRAG balances local detail and global information and supports incremental updates [2, 3]. Both must integrate with document parsing, vector indexing, graph databases, prompt construction, and evaluation modules, as shown in Figure 1. The index should maintain provenance for traceability and define a graph schema to constrain arbitrary extraction by the LLM. Rather than attempting to cover all knowledge at once, the schema should be expanded gradually around frequently asked questions to control complexity.

2.2. Indexing stage

During indexing, the system first parses, cleans, and segments PDFs, web pages, or business documents while retaining metadata such as file names, page numbers, paragraph locations, and update times. Named entity recognition and relation extraction are then used to obtain entity-relation-entity triples, followed by entity disambiguation, synonym merging, and conflict detection. The graph database stores entities, relations, and their evidence sources, while the vector database stores text chunks, entity descriptions, and summary embeddings. Dense vector retrieval can employ the query-passage encoding approach proposed in Dense Passage Retrieval [6]. Microsoft GraphRAG also applies community detection to the entity graph and generates summaries at multiple community levels. The Leiden algorithm can produce well-connected communities and therefore provides a suitable foundation for graph partitioning [7].

2.3. Retrieval stage

During retrieval, the system first identifies the entities, topics, and intent in a question and then selects a local or global retrieval route. For local questions, candidate subgraphs are typically extracted from one- or multi-hop neighborhoods of linked entities under constraints such as relation type and edge weight. Path scores, textual similarity, and evidence reliability are then combined for ranking. For global questions, the system retrieves multiple topic-related community summaries and aggregates them hierarchically. Multi-step questions can use an iterative retrieval-reasoning-retrieval process, allowing intermediate reasoning results to guide the next round of evidence acquisition [8]. Consequently, GraphRAG can retrieve not only text chunks but also entities, paths, subgraphs, and community summaries.

2.4. Generation stage

During generation, the user question, textual evidence, relation paths, or community summaries are organized into a context that an LLM can interpret. Local answers should retain key triples and their provenance to support fact checking. Global answers can adopt a map-reduce strategy by first generating partial answers from multiple community summaries and then combining them into an overall conclusion [2]. Prompts should explicitly instruct the model to answer only from the supplied evidence, cite sources, and abstain when evidence is insufficient. The retrieval, generation, and self-reflection principles embodied in Self-RAG indicate that outputs should be checked for relevance and factual support rather than equating fluency as correctness [9].

DOI: 10.54254/2755-2721/2026.AST36546

3. Key methods and technologies

3.1. Graph traversal-based retrieval

Graph traversal-based retrieval requires determining starting nodes, expansion scope, and ranking rules. Entity linking identifies the starting nodes, after which neighborhoods are expanded according to relation type, time range, path length, and access constraints. Path ranking can combine vector similarity between the query and node descriptions, edge confidence, node centrality, and source quality, while subgraph extraction must balance information coverage against context length. For complex questions, the system can first identify shortest or high-confidence paths connecting multiple query entities and then supplement them with nearby textual evidence. This process improves relational interpretability, but erroneous edges can propagate noise along incorrect paths, making reranking and evidence verification necessary.

3.2. Graph-augmented generation

The most direct graph-augmented generation method serializes a subgraph as text, for example by using an entity-relation-entity-source format in the prompt. This approach does not require modifying the LLM architecture, but large subgraphs can create contextual redundancy. Another approach first uses a graph neural network to learn node representations and then applies graph vectors to retrieval, reranking, or fusion with text representations. GCN, GraphSAGE, and GAT offer neighborhood aggregation, inductive sampling, and attention-weighted aggregation [10-12], while Graph Transformers help combine local message passing with global dependencies [13]. However, these models are optional enhancement modules rather than required components of Microsoft GraphRAG or LightRAG. For general question answering systems, auditable graph serialization and hybrid retrieval are often easier to deploy than end-to-end training of complex graph models.

4. Application scenarios

4.1. Medical question answering

Medical knowledge involves numerous entity types, strong relational constraints, and strict evidentiary requirements. GraphRAG can organize diseases, symptoms, tests, medications, contraindications, and clinical guidelines into a graph and retrieve relation paths relevant to a patient's question. For example, when asked why a medication is unsuitable for a particular comorbidity, the system can jointly retrieve drug actions, disease mechanisms, contraindication relations, and guideline sources rather than returning passages that merely share keywords. Studies such as QA-GNN have shown that joint reasoning over language models and knowledge graphs can improve knowledge-intensive question answering [14]. Nevertheless, medical applications require source grading, version control, privacy safeguards, and human review. System outputs should serve only as informational assistance and must not replace professional diagnosis.

4.2. Enterprise and research knowledge base question answering

Enterprise knowledge is distributed across policy documents, project reports, and product manuals. Conventional RAG can confuse entities with identical names, encounter conflicts across document versions, and miss cross-departmental relationships. GraphRAG can connect people, projects,

DOI: 10.54254/2755-2721/2026.AST36546

products, processes, and document sources to answer questions such as "Which materials support this decision?" and "What are the dependencies between product solutions?" Similarly, course materials can be organized into a concept graph for concept comparison and research-roadmap analysis. In such settings, the key value lies not only in improving recall but also in generating a traceable chain of knowledge.

For deployment, organizations should begin with a clearly bounded knowledge domain, such as a course syllabus and a designated set of papers. A small manually verified graph and a benchmark question set can then be built and compared with conventional RAG in terms of retrieval hit rate, evidence completeness, and response time. If structured relations genuinely improve comparative, multi-hop, and summarization questions, the knowledge sources can be expanded gradually. This staged approach reduces the risk of constructing a large graph all at once and makes error sources easier to locate.

5. Challenges and future directions

5.1. Key challenges

First, large-scale dynamic graphs require frequent additions or corrections to entity relations. Complete index rebuilding is expensive, whereas excessive compression may discard critical evidence. Second, a semantic gap exists between natural language questions and graph structures. The same question may correspond to different entities, relations, and retrieval scopes, so entity-linking or routing errors directly affect answers. Third, entity extraction, community summarization, multi-channel retrieval, and LLM generation incur substantial computational and API costs. Fourth, errors, biases, access-controlled information, and outdated relations may propagate through the graph structure. Accordingly, evaluation must address source reliability, privacy, and abstention in addition to accuracy.

5.2. Future directions

Future improvements can be pursued along four directions. First, hierarchical graph indexing, incremental community updates, and hot/cold data tiering can balance retrieval efficiency and knowledge completeness. Second, joint learning of text and graph representations, together with query decomposition and interpretable routing, can improve semantic alignment. Third, small-model extraction, cached community summaries, on-demand graph expansion, and tiered invocation of LLMs can reduce cost. Fourth, end-to-end evaluation systems should cover graph quality, retrieval paths, answer evidence, safety, and compliance. Public benchmarks for Chinese-domain materials are particularly needed so that GraphRAG systems can be compared on unified question sets and evidence standards.

Evaluation should also distinguish local factual question answering from global synthesis questions. The former can use recall, path hit rate, and factual consistency, whereas the latter often lacks a single standard answer and is better assessed through topic coverage, information diversity, citation sufficiency, and human evaluation. Future benchmarks should also record index construction cost, per-query latency, and knowledge update cost, so that answer-quality comparisons do not overlook the sustainability of system operation.

DOI: 10.54254/2755-2721/2026.AST36546

6. Conclusion

This paper presents a literature review of the application of knowledge graph-based GraphRAG in intelligent question answering systems. Existing research indicates that GraphRAG extends the retrieval units of conventional RAG through entity relations, subgraph paths, and community summaries. It can mitigate information fragmentation in complex-relation questions, multi-hop reasoning, and global topic synthesis while improving answer interpretability and source traceability. GraphRAG shows potential in medical, enterprise, and research knowledge question answering, but its practical effectiveness depends heavily on graph quality, retrieval routing, and evidence governance. As this paper is a review and framework analysis and does not include quantitative experiments on a unified dataset, its conclusions still require validation through prototype systems and comparative evaluation. Future research should prioritize incremental maintenance of dynamic graphs, text-graph semantic alignment, low-cost inference, and trustworthy evaluation, enabling GraphRAG to progress from methodological exploration toward stable and auditable real-world applications.

References

[1] Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474.
[2] Edge, D., Trinh, H., Cheng, N., et al. (2024). From local to global: A graph RAG approach to query-focused summarization. arXiv preprint arXiv:2404.16130.
[3] Guo, Z., Xia, L., Yu, Y., et al. (2024). LightRAG: Simple and fast retrieval-augmented generation. arXiv preprint arXiv:2410.05779.
[4] Hogan, A., Blomqvist, E., Cochez, M., et al. (2021). Knowledge graphs. *ACM Computing Surveys, 54*(4), 1–37.
[5] Gao, Y., Xiong, Y., Gao, X., et al. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.
[6] Karpukhin, V., Oguz, B., Min, S., et al. (2020). Dense passage retrieval for open-domain question answering. *Proceedings of EMNLP*, 6769–6781.
[7] Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports, 9*, 5233.
[8] Trivedi, H., Balasubramanian, N., Khot, T., et al. (2023). Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *Proceedings of ACL*, 10014–10037.
[9] Asai, A., Wu, Z., Wang, Y., et al. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *Proceedings of ICLR*.
[10] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *Proceedings of ICLR*.
[11] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems, 30*.
[12] Velickovic, P., Cucurull, G., Casanova, A., et al. (2018). Graph attention networks. *Proceedings of ICLR*.
[13] Rampasek, L., Galkin, M., Dwivedi, V. P., et al. (2022). Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems, 35*, 14501-14515.
[14] Yasunaga, M., Ren, H., Bosselut, A., et al. (2021). QA-GNN: Reasoning with language models and knowledge graphs for question answering. *Proceedings of NAACL-HLT*, 535-546.