A Study of the GraphRAG systems: Korean Art Question and Answering

Yebeen Lee, Minjun Song, Sumin Kim, Heung-No Lee
Gwangju Institute of Science and Technology
leeyebeen_ug@gm.gist.ac.kr, paulmjsong@gm.gist.ac.kr, smkim6927@gm.gist.ac.kr, heungno@gist.ac.kr

ABSTRACT

Retrieval-Augmented Generation (RAG) has emerged as a key technique for mitigating hallucinations in Large Language Models (LLMs). However, conventional vector–similarity-based RAG systems struggle to use explicit relationships among entities and to answer complex multi-hop questions that require "multi-hop" reasoning. They also handle structured relational constraints less effectively than relational databases. To address these limitations, this study proposes a Korean Art GraphRAG system, which constructs a knowledge graph from specific domain of Korean art API and uses structured graph retrieval to guide LLM responses. Through a pilot study comparing our system with a vector-based QA baseline and GPT-4o, we observed instances where the proposed system effectively handled relational queries and multi-hop questions. This research demonstrates the effectiveness of GraphRAG in specialized domains where relational reasoning over structured knowledge is essential.

1. Introduction

Retrieval-Augmented Generation (RAG) has emerged as a practical solution for mitigating hallucinations in Large Language Models (LLMs). However, retrieval methods that rely solely on vector similarity exhibit clear limitations in reliably handling relational and multi-hop reasoning, particularly in the visual arts domain, which requires the precise navigation of multi-layered relationships among artists, artworks, years, genres, and historical contexts.

To address these issues, this paper proposes the Korean Art GraphRAG framework, specialized for the Korean art domain, built upon a Knowledge Graph constructed using collection data from the National Museum of Modern and Contemporary Art (MMCA) API. When a query is given, it dynamically extracts a relevant subgraph through a multi-granularity, structured search, and provides this subgraph.

The performance of the proposed model was evaluated on two tasks: Simple Question Answering (Simple QA) and Multi-hop Question Answering (Multi-hop QA). Comparative experiments were conducted against strong baselines, including a vector-based RAG and GPT-4o, using the automated evaluation metric BLEU.

The contributions of this study are as follows. First, we constructed a Neo4j-based Knowledge Graph for Korean art using the MMCA API. Second, we designed a structured context retrieval pipeline that preserves relational paths tailored to the query's intent. Third, through a preliminary evaluation on selected queries, we verified the feasibility of the proposed model in handling complex relational information compared to vector-based RAG. This suggests that in question-answering systems for highly structured domains like art, an explicit knowledge structure serves as a powerful inductive bias.

2. Method

(1) construction of a Korean art knowledge graph, and (2) structured context retrieval at multiple levels of granularity.

2.1. Knowledge Graph

The GraphRAG system in this study is founded on a systematic knowledge graph to enable a deep understanding of Korean art history data.

Fig. 1 Pipeline of building graphDB

2.1.1. Information Extraction and Refinement

The collected data includes unstructured text where artist and artwork information are commingled, making a refinement process to convert it into essential structured information. Each of these is structured and extracted into the necessary data. This process structures the various art-related texts based on four key entities: Artist, Artwork, Year, and Genre (movements).

2.1.2. Graph Transformation and Loading

Based on the refined data, a graph model was defined and loaded into Neo4j.

Relationships: The relationships between nodes were defined as CREATED (Artist → Artwork), BELONGS (Artwork → Genre), CREATED_IN (Artwork → Year), BORN_IN (Artist → Birth Year), and DIED_IN (Artist → Death Year).

Data Loading (Upsert): Cypher query functions corresponding to each node and relationship type, such as upsert_artists and upsert_artworks, were written to merge (MERGE) data into the Neo4j database. This approach prevents data duplication and maintains consistency.

2.2. Traversal

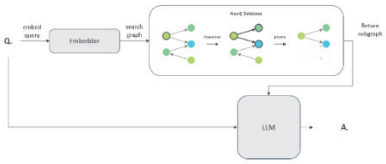

Fig.2 Pipeline of Question and Answering

2.2.1. Hybrid Retrieval Initiation

The process begins when a user submits a natural language query q . The system first uses an embedding model to convert this query into a high-dimensional vector $\square\ \square q = fembed\ (q) \in \mathbb{R}d$. This initial step identifies the most semantically relevant nodes $S = \{vi \mid \text{sim}(\mathbf{v}q, \mathbf{v}i) \geq \theta\}$ or the top-k nodes, known as seed nodes, which serve as the starting points for the subsequent graph traversal.

2.2.2. Graph Traversal and Path Ranking

Multi-hop Path Finding: Starting from each seed node $s \in S$, the query performs a multi-hop traversal, searching for all paths $P_s = \{p \mid p \text{ is a path from } s, |p| \leq d_{max}\}$ up to a predefined depth (e.g., $d_{max} = 3$ hops). This allows the system to uncover not just direct relationships but also indirect, multi-step connections between entities.

Hybrid Scoring and Ranking: A simple traversal can yield an overwhelming amount of information. To address this, the system calculates a rankScore for each discovered path. This hybrid score is a weighted combination of three key factors:

$$\text{rankScore}(\boldsymbol{p}) = \boldsymbol{w_1} \cdot \text{seedScore}(\boldsymbol{p}) + \boldsymbol{w_2} \cdot \text{hopDecay}(\boldsymbol{p}) + \boldsymbol{w_3} \cdot \text{edgeDegree}(\boldsymbol{p}) \quad (1)$$

where $w_1 + w_2 + w_3 = 1$, and the three key factors are defined.

Seed Score:

$$\text{seedScore}(\boldsymbol{p}) = \text{sim}(\mathbf{v}_{\boldsymbol{q}}, \mathbf{v}_{\boldsymbol{seed(p)}}) \quad (2)$$

The initial semantic similarity score from the vector search.

Hop Decay: A penalty is applied to paths based on their length (hops), giving higher importance to closer relationships.

Edge Degree:

$$\text{edgeDegree}(\boldsymbol{p}) = \frac{\mathbf{1}}{|\boldsymbol{V(p)}|} \sum_{\boldsymbol{v} \in \boldsymbol{V(p)}} \text{degree}(\boldsymbol{v}) \quad (3)$$

where $\mathbf{V(p)}$ denotes the set of all vertices included in path p - The query considers the connectivity of the nodes within a path, prioritizing paths that involve more central or highly connected entities.

Top Path Selection: After ranking all paths by their rankScore, the system selects a limited number of the highest-scoring paths $P_{top} = \text{TopK}(\bigcup_{s \in S} P_s, L \times |S|)$ (per_seed_limit) to form the basis of the context, filtering out less relevant information.

2.2.3. Subgraph Reconstruction

The top-ranked paths are then used to reconstruct a concise and relevant subgraph $G_{sub} = (V_{sub}, E_{sub})$. This involves collecting all unique nodes $V_{sub} = \bigcup_{p \in P_{top}} V(p)$ and relationships $E_{sub} = \bigcup_{p \in P_{top}} E(p)$ present in these paths. The system ensures that each node and relationship in the final subgraph is assigned the highest rank it received across all the paths it appeared in.

$$\text{rank}(\boldsymbol{v}) = \mathbf{max}_{\boldsymbol{p}: \boldsymbol{v} \in \boldsymbol{p}, \boldsymbol{p} \in \boldsymbol{P_{top}}} \text{rankScore}(\boldsymbol{p}) \quad (4)$$

for each node $v \in V_{sub}$

$$\text{rank}(\boldsymbol{e}) = \mathbf{max}_{\boldsymbol{p}: \boldsymbol{e} \in \boldsymbol{p}, \boldsymbol{p} \in \boldsymbol{P_{top}}} \text{rankScore}(\boldsymbol{p}) \quad (5)$$

for each edge $e \in E_{sub}$

2.2.4. Context Generation for LLM

Finally, the reconstructed subgraph is passed to a formatter function

$$\mathbf{C} = \boldsymbol{f_{fm}}(\boldsymbol{G_{sub}}, \{\text{rank}(\boldsymbol{v})\}_{\boldsymbol{v} \in \boldsymbol{V_{sub}}}, \{\text{rank}(\boldsymbol{e})\}_{\boldsymbol{e} \in \boldsymbol{E_{sub}}}) \quad (7)$$

This text, which explicitly lists the entities and their connections (e.g., Artist -[DIED_IN]-> Year).

3. Result

We executed three single-hop and three multi-hop questions for each model and measured their BLEU scores. The results are presented in Table 1.

	Multi-Hop QA		
	1	2	3
Retrieval			
VectorRAG	0.0308	0.2388	0.2170
GraphRAG	0.3088	0.7335	0.5905
ChatGPT 4o	0.0131	0.0385	0.0156

Table 1. BLEU Scores for Multi-Hop QA Tasks

Multi-Hop QA Analysis: The performance gap between models became more pronounced on multi-hop questions that require complex reasoning. Graph RAG outperformed the other models across all three multi-hop questions and showed strong results when answers had to be synthesized across multiple relations. In particular, for Question B ("tell me another painter who died in the same year Jinhwan died," BLEU 0.7335) and Question C ("tell me a painter born in the year the Official Portrait of Emperor Gojong was painted," BLEU 0.5905) Graph RAG was the only system to produce meaningfully correct answers.

By contrast, Vector RAG and ChatGPT-4o scored below 0.05 BLEU on all multi-hop questions, effectively failing to capture the intent and perform the necessary reasoning. This highlights a clear limitation of vector-based retrieval: while it can locate individual fragments such as ″A and B″ or ″B and C,″ it struggles to follow the logical chain that connects them as ″A → B → C.

4. Conclusion

This study proposes Korean Art GraphRAG, a specialized framework for the Korean art domain, demonstrating that knowledge graphs enable accurate responses to relational queries. By leveraging query-specific subgraphs derived from the MMCA API, our approach facilitates relational path inference, demonstrating an alternative approach to address the limitations of simple text matching. Experimental results showed excellent performance across both Simple and Multi-hop categories and significantly overcame the fragmented and inconsistent responses of vector-based RAG and general-purpose LLMs, particularly on multi-hop queries that trace temporal and semantic relationships.

Admittedly, this approach is constrained by the knowledge graph's coverage, with limitations in handling long-tail information or unstructured text not yet encoded. Furthermore, refining the subgraph extraction mechanism to accurately reflect query intent remains a future task. Future research will focus on expanding the knowledge base by integrating external corpora, enriching the graph to include more flexible contextual relations, and introducing human evaluation to assess discourse-level response quality.

ACKNOWLEDGMENT

This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2026-RS-2021-II211835) and This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (RS-2026-22932973)

REFERENCE

[1] Gutiérrez, B. J., Shu, Y., Qi, W., Zhou, S., & Su, Y. (2025). From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

[2] Wang, S., Najdenkoska, I., Zhu, H., Rudinac, S., Kackovic, M., Wijnberg, N., & Worring, M. (2025). ArtRAG: Retrieval-Augmented Generation with Structured Context for Visual Art Understanding. *arXiv preprint arXiv:2505.06020*.

[3] Gutierrez, B. J. (2025). From RAG to Memory: Non-Parametric Continual Learning for Large Language Models. *ICML 2025*. https://doi.org/10.48550/arXiv.2502.14802