ReGraphRAG: Reorganizing Fragmented Knowledge Graphs for Multi-Perspective Retrieval-Augmented Generation

Soohyeong Kim[1], Seok Jun Hwang[1], JungHyoun Kim[2], Jeonghyeon Park[3], Yong Suk Choi[3]*

[1]Department of Artificial Intelligence, Hanyang University, Seoul, Korea
[2]Department of Intelligence and Convergence, Hanyang University, Seoul, Korea
[3]Department of Computer Science, Hanyang University, Seoul, Korea
{ksh970404, cody628, kkksk1157, shshjhjh4455, cys}@hanyang.ac.kr

Abstract

Recent advancements in Retrieval-Augmented Generation (RAG) have improved large language models (LLMs) by incorporating external knowledge at inference time. Graph-based RAG systems have emerged as promising approaches, enabling multi-hop reasoning by organizing retrieved information into structured graphs. However, when knowledge graphs are constructed from unstructured documents using LLMs, they often suffer from fragmentation—resulting in disconnected subgraphs that limit inferential coherence and undermine the advantages of graph-based retrieval. To address these limitations, we propose ReGraphRAG, a novel framework designed to reconstruct and enrich fragmented knowledge graphs through three core components: Graph Reorganization, Perspective Expansion, and Query-aware Reranking. Experiments on four benchmarks show that ReGraphRAG outperforms state-of-the-art baselines, achieving over 80% average diversity win rate. Ablation studies highlight the key contributions of graph reorganization and especially perspective expansion to performance gains. Our code is available at: https://github.com/ToBeSuperior/ReGraphRAG

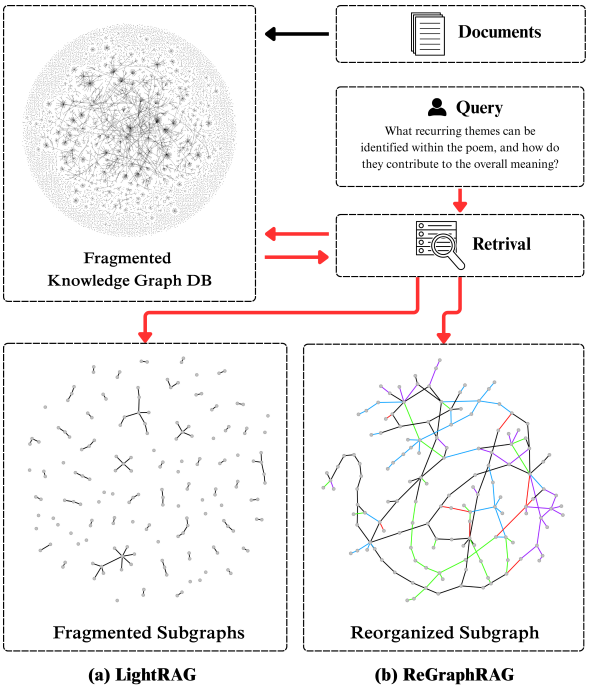

Figure 1: Comparison of retrieved subgraphs between LightRAG (Guo et al., 2024) and ReGraphRAG. (a) LightRAG retrieve disconnected and fragmented subgraphs from the knowledge graph, which limits coherent reasoning. (b) ReGraphRAG reorganizes the retrieved fragments into a unified and semantically enriched subgraph, enabling improved multi-hop reasoning and contextual understanding.

1 Introduction

Recent advances in artificial intelligence (AI), particularly the emergence and widespread adoption of large language models (LLMs) (Achiam et al., 2023; Hagos et al., 2024; Matarazzo and Torlone, 2025), have significantly transformed the landscape of natural language processing and knowledge-intensive tasks. While LLMs have demonstrated impressive capabilities across a wide range of applications, their reliance on static, pre-trained knowledge introduces limitations when addressing queries that require up-to-date, domain-specific, or contextually nuanced information (Ling et al., 2023). These challenges have spurred the development and adoption of Retrieval-Augmented Generation (RAG) frameworks (Lewis et al., 2020; Gao et al., 2023b), which enhance the reasoning and generation capabilities of LLMs by dynamically incorporating external information from large-scale knowledge sources during inference.

Building upon the foundational paradigm of RAG, graph-based RAG has emerged as a prominent structured extension that organizes retrieved information into graph structures (Han et al., 2024; Peng et al., 2024; Zhang et al., 2025). Unlike conventional RAG methods that treat retrieved information as independent text chunks, graph-

* Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5426–5443
November 4-9, 2025

based RAG advances this framework by retrieving and organizing knowledge in the form of graphs, thereby supporting multi-hop reasoning and improved contextual coherence. Notably, contemporary approaches have moved beyond relying solely on pre-existing knowledge graphs; instead, they employ LLMs to construct knowledge graphs directly from unstructured documents, enabling more flexible and adaptive knowledge integration (Edge et al., 2024; Guo et al., 2024). This graph-based retrieval and reasoning mechanism not only strengthens the logical flow of generated responses but also contributes significantly to the overall quality and reliability of LLMs outputs.

Despite the growing interest in graph-based RAG and the emergence of various methodological advancements, significant limitations remain unaddressed. A key challenge lies in the fragmentation that arises when indexing documents into knowledge graphs. When LLMs are used to extract structured information from unstructured text, the resulting graphs often consist of numerous disconnected nodes and isolated components, resulting in sparse and incoherent structures. This undermines the core strength of graph-based retrieval—its ability to support multi-hop reasoning across semantically linked entities. Consequently, as shown in Figure 1(a), the retrieval process itself becomes disjointed, diminishing the effectiveness of downstream reasoning and generation. To address this issue, we propose **ReGraphRAG**, a novel framework that reorganizes retreived fragmented knowledge into connected graphs, thereby enhancing the comprehensiveness and inferential depth of generated responses.

ReGraphRAG, a novel framework comprising three key components: **Graph Reorganization**, **Perspective Expansion**, and **Query-aware Reranking**. First, as shown in Figure 1(b), Graph Reorganization ensures that the retrieved subgraphs are transformed into a single connected graph by identifying meaningful shortest paths between them. This process not only bridges missing connections among disjoint subgraphs but also introduces new edges to support coherent reasoning flows, thereby enhancing both the logical consistency and inferential strength of the generated answers. Given that this reorganization guarantees a connected structure, we further enhance the breadth of retrieved knowledge through Perspective Expansion, which decomposes the original query into multiple interpretive angles and generates sub-queries accordingly. Each sub-query guides the retrieval of complementary subgraphs that collectively broaden the semantic coverage. Finally, all retrieved and reorganized information is converted into a structured triplet form ($< node_i, edge_{ij}, node_j >$) and subjected to Query-aware Reranking, which scores and rerank the most relevant triplets based on their semantic similarity to the original query.

One of the core contributions of this work is the introduction of **ReGraphRAG**, a novel framework that reconstructs knowledge graphs from multiple interpretive perspectives based on the input query. This approach enables the generation of contextually enriched and structurally coherent graphs that are well-suited for inference-driven answer generation. Experimental evaluations conducted on four benchmark datasets demonstrate that ReGraphRAG consistently outperforms state-of-the-art baselines across all four evaluation dimensions—comprehensiveness, diversity, empowerment, and overall. In particular, the average diversity win rate exceeds 80% compared to existing methods, highlighting the framework's strong capacity to capture and integrate semantically diverse information. Furthermore, ablation studies and in-depth analyses reveal that both Graph Reorganization and Perspective Expansion are key contributors to performance gains. Notably, Perspective Expansion is shown to produce significant improvements, underscoring the importance of multi-perspective reasoning in enhancing the quality of generated responses.

2 Related Work

2.1 RAG and Graph-based RAG

Despite the rapid advancement and widespread use of LLMs (Achiam et al., 2023; Grattafiori et al., 2024), they face key limitations such as outdated knowledge, hallucinations, and lack of verifiable outputs (Zhao et al., 2023). To address these issues, retrieval-based knowledge augmentation has been explored, notably with early models like RAG using Dense Passage Retrieval (DPR) (Karpukhin et al., 2020; Lewis et al., 2020).

Existing RAG methods embed documents into vectors stored in a database (Fan et al., 2024), improving LLM performance to some extent. However, they still struggle with multi-hop QA, capturing structural relations, and handling ambiguous queries (Gupta et al., 2024; Das et al., 2025).

To address these limitations, graph-based RAG has been proposed, replacing vector databases with graph-based ones for knowledge augmentation (Peng et al., 2024; Han et al., 2024; Zhang et al., 2025). Approaches include using existing open-domain knowledge graphs (LUO et al., 2024; Sun et al., 2024) or constructing knowledge graphs by extracting entities and relations from corpora via LLMs (Edge et al., 2024; Guo et al., 2024; He et al., 2024b).

2.2 Knowledge Graph Construction from Corpus

Knowledge Graph Construction (KGC) has long been studied, evolving from rule-based and statistical methods to more efficient machine learning and deep learning approaches (Kim et al., 2016; Ji et al., 2021). More recently, LLM-based KGC techniques that utilize prompt-based or few-shot learning have been proposed (Hu et al., 2022; Trajanoska et al., 2023; Pan et al., 2024; Zhang et al., 2024), driven by LLMs' improved ability to understand context and handle long documents.

LLM-based KGC offers high efficiency and domain adaptability, effectively structuring large document-based datasets for QA and reasoning tasks (Peng et al., 2024; Han et al., 2024; Zhang et al., 2025). Many graph-based RAG studies with LLM-based KGC (Edge et al., 2024; Guo et al., 2024; Xu et al., 2025; Chen et al., 2025) show great improvements on these tasks; however, construction issues —such as missing entities and relation errors— are not sufficiently addressed, often resulting in fragmented graphs with disconnected subgraphs that hinder meaningful structural inference (Liu and Li, 2020; Pan et al., 2023; Meyer et al., 2023).

3 Background

Graph-based RAG extends the traditional RAG framework by replacing or enriching flat text indices with explicit graph structures that capture entities and their interrelationships. To construct such graphs from input text documents, LLMs are employed to recognize and extract entities and the relationships among them, resulting in an entity set $\mathcal{V}$ and a relationship set $\mathcal{E}$*. Specifically, the resulting knowledge graph, constructed from textual data, is denoted as $G = (\mathcal{V}, \mathcal{E})$ and consists of k subgraphs. Each node $v \in \mathcal{V}$ represents an entity and contains associated information such as its name, description, and type. Each edge $e \in \mathcal{E}$ indicates a semantic relationship between entities, including metadata such as the source and target entities and a textual description of the relation. All nodes and edges are embedded into a vector space using a pre-trained embedding model, yielding vector representations $\Phi(\mathcal{V})$ and $\Phi(\mathcal{E})$, respectively.

Given a query q, the retriever R identifies semantically relevant nodes and edges from the knowledge graph G based on similarity measures. These retrieved nodes and edges, along with their associated metadata, are integrated into a predefined prompt template P, which, together with q, serves as input to a LLM. This process can be formally represented as:

$$Response(q, G) = LLM(P(q; R(q, G))) \quad (1)$$

where $Response$ denotes the final answer conditioned on the retrieved results.

In this paper, we focus on reorganizing the fragmented granularities retrieved from the graph to enhance the structural connectivity—one of the key advantages of the graph-based approach. Furthermore, we aim to integrate multiple interpretive perspectives to generate responses that are enriched with diverse and contextually grounded knowledge.

4 Method

As illustrated in Figure 2, ReGraphRAG comprises three core components: **Perspective Expansion** (Section 4.1), which decomposes the input query into multiple interpretive angles to retrieve diverse granularities (nodes and edges); **Graph Reorganization** (Section 4.2), which reconstructs the fragmented subgraphs into a single connected graph structure; and **Query-aware Reranking** (Section 4.3), which reorders the retrieved triplets in alignment with the query's semantic intent. Finally, the reorganized graph is converted into a graph-oriented prompt (Section 4.4), which serves as the input to the language model for answer generation.

4.1 Perspective Expansion

Even the simplest questions can typically be addressed from multiple perspectives. For example, the question "Is 1 plus 1 equal to 2?" may be answered affirmatively from a mathematical standpoint, yet a linguistic relativism perspective might argue that "numbers are symbolic constructs whose

*The prompt used for entity and relation extraction is detailed in Appendix A.1

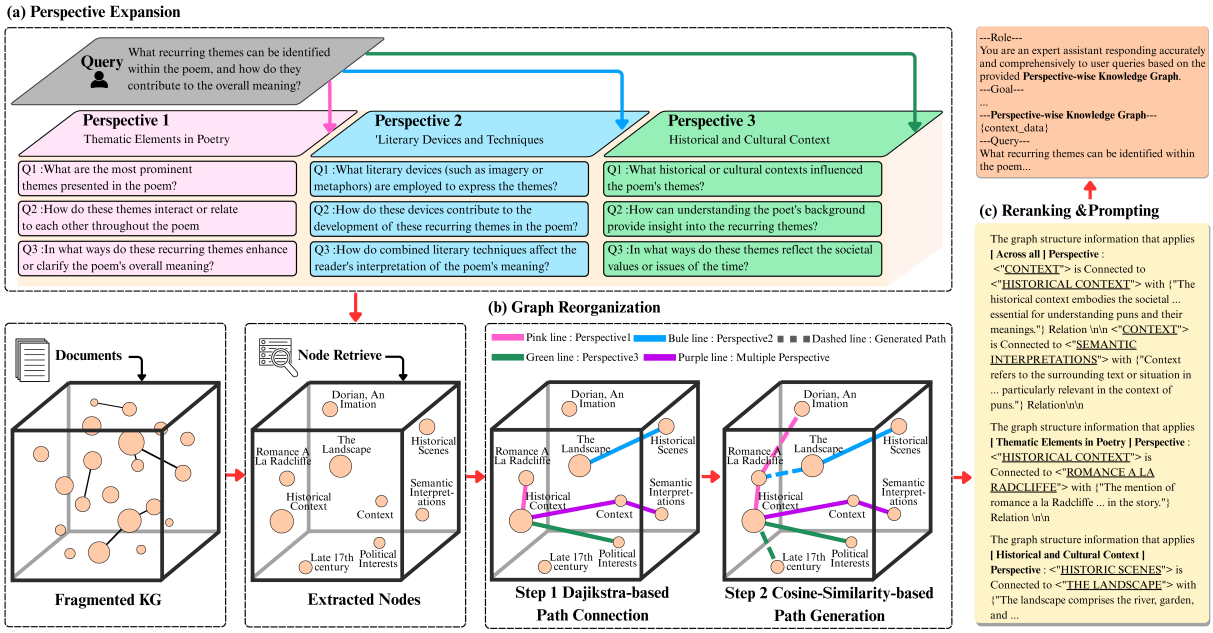

Figure 2: An overview of the **ReGraphRAG** framework. Given a user query, the system first performs **(a) Perspective Expansion**, generating multiple sub-queries across distinct interpretive angles. Retrieved nodes from the fragmented knowledge graph are then reorganized via a two-step process of **(b) Graph Reorganization**: (1) shortest-path connections between subgraphs, and (2) similarity-based path generation. Finally, **(c) Reranking & Prompting** transforms the resulting graph into structured triplets and reranks based on query relevance to construct the LLM prompt.

interpretation can vary across linguistic systems." To implement **Perspective Expansion**, we utilize a LLM to decompose the original query into m distinct interpretive angles, each representing a semantically meaningful subspace of reasoning. For each perspective, we generate n sub-queries using chain-of-thought prompting with exemplars. The prompts used for perspective expansion are detailed in Appendix A.2.

Rather than using the original query, the retriever R uses each generated sub-query to retrieve semantically relevant nodes from the knowledge graph. Specifically, the retrieved nodes corresponding to each sub-query are represented as individual subgraphs $S = (\mathcal{V}, \mathcal{E})$. After eliminating duplicate subgraphs, the collection of subgraphs for a single perspective is denoted by $H = \{S_1, S_2, \ldots, S_k\}$, where each $S_i = (\mathcal{V}_i, \mathcal{E}_i)$.

Consequently, the set $\{H_1, H_2, \ldots, H_m\}$ represents the collection of subgraphs constructed from the m perspectives. Each H_i corresponds to a fragmented, unconnected knowledge graph generated from the sub-queries of a single perspective. In the following section, we describe the method for reorganizing these fragmented subgraphs into a unified, connected graph structure.

4.2 Graph Reorganization

Graph-based RAG systems extract semantically relevant granularities from a knowledge graph in response to a given query. However, the extracted granularities frequently exhibits discontinuities in informational flow, hindering the effective utilization of multi-hop reasoning — one of the key advantages of graph-based approaches. To address this limitation, we aim to reorganize these fragmented subgraphs into a single, connected graph structure, thereby enabling the LLM to leverage the underlying graph topology for more coherent and logically grounded answer generation.

The subgraph set H, extracted in the previous section, consists solely of nodes. Due to the fragmented knowledge graph, there may be no existing paths between these nodes, resulting in disconnected components. To address this, we reconstruct a connected graph from the fragmented subgraphs through a **two-step Graph Reorganization process**.

In the first step, we connect subgraphs that are already linked through existing paths in the original knowledge graph. For each pair of subgraphs, we apply a Dijkstra-based shortest path algorithm and prioritize connections in order of increasing path length. These paths may include intermediary

nodes that were not retrieved during the initial retrieval phase, thereby filling in missing links in the overall reasoning flow.

Algorithm 1 Graph Reorganization

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, list of subgraphs $\{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$ where $\mathcal{S}_i = (\mathcal{V}_i, \mathcal{E}_i)$
Ensure: Connected and merged graph $\mathcal{R}$

$\mathcal{P} \leftarrow \emptyset$ ▷ Step 1: Merge subgraphs based on Dijkstra-based shortest path distances
for $i \leftarrow 1$ to $k-1$ **do**
  for $j \leftarrow i+1$ to k **do**
    $\mathcal{V}_i \leftarrow$ entities of $\mathcal{S}_i$, $\mathcal{V}_j \leftarrow$ entities of $\mathcal{S}_j$
    if paths exist between any $u \in \mathcal{V}_i$ and $v \in \mathcal{V}_j$ **then**
      $p^* \leftarrow$ shortest path between some $u \in \mathcal{V}_i$ and $v \in \mathcal{V}_j$ using Dijkstra
      Add $(i, j, p^*, |p^*|)$ to $\mathcal{P}$
    end if
  end for
end for
Sort $\mathcal{P}$ by path length ascending
for each (i, j, p^*) in $\mathcal{P}$ **do**
  if $\mathcal{S}_i$ and $\mathcal{S}_j$ are not yet merged **then**
    Merge $\mathcal{S}_i$ and $\mathcal{S}_j$, including nodes and edges from p^*
  end if
end for
▷ Step 2: Iteratively merge most similar subgraph pairs based on cosine similarity
$\mathcal{U} \leftarrow$ list of unmerged subgraphs
Extract node embeddings $\Phi(\mathcal{V})$ from subgraphs in $\mathcal{U}$
Compute cosine similarity matrix $\mathcal{C}$ from $\Phi(\mathcal{V})$
while there are disconnected subgraphs in $\mathcal{U}$ **do**
  $(\mathcal{S}_i, \mathcal{S}_j) \leftarrow$ most similar pair in $\mathcal{C}$
  $(u, v) \leftarrow$ most similar entity pair between $\mathcal{S}_i$ and $\mathcal{S}_j$
  Add edge (u, v) to $\mathcal{G}$
  Merge $\mathcal{S}_i$ and $\mathcal{S}_j$ into $\mathcal{S}_{ij}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathcal{S}_i, \mathcal{S}_j\} \cup \{\mathcal{S}_{ij}\}$
  Update $\mathcal{C}$ accordingly
end while
return final connected graph $\mathcal{R}$

In the second step, for subgraphs that remain disconnected due to the absence of any path, we establish new edges based on node-level semantic similarity. Specifically, we extract the node embeddings $\Phi(\mathcal{V})$ for each subgraph and compute pairwise cosine similarity between nodes belonging to different subgraphs. We then iteratively construct edges between the most similar node pairs across different subgraphs. For each selected pair, one node serves as the source entity and the other as the target entity; their respective textual descriptions are concatenated to form the description of the new edge. This step is repeated until the subgraph set H is fully merged into a single connected graph, ensuring the integration of all fragmented knowledge.

The complete process is detailed in Algorithm 1. As a result, the m connected graphs—each corresponding to a distinct interpretive perspective—serve as reorganized knowledge graphs that facilitate multi-perspective reasoning.

4.3 Query-aware Reranking

Recent studies on prompting in RAG systems (Liu et al., 2023; He et al., 2024a; Park et al., 2025) suggest that the placement of critical information—particularly at the beginning or end of the prompt—can significantly influence the performance of LLMs. Motivated by this insight, we rerank the graph structure based on its semantic relevance to the main query, ensuring that the most important information is positioned effectively within the prompt.

To preserve the original structure of the extracted knowledge graph while adapting it for LLM input, we decompose the graph into a set of triplets in the form of $(< node_i, edge_{ij}, node_j >)$. Each edge embedding $\Phi(\mathcal{E})$ captures not only the relationship between the source and target entities but also incorporates their descriptive information. We then perform query-aware reranking of these triplets based on the cosine similarity between the edge embeddings and the main query representation. This process ensures that graph elements most semantically aligned with the query are prioritized during answer generation.

4.4 Prompt Instruction

In graph-based RAG systems, the retrieved granularities must be transformed into textual representations to serve as input to LLMs. Consequently, prior works (Edge et al., 2024; Guo et al., 2024) incorporate all extracted information (e.g., description, chunk, and metadata) directly into the prompt. However, recent research (Chen et al., 2025) argues that the primary limitation of graph-based RAG methods lies not in the insufficiency but in the redundancy of the retrieved information. To

	Agriculture		Legal		CS		Mix		Average Win Rate	
	Naïve	ReGraphRAG	Naïve	ReGraphRAG	Naïve	ReGraphRAG	Naïve	ReGraphRAG	Naïve	ReGraphRAG
Comprehensiveness	23.2%	**76.8%**	28.8%	**71.2%**	18.4%	**81.6%**	28.5%	**71.5%**	24.7%	**75.3%**
Diversity	8.0%	**92.0%**	4.8%	**95.2%**	4.0%	**96.0%**	9.8%	**90.2%**	6.7%	**93.4%**
Empowerment	18.4%	**81.6%**	20.8%	**79.2%**	13.6%	**86.4%**	26.8%	**73.2%**	19.9%	**80.1%**
Overall	17.6%	**82.4%**	23.2%	**76.8%**	13.6%	**86.4%**	26.0%	**74.0%**	20.1%	**79.9%**
	HyDE	ReGraphRAG	HyDE	ReGraphRAG	HyDE	ReGraphRAG	HyDE	ReGraphRAG	HyDE	ReGraphRAG
Comprehensiveness	31.2%	**68.8%**	36.0%	**64.0%**	32.0%	**68.0%**	26.8%	**73.2%**	31.5%	**68.5%**
Diversity	17.6%	**82.4%**	15.2%	**84.8%**	7.2%	**92.8%**	10.6%	**89.4%**	12.7%	**87.4%**
Empowerment	26.4%	**73.6%**	28.8%	**71.2%**	26.4%	**73.6%**	26.0%	**74.0%**	26.9%	**73.1%**
Overall	27.2%	**72.8%**	30.4%	**69.6%**	25.6%	**74.4%**	24.4%	**75.6%**	26.9%	**73.1%**
	GraphRAG	ReGraphRAG	GraphRAG	ReGraphRAG	GraphRAG	ReGraphRAG	GraphRAG	ReGraphRAG	GraphRAG	ReGraphRAG
Comprehensiveness	20.0%	**80.0%**	35.2%	**64.8%**	32.0%	**68.0%**	35.8%	**64.2%**	30.8%	**69.3%**
Diversity	10.4%	**89.6%**	22.4%	**77.6%**	8.0%	**92.0%**	19.5%	**80.5%**	15.1%	**84.9%**
Empowerment	19.2%	**80.8%**	32.8%	**67.2%**	30.4%	**69.6%**	42.3%	**57.7%**	31.2%	**68.8%**
Overall	17.6%	**82.4%**	33.6%	**66.4%**	30.4%	**69.6%**	36.6%	**63.4%**	29.6%	**70.5%**
	LightRAG	ReGraphRAG	LightRAG	ReGraphRAG	LightRAG	ReGraphRAG	LightRAG	ReGraphRAG	LightRAG	ReGraphRAG
Comprehensiveness	20.0%	**80.0%**	30.4%	**69.6%**	23.2%	**76.8%**	26.8%	**73.2%**	25.1%	**74.9%**
Diversity	8.0%	**92.0%**	6.4%	**93.6%**	8.0%	**92.0%**	8.1%	**91.9%**	7.6%	**92.4%**
Empowerment	16.8%	**83.2%**	27.2%	**72.8%**	17.6%	**82.4%**	26.8%	**73.2%**	22.1%	**77.9%**
Overall	15.2%	**84.8%**	26.4%	**73.6%**	17.6%	**82.4%**	25.2%	**74.8%**	21.1%	**78.9%**

Table 1: Pairwise win rates of ReGraphRAG against four strong baselines (NaïveRAG, HyDE, GraphRAG, LightRAG) across four domains: Agriculture, Legal, Computer Science (CS), and a Mixed domain sampled from the Ultradomain dataset. Evaluation is conducted along four dimensions: Comprehensiveness, Diversity, Empowerment, and Overall quality. ReGraphRAG consistently outperforms all baselines, achieving particularly strong gains in Diversity and Overall scores. The rightmost column reports average win rates across all domains.

address this, we design a prompt that facilitates chain-of-thought reasoning using only the minimal necessary information encoded in the knowledge graph. Specifically, each triplet is formatted as:

`[Perspective]: <Node_i> is connected to <Node_j> with {Edge Description} relation.`

This format explicitly defines the semantic context—denoted by the [Perspective]—in which the relationship is grounded. Triplets are grouped by their corresponding perspective, and if a triplet occurs in multiple perspectives, it is included only once under the unified [Across all] perspective. This deduplication strategy preserves semantic coverage across the m generated knowledge graphs while effectively eliminating redundancy.

By compressing the node and edge information extracted through our Perspective Expansion and Graph Reorganization procedures, this instruction format significantly reduces the token length required for LLM input—resulting in a more efficient and scalable prompting scheme.

5 Experiments

5.1 Experimental Setup

Baselines: Although numerous graph-based RAG systems have been recently proposed, we exclude those lacking publicly available code, as their inclusion would hinder fair and reproducible comparisons. We evaluate ReGraphRAG against several strong and widely recognized baseline models, including NaiveRAG (Gao et al., 2023b), GraphRAG (Edge et al., 2024), HyDE (Gao et al., 2023a), and LightRAG (Guo et al., 2024). Further details regarding the selection and exclusion criteria for the baseline methods are provided in Appendix B.

Datasets and metrics: For a fair comparison, we apply the evaluation protocol of LightRAG, utilizing the Ultradomain (Qian et al., 2025), which is sourced from college textbooks spanning 18 distinct domains. Our evaluation focuses on three specific domains—Agriculture, Computer Science, and Legal—as well as a Mix domain that includes representative samples from various fields.

We also follow LightRAG's LLM-based evaluation method (Zheng et al., 2023; Gu et al., 2024), which performs pairwise comparison to determine which of two responses—generated by different RAG systems—is superior. The language model evaluates the responses across four dimensions: Comprehensiveness, Diversity, Empowerment, and Overall quality. For each dimension, the win rate is computed based on the number of times a system's response is preferred.

A detailed statistical analysis of the dataset is provided in Appendix C, and further explanations of the evaluation metrics are presented in Appendix D.

Implementation Details: To ensure a rigorous and fair comparison, LightRAG and ReGraphRAG share the same knowledge graph indexed from the underlying documents. This setup allows both models to operate over an identical retrieval space. For ReGraphRAG, we set the number of perspectives

	Agriculture		Legal		CS		Mix		Average Win Rate	
	w/o P-exp	ReGraphRAG	w/o P-exp	ReGraphRAG	w/o P-exp	ReGraphRAG	w/o P-exp	ReGraphRAG	w/o P-exp	ReGraphRAG
Comprehensiveness	38.4%	**61.6%**	42.4%	**57.6%**	41.6%	**58.4%**	42.4%	**57.6%**	41.2%	**58.8%**
Diversity	29.6%	**70.4%**	33.6%	**66.4%**	27.2%	**72.8%**	38.4%	**61.6%**	32.2%	**67.8%**
Empowerment	34.4%	**65.6%**	35.2%	**64.8%**	40.0%	**60.0%**	40.0%	**60.0%**	37.4%	**62.6%**
Overall	35.2%	**64.8%**	36.0%	**64.0%**	40.0%	**60.0%**	40.8%	**59.2%**	38.0%	**62.0%**
	w/o Reorg	ReGraphRAG	w/o Reorg	ReGraphRAG	w/o Reorg	ReGraphRAG	w/o Reorg	ReGraphRAG	w/o Reorg	ReGraphRAG
Comprehensiveness	48.8%	**51.2%**	48.8%	**51.2%**	46.8%	**53.2%**	48.0%	**52.0%**	48.1%	**51.9%**
Diversity	48.8%	**51.2%**	48.0%	**52.0%**	47.6%	**52.4%**	48.4%	**51.6%**	48.2%	**51.8%**
Empowerment	44.8%	**55.2%**	47.8%	**52.2%**	43.5%	**56.5%**	41.8%	**58.2%**	44.5%	**55.5%**
Overall	46.4%	**53.6%**	46.0%	**54.0%**	45.2%	**54.8%**	46.4%	**53.6%**	46.0%	**54.0%**
	w/o Rerank	ReGraphRAG	w/o Rerank	ReGraphRAG	w/o Rerank	ReGraphRAG	w/o Rerank	ReGraphRAG	w/o Rerank	ReGraphRAG
Comprehensiveness	**52.0%**	48.0%	**52.8%**	47.2%	**52.0%**	48.0%	**52.0%**	48.0%	**52.2%**	47.8%
Diversity	**52.0%**	48.0%	**56.0%**	44.0%	**54.4%**	45.6%	**56.1%**	43.9%	**54.6%**	45.4%
Empowerment	**52.8%**	47.2%	**54.4%**	45.6%	**52.0%**	48.0%	**57.7%**	42.3%	**54.2%**	45.8%
Overall	**54.4%**	45.6%	**54.4%**	45.6%	**52.8%**	47.2%	**54.5%**	45.5%	**54.0%**	46.0%

Table 2: Ablation study evaluating the contribution of each component in ReGraphRAG: Perspective Expansion (P-exp), Graph Reorganization (Reorg), and Query-aware Reranking (Rerank). For each domain, we report win rates of ReGraphRAG and its ablated variants across four dimensions: Comprehensiveness, Diversity, Empowerment, and Overall. Removing Perspective Expansion leads to the largest performance drop, particularly in Diversity, highlighting its importance in retrieving semantically rich information. Graph Reorganization also contribute consistently across domains by enhancing logical coherence and relevance.

$m = 4$ and generate 3 sub-queries per perspective. These hyperparameters were selected based on the point of performance convergence observed in our analysis (Section 5.5). To ensure a comparable total number of retrieved granularities with LightRAG, we extract the top 30 nodes from the knowledge graph database for each subquery. All RAG systems are configured with "GPT-4o-mini" as the language model and "text-embedding-3-small" as the embedding model for all components, using a chunk size of 1200 consistently across all datasets.

5.2 Main Results

Table 1 presents the pairwise comparison results between ReGraphRAG and each baseline across four distinct domain-specific datasets. Empirically, ReGraphRAG demonstrates consistently strong performance across all domains. As shown by the Average Win Rate, it outperforms all baselines by a margin exceeding 70%, with particularly notable gains in the Diversity approaching a 90% win rate.

The remarkably high Diversity score, compared to Comprehensiveness and Empowerment, underscores the strength of the Perspective Expansion module in retrieving knowledge from varied interpretive angles. However, increasing Diversity alone can sometimes lead to less coherent or harder-to-follow responses, which may negatively affect Empowerment. This trade-off is effectively mitigated by the Graph Reorganization component, which integrates fragmented subgraphs into a coherent and logically structured graph. By restoring information flow and reinforcing reasoning paths, Graph Reorganization plays a crucial role in preserving Empowerment while maintaining the benefits of high Diversity.

To further assess the individual contributions of each component, we conduct an ablation study, with detailed results presented in the following section.

5.3 Ablation Study

We conduct an ablation study to evaluate the impact of each ReGraphRAG component on answer generation. Using the same hyperparameters as the main model, we remove one module at a time and compare performance pairwise with the full model (Table 2). P-exp, Reorg, and Rerank denote Perspective Expansion, Graph Reorganization, and Query-aware Reranking, respectively.

Removing Perspective Expansion leads to consistent performance drops across all domains and metrics, with Diversity showing the most significant decline (32.3% average win rate), highlighting its key role in enriching responses with diverse information. Excluding Graph Reorganization leads to a notable degradation in Empowerment (44.5% average win rate), which evaluates how clearly and coherently the answer is articulated. This suggests that while the key content remains relatively intact, the logical structure and clarity of the answer suffer without this component.

5.4 Discussion

Unexpectedly, the model without Query-aware Reranking performs better. This may stem from the graph structure or limitations in prompt tuning. As our prompts preserve graph form via triplets (section 4.4) without deeply modeling multi-hop priorities, reranking effects may be diminished. While

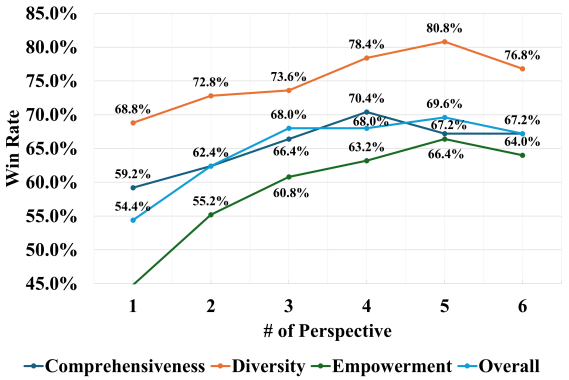

Figure 3: Win rate of ReGraphRAG compared to GraphRAG as **the number of perspectives** increases, measured across four evaluation dimensions: Comprehensiveness, Diversity, Empowerment, and Overall.

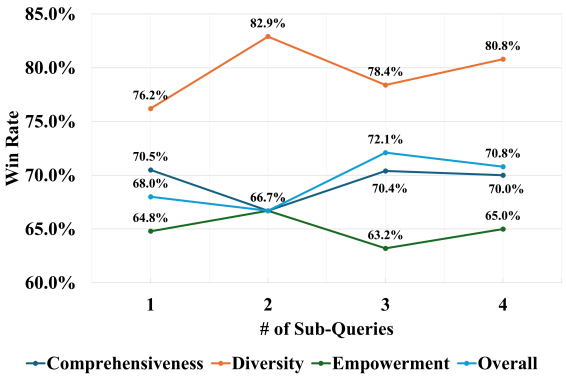

Figure 4: Win rate of ReGraphRAG compared to GraphRAG as **the number of sub-queries** per perspective increases, evaluated across Comprehensiveness, Diversity, Empowerment, and Overall dimensions.

reranking is common in graph-based RAG systems (Glass et al., 2022; Chen et al., 2024; Sun et al., 2025), its interaction with graph-structured prompt design remains underexplored—an area we leave for future work.

5.5 Analysis

In this section, we analyze the performance variation of ReGraphRAG with respect to the number of perspectives and sub-queries used during inference. The evaluation is conducted on the Mix domain subset of the Ultradomain dataset. For each hyperparameter setting, we perform a pairwise comparison between the responses generated by ReGraphRAG and those of GraphRAG, measuring relative performance. We vary the number of perspectives from 1 to 6 and the number of sub-queries per perspective from 1 to 4, while keeping all other hyperparameters consistent with those used in the main experimental results.

Number of Perspectives: As shown in Figure 3, increasing the number of perspectives generally leads to performance improvements across all evaluation criteria. Performance tends to converge around four perspectives, suggesting that incorporating multiple perspectives enhances response quality up to a point, after which marginal gains diminish. Specifically, ReGraphRAG consistently outperforms GraphRAG, particularly in terms of diversity. Even with only one perspective, ReGraphRAG achieves a high diversity win rate (68.8%) due to the presence of three subqueries, which ensures that multiple evidence paths are still explored.
As the number of perspectives increases, the comprehensiveness and empowerment scores also rise steadily. This indicates that incorporating diverse interpretive angles allows the model to generate responses that are both more detailed and more helpful for the user. For instance, when answering a question like "What are the implications of climate change on agriculture?", considering perspectives such as economic impact, ecological transformation, and food security allows the system to construct a more nuanced and informative answer.
Number of Sub Queries: Figure 4 presents the changes in win rate as a function of the number of sub-queries generated per perspective. As the number of sub-queries increases, we observe a modest improvement in Diversity and Overall scores. However, a substantial increase in sub-query count within a single perspective appears to offer diminishing returns in terms of Comprehensiveness and Empowerment. This suggests that excessively expanding sub-queries within the same interpretive angle may introduce redundancy rather than enhancing informativeness or clarity. Based on these findings, we set the number of sub-queries $n = 3$ in ReGraphRAG, as it yields the highest overall win rate.

6 Conclusion

In this work, we presented ReGraphRAG, a novel graph-based RAG framework designed to address the fragmentation problem inherent in LLM-generated knowledge graphs. By integrating three key components—Perspective Expansion, Graph Reorganization, and Query-aware Reranking—ReGraphRAG reconstructs and enriches disconnected knowledge graphs into coherent and inference-friendly structures. Our extensive experiments on diverse benchmark domains demonstrate that ReGraphRAG consistently outperforms strong baselines in all evaluation dimensions, especially

achieving substantial gains in diversity and empowerment.

Limitations

The effectiveness of the Query-aware Reranking module remains limited. As observed in our ablation study, removing the reranking step occasionally led to better performance, contrary to expectations. This outcome implies that the current reranking strategy—based on cosine similarity between edge embeddings and query representations—may not fully capture the multi-hop reasoning potential afforded by graph-structured information. As also noted in the ablation analysis (section 5.3), research focusing on the interplay between graph-structured prompt design and reranking in graph-based RAG systems remains scarce. We leave this exploration as an important direction for future work.

The framework incurs computational overhead due to its multi-perspective expansion strategy. Decomposing the original query into multiple interpretive angles and generating several sub-queries per perspective increases both retrieval and inference time. While this approach enhances diversity and semantic coverage, it may limit the practicality of ReGraphRAG in real-time or resource-constrained environments. Future work may focus on adaptive perspective selection or budget-aware expansion to balance performance gains with efficiency.

Acknowledgments

This work was supported by the Institute of Information and communications Technology Planning and evaluation (IITP) grant (No.RS-2025-25422680, No. RS-2020-II201373), and the National Research Foundation of Korea (NRF) grant (No. RS-2025-00520618) funded by the Korean Government (MSIT).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2025. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *arXiv preprint arXiv:2502.14902*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. 2025. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2g: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current

landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Desta Haileselassie Hagos, Rick Battle, and Danda B Rawat. 2024. Recent advances in generative ai and large language models: Current status, challenges, and perspectives. *IEEE Transactions on Artificial Intelligence*.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, and 1 others. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024a. Does prompt formatting have any impact on llm performance? *arXiv preprint arXiv:2411.10541*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024b. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Ziniu Hu, Yichong Xu, Wenhao Yu, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Kai-Wei Chang, and Yizhou Sun. 2022. Empowering language models with knowledge graph reasoning for open-domain question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9562–9581.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Dongwoo Kim, Lexing Xie, and Cheng Soon Ong. 2016. Probabilistic knowledge graph construction: Compositional and incremental approaches. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2257–2262.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, and 1 others. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*.

Guliu Liu and Lei Li. 2020. Knowledge fragment cleaning in a genealogy knowledge graph. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*, pages 521–528.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

LINHAO LUO, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Andrea Matarazzo and Riccardo Torlone. 2025. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*.

Lars-Peter Meyer, Claus Stadler, Johannes Frey, Norman Radtke, Kurt Junghanns, Roy Meissner, Gordian Dziwis, Kirill Bulert, and Michael Martin. 2023. Llm-assisted knowledge graph engineering: Experiments with chatgpt. In *Working conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow*, pages 103–115. Springer Fachmedien Wiesbaden Wiesbaden.

Jeff Z Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, and 1 others. 2023. Large language models and knowledge graphs: Opportunities and challenges. *arXiv preprint arXiv:2308.06374*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Joon Park, Kyohei Atarashi, Koh Takeuchi, and Hisashi Kashima. 2025. Emulating retrieval augmented generation via prompt engineering for enhanced long context comprehension in llms. *arXiv preprint arXiv:2502.12462*.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pages 2366–2377.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Jiashuo Sun, Xianrui Zhong, Sizhe Zhou, and Jiawei Han. 2025. Dynamicrag: Leveraging outputs of large language model as feedback for dynamic reranking in retrieval-augmented generation. *arXiv preprint arXiv:2505.07233*.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models. *arXiv preprint arXiv:2305.04676*.

Tianyang Xu, Haojie Zheng, Chengze Li, Haoxiang Chen, Yixin Liu, Ruoxi Chen, and Lichao Sun. 2025. Noderag: Structuring graph-based rag with heterogeneous nodes. *arXiv preprint arXiv:2504.11544*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Yuzhe Zhang, Yipeng Zhang, Yidong Gan, Lina Yao, and Chen Wang. 2024. Causal graph discovery with retrieval-augmented generation based large language models. *arXiv preprint arXiv:2402.15301*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Overview of the used prompts

A.1 Entity and Relationship Extraction Prompt

-Goal-
Given a text document that is potentially relevant to this activity and a list of entity types, identify all entities of those types from the text and all relationships among the identified entities.
Use {language} as output language.

-Steps-
1. Identify all entities. For each identified entity, extract the following information:
- entity_name: Name of the entity, use same language as input text. If English, capitalized the name.
- entity_type: One of the following types: [{entity_types}]
- entity_description: Comprehensive description of the entity's attributes and activities
Format each entity as ("entity"{tuple_delimiter}<entity_name>{tuple_delimiter}<entity_type>{tuple_delimiter}<entity_description>)

2. From the entities identified in step 1, identify all pairs of (source_entity, target_entity) that are *clearly related* to each other.
For each pair of related entities, extract the following information:
- source_entity: name of the source entity, as identified in step 1
- target_entity: name of the target entity, as identified in step 1
- relationship_description: explanation as to why you think the source entity and the target entity are related to each other
- relationship_strength: a numeric score indicating strength of the relationship between the source entity and target entity
- relationship_keywords: one or more high-level key words that summarize the overarching nature of the relationship, focusing on concepts or themes rather than specific details
Format each relationship as ("relationship"{tuple_delimiter}<source_entity>{tuple_delimiter}<target_entity>{tuple_delimiter}<relationship_description>{tuple_delimiter}<relationship_keywords>{tuple_delimiter}<relationship_strength>)

3. Identify high-level key words that summarize the main concepts, themes, or topics of the entire text. These should capture the overarching ideas present in the document.
Format the content-level key words as ("content_keywords"{tuple_delimiter}<high_level_keywords>)

4. Return output in {language} as a single list of all the entities and relationships identified in steps 1 and 2. Use **{record_delimiter}** as the list delimiter.

5. When finished, output {completion_delimiter}

######################
-Examples-
######################
{examples}

##############################
-Real Data-
######################
Entity_types: {entity_types}
Text: {input_text}
######################
Output:

Entity & Relationship Extraction Prompt

A.2 Perspective Expansion Prompt

---Role---
You are a helpful assistant tasked with identifying four different perspectives in the user's query and conversation history and writing three questions what knowledge you need to gain to answer the query.

---Goal---
Given the query and conversation history, list three questions from **four different perspectives**. Each perspective should be clearly distinct from the others, and each question within a perspective should follow a sequential set of steps to address the original query.

---Instructions----
Consider both the current query and relevant conversation history when extracting keywords
Output the keywords in JSON format
The JSON should have eight keys: **"first_perspective", "second_perspective", "third_perspective", "fourth_perspective"**

-Examples-

Example 1:
Query: "How does international trade influence global economic stability?"
Output: {{ "first_perspective": {"International Trade and Financial Market Stability": ["What specific aspects of international trade significantly influence financial market stability?", "How do fluctuations in trade volumes impact currency valuation and exchange rates?", "What historical cases show how trade disruptions led to financial market instability?"]}, "second_perspective": {...}, ... , "fourth_perspective": {...}}}

Example 2:
Query: "What are the environmental consequences of deforestation on biodiversity?"
Output: {{ "first_perspective": {"Impact on Ecosystem Stability": ["Which ecosystems are primarily affected by deforestation?", "How does deforestation disrupt interactions among species within these ecosystems?", "What are the long-term implications of disrupted ecosystems for overall biodiversity?"]}, "second_perspective": {...}, ... , "fourth_perspective": {...}}}

-Real Data-
Conversation History: {history}
Current Query: {query}
Output: Query Expansion Input Prompt

Perspective Expansion Instruction Prompt

The `Output` should be human text, not unicode characters. Keep the same language as `Query`.

A.3 Response Prompt

---Role---

You are an expert assistant responding accurately and comprehensively to user queries based on the provided **Perspective-wise Knowledge Graph**.

---Goal---

Your goal is to synthesize a concise, accurate response that strictly adheres to the provided **Perspective-wise Knowledge Graph** and clearly addresses the user's current query.
You must reflect and integrate the provided "Perspectives" into your response to fully cover the query from multiple angles.

---Instructions---

Summarize: Provide a clear and concise summary that fully encapsulates all relevant information from the **Perspective-wise Knowledge Graph.**
Perspective Integration: Clearly indicate each **"Perspective"** given in the **Perspective-wise Knowledge Graph** and address the query specifically through these lenses.
Clarity and Precision: Use precise and unambiguous language.
Continuity: If relevant, consider previous conversation history to maintain contextual continuity.
General Knowledge: Include general knowledge contextually related to the **Perspective-wise Knowledge Graph** only if it enhances clarity or explanation, but ensure no contradiction or addition of new factual claims not explicitly stated in the Perspective-wise Knowledge Graph.
Response Format:
- Use markdown formatting.
- Employ clear headings for different perspectives.
- Answer in the same language as the user's question.

---Constraints---

Do NOT fabricate information or introduce details not explicitly stated or implied in the **Perspective-wise Knowledge Graph.**
If information is insufficient to respond fully, clearly state the limitation instead of guessing.

---Target response length and format---
{response_type}

---Conversation History---
{history}

---**Perspective-wise Knowledge Graph**---
{context_data}

RAG Response Instruction Prompt

A.4 LLM-based Evaluation Prompt

---Role---
You are an expert tasked with evaluating two answers to the same question based on three criteria: **Comprehensiveness**, **Diversity**, and **Empowerment**.

You will evaluate two answers to the same question based on three criteria: **Comprehensiveness**, **Diversity**, and **Empowerment**.

- **Comprehensiveness**: How much detail does the answer provide to cover all aspects and details of the question?
- **Diversity**: How varied and rich is the answer in providing different perspectives and insights on the question?
- **Empowerment**: How well does the answer help the reader understand and make informed judgments about the topic?

For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain why. Then, select an overall winner based on these three categories.

Here is the question: {query}

Here are the two answers:
Answer 1:
{answer1}

Answer 2:
{answer2}

Evaluate both answers using the three criteria listed above and provide detailed explanations for each criterion.

Output your evaluation in the following JSON format:
{{
"Comprehensiveness": {{ "Winner": "[Answer 1 or Answer 2]", "Explanation": "[Provide explanation here]" }},
"Diversity": {{ "Winner": "[Answer 1 or Answer 2]", "Explanation": "[Provide explanation here]" }},
"Empowerment": {{ "Winner": "[Answer 1 or Answer 2]", "Explanation": "[Provide explanation here]" }},
"Overall Winner": {{ "Winner": "[Answer 1 or Answer 2]", "Explanation": "[Summarize why this answer is the overall winner based on the three criteria]" }}
}}

Response Evaluation Prompt

B Explanation for Baseline

B.1 Selected Baseline

NaiveRAG (Lewis et al., 2020; Gao et al., 2023b) is the most standard baseline. It separates the corpus into chunks, converts them into embedding vectors, stores them in the database, and performs retrieval based on the similarity of the embedding vectors to the query. Since it is an intuitive and efficient method, it is still the choice of many studies.
HyDE (Gao et al., 2023a) is one of the advanced versions of NaiveRAG, which follows the method of generating hypotheses by looking at the query and then performing retrieval on the embedding vector database with the hypotheses. HyDE can be interpreted as an extension of query expansion method, and it has shown an effective way to retrieve documents that do not have relevance labels. The license for HyDE is unknown, but we denote the source code URL: https://github.com/texttron/hyde
GraphRAG (Edge et al., 2024) extracts entities and relationships from the corpus with LLM to build a database with a graph structure. Then, it uses Leiden community detection[Leiden] to community the graph and generates community summaries from leaf-level to higher-level. After that, the final answer is generated by the map-reduce step. GraphRAG is released under the MIT License.
LightRAG (Guo et al., 2024) is the successor of GraphRAG. LightRAG builds a framework that is much more efficient than GraphRAG by using key-value pair generation and dual-level retrieval instead of summarization after graph indexing. This is the most direct baseline for comparison with our work. LightRAG is released under the MIT License.

B.2 Excepted Baseline

PathRAG (Chen et al., 2025) proposed a methodology that applied pruning techniques to retain only the information that is core to generating answers, revealing that excessive retrieved information actually degrades the performance of the generator. However, we decided to exculde it from the baseline due to incomplete publicly available code.

NodeRAG (Xu et al., 2025) uses a heterogeneous KGC method and unified-level information retrieval to achieve higher performance than GraphRAG, LightRAG, etc. However, we decided to exculde it from the baseline due to incomplete publicly available code.

Statistics	Agriculture	CS	Legal	Mix
Total Documents	12	10	94	61
Total Tokens	1,923,151	2,039,189	4,719,432	602,560
Max Tokens	378,588	433,563	79,095	18,797
Min Tokens	75,907	51,677	31,778	1,848

Table 3: Statistics of the Ultradomain dataset

C Explanation for Dataset

We used the Ultradomain dataset (Qian et al., 2025), which has been used in several graph-based RAG studies, including LightRAG. Ultradomain is a domain-specific QA benchmark built from college textbooks. It consists of a total of 20 QA datasets, and in this study, we chose 4 datasets: Agriculture, Computer Science, Legal, and Mix, as per LightRAG's method. Among them, the Legal Dataset consists of legal contracts and requires professional and structured answers. Ultradomain is released under the Apache License 2.0. Table 3 shows the statistical information such as the number of documents and tokens for each dataset.

D Explanation for Evaluation Metric

There are many methods for LLM evaluations, but the LLM-as-a-Judge method (Zheng et al., 2023; Gu et al., 2024), which directly compares two answers, is evaluated as objective and efficient. In this study, we used the same evaluation metrics of Comprehensiveness, Diversity, and Empowerment for fair comparison with Edge et al. (2024) and Guo et al. (2024). The three metrics can evaluate the completeness of the answer in different ways, and LLMs should compare two answers for each of the three metrics, choose the better answer, and give reasons. Finally, an overall winner is selected. The prompt used in the evaluation procedure is provided in Appendix A.4.

Model	Time Cost (sec)	Token Cost
Naïve	0.8	3,800
GraphRAG	8.4	360,000
LightRAG	7.2	29,000
ReGraphRAG w/o expansion	4.8	3,700
ReGraphRAG (full)	19.8	18,000

Table 4: Comparison of approximate time cost and token cost across representative baselines

E Computational Efficiency

As shown in Table 4, GraphRAG achieves efficient response generation through multithreaded retrieval and generation from multiple communities, but this strategy incurs extremely high token costs (360,000 tokens). LightRAG, although faster, retrieves 1-hop nodes from fragmented graphs and requires a larger set of nodes and context to compensate for the lack of coherent multi-hop structure. Despite this, its performance remains limited, as demonstrated in our main results. In contrast, ReGraphRAG without perspective expansion already achieves lower time and token costs, suggesting that our graph reorganization strategy can be applied selectively for efficiency gains. While the full ReGraphRAG with multi-perspective expansion shows higher time cost (19.8 sec), it is worth noting that this value reflects a sequential execution setup. Adopting the same multithreaded parallel retrieval strategy used by GraphRAG could reduce time cost to the 4–5 sec range without changing the algorithm itself. Importantly, even with additional processing steps, our method requires far fewer tokens (18,000) compared to GraphRAG and LightRAG, indicating a more efficient use of prompt resources relative to performance.

F Case Study

Query : How can understanding bee behavior improve hive management practices?

GraphRAG

Introduction : Understanding bee behavior is crucial for effective hive management. By observing and interpreting the behaviors of bees ...
Enhancing Colony Health Recognition of stress signs or disease behaviors in bees enables timely interventions. Beekeepers can address issues such as pest infestations, including Varroa mites, before they negatively impact the hive. By understanding the social dynamics and needs of the colony, beekeepers can...
Optimizing Resource Management ... **For instance, when beekeepers align** feeding schedules with bees' active foraging periods, they ensure that colonies receive adequate nutrition. **Additionally, awareness of seasonal shifts** in bee activity allows beekeepers to manage food stores effectively, preventing starvation during low-resource periods.
Supporting Reproductive Success ... The Queen's health directly influences the hive's population and structure. Beekeepers who prioritize the well-being of the Queen can enhance colony stability and productivity. ...
Pest Management Strategies Awareness of threats such as pests and diseases informs ...

ReGraphRAG

Understanding bee behavior is crucial for improving hive management practices in several ways, which can be examined **through the following perspectives**:\
Hive Management Understanding bee behavior enhances hive management... **For example, recognizing how bees** communicate through pheromones and dances can inform beekeepers about resource availability and hive health. ... **Additionally, by observing behaviors related** to foraging, cleaning, and brood care, beekeepers can align ...
Pest and Disease Management Awareness of bee behavior plays a vital role in pest and disease management. **For instance, hygienic behavior**—where bees remove diseased or dead brood—can be encouraged through appropriate hive condition ... By supporting this natural behavior, beekeepers can reduce the incidence of diseases like Chalkbrood. **Furthermore, understanding how stress** factors or disturbances affect bee behavior can help prevent problems ...
Beekeeping Management Techniques ... By understanding the natural instincts of bees ... , beekeepers can make informed decisions about hive placement, resource allocation, and seasonal preparations. **For example, ensuring that hives are** adequately ventilated and insulated according to the bees' behavioral responses ...
Behavioral Adaptation to Environmental Changes Bees exhibit resilience through behavioral adaptations

LLM Decision

Comprehensiveness *Answer 2* provides a broader and deeper exploration of how understanding bee behavior can inform hive management. *It touches on five distinct domains* (hive management, pest and disease management, beekeeping techniques, behavioral adaptation to environmental changes, and pollination services) *and offers specific examples within each, showing how behavioral insights translate to practice*. This makes the response highly detailed and well-rounded.
Diversity *Answer 2 demonstrates greater diversity by covering a wider range of perspectives, including pollination services and environmental adaptation—topics not directly addressed in Answer 1*. It presents both internal hive dynamics and external agricultural implications, offering a richer array of angles from which bee behavior impacts management.
Empowerment *Answer 2 better empowers the reader by linking behavioral insights to actionable strategies across various areas.* The explanation of how specific behaviors (e.g., hygienic behavior, temperature regulation, foraging) can guide interventions gives beekeepers a clear sense of how to make informed decisions. The multi-perspective framing also helps readers anticipate challenges and opportunities in diverse beekeeping contexts.
Overall Winner *Answer 2 is the overall winner because it excels in all three evaluation criteria.* It offers more comprehensive coverage, explores a wider diversity of perspectives, and provides practical, empowering guidance. While *Answer 1 is structured and informative, it is more limited in scope and depth* compared to the nuanced and multifaceted insights presented in Answer 2.