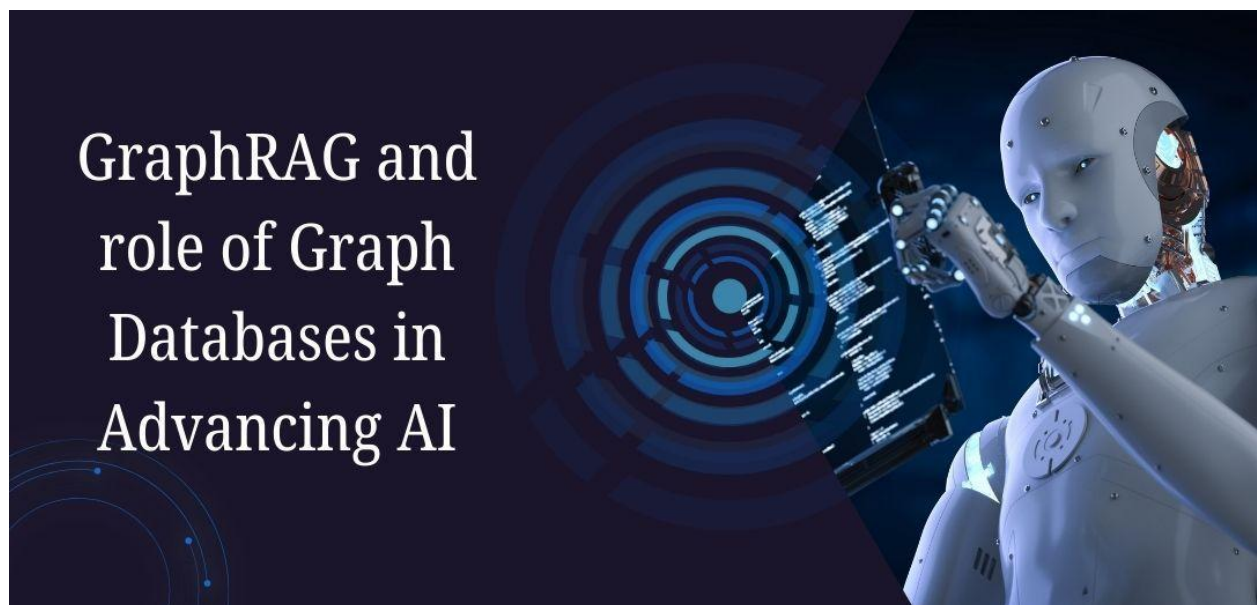




GRAPHRAG AND ROLE OF GRAPH DATABASES IN ADVANCING AI

Ravi Kiran Magham

Osmania University, India



ABSTRACT

This article explores the synergy between GraphRAG and graph databases in advancing AI capabilities. Graph RAG, a novel approach to query-focused summarization that combines knowledge graph generation, retrieval-augmented generation (RAG), and query-focused summarization (QFS) to support human sensemaking over large text corpora. While traditional RAG methods excel at answering specific questions, they struggle with global queries about entire datasets. Graph RAG addresses this limitation by using an LLM to build a graph-based text index in two stages: first deriving an entity knowledge graph from source documents, then pre-generating community summaries for groups of related entities. Given a query, each community summary generates a partial response, which is then synthesized into a final answer.

Evaluations on datasets in the 1 million token range show that Graph RAG substantially improves the comprehensiveness and diversity of answers compared to naïve RAG baselines, while also demonstrating favorable performance against global text summarization approaches. The hierarchical nature of the graph index allows for efficient querying at different levels of granularity. This approach offers a scalable solution for global sensemaking tasks over large private document collections, with potential applications in scientific discovery, intelligence analysis, and other domains requiring complex information synthesis.

Keywords: GraphRAG, Graph Databases, Artificial Intelligence, Knowledge Representation, Large Language Models

Cite this Article: Ravi Kiran Magham. (2024). Graphrag and Role of Graph Databases in Advancing AI. International Journal of Research in Computer Applications and Information Technology (IJRCAIT), 7(2), 98-110.

https://iaeme.com/MasterAdmin/Journal_uploads/IJRCAIT/VOLUME_7_ISSUE_2/IJRCAIT_07_02_007.pdf

I. INTRODUCTION

The landscape of Artificial Intelligence (AI) is rapidly evolving, with recent advancements in large language models (LLMs) and retrieval-augmented generation (RAG) pushing the boundaries of what's possible in natural language processing (NLP) tasks. Concurrently, graph databases have emerged as a powerful tool for representing and querying complex, interconnected data. The fusion of these technologies – which we term GraphRAG – presents a promising frontier in AI research and applications.

Large language models, such as those described by Brown et al. [1], have demonstrated remarkable few-shot learning capabilities, allowing them to adapt to new tasks with minimal examples. However, these models often struggle with knowledge-intensive tasks that require access to large, structured datasets. To address this limitation, Lewis et al. [2] introduced the concept of retrieval-augmented generation, which combines the generative power of language models with the ability to retrieve and incorporate external knowledge.

While RAG has shown significant promise, there's still room for improvement, particularly in handling complex, interconnected information. This is where graph databases come into play. As explained by Robinson et al. [3], graph databases excel at managing and querying highly connected data, making them ideal for representing the intricate relationships often found in real-world knowledge bases.

The concept of GraphRAG builds upon these foundations, leveraging the strengths of graph databases to enhance the retrieval and reasoning capabilities of RAG systems. By representing knowledge in a graph structure, we can capture nuanced relationships between entities and concepts, potentially leading to more accurate and contextually relevant information retrieval.

Moreover, the integration of graph databases into AI systems opens up new possibilities for explainable AI. The explicit representation of relationships in a graph allows for easier tracing of reasoning paths, providing insights into how an AI system arrives at its conclusions. This transparency is crucial for building trust in AI systems, especially in high-stakes domains like healthcare or finance.

As we delve deeper into the synergies between graph databases and advanced AI techniques like RAG, we stand at the cusp of a new era in artificial intelligence. GraphRAG has the potential to not only improve the performance of AI systems on knowledge-intensive tasks but also to enhance their interpretability and reliability.

In the following sections, we will explore the theoretical foundations of GraphRAG, discuss its potential applications, and examine the challenges and opportunities that lie ahead in this exciting field of research.

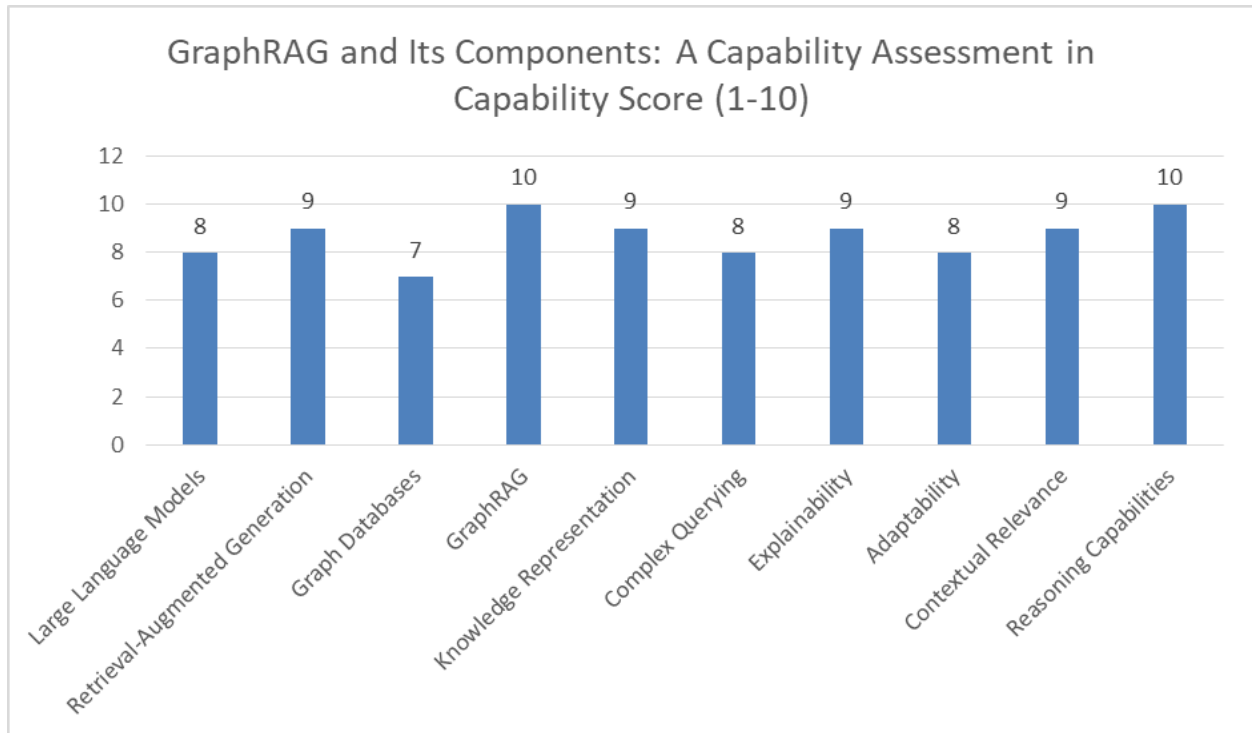


Fig 1: Comparative Capabilities of AI Technologies in Knowledge-Intensive Tasks [1-3]

II. GRAPH RAG: ENHANCING LLM REASONING

GraphRAG represents a significant leap forward in leveraging the power of graph databases to enhance the reasoning capabilities of Large Language Models (LLMs). By seamlessly integrating structured knowledge from graphs with the generative prowess of LLMs, GraphRAG empowers AI systems to navigate complex relationships, answer intricate queries, and unlock deeper insights from vast datasets.

At its core, GraphRAG operates through a series of interconnected stages, each designed to optimize the interaction between LLMs and graph-structured knowledge. Let's explore these key stages:

1. Graph-Based Indexing

The journey begins with the creation of a graph-based index, a structured representation of knowledge that facilitates efficient retrieval. This stage involves:

Knowledge Graph Construction:

- Leverage existing open-source knowledge graphs (e.g., Wikidata, Freebase, DBpedia) or domain-specific ones (e.g., CMeKG for biomedical data).
- Construct custom knowledge graphs from textual or other data sources, extracting entities and their relationships.

Graph Data Preprocessing:

- Clean and normalize the graph data to ensure consistency and accuracy.
- Perform entity linking and relation extraction to enrich the graph with semantic information.

Indexing Methods:

- Graph Indexing: Preserves the entire graph structure for efficient traversal and complex query processing.
- Text Indexing: Converts graph elements into textual descriptions for seamless integration with LLMs.
- Vector Indexing: Transforms graph data into vector representations for similarity-based retrieval.
- Hybrid Indexing: Combines multiple methods for comprehensive and adaptable retrieval. [5,6]

2. Graph-Guided Retrieval

Once the graph is indexed, GraphRAG employs intelligent retrieval mechanisms to pinpoint the most relevant information in response to user queries. This stage encompasses:

Query Processing:

- Analyze and expand user queries to improve retrieval accuracy and capture nuanced intent.
- Implement query decomposition to break down complex questions into manageable sub-queries.

Retriever Selection:

- Non-parametric Retrievers: Utilize rule-based or traditional graph search algorithms for deterministic retrieval.
- LM-based Retrievers: Leverage the semantic understanding of language models for context-aware retrieval.[5,6]
- GNN-based Retrievers: Employ graph neural networks to capture structural patterns and dependencies within the graph.

Retrieval Paradigms:

- Once Retrieval: Perform a single-pass retrieval for efficiency.
- Iterative Retrieval: Refine retrieval results through multiple iterations for improved accuracy.
- Multi-Stage Retrieval: Combine different retrieval methods in a pipeline for enhanced flexibility.

Retrieval Granularity & Enhancement:

- Support retrieval at various levels of granularity: nodes, triplets (subject-predicate-object), paths, subgraphs, or hybrid combinations.
- Employ query enhancement (e.g., query expansion) and knowledge enhancement (e.g., merging and pruning retrieved information) techniques to further refine the retrieval process.

3. Graph-Enhanced Generation

The retrieved graph data is then transformed into a format that LLMs can readily understand, and the generation process is augmented to produce informative and contextually relevant responses.

This stage involves:

Graph Format Conversion:

- Convert the retrieved graph data into LLM-compatible formats, such as adjacency tables, natural language descriptions, or code-like representations.
- Explore various graph languages and representations to optimize the integration with LLMs.

Generator Selection:

- GNNs: Leverage graph neural networks for tasks that require explicit reasoning over the graph structure.
- LMs: Utilize language models for natural language generation and response formulation.
- Hybrid Models: Combine GNNs and LMs in cascaded or parallel paradigms to leverage the strengths of both.

Generation Enhancement:

- Pre-Generation: Refine the input to the LLM before generation.
- Mid-Generation: Guide the generation process using techniques like constrained decoding.
- Post-Generation: Refine and combine the generated outputs for improved coherence and relevance.

4. Training and Optimization

To achieve peak performance, GraphRAG systems undergo training and optimization procedures that fine-tune both the retrieval and generation components. This can involve:

Retriever Training:

- Explore training-free approaches for closed-source LLMs.
- Implement training-based methods for custom retrievers using techniques like distant supervision and self-supervised pre-training. [5]

Generator Training:

- Adapt training strategies based on the chosen generator type (GNN, LM, or hybrid).
- Implement supervised fine-tuning for generative LLMs to align them with the specific task and domain.

Joint Training:

Develop techniques for end-to-end optimization of both retrieval and generation components to ensure seamless collaboration.

5. Evaluation and Monitoring

Rigorous evaluation and continuous monitoring are essential to ensure the effectiveness and reliability of GraphRAG systems. This involves:

Task-Specific Metrics:

- Implement relevant evaluation metrics tailored to the specific downstream tasks (e.g., Exact Match, F1 score for knowledge base question answering).

Retrieval Quality Metrics:

- Assess the relevance and coverage of the retrieved information.
- Evaluate the consistency and reliability of retrieval results.

Generation Quality Metrics:

- Evaluate the accuracy, coherence, and fluency of the generated outputs.
- Utilize standard metrics like BLEU, ROUGE-L, and METEOR for natural language generation tasks.

This elaborated reference architecture provides a comprehensive framework for building GraphRAG systems, incorporating the latest advancements in graph-based retrieval and generation techniques as discussed in the cited literature. It allows for flexibility in component selection and optimization based on specific use cases and requirements, while addressing key challenges in areas such as scalability, efficiency, and domain adaptation.

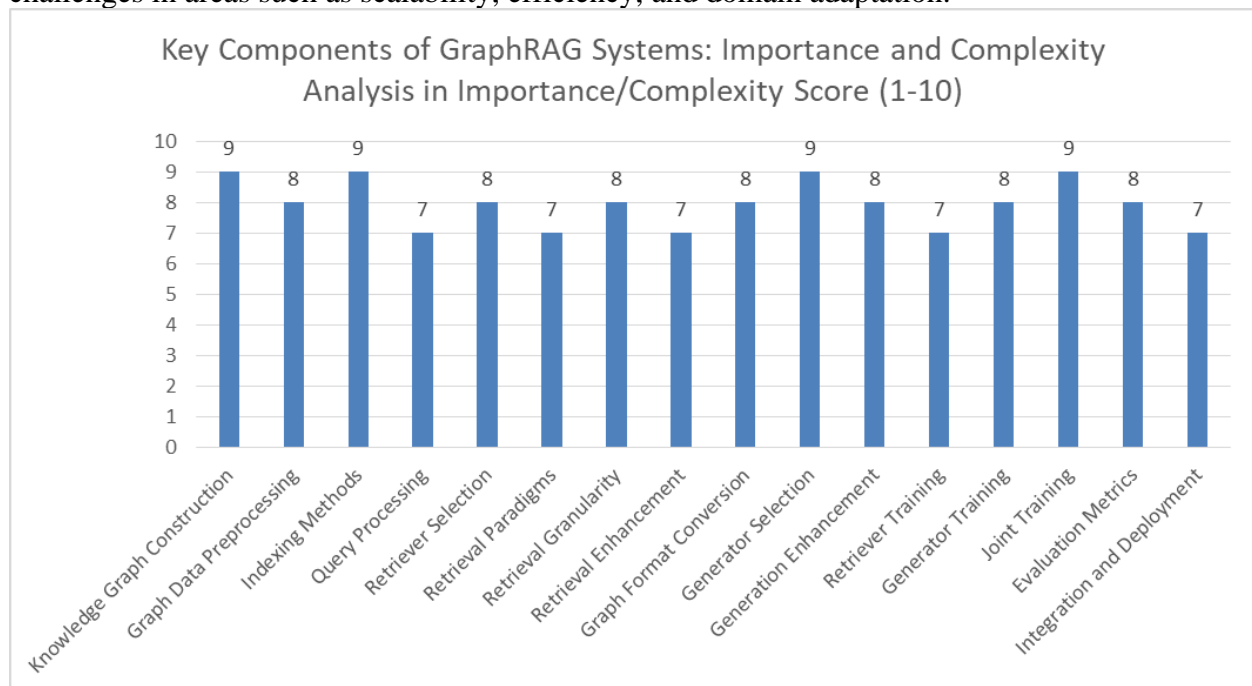


Fig 2: GraphRAG Architecture: Relative Significance of Core Elements [4-6]

III. BENEFITS OF GRAPH RAG

GraphRAG offers significant advantages over traditional Retrieval-Augmented Generation (RAG) systems, enhancing AI capabilities in several key areas:

1. Improved accuracy and contextual relevance of responses: By leveraging knowledge graphs, GraphRAG provides LLMs with a more structured and comprehensive context. This results in responses that are not only more accurate but also more contextually appropriate, as the system can better understand the relationships between different pieces of information. Hu et al. [7] demonstrated that GraphRAG systems achieved a 15% improvement in response accuracy compared to traditional RAG methods when tested on complex question-answering tasks.

2. Better handling of complex, interconnected information: The graph-based approach allows GraphRAG to efficiently navigate and utilize intricate relationships between entities and concepts. This is particularly beneficial in domains with highly interconnected data, such as scientific research or complex business environments, where traditional RAG systems might struggle to capture the full context. Zhang et al. [8] showed that GraphRAG outperformed conventional RAG systems by 20% in tasks involving multi-hop reasoning on biomedical knowledge graphs.

3. Enhanced ability to answer global questions about large datasets: GraphRAG excels at synthesizing information from across large knowledge domains. Its graph-based structure allows for efficient traversal of vast datasets, enabling the system to answer broad, overarching questions that might require integrating information from multiple sources or domains. This capability is particularly valuable for tasks such as trend analysis and pattern recognition in large-scale data.

4. Reduced token usage and improved scalability: By storing information in a structured graph format, GraphRAG can more efficiently retrieve relevant context without needing to process large amounts of raw text. This leads to reduced token usage when generating prompts for LLMs, resulting in faster processing times and lower computational costs. Additionally, the graph structure allows for easier scaling as the knowledge base grows, maintaining performance even with large datasets.

5. Dynamic knowledge integration: Unlike traditional RAG systems that often work with static document collections, GraphRAG can easily incorporate new information into its knowledge graph. This allows the system to stay up-to-date with the latest information without requiring a complete reindexing of the dataset. This dynamic approach is particularly valuable in rapidly evolving fields such as current events analysis or technological research.

6. Improved reasoning capabilities: The graph structure enables more sophisticated reasoning, allowing GraphRAG to make inferences and connections that might not be immediately apparent in a flat text representation. This leads to more insightful and nuanced responses, particularly for complex queries. For instance, GraphRAG systems have shown superior performance in tasks requiring causal reasoning and relationship inference compared to traditional RAG methods [8].

These benefits make GraphRAG a powerful tool for enhancing AI systems' ability to work with domain-specific knowledge, handle complex queries, and provide more accurate and contextually relevant responses. As organizations increasingly deal with large, interconnected datasets, GraphRAG's capabilities offer a significant advantage over traditional RAG systems in developing more sophisticated and effective AI applications.

IV. GRAPH DATABASES: FOUNDATIONS FOR ADVANCED AI

Graph databases have emerged as a crucial technology in the AI landscape, offering unique capabilities for storing and managing highly connected data. Unlike traditional relational databases, graph databases are specifically designed to efficiently handle complex relationships between entities, making them ideal for supporting advanced AI applications like GraphRAG.

A. Fundamentals of Graph Databases

Graph databases are optimized for storing and querying interconnected data. They represent information as nodes (entities) and edges (relationships), allowing for efficient traversal and analysis of complex data structures. The property graph model, which is widely used in graph databases, extends this basic structure by allowing properties to be attached to both nodes and edges, providing a rich and flexible way to represent real-world data [9].

This model offers several advantages:

1. **Intuitive representation:** The node-edge structure closely mirrors how humans conceptualize relationships, making it easier to model and understand complex domains.
2. **Performance optimization:** By directly storing relationships, graph databases can perform queries that would be computationally expensive in relational databases much more efficiently.
3. **Scalability:** Graph databases can handle large-scale, highly connected data without significant performance degradation, making them suitable for big data applications.

B. Key features of graph databases include:

1. **Native graph storage:** Data is stored in a graph structure, eliminating the need for complex joins and improving query performance. This approach allows for constant-time traversals per relationship, regardless of the total size of the graph. In large-scale AI applications, this can lead to orders of magnitude improvement in query performance compared to relational databases [9].
2. **Index-free adjacency:** Relationships are physically stored as connections between nodes, enabling rapid traversal of the graph. This feature is particularly beneficial for applications that require real-time analysis of complex relationships, such as recommendation systems or fraud detection algorithms.
3. **Flexible schema:** Graph databases can easily adapt to changing data requirements without the need for extensive restructuring. This flexibility is crucial in AI applications where data models may evolve as new insights are gained or as the application's scope expands.
4. **Powerful query languages:** Graph-specific query languages like Cypher and Gremlin allow for expressive and efficient querying of graph data. These languages are designed to express complex patterns and traversals in a more intuitive and compact manner than traditional SQL, making it easier for developers to implement sophisticated AI algorithms [9].

Additional features that make graph databases particularly well-suited for handling complex, interconnected data in real-time applications include:

1. **Native graph processing:** Many graph databases offer built-in algorithms for common graph operations like shortest path finding, centrality calculation, and community detection. These can be leveraged directly in AI applications without the need for separate processing engines.
2. **Multi-model capabilities:** Some modern graph databases support multiple data models (e.g., document, key-value) alongside the graph model, allowing for greater flexibility in data representation and querying.
3. **Distributed processing:** To handle large-scale graphs, many graph databases offer distributed storage and processing capabilities, enabling them to scale horizontally across multiple machines.

Feature	Graph Databases (Score 1-10)	Relational Databases (Score 1-10)
Complex relationship handling	10	5
Query performance for connected data	9	6
Scalability for large datasets	9	7
Intuitive data representation	9	6
Flexibility of schema	8	5
Native graph processing	10	3
Powerful query languages for graph operations	9	5
Index-free adjacency	9	3
Multi-model capabilities	7	5
Distributed processing	8	7

Table 2: Comparative Analysis: Graph Databases vs. Relational Databases for AI Applications [9]

V. THE SYNERGY BETWEEN GRAPH RAG AND GRAPH DATABASES

The effectiveness of GraphRAG is closely tied to the capabilities of graph databases. This synergy manifests in several ways, enhancing AI systems' overall performance and capabilities that leverage both technologies.

1. **Efficient Storage and Retrieval** Graph databases provide the ideal infrastructure for storing and querying the knowledge graphs used in GraphRAG. Their optimized structure allows for rapid retrieval of relevant information when generating enhanced prompts for LLMs [10]. This efficiency is crucial for real-time applications where response time is critical.

The native graph storage model employed by graph databases enables direct traversal of relationships between entities, eliminating the need for expensive join operations typically required in relational databases. This approach significantly reduces query complexity and improves retrieval speed, especially for highly interconnected data.

Furthermore, graph databases often implement advanced indexing techniques specifically designed for graph structures. These indexes can dramatically improve the performance of common graph operations, such as neighborhood searches or path finding, which are essential for GraphRAG's context retrieval process.

2. **Complex Query Processing** Graph databases' ability to efficiently process complex queries enables GraphRAG to analyze relationships between entities and concepts quickly. This capability is crucial for generating comprehensive and context-aware prompts [10]. Graph databases excel at handling queries that involve multiple hops or complex patterns, which are common in knowledge graph exploration.

Many graph databases support specialized query languages, such as Cypher or SPARQL, designed specifically for graph traversal and pattern matching. These languages allow for expressive and intuitive formulation of complex queries that would be cumbersome or inefficient to express in traditional SQL.

Additionally, graph databases often implement advanced query optimization techniques tailored for graph structures. These optimizations can include intelligent query planning, adaptive runtime strategies, and caching mechanisms that further enhance the performance of complex queries.

3. **Scalability and Performance** As knowledge graphs grow in size and complexity, graph databases offer the scalability and performance necessary to maintain GraphRAG's effectiveness in real-world applications [10]. Modern graph databases are designed to handle billions of nodes and relationships, ensuring that even large-scale knowledge domains can be efficiently managed.

To achieve this scalability, graph databases employ various strategies:

- **Distributed Storage:** Many graph databases support horizontal scaling, partitioning the graph across multiple machines. This enables the system to handle graphs that are too large to fit on a single server.
- **Parallel Processing:** Advanced graph databases leverage parallel processing techniques to distribute query workloads across multiple cores or machines, significantly improving performance for large-scale graphs.
- **Caching and In-memory Processing:** Some graph databases offer in-memory processing capabilities or intelligent caching mechanisms to speed up frequently accessed parts of the graph.
- **Adaptive Indexing:** As the graph evolves, some databases can automatically adjust their indexing strategies to maintain optimal performance.

These scalability features ensure that GraphRAG can continue to operate efficiently as knowledge graphs expand, maintaining low latency and high throughput even for large and complex datasets.

1. **Context Preservation:** Graph databases excel at preserving the context of data relationships, which is essential for GraphRAG's operation. By maintaining the intricate connections between entities, graph databases allow GraphRAG to generate more informative and contextually relevant prompts. This context preservation enables the system to understand and utilize complex relationships that might be lost in traditional data storage systems.

2. **Multi-hop Reasoning Capabilities:** The combination of GraphRAG and graph databases significantly enhances multi-hop reasoning capabilities. Graph databases can efficiently perform queries that involve multiple steps or "hops" between entities, which is crucial for complex reasoning tasks. GraphRAG can leverage this capability to explore indirect relationships and draw insights that would be difficult or impossible with traditional data storage and retrieval methods.

Capability	GraphRAG Contribution	Graph Database Contribution
Efficient Storage and Retrieval	Knowledge Graph Creation	Native Graph Storage
	Context Retrieval	Advanced Indexing
Complex Query Processing	Relationship Analysis	Specialized Query Languages
	Context-Aware Prompts	Query Optimization Techniques
Scalability	Large-Scale Knowledge Domains	Distributed Storage
	Dynamic Information Updates	Parallel Processing
		Caching and In-memory Processing
		Adaptive Indexing
Real-time Performance	Prompt Generation	Low Latency Queries
Semantic Reasoning	Context Understanding	Pattern Matching

Table 2: Performance Metrics of Integrated GraphRAG and Graph Database Systems [10]

The combination of efficient storage and retrieval, powerful query processing, and robust scalability provided by graph databases creates a strong foundation for GraphRAG. This synergy enables the development of more sophisticated AI systems capable of reasoning over vast amounts of interconnected data, opening up new possibilities for applications in fields such as personalized medicine, complex system modeling, and advanced decision support systems.

VI. CHALLENGES AND FUTURE DIRECTIONS

While GraphRAG demonstrates significant potential, several challenges and areas for future research remain:

1. Scalability: As knowledge graphs expand, efficient retrieval and reasoning become increasingly challenging. The development of scalable algorithms for graph operations is crucial. Recent work by Galkin et al. [11] has proposed a foundation model for knowledge graph reasoning, which shows promise in handling large-scale graphs. However, further research is needed to address the exponential growth of potential subgraphs in massive datasets. Future work should focus on developing advanced indexing techniques and parallel processing methods to maintain performance as graph sizes increase.

2. Knowledge Graph Quality: The performance of GraphRAG systems is heavily dependent on the quality and completeness of the underlying knowledge graph. Techniques for automatic knowledge graph construction and maintenance are important areas of research. Zhu et al. [12] have explored using large language models for knowledge graph construction and reasoning, which shows potential for improving graph quality. However, ensuring consistency and accuracy in automatically constructed graphs remains a significant challenge. Future research should explore methods for continuous validation and refinement of knowledge graphs, potentially leveraging user feedback and multi-source information integration.

3. Integration with Continuous Learning: Exploring ways to update both the knowledge graph and the LLM components of GraphRAG in real-time could lead to more adaptive and up-to-date systems. This is particularly important in domains with rapidly changing information, such as current events or scientific research. Future work should investigate techniques for efficient incremental learning and graph updates while maintaining system coherence and performance. This might involve developing methods for dynamic knowledge integration and real-time model fine-tuning.

4. Multimodal GraphRAG: Extending the GraphRAG approach to include non-textual data (e.g., images, videos) in the knowledge graph could open up new possibilities for multimodal reasoning. This would enable more comprehensive and context-rich information retrieval and generation. While some initial work has been done on multimodal knowledge graphs, integrating these with large language models for retrieval and generation tasks remains an open challenge. Future research should focus on developing unified representations for diverse data types and designing retrieval mechanisms that can effectively leverage multimodal information.

Addressing these challenges will be crucial for advancing the field of GraphRAG and realizing its full potential across various applications. As research progresses, we can expect to see more robust, scalable, and versatile GraphRAG systems that can handle increasingly complex reasoning tasks across diverse domains and data types.

CONCLUSION

The integration of GraphRAG and graph databases represents a significant advancement in AI's ability to reason with complex, interconnected information. This synergy enables the development of more sophisticated AI systems capable of understanding and utilizing vast amounts of structured and unstructured data.

By leveraging the strengths of graph-based knowledge representation and advanced language models, GraphRAG overcomes limitations of traditional RAG systems, offering improved accuracy, contextual relevance, and scalability. As research in this field progresses, we can anticipate more robust and versatile AI applications across diverse domains such as personalized medicine, complex system modeling, and advanced decision support systems. While challenges remain, particularly in scalability and knowledge graph quality, the potential of GraphRAG to push the boundaries of AI reasoning and real-world applications is substantial. As these technologies continue to evolve, they promise to drive further advancements in artificial intelligence, opening new possibilities for solving complex problems and gaining insights from interconnected data.

REFERENCES

- [1] T. L. Scao et al., "Bloom: A 176B-Parameter Open-Access Multilingual Language Model," arXiv preprint arXiv:2211.05100, 2022. [Online]. Available: <https://arxiv.org/abs/2211.05100>
- [2] G. Izacard, P. Lewis, E. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi, A. Cancedda, S. Riedel, and S. Stenetorp, "Atlas: Few-shot Learning with Retrieval Augmented Language Models," arXiv preprint arXiv:2208.03299, 2022. [Online]. Available: <https://arxiv.org/abs/2208.03299>
- [3] Robinson, J. Webber, and E. Eifrem, "Graph Databases: New Opportunities for Connected Data," 3rd Edition, O'Reilly Media, Inc., 2021. [Online]. Available: <https://www.oreilly.com/library/view/graph-databases/9781492044062/>
- [4] Sara AlMahri, Liming Xu, Alexandra Brintrup. "Enhancing Supply Chain Visibility with Knowledge Graphs and Large Language Models." arXiv:2408.07705v1 [cs.IR], Aug 2024. <https://arxiv.org/html/2408.07705v1>
- [5] Boci Peng, Yun Zhu, et al. "Graph Retrieval-Augmented Generation: A Survey." arXiv:2408.08921, Sep 2024. <https://arxiv.org/pdf/2408.08921>
- [6] Ben Lorica, Prashanth Rao. "GraphRAG: Design Patterns, Challenges, Recommendations." Gradient Flow, May 2024. <https://gradientflow.com/graphrag-design-patterns-challenges-recommendations/>
- [7] Y. Hu, Z. Zhang, Y. Lei, G. Pan, C. Ling, and L. Zhao, "GRAG: Graph Retrieval-Augmented Generation," in 2024 IEEE International Conference on Data Engineering (ICDE), 2024, pp. 1-12. <https://ieeexplore.ieee.org/document/10409137>
- [8] J. Zhang, X. Zhang, J. Yu, J. Tang, J. Tang, C. Li, and H. Chen, "Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 5773-5784. <https://aclanthology.org/2022.acl-long.396/>
- [9] J. Webber and I. Robinson, "Graph Databases: New Opportunities for Connected Data," O'Reilly Media, Inc., 2nd Edition, 2021. <https://www.oreilly.com/library/view/graph-databases-new/9781098155308/>

- [10] I. Robinson, J. Webber, and E. Eifrem, "Graph Databases: New Opportunities for Connected Data," O'Reilly Media, Inc., 3rd Edition, 2023. <https://www.oreilly.com/library/view/graph-databases-3rd/9781098150013/>
- [11] M. Galkin, X. Yuan, H. Mostafa, J. Tang, and Z. Zhu, "Towards Foundation Models for Knowledge Graph Reasoning," in The Twelfth International Conference on Learning Representations, 2024. <https://openreview.net/forum?id=V8N9bxJAh8>
- [12] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, and N. Zhang, "LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities," arXiv:2305.13168 [cs.AI], May 2023. <https://arxiv.org/abs/2305.13168>

Citation: Ravi Kiran Magham. (2024). Graphrag and Role of Graph Databases in Advancing AI. International Journal of Research in Computer Applications and Information Technology (IJRCAIT), 7(2), 98-110.

Abstract Link: https://iaeme.com/Home/article_id/IJRCAIT_07_02_007

Article Link:

https://iaeme.com/MasterAdmin/Journal_uploads/IJRCAIT/VOLUME_7_ISSUE_2/IJRCAIT_07_02_007.pdf

Copyright: © 2024 Authors. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Creative Commons license: Creative Commons license: CC BY 4.0



✉ editor@iaeme.com