arXiv:2508.10391v4 [cs.AI] 12 Nov 2025

LeanRAG: Knowledge-Graph-Based Generation with Semantic Aggregation and Hierarchical Retrieval

Yaoze Zhang[1,2*], **Rong Wu**[1,3*], **Pinlong Cai**[1†], **Xiaoman Wang**[4], **Guohang Yan**[1]
Song Mao[1], **Ding Wang**[1], **Botian Shi**[1]

[1]Shanghai Artificial Intelligence Laboratory
[2]University of Shanghai for Science and Technology
[3]Zhejiang University
[4]East China Normal University

Abstract

Retrieval-Augmented Generation (RAG) plays a crucial role in grounding Large Language Models by leveraging external knowledge, whereas the effectiveness is often compromised by the retrieval of contextually flawed or incomplete information. To address this, knowledge graph-based RAG methods have evolved towards hierarchical structures, organizing knowledge into multi-level summaries. However, these approaches still suffer from two critical, unaddressed challenges: high-level conceptual summaries exist as disconnected "semantic islands", lacking the explicit relations needed for cross-community reasoning; and the retrieval process itself remains structurally unaware, often degenerating into an inefficient flat search that fails to exploit the graph's rich topology. To overcome these limitations, we introduce LeanRAG, a framework that features a deeply collaborative design combining knowledge aggregation and retrieval strategies. LeanRAG first employs a novel semantic aggregation algorithm that forms entity clusters and constructs new explicit relations among aggregation-level summaries, creating a fully navigable semantic network. Then, a bottom-up, structure-guided retrieval strategy anchors queries to the most relevant fine-grained entities and then systematically traverses the graph's semantic pathways to gather concise yet contextually comprehensive evidence sets. The LeanRAG can mitigate the substantial overhead associated with path retrieval on graphs and minimize redundant information retrieval. Extensive experiments on four challenging QA benchmarks with different domains demonstrate that LeanRAG significantly outperforms existing methods in response quality while reducing 46% retrieval redundancy. Our code is available at: https://github.com/RaZzzyz/LeanRAG.

Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation. Yet their effectiveness is often undermined by their static internal knowledge, leading to factual inaccuracies and hallucinations (Huang et al. 2025b; Li et al. 2024). Retrieval-Augmented Generation (RAG) was introduced as a potential solution, dynamically grounding LLMs in external, up-to-date information (Gao et al. 2023). However, the effectiveness of naive RAG approaches is frequently compromised. The retrieved text chunks often lack precise alignment with the user's true intent, and the reliance on embedding-based similarity alone is often insufficient to capture the deep semantic relevance required for complex reasoning, resulting in responses that are either incomplete or contextually flawed (Zhao et al. 2024; Wang et al. 2025).

To overcome the limitations of unstructured retrieval, researchers have increasingly explored knowledge graph-based RAG methods. Initial efforts, such as GraphRAG (Edge et al. 2024), successfully organized documents into community-based knowledge graphs, which helped preserve local context better than disconnected text chunks. However, these methods often generated large, coarse-grained communities, leading to significant information redundancy during retrieval. Subsequently, more advanced works like HiRAG (Huang et al. 2025a) refined this paradigm by introducing hierarchical structures, clustering entities into multi-level summaries. This represented a significant step forward in organizing knowledge. Despite this progress, our analysis reveals that two critical challenges remain unaddressed currently (as Figure 1 shows). First, the high-level summary nodes in these hierarchies exist as "semantic islands". They lack explicit relational connections between each other, making it hard to reason across different conceptual communities within the knowledge base. Second, the retrieval process itself remains structurally unaware, often degenerating into a simple semantic search over a flattened list of nodes, failing to exploit the rich topological information encoded in the graph. This leads to a retrieval process that is both inefficient and imprecise.

To address these challenges, we propose LeanRAG, a novel retrieval-augmented generation framework that synergistically integrates deeply collaborative knowledge structuring with a lean, structure-guided retrieval strategy. At its core, LeanRAG introduces a semantic aggregation algorithm that constructs a hierarchical knowledge graph by organizing retrieved entities into semantically coherent clusters. Its key innovation lies not only in clustering entities based on semantic similarity but also in automatically inferring explicit inter-cluster summary relations, leveraging the underlying knowledge's contextual and relational semantics to es-

[*]These authors contributed equally.
[†]Corresponding author

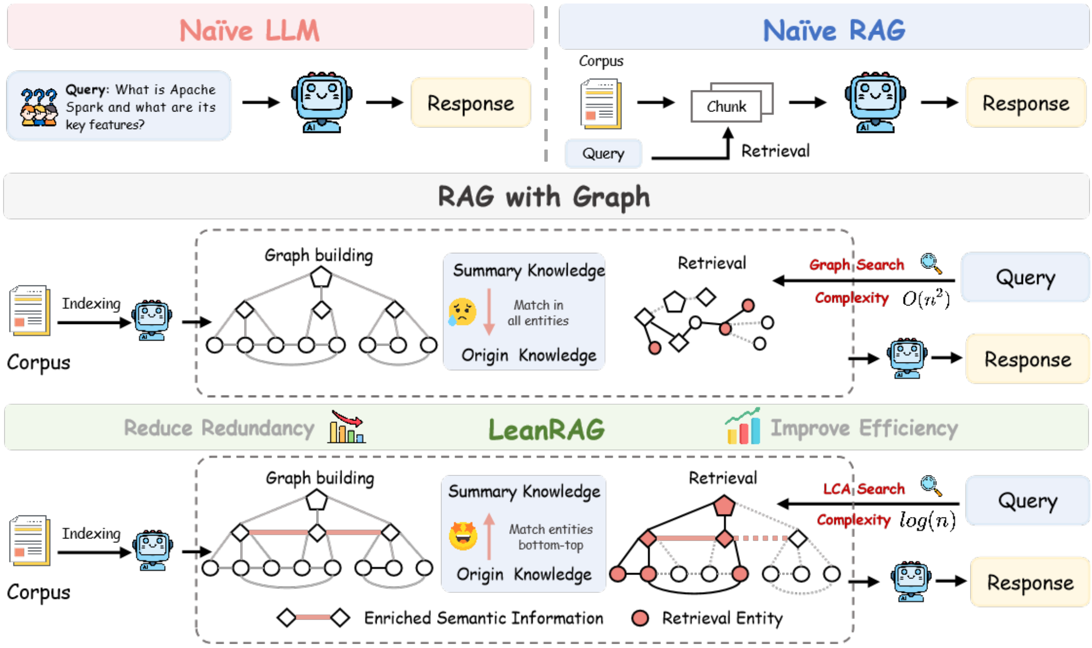

Figure 1: Comparison of typical LLM retrieval-augmented generation frameworks.

tablish higher-order abstractions. This process transforms fragmented, isolated hierarchies into a unified, fully navigable semantic network, where both fine-grained details and abstracted knowledge are seamlessly interconnected.

Building upon this enriched structure, LeanRAG employs a bottom-up, structure-aware retrieval mechanism that strategically navigates the graph to maximize relevance while minimizing redundancy. The retrieval process begins by anchoring the query to the most contextually pertinent fine-grained entities at the leaf level. It then systematically traverses relational pathways across both the original entity layer and the derived summary layer, propagating evidence upward through the hierarchy. This dual-level traversal ensures that the retrieved evidence set is not only concise and focused but also contextually comprehensive, capturing both specific details and broader conceptual relations essential for accurate and coherent generation.

Our primary contributions can be summarized as follows:

- A novel semantic aggregation algorithm designed for superior knowledge condensation. This method constructs a multi-resolution knowledge map by modeling and building new relational edges between summary-level conceptual nodes, effectively preserving both fine-grained facts and high-level thematic connections within a single, coherent structure.
- The introduction of a bottom-up entity retrieval strategy to mitigate information redundancy. By initiating retrieval from high-relevance "anchor" nodes and expanding context strictly along relevant semantic pathways, this strategy yields a precise and compact evidence subgraph for LLMs.
- We demonstrate through extensive experiments that LeanRAG achieves a new state-of-the-art on multiple challenging QA tasks, significantly outperforming existing methods in both response performance and efficiency.

Related Work

Retrieval-Augmented Generation

Retrieval-Augmented Generation was introduced as a powerful paradigm to mitigate the intrinsic knowledge limitations of LLMs by grounding them in external information (Lewis et al. 2020). The standard RAG framework operates by retrieving relevant text chunks from a corpus and providing them as context to an LLM for answer generation (Wang et al. 2024). While effective, this approach is fundamentally constrained by the "chunking dilemma": small, fine-grained chunks risk losing critical context, whereas larger chunks often introduce significant noise and dilute the LLM's focus (Tonellotto et al. 2024).

Substantial research has been dedicated to overcoming this limitation. One line of work improves the retriever itself, evolving from sparse methods like BM25 (Robertson, Zaragoza et al. 2009) to dense models such as DPR (Karpukhin et al. 2020) and Contriever (Izacard et al. 2021), which better capture semantic relevance. Another focuses on indexing and organizing source documents (Jiang et al. 2023), with recent methods creating hierarchical summaries of text chunks to enable multi-level retrieval. For example, RAPTOR builds a tree of recursively summarized clusters, allowing retrieval of fine-grained details and high-level summaries (Sarthi et al. 2024). However, these approaches still treat knowledge as linear or simple hierarchical structures and do not explicitly model complex, non-hierarchical relations between entities and concepts, limiting their ability to answer queries requiring reasoning over such connections—motivating KG-based RAG methods.

Knowledge Graph Based Retrieval-Augmented Generation

To better capture the relational nature of information, KG-based RAG has emerged as a prominent research direction. By representing knowledge as a graph of entities and relations, these methods aim to provide a more structured and semantically rich context for the LLM (Peng et al. 2024). Early approaches in this domain focused on leveraging graph structures for improved retrieval. For instance, GraphRAG (Edge et al. 2024) organizes documents into community-based KGs to preserve local context, while other methods like FastGraphRAG utilize graph-centrality metrics such as PageRank (Page et al. 1999) to prioritize more important nodes during retrieval. This subgraph retrieval approach has also proven effective in industrial applications like customer service, where KGs are constructed from historical support tickets to provide structured context (Xu et al. 2024). These methods marked a significant step forward by imposing a macro-structure onto the knowledge base, moving beyond disconnected text chunks.

Recognizing the need for more fine-grained control and abstraction, subsequent works have explored more sophisticated hierarchical structures. HiRAG, the current state-of-the-art, clusters entities to form multi-level summaries (Huang et al. 2025a), while LightRAG (Guo et al. 2024) proposes a dual-level framework to balance global and local information retrieval. While these hierarchical methods have progressively improved retrieval quality, a critical gap persists in how the constructed graph structures are leveraged at query time. The retrieval process is often decoupled from the indexing structure; for instance, an initial search may be performed over a "flattened" list of all nodes, rather than being directly guided by the indexed community or hierarchical relations. This decoupling means the rich structural information is primarily used for post-retrieval context expansion, rather than for guiding the initial, crucial step of identifying relevant information. This can limit performance on complex queries where the relations between entities are paramount, highlighting the need for a new paradigm where the retrieval process is natively co-designed with the knowledge structure.

Preliminary

In this section, we will introduce and give a formal definition of a RAG system with a specific knowledge graph.

Given a rich knowledge graph with the description of vertexes and relations $\mathcal{G} = (V, R, D_{(ver)}, D_{(rel)})$, where V and R denote the set of entities and relations, $D_{(ver)}$ represents the collection of entity descriptions and $D_{(rel)}$ represents the collection of relationship descriptions. The goal of KG-based RAG is to leverage existing information to build a query-relevant sub-graph that helps LLMs generate high-quality responses. Given a query q, the searching process can be formulated as:

$$\tilde{V} = \text{Top-n}_{v \in V}(\text{Sim}(q, d_v)) \tag{1}$$

where $Sim(\cdot, \cdot)$ is the embedding similarity metric function, and n is the choice number of similarity entities. Based on the metric, $\tilde{V}$ contains the top n entities. Then we can search the relational paths L between nodes $v \in \tilde{V}$. All relations r that constitute the path L belong to the relation set R.

$$L = \bigcup_{x, y \in \tilde{\mathcal{V}}} Path(x, y) = (r_1, r_2, \dots) \tag{2}$$

By leveraging $\tilde{V}$ and L, the sub-graph $\tilde{\mathcal{G}}$ is constructed to support RAG systems with focused, query-relevant, and semantically enriched knowledge retrieval.

Method

The performance of a generic KG-augmented retrieval framework is fundamentally determined by the structural and semantic quality of the underlying knowledge graph $\mathcal{G}$, as well as the precision and efficiency of the retrieval strategy. To address the limitations of a flat graph structure and naive path search strategy, we introduce **LeanRAG**, a framework built on the principle of tightly **co-designing** its aggregation and retrieval processes. As illustrated in Figure 2, LeanRAG consists of two core innovations: (1) a **Hierarchical Graph Aggregation** method that recursively builds a multi-level, navigable semantic network from the base KG; and (2) a **Structured Retrieval** strategy that leverages this hierarchy via Lowest Common Ancestor (LCA) path search approach to construct a compact and coherent context.

Hierarchical Knowledge Graph Aggregation

The foundation of LeanRAG is the transformation of a flat knowledge graph $\mathcal{G}_0$ into a multi-level, semantically rich hierarchy $\mathcal{H}$. This hierarchy allows for retrieval at varying levels of abstraction. We construct this hierarchy, denoted as $\mathcal{H} = \{\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_k\}$, in a bottom-up, layer-by-layer fashion. Each layer $\mathcal{G}_i = (V_i, R_i, D_{(ver)_i}, D_{(rel)_i})$ represents a more abstract view of the layer below it, $\mathcal{G}_{i-1}$. The core of this construction lies in a recursive aggregation process that clusters nodes based on semantic similarity and then intelligently generates new, more abstract entities and relations to form the next layer.

Recursive Semantic Clustering. Given a knowledge graph layer $\mathcal{G}_{i-1}$, the first step is to identify groups of semantically related entities that can be abstracted into a single, higher-level concept. We leverage the rich descriptive text $d_v \in D_{(ver)_{i-1}}$ associated with each entity $v \in V_{i-1}$ for this purpose. Following recent works in clustering text representation (Sarthi et al. 2024), we employ a two-step process:

1. **Semantic Embedding:** We first encode the textual description of each entity into a dense vector representation using a pre-trained embedding model $\Phi(\cdot)$. This yields a set of embeddings for the entire KG layer:

$$\mathbf{E}_{i-1} = \{\Phi(d_v) \mid v \in V_{i-1}\} \tag{3}$$

2. **Gaussian Mixture Clustering:** We then apply a Gaussian Mixture Model (GMM) (Reynolds 2015) to the set of embeddings $\mathbf{E}_{i-1}$. The GMM partitions the entities V_{i-1} into m disjoint clusters $\mathcal{C}_{i-1} = \{C_1, C_2, \dots, C_m\}$,

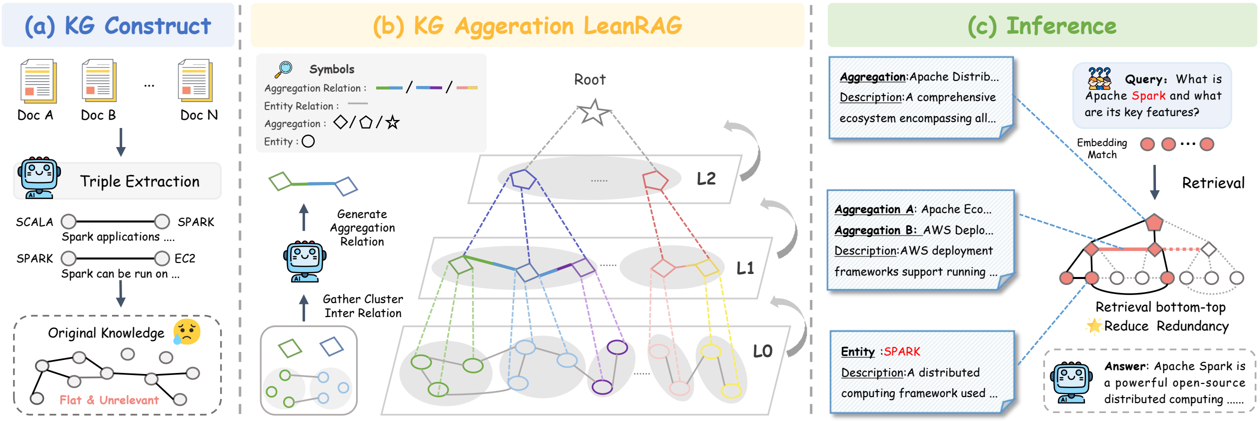
(a) KG Construct
Doc A
Doc B
...
Doc N
Triple Extraction
SCALA
SPARK
Spark applications ....
SPARK
EC2
Spark can be run on ...
Original Knowledge
Flat & Unrelevant
(b) KG Aggeration LeanRAG
Symbols
Aggregation Relation :
Entity Relation :
Aggregation : ◇/⬠/☆
Entity : ○
Generate Aggregation Relation
Gather Cluster Inter Relation
Root
L2
L1
L0
......
(c) Inference
Aggregation:Apache Distrib...
Description:A comprehensive ecosystem encompassing all...
Aggregation A: Apache Eco...
Aggregation B: AWS Deplo...
Description:AWS deployment frameworks support running ...
Entity :SPARK
Description:A distributed computing framework used ...
Query: What is Apache Spark and what are its key features?
Embedding Match
Retrieval
Retrieval bottom-top
Reduce Redundancy
Answer: Apache Spark is a powerful open-source distributed computing ......

Figure 2: Overview of the LeanRAG framework.

where each cluster C_j ($j \in [1, m]$) contains entities that are semantically similar in the embedding space.

This clustering provides a principled grouping of fine-grained entities, setting the stage for conceptual abstraction.

Generation of Aggregated Entities and Relations. A key limitation of prior hierarchical methods is that they often only cluster entities, losing the rich relational information in the process. LeanRAG overcomes this by using LLMs to intelligently generate both new entities and new relations for the subsequent layer $\mathcal{G}_i$.

Aggregated Entity Generation. For each cluster $C_j \in \mathcal{C}_{i-1}$, we generate a single, more abstract aggregated entity α_j that represents the cluster's collective semantics. This abstraction is achieved via a generation function $\mathcal{F}_{\text{entity}}$, which synthesizes a new concept by considering both the entities within the cluster and the relations that exist among them. Let R_{C_j} be the set of relations in $\mathcal{G}_{i-1}$ among entities within cluster C_j.

$$(\alpha_j, d_{\alpha_j}) = \mathcal{F}_{\text{entity}}(C_j, R_{C_j}) \tag{4}$$

The new entity set $V_i = \{\alpha_j\}_{j=1}^{m}$ and their associated descriptions $D_{V_i}\{d_{\alpha_j}\}_{j=1}^{m}$ are defined as the parent nodes of $\{C_1, C_2, \ldots, C_m\}$ in the hierarchy, i.e., the nodes located at the immediate higher level in the hierarchical structure.

In practice, the generation function $\mathcal{F}_{\text{entity}}$ is implemented by LLMs guided by a carefully designed prompt $\mathcal{P}_{\text{entity}}$. We prompt LLMs to produce a concise name for the new entity α_j and a comprehensive description d_{α_j} that summarizes its components. Each entity $v \in C_j$ is then linked to its new parent entity α_j, forming the parent-child connections in the hierarchy.

Aggregated Relation Generation. To prevent the formation of "semantic islands" at higher layers, we explicitly create new relations between the aggregated entities in V_i. This ensures that the graph remains connected and navigable at all levels of abstraction. For any pair of aggregated entities (α_j, α_k), we confirm the inter-cluster relations $R_{<C_j,C_k>}$ that contains the relations between nodes that belong to the C_j and C_k, respectively. Then, we constitute the inter-cluster aggregated relation $r_{<C_j,C_k>}$ by $R_{<C_j,C_k>}$. This paper defines the number of $R_{<C_j,C_k>}$ as the connectivity strength, $\lambda_{j,k}$. If $\lambda_{j,k}$ exceeds a dynamically defined threshold τ, we infer that a meaningful high-level relationship exists, which is summarized by the LLM-driven function $\mathcal{F}_{\text{rel}}$. Otherwise, the inter-cluster aggregated relation is simply regarded as the text concatenation of $R_{<C_j,C_k>}$.

$$r_{<\alpha_j,\alpha_k>} = \begin{cases} \mathcal{F}_{(\text{rel})}(\alpha_j, \alpha_k, R_{<C_j,C_k>}), & \text{if } \lambda_{j,k} > \tau \\ Concate(R_{<C_j,C_k>}), & \text{otherwise} \end{cases} \tag{5}$$

In practice, the generation function $\mathcal{F}_{\text{rel}}$ is implemented by LLMs guided by a specific prompt $\mathcal{P}_{\text{rel}}$.

The threshold τ is a data-dependent hyper-parameter that may vary with the layer index to reflect the knowledge graph's density at different abstraction levels, ensuring only salient, well-supported relations are propagated.

By recursively applying this process of clustering and generation, we construct a rich, multi-layered KG where each layer provides a progressively more abstract, yet semantically coherent, view of the original information.

Structured Retrieval via Lowest Common Ancestor

The hierarchical knowledge graph $\mathcal{H}$ enables a retrieval strategy that is fundamentally more structured and efficient than searching over a flat graph. Our approach moves beyond simple similarity-based retrieval by leveraging the graph's topology to construct a compact and contextually coherent subgraph. This process consists of two main phases: initial entity anchoring at the base layer, followed by a structured traversal of the hierarchy to gather context.

Initial Entity Anchoring. Given a user query q, the first step is to ground the query in the most specific, fine-grained facts available. We achieve this by performing a dense retrieval search exclusively over the entities of the original graph, including the initial entities, that is, the base-layer graph $\mathcal{G}_0$. We identify the top n entities whose textual descriptions are most semantically similar to the query:

$$V_{\text{seed}} = \text{Top-n}_{v \in V_0}(\text{sim}(q, d_v)) \tag{6}$$

This set of “seed entities”, V_{seed}, serves as the starting point for structured traversal, ensuring our retrieval process is anchored in the most relevant parts of the knowledge base.

Contextualization via LCA Path Traversal. Graph retrieval methods in the prior KG-based RAG would typically find all paths between entities in V_{seed} on the flat graph $\mathcal{G}_0$. This approach often retrieves a large number of intermediate nodes that add noise and redundancy. In contrast, LeanRAG utilizes the entire hierarchy $\mathcal{H}$ to define a much more focused and meaningful context. Our core idea is to construct a minimal subgraph that connects the seed entities through their most immediate shared concepts in the hierarchy. We achieve this using the principle of the LCA. For two seed entities in V_{seed}, their lowest common ancestor (LCA) v_{lca} is defined as the common ancestor with the minimum depth in the hierarchy $\mathcal{H}$ among all their ancestors. This ensures that the combined path length from the two seed entities to v_{lca} is minimized to avoid information redundancy.

The retrieval path $\mathcal{P}_{\text{lca}}$ is then defined as the union of all shortest paths in the hierarchy from each seed entity $v \in V_{\text{seed}}$ to the common ancestor v_{lca}:

$$\mathcal{P}_{\text{lca}}(V_{\text{seed}}, \mathcal{H}) = \bigcup_{v \in V_{\text{seed}}} \text{ShortestPath}_{\mathcal{H}}(v, v_{\text{lca}}) \tag{7}$$

where $\text{ShortestPath}_{\mathcal{H}}(\cdot,\cdot)$ denotes the shortest path between two nodes within the hierarchical graph $\mathcal{H}$. Since our hierarchy is tree-like, this path consists of the direct chain of from child nodes to parent nodes. Finally, the retrieved subgraph for RAG context $\mathcal{G}_{\text{ret}}$ is composed of all entities and relations that lie on these LCA paths:

$$\mathcal{G}_{\text{ret}} = (V_{\text{ret}}, R_{\text{ret}}) \tag{8}$$

$$V_{\text{ret}} = \{v \mid v \in \mathcal{P}_{\text{lca}}\} \tag{9}$$

$$R_{\text{ret}} = R_{\text{lca}} \cup R_{\text{inter-cluster}} \tag{10}$$

where R_{lca} contains the relations within the retrieval path $\mathcal{P}_{\text{lca}}$ and $R_{\text{inter-cluster}}$ contains the inter-cluster relations between aggregation entities that are in the same level in the hierarchical knowledge graph. For example, $r_{<\alpha_j,\alpha_k>} \in R_{\text{inter-cluster}}$, where $\alpha_j \in \mathcal{G}_i$ and $\alpha_k \in \mathcal{G}_i$.

This LCA-based traversal strategy ensures that the retrieved context is not just a collection of relevant entities, but a connected, coherent narrative structure, spanning from specific facts to their shared abstract concepts. This significantly reduces information redundancy and provides a much richer, more structured context to the final LLM generator. Furthermore, we return the original chunks from which the entities were sourced as supporting evidence. The illustration of this process is provided in Figure 2.

Experiments

In our experiments, we aim to answer the following research questions:

- RQ1: How does LeanRAG’s **QA performance** compare against state-of-the-art baselines across diverse domains?
- RQ2: Does LeanRAG’s retrieval strategy **reduce information redundancy** while improving generation quality?
- RQ3: To what extent does the explicit generation of **relations between aggregated entities** contribute to the quality of the response?
- RQ4: Is the structured knowledge retrieved from the graph sufficient for high-quality generation, or is the inclusion of the entities **original textual context essential**?

Baselines. To evaluate the performance of LeanRAG, we compare it against a comprehensive suite of representative and state-of-the-art KG-based RAG methods. The selected baselines include:

- **NaiveRAG** (Lewis et al. 2020): The foundational RAG approach, which retrieves semantically similar text chunks from a document corpus.
- **GraphRAG** (Edge et al. 2024): A prominent KG-based method that organizes knowledge into communities. We utilize its local search mode, as the global mode has significant computational overhead and does not leverage local entity context.
- **LightRAG** (Guo et al. 2024): Uses a dual-level retrieval framework based on a KG-based text indexing paradigm.
- **KAG** (Liang et al. 2025): A pipeline that aligns LLM generation with structured KG reasoning through mutual knowledge-text indexing and logic-form guidance.
- **FastGraphRAG**: An enhancement of graph retrieval that uses the PageRank algorithm (Page et al. 1999) to prioritize nodes of higher importance.
- **HiRAG** (Huang et al. 2025a): The current state-of-the-art, which introduces hierarchical structures by clustering entities into multi-level summaries.

Datasets and Evaluation Metrics. We used four datasets from the UltraDomain benchmark (Qian et al. 2024), which is designed to evaluate RAG systems across diverse applications, focusing on long context tasks and high-level queries in specialized domains. We used Mix, CS, Legal, and Agriculture datasets following the prior work (Guo et al. 2024).

Evaluation Metrics. To provide a multi-faceted and in-depth analysis of system performance, we evaluate the generated answers along four crucial dimensions, following the prior work (Huang et al. 2025a):

- **Comprehensiveness:** Measures how thoroughly the answer addresses the user’s query.
- **Empowerment:** Evaluates the answer’s practical utility and its ability to provide actionable information.
- **Diversity:** Assesses the breadth of information and perspectives presented in the answer.
- **Overall:** Provides a single, holistic quality score to measure how the answer performs overall, considering comprehensiveness, empowerment, diversity, and any other relevant factors.

Following recent best practices in automated evaluation, we employ powerful LLMs as judges to score the outputs of all methods on the 1 to 10 scale defined by our metrics. In order to directly reflect the quality of the answers, we will also use LLM to directly evaluate the two answers to obtain their win rates. Specifically, we use DeepSeek-V3 (Liu et al. 2024) as our evaluators, providing them with carefully designed prompts to ensure consistent and unbiased scoring, and each query and answer is scored 5 times.

Implementation Details. Across all experiments, we use DeepSeek-V3 as an LLM generator for all models to ensure a fair comparison. The text embedding for retrieval is computed using BGE-M3 (Chen et al. 2024). The number of clusters for the GMM and other key hyperparameters are tuned on a held-out validation set. All main experiments were conducted by leveraging commercial API services. For our main experiments, we utilized the Deepseek-V3 model as the backbone for all models, following the prior work (Huang et al. 2025a), ensuring a fair comparison. In addition, in order to evaluate RQ2 efficiently, we reproduced the baseline methods on the Qwen3-14b (Yang et al. 2025) model to evaluate the redundancy between LeanRAG and other methods. All implementation details are provided in the extended version (**?**).

Overall Performance Comparison (RQ1)

To address RQ1 , we compare LeanRAG against all baseline models across four benchmarks, as presented in Table 1. The experimental results demonstrate that LeanRAG almost outperforms all baselines across the evaluated datasets.

From a *Comprehensiveness* perspective, even after removing the information-intensive community structure of traditional KG-based RAG, the aggregation used by LeanRAG still provides sufficient query-related information. Furthermore, *Empowerment* and *Diversity* effectively measure the relevance of the provided information. These indicate that LeanRAG effectively enhances the breadth of information by establishing inter-cluster relations, resulting in optimal performance. In summary, LeanRAG demonstrates state-of-the-art performance on the majority of metrics across four evaluated datasets and achieves highly competitive results on the remaining ones.

Analysis of Information Redundancy (RQ2)

Experimental Setup. To answer RQ2, we evaluate the information redundancy of different methods. We use the token count of the retrieved context as a metric for redundancy, where a lower token count at a comparable performance level signifies a less redundant context. We re-implemented all baselines with Qwen3-14B-Instruct.

Retrieved Context Size. Figure 3 shows the number of tokens in the context retrieved by each method. The results indicate that LeanRAG retrieves a substantially more compact context compared to all baselines. On average, its retrieved context is 46% smaller than baselines. This result can be attributed to our LCA-based traversal strategy, which constructs a focused subgraph by navigating the hierarchy, in contrast to methods that retrieve larger communities.

Cluster Relation Effectiveness Analysis (RQ3)

The core innovation of LeanRAG is not only its use of fine-grained, controllable aggregate entities but also its establishment of paths between them, which creates a fully navigable semantic network for retrieval. This design directly addresses RQ3: whether the inter-cluster relationships, which break the traditional “semantic islands” problem, can truly improve retrieval quality. To test this, we conducted experiments on four datasets, comparing the retrieval results of LeanRAG with and without the inclusion of path information. The win rates across four different metrics were then analyzed, with the results summarized in Table 2.

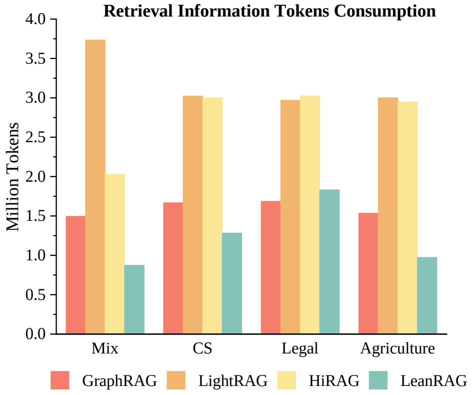

Figure 3: Comparison in retrieval tokens across four datasets

The data in Table 3 clearly shows that when relational paths are removed, LeanRAG’s retrieval diversity, or the breadth of its information, decreases significantly. This result confirms that establishing relationships between clusters effectively connects isolated entities, thereby enriching the information available for retrieval. Furthermore, by explicitly returning these relationships, the retrieval process is enhanced, leading to a demonstrable improvement in the overall quality of the retrieved answers.

Necessity Analysis of Textual Context (RQ4)

Motivation and Setup. To answer RQ4, we investigate the role of the original unstructured text chunks in our framework. While the graph structure serves as an effective retrieval guide, it is crucial to assess whether structured information alone suffices for the generator or if the source text remains essential. To this end, we conduct an ablation study using a variant of our model, denoted as **LeanRAG w/o Context**. This variant follows the same hierarchical retrieval process, but the final context provided to the LLM generator includes only the names and descriptions of the retrieved graph entities, excluding the original text chunks linked to the base-level entities. We then compare its performance with that of the full LeanRAG model.

Results and Analysis. The results of this comparison are presented in Table 3. Across all four datasets and nearly every evaluation metric, the performance of LeanRAG drops significantly when the original textual context is removed. On average, the overall quality score decreases from 8.59 to 7.93 on the Mix dataset, and similar degradations are observed on the CS, Legal, and Agriculture datasets.

Dataset	Metric ↑	LeanRAG	HiRAG	Naive	GraphRAG	LightRAG	FastGraphRAG	KAG
Mix	Comprehensiveness	**8.89±0.01**	8.72±0.02	8.20±0.01	8.52±0.01	8.19±0.02	6.56±0.02	7.90±0.03
	Empowerment	**8.16±0.02**	7.86±0.03	7.52±0.03	7.73±0.02	7.56±0.03	5.82±0.03	7.41±0.04
	Diversity	**7.73±0.01**	7.21±0.02	6.65±0.03	7.04±0.02	6.69±0.04	4.88±0.03	6.42±0.04
	Overall	**8.59±0.01**	8.08±0.02	7.47±0.02	7.87±0.01	7.61±0.04	5.76±0.02	7.25±0.03
CS	Comprehensiveness	8.92±0.01	8.92±0.01	**8.94±0.01**	8.55±0.02	8.76±0.02	6.79±0.01	8.22±0.02
	Empowerment	8.68±0.02	8.66±0.02	**8.69±0.04**	8.28±0.04	8.50±0.04	6.67±0.04	8.52±0.05
	Diversity	**7.87±0.02**	7.84±0.02	7.79±0.02	7.42±0.02	7.63±0.04	5.45±0.04	7.03±0.02
	Overall	**8.82±0.02**	8.77±0.02	8.77±0.03	8.37±0.04	8.59±0.04	6.31±0.03	7.99±0.03
Legal	Comprehensiveness	8.88±0.02	8.68±0.02	8.85±0.01	**8.95±0.01**	8.24±0.02	3.87±0.02	8.41±0.02
	Empowerment	**8.42±0.03**	8.18±0.06	8.28±0.03	8.33±0.02	7.83±0.05	3.53±0.03	8.20±0.03
	Diversity	**7.49±0.03**	7.00±0.03	7.10±0.04	7.47±0.03	6.87±0.01	2.87±0.02	6.71±0.01
	Overall	**8.49±0.04**	8.00±0.04	8.21±0.03	8.44±0.01	7.74±0.03	3.43±0.02	7.83±0.03
Agriculture	Comprehensiveness	8.94±0.06	**8.99±0.00**	8.85±0.01	8.97±0.01	8.71±0.01	3.28±0.01	8.22±0.01
	Empowerment	**8.66±0.02**	8.52±0.02	8.51±0.03	8.52±0.02	8.23±0.02	3.29±0.05	8.33±0.06
	Diversity	**8.06±0.03**	7.98±0.02	7.76±0.06	7.95±0.02	7.68±0.03	3.01±0.03	7.07±0.02
	Overall	**8.87±0.02**	8.87±0.03	8.69±0.03	8.85±0.01	8.56±0.02	3.17±0.02	7.95±0.03

Table 1: Evaluation scores (1–10 scale) of LeanRAG compared to baseline methods, assessed by an LLM

	Mix		CS		Legal		Agriculture	
Comprehensiveness	**51.5%**	48.6%	**54.5%**	45.5%	**55.5%**	44.5%	**54.0%**	46.0%
Empowerment	**55.0%**	45.0%	**55.5%**	44.5%	**56.5%**	43.5%	**59.5%**	40.5%
Diversity	**59.6%**	40.4%	**66.0%**	34.0%	**57.0%**	43.0%	**63.0%**	37.0%
Overall	**53.8%**	46.2%	**58.5%**	41.5%	**56.5%**	43.5%	**58.0%**	42.0%

Table 2: Win rates (%) between LeanRAG and LeanRAG w/o Relation (Left: LeanRAG; Right: w/o Relation)

The most pronounced drops are consistently seen in the *Comprehensiveness* and *Empowerment* metrics. This is expected, as raw text chunks contain the detailed explanations, evidence, and nuanced language necessary for generating thorough and actionable answers. In contrast, a context composed solely of structured entity information, while semantically focused, lacks the narrative richness required by the LLM. These findings confirm our hypothesis: the hierarchical graph in LeanRAG acts as an effective semantic index and navigation system whose primary function is to precisely locate critical segments of unstructured text. The collaboration between structured graph traversal for guidance and the rich content of unstructured text for generation is essential to achieving state-of-the-art performance.

Dataset	Metric ↑	LeanRAG	LeanRAG w/o Context
Mix	Comprehensiveness	**8.89±0.01**	8.15±0.02 ↓
	Empowerment	**8.16±0.02**	7.80±0.01 ↓
	Diversity	**7.73±0.01**	7.26±0.02 ↓
	Overall	**8.59±0.01**	7.93±0.01 ↓
CS	Comprehensiveness	**8.92±0.01**	8.66±0.02 ↓
	Empowerment	**8.68±0.02**	8.19±0.03 ↓
	Diversity	**7.87±0.02**	7.57±0.02 ↓
	Overall	**8.82±0.02**	8.34±0.02 ↓
Legal	Comprehensiveness	**8.88±0.02**	8.49±0.01 ↓
	Empowerment	**8.42±0.03**	8.11±0.04 ↓
	Diversity	**7.49±0.03**	7.09±0.04 ↓
	Overall	**8.49±0.04**	8.00±0.04 ↓
Agriculture	Comprehensiveness	**8.94±0.06**	8.65±0.01 ↓
	Empowerment	**8.66±0.02**	8.16±0.05 ↓
	Diversity	**8.06±0.03**	7.88±0.05 ↓
	Overall	**8.87±0.02**	8.53±0.03 ↓

Table 3: Necessity analysis of textual context

Conclusions

To address the critical challenges of "semantic islands" and the structure-retrieval mismatch prevalent in the KG-based RAG systems, we introduced **LeanRAG**, a novel framework that resolves these issues through a tight co-design of its knowledge aggregation and retrieval mechanisms. Our approach features a hierarchical aggregation algorithm that constructs a fully navigable semantic network by generating explicit relations between abstract summary concepts, and a complementary bottom-up, LCA-based retrieval strategy that efficiently traverses this structure. Extensive experiments validated our design, demonstrating that LeanRAG achieves state-of-the-art performance while significantly reducing information redundancy. Furthermore, our ablation studies confirmed that both the generation of summary information and the original textual context are essential for producing comprehensive and diverse answers.

Acknowledgments

The research was supported by Shanghai Artificial Intelligence Laboratory, the National Key R&D Program of China

(Grant No. 2022ZD0160201) and the Science and Technology Commission of Shanghai Municipality (Grant Nos. 22DZ1100102).

References

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; and Liu, Z. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Guo, Q.; Wang, M.; and Wang, H. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *ArXiv*, abs/2312.10997.

Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779*.

Huang, H.; Huang, Y.; Yang, J.; Pan, Z.; Chen, Y.; Ma, K.; Chen, H.; and Cheng, J. 2025a. HiRAG: Retrieval-Augmented Generation with Hierarchical Knowledge. arXiv:2503.10150.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2025b. A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55.

Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Jiang, Z.; Xu, F. F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Active retrieval augmented generation. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7969–7992.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P. S.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Li, J.; Chen, J.; Ren, R.; Cheng, X.; Zhao, W. X.; yun Nie, J.; and Wen, J.-R. 2024. The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models. In *Annual Meeting of the Association for Computational Linguistics*, 10879–10899.

Liang, L.; Bo, Z.; Gui, Z.; Zhu, Z.; Zhong, L.; Zhao, P.; Sun, M.; Zhang, Z.; Zhou, J.; Chen, W.; et al. 2025. Kag: Boosting llms in professional domains via knowledge augmented generation. In *Companion Proceedings of the ACM on Web Conference 2025*, 334–343.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford infolab.

Peng, B.; Zhu, Y.; Liu, Y.; Bo, X.; Shi, H.; Hong, C.; Zhang, Y.; and Tang, S. 2024. Graph Retrieval-Augmented Generation: A Survey. *ArXiv*, abs/2408.08921.

Qian, H.; Zhang, P.; Liu, Z.; Mao, K.; and Dou, Z. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*, 1.

Reynolds, D. 2015. Gaussian mixture models. In *Encyclopedia of biometrics*, 827–832. Springer.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; and Manning, C. D. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Tonellotto, N.; Trappolini, G.; Silvestri, F.; Campagnano, C.; Siciliano, F.; Cuconasu, F.; Maarek, Y.; and Filice, S. 2024. The Power of Noise: Redefining Retrieval for RAG Systems. *ACM International Conference on Research and Development in Information Retrieval (SIGIR)*.

Wang, X.; Wang, Z.; Gao, X.; Zhang, F.; Wu, Y.; Xu, Z.; Shi, T.; Wang, Z.; Li, S.; Qian, Q.; et al. 2024. Searching for best practices in retrieval-augmented generation. *arXiv preprint arXiv:2407.01219*.

Wang, Z. R.; Wang, Z.; Le, L.; Zheng, H. S.; Mishra, S.; Perot, V.; Zhang, Y.; Mattapalli, A.; Taly, A.; Shang, J.; Lee, C.-Y.; and Pfister, T. 2025. Speculative RAG: Enhancing Retrieval Augmented Generation through Drafting. In Yue, Y.; Garg, A.; Peng, N.; Sha, F.; and Yu, R., eds., *International Conference on Representation Learning*, volume 2025, 18483–18505.

Xu, Z.; Cruz, M. J.; Guevara, M.; Wang, T.; Deshpande, M.; Wang, X.; and Li, Z. 2024. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. *ACM International Conference on Research and Development in Information Retrieval (SIGIR)*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Zhao, P.; Zhang, H.; Yu, Q.; Wang, Z.; Geng, Y.; Fu, F.; Yang, L.; Zhang, W.; and Cui, B. 2024. Retrieval-Augmented Generation for AI-Generated Content: A Survey. *ArXiv*, abs/2402.19473.

Appendix

The appendix provides supplementary materials and detailed information to support the main findings of this paper. It includes a comprehensive breakdown of our methodology, covering the following key sections: More Results of Model Performance Across Four Datasets, Experimental Implementation Details, QA Cases of LeanRAG, Prompt Templates used in LeanRAG.

This section is designed to ensure the reproducibility of our work by offering an in-depth look at the implementation specifics, including a detailed description of the datasets, the graph construction process, and the retrieval strategies employed. The provided prompt examples and templates further illustrate the mechanisms used to guide the Large Language Model (LLM) in generating high-quality responses.

A. More Results of Model Performance Across Four Datasets

The results consistently demonstrate that LeanRAG outperforms all baseline methods across all four datasets and evaluation metrics, achieving notably higher win rates. This suggests that the proposed approach offers a more efficient and reliable framework for retrieval and generation compared to existing methods.

LeanRAG significantly outperforms NaiveRAG, FastGraphRAG, and KAG: Across these baselines, LeanRAG exhibits overwhelming superiority, with win rates often exceeding 95%, and reaching 100% in some cases. This advantage is particularly pronounced in the Empowerment and Diversity metrics, underscoring LeanRAG's ability to leverage structured knowledge graphs to provide more relevant and diverse information. These findings validate the fundamental advantage of graph-based methods over simple text retrieval approaches.

LeanRAG demonstrates strong performance against more advanced baselines such as GraphRAG, LightRAG, and HiRAG: Although the win rates are comparatively lower than against simpler baselines, LeanRAG still maintains a substantial performance margin. When compared to other graph-based methods like GraphRAG and HiRAG, LeanRAG achieves win rates consistently between 50% and 80%. This highlights the competitive advantage of LeanRAG's strategy in aggregating entities and constructing multi-level semantic networks, surpassing conventional graph-based or hierarchy-based RAG techniques. On the Comprehensiveness metric within the Legal domain, LeanRAG's win rate against GraphRAG is the lowest (51.0% vs. 49.0%), indicating that the dense and domain-specific nature of legal texts poses similar challenges for both models. In comparison with LightRAG, LeanRAG consistently outperforms it across the Mix and Legal datasets, with win rates frequently exceeding 80%, particularly in the Empowerment and Diversity categories. This indicates that LeanRAG's enhanced graph construction and retrieval mechanisms are more effective than those employed by LightRAG.

Consistency across datasets and metrics: LeanRAG's superior performance is not limited to any single dataset or metric. It consistently outperforms baselines across diverse domains, including Mix, Computer Science, Legal, and Agriculture, demonstrating both robustness and generalizability. Notably, its highest win rates are often observed in the Empowerment and Diversity metrics, which are critical for generating high-quality, non-redundant, and actionable responses. This underscores the effectiveness of LeanRAG's core design in producing meaningful outputs.

B. Experimental Implementation Details

B.1 Dataset Details

This subsection provides a comprehensive description of the dataset(s) utilized in this study. It includes details regarding the source of the data and its overall size (e.g., number of documents, total tokens).

As presented in Table 5, the datasets vary significantly in size and content. The Legal dataset is the largest, containing 94 documents and a substantial 5,279,400 tokens, reflecting the detailed and extensive nature of legal texts. In contrast, the CS (Computer Science) dataset, while having fewer documents (10), still comprises a significant 2,210,894 tokens, indicating potentially longer and more technical documents within that domain. The Agriculture dataset contributes 12 documents and 2,028,496 tokens, while the Mix dataset, serving as a general collection, includes 61 documents and 625,948 tokens. These diverse characteristics allow for a thorough assessment of our model's performance across varied information landscapes.

B.2 Graph Construction Implementation Details

To effectively manage the scale of LeanRAG, we introduced a hyperparameter, $clustersize$, which allows us to control the number of clusters generated during the Gaussian Mixture Model (GMM) clustering process by manually limiting the number of nodes within each cluster. This design choice provides a significant degree of controllability, enabling us to adjust the size of LeanRAG according to specific application requirements.

In our experiments, we performed a unified entity and relationship extraction for all documents within each dataset to build a single knowledge graph. This approach ensures a consistent graph structure for each dataset, rather than generating a separate graph for each question-answer pair.

Despite the four datasets varying considerably in both size and domain, we consistently used a $clustersize$ of 20 for graph construction. It's important to note that $clustersize$ is a pivotal factor that not only dictates the overall size of the LeanRAG graph but also profoundly impacts its retrieval efficiency and quality. Beyond this, the threshold τ, which governs the generation of inter-cluster relationships, also profoundly impacts LeanRAG's performance. For our experiments, we set this threshold to 3.To further assess the efficacy of our method and cater to diverse use cases, future work will involve a comprehensive exploration of how different $clustersize$ values and τ values influence LeanRAG's performance.

Table 4: Win rates (%) of LeanRAG, its two variants (for ablation study), and baseline methods on QFS tasks.

	Mix		**CS**		**Legal**		**Agriculture**	
	NaiveRAG	**LeanRAG**	NaiveRAG	**LeanRAG**	NaiveRAG	**LeanRAG**	NaiveRAG	**LeanRAG**
Comprehensiveness	11.9%	88.1%	41.0%	59.0%	30.0%	70.0%	34.0%	66.0%
Empowerment	1.5%	98.5%	40.5%	59.5%	24.5%	75.5%	15.5%	84.5%
Diversity	3.1%	96.9%	28.0%	72.0%	9.0%	91.0%	10.0%	90.0%
Overall	2.7%	97.3%	39.5%	60.5%	23.5%	76.5%	16.0%	84.0%
	GraphRAG	**LeanRAG**	GraphRAG	**LeanRAG**	GraphRAG	**LeanRAG**	GraphRAG	**LeanRAG**
Comprehensiveness	35.0%	65.0%	41.0%	59.0%	49.0%	51.0%	45.5%	54.5%
Empowerment	20.0%	80.0%	33.5%	66.5%	44.0%	56.0%	27.0%	73.0%
Diversity	16.5%	83.5%	34.0%	66.0%	44.0%	56.0%	22.0%	78.0%
Overall	21.9%	78.1%	37.5%	62.5%	47.0%	53.0%	28.5%	71.5%
	LightRAG	**LeanRAG**	LightRAG	**LeanRAG**	LightRAG	**LeanRAG**	LightRAG	**LeanRAG**
Comprehensiveness	28.8%	71.2%	44.5%	55.5%	25.0%	75.0%	38.0%	62.0%
Empowerment	16.5%	83.5%	35.5%	64.5%	12.0%	88.0%	17.0%	83.0%
Diversity	13.1%	86.9%	34.0%	66.0%	40.5%	59.5%	16.5%	83.5%
Overall	18.8%	81.2%	38.5%	61.5%	21.0%	79.0%	18.5%	81.5%
	FastGraphRAG	**LeanRAG**	FastGraphRAG	**LeanRAG**	FastGraphRAG	**LeanRAG**	FastGraphRAG	**LeanRAG**
Comprehensiveness	0%	100%	0.5%	99.5%	1.0%	99.0%	0.5%	99.5%
Empowerment	0%	100%	0.0%	100.0%	0.5%	99.5%	0.0%	100.0%
Diversity	0%	100%	0.8%	99.2%	2.5%	97.5%	0.0%	100.0%
Overall	0%	100%	0.0%	100.0%	4.5%	95.5%	0.0%	100.0%
	KAG	**LeanRAG**	KAG	**LeanRAG**	KAG	**LeanRAG**	KAG	**LeanRAG**
Comprehensiveness	1.5%	98.5%	5.0%	95.0%	5.0%	95.0%	2.5%	97.5%
Empowerment	1.9%	98.1%	3.0%	97.0%	4.5%	95.5%	2.5%	97.5%
Diversity	1.2%	98.8%	4.0%	96.0%	2.5%	97.5%	1.0%	99.0%
Overall	1.2%	98.8%	3.5%	96.5%	4.5%	95.5%	1.0%	99.0%
	HiRAG	**LeanRAG**	HiRAG	**LeanRAG**	HiRAG	**LeanRAG**	HiRAG	**LeanRAG**
Comprehensiveness	43.8%	56.2%	46.5%	53.5%	29.5%	70.5%	49.5%	50.5%
Empowerment	26.5%	73.5%	43.5%	56.5%	16.5%	83.5%	26.5%	73.5%
Diversity	20.4%	79.6%	44.5%	55.5%	23.5%	76.5%	23.5%	76.5%
Overall	28.1%	71.9%	45.0%	55.0%	21.5%	78.5%	28.0%	72%

Table 5: Statistics of task datasets.

Dataset	**Mix**	**CS**	**Legal**	**Agriculture**
# of Documents	61	10	94	12
# of Tokens	625,948	2,210,894	5,279,400	2,028,496

B.3 Graph Retrieval Details

B.3.1 Chunk Selection Strategy Based on our observations of traditional GraphRAG methods, we found that even after extracting structured entities, relationships, and community information, the original text chunks remain crucial for answering questions. This is because these chunks often contain incoherent semantic information that cannot be structurally extracted, yet still plays a vital role. Consequently, in LeanRAG, we also return the top-C retrieved chunks during the process.

Our specific approach is as follows: After identifying the initial seed nodes V_{seed}, we trace back to their original text chunks. We then rank these chunks in descending order based on the number of entities from V_{seed} that appear within each chunk. Finally, we return the top-C chunks from this ranked list. This method allows us to pinpoint the top-C chunks most relevant to the query by aligning with the user's intent through entity-based searching, which we find to be more effective than the similarity-based chunk retrieval employed by Naive RAG.

B.4 Experiment Settings

To ensure our method achieves optimal performance across all four datasets, we fine-tuned the hyperparameter $clustersize$, $Top-N$, and $Top-C$. The specific parameter settings used for these adjustments are detailed below:

Our observations during the retrieval phase revealed dis-

Table 6: Setting of task datasets.

Dataset	Mix	CS	Legal	Agriculture
$clustersize$	20	20	20	20
N	10	10	15	10
C	5	10	10	5

tinct characteristics across the datasets, influencing the optimal parameter settings for effective information retrieval.

Specifically, for the Mix and Agriculture datasets, a relatively smaller number of seed nodes $V_{\text{seed}}$was sufficient for robust query resolution. This can be attributed to the limited scope of content within a subset of documents and the overall stronger internal connectedness within their respective knowledge bases.

Conversely, the Computer Science (CS) dataset presented unique challenges. Its weaker intrinsic associativity and the less structured nature of its specialized terminology necessitated the retrieval of a larger number of supporting chunks. This suggests that relevant information for a given query in the CS domain might be more distributed and less directly interlinked within the graph structure.

Finally, the Legal dataset, characterized by its highly specialized and extensive terminology and greater document-level separability, required the retrieval of a larger volume of information. This indicated a need for a higher count of $V_{\text{seed}}$to achieve a comprehensive understanding of the query, as pertinent details tended to be more dispersed across a broader range of documents.

C. QA Cases of LeanRAG

To illustrate the effectiveness of our approach, this section presents a few straightforward examples comparing the performance of LeanRAG with the HiRAG method. These cases are designed to highlight how LeanRAG's optimized graph structure and retrieval strategy lead to more precise and coherent answers. By directly contrasting their outputs, we aim to demonstrate the practical benefits of our method in various query scenarios,the case can be found in Table 7.

D. Prompt Templates used in LeanRAG

This section details the specific prompt templates employed within the LeanRAG framework. While our knowledge graph (KG) generation code aligns with that of LightRAG and will not be reiterated here, this chapter focuses on the four distinct prompt templates critical to LeanRAG's operation: the Entity Aggregation Prompt, the Inter-Cluster Relation Generation Prompt, the Score Scoring Prompt, and the Win Rate Evaluation Prompt. Each prompt plays a vital role in guiding the Large Language Model (LLM) through various stages of information processing, from consolidating entities to evaluating retrieval outcomes.

D.1 Prompt Templates for Entity Aggregation

As depicted in Figure 8, we leverage the clusters generated by the Gaussian Mixture Model (GMM) to derive descriptions of all entities within a cluster, along with the relationships between these intra-cluster entities. This information is then used to generate an aggregated entity. To circumvent the limitations of traditional community concepts, which can forcibly aggregate all entities and inadvertently assign irrelevant attributes, we explicitly constrain the Large Language Model (LLM) to generate information solely based on the current set of entity descriptions. Furthermore, we emphasize the connecting role of the generated aggregated entity for its constituent sub-entities, ensuring its relevance and coherence within the broader knowledge graph.

D.2 Prompt Templates for Relation Aggregation

As illustrated in Figure 9, we employ a specialized relation prompt to generate relationships between the aggregated entities. This prompt leverages the names and descriptions of two aggregated entities, alongside the existing relationships between their constituent sub-entities, to infer and generate all relevant connections between the two aggregated entities. Given that the descriptions of the aggregated entities already encapsulate the broad information of their sub-entities, we did not incorporate additional sub-entity descriptions to enrich the input. Through this generation of relationships between sets of aggregated entities, LeanRAG effectively mitigates the problem of "semantic islands", thereby constructing a multi-level navigable semantic network.

D.3 Prompt Template for Absolute Quality Scoring

To obtain a quantitative measure of performance for each model, we designed a prompt for absolute quality scoring. This prompt instructs an evaluating LLM to assess a single generated answer based on our predefined metrics (*Comprehensiveness*, *Empowerment*, etc.) and assign a numerical score from 1 to 10 for each. To ensure transparency and facilitate analysis, the LLM is also required to provide a concise rationale for each score. All assessments are structured in a JSON format to ensure consistency and ease of parsing. The detailed template used for this scoring task is presented in Table 10.

D.4 Prompt Template for Pairwise Comparison

In addition to absolute scoring, we conducted pairwise comparisons to determine the relative performance between different models, resulting in win-rate statistics. For this purpose, we developed a separate prompt that presents the answers from two different models (e.g., LeanRAG vs. HiRAG) to an evaluating LLM. The prompt then instructs the evaluator to act as an impartial judge and determine which of the two answers is superior, considering the overall quality. The LLM must declare a "winner" and provide a detailed justification for its decision, again in a structured JSON format. The template used for these head-to-head comparisons is shown in Table 11.

Table 7: Example question, answers, and LLM-generated assessments for the Mix dataset.

Question	**What is the significance of studying puns in the context of human-computer interaction and machine translation?**
LeanRAG	The study of puns in the context of human-computer interaction (HCI) and machine translation (MT) holds significant importance for several reasons, as evidenced by the provided data tables and related research. **1. Enhancing Natural Language Understanding** [...] **2. Improving Machine Translation Systems** [...] **3. Benchmarking and Model Evaluation** [...] **4. Cross-Disciplinary Applications** [...] **5. Challenges and Future Directions** [...] In summary, studying puns contributes to deeper language understanding, better translation systems, and more sophisticated human-computer interactions by addressing ambiguity, humor, and cultural context—key hurdles in NLP.
HiRAG	Significance of Studying Puns in Human-Computer Interaction and Machine Translation **Human-Computer Interaction** [...] **Machine Translation** [...] **Real-World Applications** [...] In summary, the study of puns in the context of human-computer interaction and machine translation is significant for developing more sophisticated and effective NLP systems. By enabling machines to understand and handle puns, we can create more natural and engaging interactions and more accurate translations, thereby enhancing the overall utility and effectiveness of these technologies.
LLM Decision	**Comprehensiveness: Winner=1 (LeanRAG)**. Answer 1 provides a more detailed and structured analysis of the significance of studying puns in HCI and machine translation. It breaks down the topic into specific areas such as natural language understanding, machine translation systems, benchmarking, cross-disciplinary applications, and future challenges. This level of detail makes it more comprehensive compared to Answer 2, which, while informative, lacks the depth and structured approach of Answer 1. **Empowerment: Winner=1 (LeanRAG)**. Answer 1 empowers the reader by not only explaining the significance of studying puns but also by discussing specific tools (e.g., CMU Pronouncing Dictionary), models (e.g., Bidirectional RNN, Hidden Markov Model), and datasets (e.g., HOMOGRAPHIC DATASET, HETEROGRAPHIC DATASET) used in the research. This provides the reader with concrete information to further explore the topic or apply the knowledge in practical scenarios. Answer 2, while informative, does not provide as much actionable or specific information. **Diversity: Winner=1 (LeanRAG)**. Answer 1 offers a richer diversity of perspectives and insights by covering multiple aspects such as natural language understanding, machine translation, benchmarking, cross-disciplinary applications, and future challenges. It also mentions specific models and datasets, adding layers of depth and variety to the discussion. Answer 2, while covering the basics well, does not provide the same level of varied insights or detailed examples. **Overall Winner: Winner=1 (LeanRAG)**. Answer 1 is the overall winner because it excels in comprehensiveness, empowerment, and diversity. It provides a detailed, structured, and multi-faceted analysis of the topic, equips the reader with specific tools and models for further exploration, and offers a wide range of perspectives and insights. Answer 2 is informative but lacks the depth, specificity, and variety that make Answer 1 superior.

Table 8: The prompt template of aggregate entities from entity cluster.

Entity aggregation prompt

Role: Entity Aggregation Analyst

Profile
- author: LangGPT
- version: 1.1
- language: English
- description: You are an expert in **concept synthesis** and entity aggregation. Your task is to identify a meaningful aggregate entity from a set of related entities and extract structured, comprehensive insights based solely on provided evidence.

Skills
- Abstraction and naming of collective concepts based on entity roles, and relationships
- Structured summarization and typology recognition
- Comparative and relational analysis across multiple entities
- Strict grounding to provided data (no hallucinated content)
- Extraction of both explicit and implicit shared characteristics

Goals
- Derive a meaningful aggregate entity that broadly represents the given entity set, capturing both explicit and nuanced connections
- The aggregate entity name must not match any single entity in the set
- Provide an accurate, comprehensive, and concise description of the aggregate entity reflecting shared characteristics, structure, functions, and significance
- Extract as many structured findings as possible (at least 5, but preferably more) about the entity set based on grounded evidence, including roles, relationships, patterns, and unique features

Output Format
- All output MUST be in a well-formed JSON-formatted string, strictly following the structure below.
- Do NOT include any explanation, markdown, or extra text outside the JSON.

Format:
Input: {`input_text`}

Output:

```
{
  "entity_name": "<name>",
  "entity_description": "<description summarizing the shared traits, structure,
                         functions, and significance of the aggregation>",
  "findings": [
    {
      "summary": "<summary>",
      "explanation": "<explanation>"
    }
  ]
}
```

Rules
- Grounding Rule: All content must be based solely on the provided entity set — no external assumptions
- Naming Rule: The aggregate entity name must not be identical to any single entity; it should reflect a composite structure, function, or theme
- Each finding must include a concise summary and a detailed explanation
- Include findings about entity roles, interconnections, patterns, and any notable diversity or specialization within the set
- Avoid adding speculative or unsupported interpretations

Workflows
1. Review the list of entities, focusing on types, descriptions, and relational structure
2. Synthesize a generalized name that best represents the full entity set, emphasizing collective identity and function
3. Write a clear, evidence-based, and information-rich description of the aggregate entity
4. Extract and elaborate on key findings, emphasizing structure, purpose, interconnections, diversity, and any emergent properties, and explicitly relate these to the contributions of the sub-entities

Table 9: The prompt template of generate relation between aggregation entities.

Relation aggregation prompt

Role: Inter-Aggregation Relationship Analyst

Profile - author: LangGPT
- version: 1.2
- language: English
- description: You specialize in analyzing relationships between two aggregation entities. Your goal is to synthesize a high-level, abstract summary sentence that comprehensively covers all types of relationships between the sub-entities of two named aggregations, based solely on their descriptions and sub-entity relationships.

Skills - Aggregated reasoning across entity groups
- Abstraction and synthesis of all cross-entity relationship types
- Formal summarization under strict constraints
- Strong grounding without repetition or speculation

Goals - Produce a summary (≤ tokens} words) that comprehensively and collectively covers all types of relationships between the sub-entities of Aggregation A and Aggregation B
- Ensure the summary reflects the full diversity and scope of the sub-entity relationships, not just a single aspect
- Avoid reproducing individual sub-entity relationships
- Emphasize structural, functional, or thematic connections at the group level

Input Format
Aggregation A Name: {entity_a}
Aggregation A Description: {entity_a_description}
Aggregation B Name: {entity_b}
Aggregation B Description: {entity_b_description}
Sub-Entity Relationships: {relation_information}

Output Format
<Single-sentence explanation (≤ tokens words) summarizing the relationship between Aggregation A and Aggregation B. Use abstract group-level language. The sentence must comprehensively reflect all types of relationships present between the sub-entities.>

Rules
- DO NOT name specific sub-entities (e.g., individuals)
- DO NOT use the term “community”; always refer to “aggregation,” “group,” “collection,” or thematic equivalents
- DO use collective terms (e.g., “external reviewers,” “trade policy actors”)
- The sentence must be ≤ {tokens} words, factual, grounded, and in formal English
- The relationship must reflect an **aggregation-level abstraction**, such as:
- support/collaboration
- review/feedback
- functional alignment
- domain linkage (e.g., one produces work, the other evaluates it)
- any other relevant relationship types present in the sub-entity relationships
- The summary must comprehensively cover the diversity and scope of all sub-entity relationships, not just a single type

Example
Input:
Aggregation A Name: WTO External Contributors
Aggregation A Description: A group of economists and trade policy experts who provided feedback on early drafts of WTO reports.
Aggregation B Name: WTO Flagship Reports
Aggregation B Description: Core analytical publications from the WTO addressing international trade issues.

Sub-Entity Relationships:
- External contributors provided expert review and feedback on preliminary drafts of flagship reports.
- Feedback from the group was incorporated to enhance report quality and analytical depth.

Output:
The WTO External Contributors aggregation enhanced the analytical rigor and credibility of the WTO Flagship Reports aggregation by providing expert review, feedback, and collaborative input across multiple report drafts.

Table 10: The prompt template of scoring model response.

QA scoring prompt

Your task is to evaluate the following answer based on four criteria. For each criterion, assign a score from 1 to 10 , following the detailed scoring rubric.

When explaining your score, you must refer directly to specific parts of the answer to justify your reasoning. Avoid general statements — your explanation must be grounded in the content provided.

- **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?

- **Overall Quality**:
Provide an overall evaluation based on the combined performance across all four dimensions. Consider both content quality and answer usefulness to the question.

Scoring Guidelines:

"1-2": "Low score description: Clearly deficient in this aspect, with significant issues.",
"3-4": "Below average score description: Lacking in several important areas, with noticeable problems.",
"5-6": "Average score description: Adequate but not exemplary, meets basic expectations with some minor issues.",
"7-8": "Above average score description: Generally strong but with minor shortcomings.",
"9-10": "High score description: Outstanding in this aspect, with no noticeable issues."

Here is the question:
`{query}`

Here are the answer:

`{answer}`

Evaluate the answer using the criteria listed above and provide detailed explanations for each criterion with reference to the text. Output your evaluation in the following JSON format:

```
{{
    ``Comprehensiveness'': {{
        ``score'': ``[1-10]'',
        ``Explanation'': ``[Provide explanation here]''
    }},
    ``Empowerment'': {{
        ``score'': ``[1-10]'',
        ``Explanation'': ``[Provide explanation here]''
    }},
    ``Diversity'': {{
        ``score'': ``[1-10]'',
        ``Explanation'': ``[Provide explanation here]''
    }},
    ``Overall Quality'': {{
        ``score'': ``[1-10]'',
        ``Explanation'': ``[Summarize why this answer is the overall winner
                        based on the three criteria]''
    }}
}}
```

Table 11: The prompt template of rating model response.

QA rating prompt

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. And you need to be very fair and have no bias towards the order.

Output your evaluation in the following JSON format:

```
{{
    ``Comprehensiveness'': {{
        ``Winner'': ``[Answer 1 or Answer 2]'',
        ``Explanation'': ``[Provide explanation here]''
    }},
    ``Empowerment'': {{
        ``Winner'': ``[Answer 1 or Answer 2]'',
        ``Explanation'': ``[Provide explanation here]''
    }},
    ``Diversity'': {{
        ``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]''
    }}
}}
```