LightRAG: Simple and Fast Retrieval-Augmented Generation

Zirui Guo[1,2], **Lianghao Xia**[2], **Yanhua Yu**[1,*], **Tu Ao**[1], **Chao Huang**[2*]
Beijing University of Posts and Telecommunications[1]
University of Hong Kong[2]
zrguo101@hku.hk aka_xia@foxmail.com chaohuang75@gmail.com

Abstract

Retrieval-Augmented Generation (RAG) systems enhance large language models (LLMs) by integrating external knowledge sources, enabling more accurate and contextually relevant responses tailored to user needs. However, existing RAG systems have significant limitations, including reliance on flat data representations and inadequate contextual awareness, which can lead to fragmented answers that fail to capture complex inter-dependencies. To address these challenges, we propose LightRAG, which incorporates graph structures into text indexing and retrieval processes. This innovative framework employs a dual-level retrieval system that enhances comprehensive information retrieval from both low-level and high-level knowledge discovery. Additionally, the integration of graph structures with vector representations facilitates efficient retrieval of related entities and their relationships, significantly improving response times while maintaining contextual relevance. This capability is further enhanced by an incremental update algorithm that ensures the timely integration of new data, allowing the system to remain effective and responsive in rapidly changing data environments. Extensive experimental validation demonstrates considerable improvements in retrieval accuracy and efficiency compared to existing approaches. We have made our LightRAG open-source and available at the link: https://github.com/HKUDS/LightRAG.

1 Introduction

Retrieval-Augmented Generation (RAG) systems have been developed to enhance large language models (LLMs) by integrating external knowledge sources Sudhi et al. (2024); Es et al. (2024); Salemi & Zamani (2024). This innovative integration allows LLMs to generate more accurate and contextually relevant responses, significantly improving their utility in real-world applications. By adapting to specific domain knowledge Tu et al. (2024), RAG systems ensure that the information provided is not only pertinent but also tailored to the user's needs. Furthermore, they offer access to up-to-date information Zhao et al. (2024), which is crucial in rapidly evolving fields. Chunking plays a vital role in facilitating the retrieval-augmented generation process Lyu et al. (2024). By breaking down a large external text corpus into smaller, more manageable segments, chunking significantly enhances the accuracy of information retrieval. This approach allows for more targeted similarity searches, ensuring that the retrieved content is directly relevant to user queries.

However, existing RAG systems have key limitations that hinder their performance. **First**, many methods rely on flat data representations, restricting their ability to understand and retrieve information based on intricate relationships between entities. **Second**, these systems often lack the contextual awareness needed to maintain coherence across various entities and their interrelations, resulting in responses that may not fully address user queries. For example, consider a user asking, "How does the rise of electric vehicles influence urban air quality and public transportation infrastructure?" Existing RAG methods might retrieve separate documents on electric vehicles, air pollution, and public transportation challenges but struggle to synthesize this information into a cohesive response. They may fail to explain how the adoption of electric vehicles can improve air quality, which in turn

[*] Both Chao Huang and Yanhua Yu are corresponding authors.

could affect public transportation planning. As a result, the user may receive a fragmented answer that does not adequately capture the complex inter-dependencies among these topics.

To address these limitations, we propose incorporating graph structures into text indexing and relevant information retrieval. Graphs are particularly effective at representing the interdependencies among different entities Rampášek et al. (2022), which enables a more nuanced understanding of relationships. The integration of graph-based knowledge structures facilitates the synthesis of information from multiple sources into coherent and contextually rich responses. Despite these advantages, developing a fast and scalable graph-empowered RAG system that efficiently handles varying query volumes is crucial. In this work, we achieve an effective and efficient RAG system by addressing three key challenges: i) **Comprehensive Information Retrieval**. Ensuring comprehensive information retrieval that captures the full context of inter-dependent entities from all documents; ii) **Enhanced Retrieval Efficiency**. Improving retrieval efficiency over the graph-based knowledge structures to significantly reduce response times; iii) **Rapid Adaptation to New Data**. Enabling quick adaptation to new data updates, ensuring the system remains relevant in dynamic environments.

In response to the outlined challenges, we propose LightRAG, a model that seamlessly integrates a graph-based text indexing paradigm with a dual-level retrieval framework. This innovative approach enhances the system's capacity to capture complex inter-dependencies among entities, resulting in more coherent and contextually rich responses. LightRAG employs efficient dual-level retrieval strategies: low-level retrieval, which focuses on precise information about specific entities and their relationships, and high-level retrieval, which encompasses broader topics and themes. By combining both detailed and conceptual retrieval, LightRAG effectively accommodates a diverse range of quries, ensuring that users receive relevant and comprehensive responses tailored to their specific needs. Additionally, by integrating graph structures with vector representations, our framework facilitates efficient retrieval of related entities and relations while enhancing the comprehensiveness of results through relevant structural information from the constructed knowledge graph.

In summary, the key contributions of this work are highlighted as follows:

- **General Aspect**. We emphasize the importance of developing a graph-empowered RAG system to overcome the limitations of existing methods. By integrating graph structures into text indexing, we can effectively represent complex interdependencies among entities, fostering a nuanced understanding of relationships and enabling coherent, contextually rich responses.
- **Methodologies**. To enable an efficient and adaptive RAG system, we propose LightRAG, which integrates a dual-level retrieval paradigm with graph-enhanced text indexing. This approach captures both low-level and high-level information for comprehensive, cost-effective retrieval. By eliminating the need to rebuild the entire index, LightRAG reduces computational costs and accelerates adaptation, while its incremental update algorithm ensures timely integration of new data, maintaining effectiveness in dynamic environments.
- **Experimental Findings**. Extensive experiments were conducted to evaluate the effectiveness of LightRAG in comparison to existing RAG models. These assessments focused on several key dimensions, including retrieval accuracy, model ablation, response efficiency, and adaptability to new information. The results demonstrated significant improvements over baseline methods.

2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) integrates user queries with a collection of pertinent documents sourced from an external knowledge database, incorporating two essential elements: the **Retrieval Component** and the **Generation Component**. 1) The retrieval component is responsible for fetching relevant documents or information from the external knowledge database. It identifies and retrieves the most pertinent data based on the input query. 2) After the retrieval process, the generation component takes the retrieved information and generates coherent, contextually relevant responses. It leverages the capabilities of the language model to produce meaningful outputs. Formally, this RAG framework, denoted as $\mathcal{M}$, can be defined as follows:

$$\mathcal{M} = \Big(\mathcal{G},\ \mathcal{R} = (\varphi, \psi)\Big), \quad \mathcal{M}(q; \mathcal{D}) = \mathcal{G}\Big(q, \psi(q; \hat{\mathcal{D}})\Big), \quad \hat{\mathcal{D}} = \varphi(\mathcal{D}) \tag{1}$$

In this framework, $\mathcal{G}$ and $\mathcal{R}$ represent the generation module and the retrieval module, respectively, while q denotes the input query and D refers to the external database. The retrieval module $\mathcal{R}$

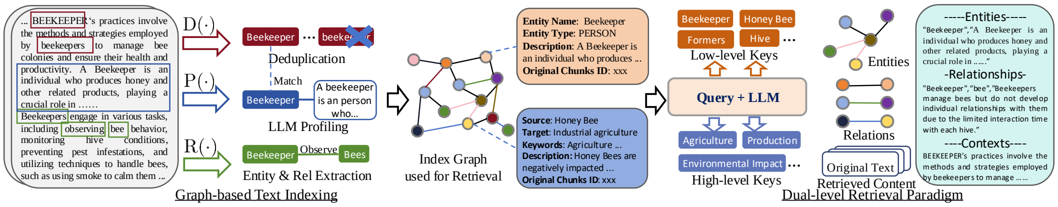
... BEEKEEPER's practices involve the methods and strategies employed by beekeepers to manage bee colonies and ensure their health and productivity. A Beekeeper is an individual who produces honey and other related products, playing a crucial role in ……
Beekeepers engage in various tasks, including observing bee behavior, monitoring hive conditions, preventing pest infestations, and utilizing techniques to handle bees, such as using smoke to calm them ...
D(·)
P(·)
R(·)
Beekeeper
beekeeper
Deduplication
Match
Beekeeper
A beekeeper is an person who...
LLM Profiling
Beekeeper
Observe
Bees
Entity & Rel Extraction
Index Graph used for Retrieval
Graph-based Text Indexing
Entity Name: Beekeeper
Entity Type: PERSON
Description: A Beekeeper is an individual who produces ...
Original Chunks ID: xxx
Source: Honey Bee
Target: Industrial agriculture
Keywords: Agriculture ...
Description: Honey Bees are negatively impacted ...
Original Chunks ID: xxx
Beekeeper
Honey Bee
Formers
Hive
Low-level Keys
Query + LLM
Agriculture
Production
Environmental Impact
High-level Keys
Entities
Relations
Original Text
Retrieved Content
-----Entities-----
"Beekeeper","A Beekeeper is an individual who produces honey and other related products, playing a crucial role in ……"
-Relationships-
"Beekeeper","bee","Beekeepers manage bees but do not develop individual relationships with them due to the limited interaction time with each hive."
----Contexts----
BEEKEEPER's practices involve the methods and strategies employed by beekeepers to manage ……
Dual-level Retrieval Paradigm

Figure 1: Overall architecture of the proposed LightRAG framework.

includes two key functionalities: i) **Data Indexer** $\varphi(\cdot)$: which involves building a specific data structure $\hat{\mathcal{D}}$ based on the external database D. ii) **Data Retriever** $\psi(\cdot)$: The relevant documents are obtained by comparing the query against the indexed data, also denoted as "relevant documents". By leveraging the information retrieved through $\psi(\cdot)$ along with the initial query q, the generative model $\mathcal{G}(\cdot)$ efficiently produces high-quality, contextually relevant responses.

In this work, we target several key points essential for an efficient and effective Retrieval-Augmented Generation (RAG) system which are elaborated below:

- **Comprehensive Information Retrieval**: The indexing function $\varphi(\cdot)$ must be adept at extracting global information, as this is crucial for enhancing the model's ability to answer queries effectively.
- **Efficient and Low-Cost Retrieval**: The indexed data structure $\hat{\mathcal{D}}$ must enable rapid and cost-efficient retrieval to effectively handle a high volume of queries.
- **Fast Adaptation to Data Changes**: The ability to swiftly and efficiently adjust the data structure to incorporate new information from the external knowledge base, is crucial for ensuring that the system remains current and relevant in an ever-changing information landscape.

3 The LightRAG Architecture

3.1 Graph-based Text Indexing

Graph-Enhanced Entity and Relationship Extraction. Our LightRAG enhances the retrieval system by segmenting documents into smaller, more manageable pieces. This strategy allows for quick identification and access to relevant information without analyzing entire documents. Next, we leverage LLMs to identify and extract various entities (e.g., names, dates, locations, and events) along with the relationships between them. The information collected through this process will be used to create a comprehensive knowledge graph that highlights the connections and insights across the entire collection of documents. We formally represent this graph generation module as follows:

$$\hat{\mathcal{D}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}}) = \text{Dedupe} \circ \text{Prof}(\mathcal{V}, \mathcal{E}), \quad \mathcal{V}, \mathcal{E} = \cup_{\mathcal{D}_i \in \mathcal{D}} \text{Recog}(\mathcal{D}_i) \tag{2}$$

where $\hat{\mathcal{D}}$ represents the resulting knowledge graphs. To generate this data, we apply three main processing steps to the raw text documents $\mathcal{D}_i$. These steps utilize a LLM for text analysis and processing. Details about the prompt templates and specific settings for this part can be found in Appendix 7.3.2. The functions used in our graph-based text indexing paradigm are described as:

- **Extracting Entities and Relationships**. R$(\cdot)$: This function prompts a LLM to identify entities (nodes) and their relationships (edges) within the text data. For instance, it can extract entities like "Cardiologists" and "Heart Disease," and relationships such as "Cardiologists diagnose Heart Disease" from the text: "Cardiologists assess symptoms to identify potential heart issues." To improve efficiency, the raw text $\mathcal{D}$ is segmented into multiple chunks $\mathcal{D}_i$.
- **LLM Profiling for Key-Value Pair Generation**. P$(\cdot)$: We employ a LLM-empowered profiling function, P$(\cdot)$, to generate a text key-value pair (K, V) for each entity node in $\mathcal{V}$ and relation edge in $\mathcal{E}$. Each index key is a word or short phrase that enables efficient retrieval, while the corresponding value is a text paragraph summarizing relevant snippets from external data to aid in text generation. Entities use their names as the sole index key, whereas relations may have multiple index keys derived from LLM enhancements that include global themes from connected entities.
- **Deduplication to Optimize Graph Operations**. D$(\cdot)$: Finally, we implement a deduplication function, D$(\cdot)$, that identifies and merges identical entities and relations from different segments of

the raw text $\mathcal{D}_i$. This process effectively reduces the overhead associated with graph operations on $\hat{\mathcal{D}}$ by minimizing the graph's size, leading to more efficient data processing.

Our LightRAG offers two advantages through its graph-based text indexing paradigm. *First*, **Comprehensive Information Understanding**. The constructed graph structures enable the extraction of global information from multi-hop subgraphs, greatly enhancing LightRAG's ability to handle complex queries that span multiple document chunks. *Second*, **Enhanced Retrieval Performance**. the key-value data structures derived from the graph are optimized for rapid and precise retrieval. This provides a superior alternative to less accurate embedding matching methods (Gao et al., 2023) and inefficient chunk traversal techniques (Edge et al., 2024) commonly used in existing approaches.

Fast Adaptation to Incremental Knowledge Base. To efficiently adapt to evolving data changes while ensuring accurate and relevant responses, our LightRAG incrementally updates the knowledge base without the need for complete reprocessing of the entire external database. For a new document $\mathcal{D}'$, the incremental update algorithm processes it using the same graph-based indexing steps φ as before, resulting in $\hat{\mathcal{D}}' = (\hat{\mathcal{V}}', \hat{\mathcal{E}}')$. Subsequently, LightRAGcombines the new graph data with the original by taking the union of the node sets $\hat{\mathcal{V}}$ and $\hat{\mathcal{V}}'$, as well as the edge sets $\hat{\mathcal{E}}$ and $\hat{\mathcal{E}}'$.

Two key objectives guide our approach to fast adaptation for the incremental knowledge base: **Seamless Integration of New Data**. By applying a consistent methodology to new information, the incremental update module allows the LightRAG to integrate new external databases without disrupting the existing graph structure. This approach preserves the integrity of established connections, ensuring that historical data remains accessible while enriching the graph without conflicts or redundancies. **Reducing Computational Overhead** . By eliminating the need to rebuild the entire index graph, this method reduces computational overhead and facilitates the rapid assimilation of new data. Consequently, LightRAG maintains system accuracy, provides current information, and conserves resources, ensuring users receive timely updates and enhancing the overall RAG effectiveness.

3.2 Dual-level Retrieval Paradigm

To retrieve relevant information from both specific document chunks and their complex interdependencies, our LightRAG proposes generating query keys at both detailed and abstract levels.

- **Specific Queries**. These queries are detail-oriented and typically reference specific entities within the graph, requiring precise retrieval of information associated with particular nodes or edges. For example, a specific query might be, "Who wrote 'Pride and Prejudice'?"
- **Abstract Queries**. In contrast, abstract queries are more conceptual, encompassing broader topics, summaries, or overarching themes that are not directly tied to specific entities. An example of an abstract query is, "How does artificial intelligence influence modern education?"

To accommodate diverse query types, the LightRAG employs two distinct retrieval strategies within the dual-level retrieval paradigm. This ensures that both specific and abstract inquiries are addressed effectively, allowing the system to deliver relevant responses tailored to user needs.

- **Low-Level Retrieval**. This level is primarily focused on retrieving specific entities along with their associated attributes or relationships. Queries at this level are detail-oriented and aim to extract precise information about particular nodes or edges within the graph.
- **High-Level Retrieval**. This level addresses broader topics and overarching themes. Queries at this level aggregate information across multiple related entities and relationships, providing insights into higher-level concepts and summaries rather than specific details.

Integrating Graph and Vectors for Efficient Retrieval. By combining graph structures with vector representations, the model gains a deeper insight into the interrelationships among entities. This synergy enables the retrieval algorithm to effectively utilize both local and global keywords, streamlining the search process and improving the relevance of results.

- (i) **Query Keyword Extraction**. For a given query q, the retrieval algorithm of LightRAG begins by extracting both local query keywords $k^{(l)}$ and global query keywords $k^{(g)}$.
- (ii) **Keyword Matching**. The algorithm uses an efficient vector database to match local query keywords with candidate entities and global query keywords with relations linked to global keys.

- (iii) **Incorporating High-Order Relatedness**. To enhance the query with higher-order relatedness, LightRAGfurther gathers neighboring nodes within the local subgraphs of the retrieved graph elements. This process involves the set $\{v_i | v_i \in \mathcal{V} \wedge (v_i \in \mathcal{N}_v \vee v_i \in \mathcal{N}_e)\}$, where $\mathcal{N}_v$ and $\mathcal{N}_e$ represent the one-hop neighboring nodes of the retrieved nodes v and edges e, respectively.

This dual-level retrieval paradigm not only facilitates efficient retrieval of related entities and relations through keyword matching, but also enhances the comprehensiveness of results by integrating relevant structural information from the constructed knowledge graph.

3.3 Retrieval-Augmented Answer Generation

Utilization of Retrieved Information. Utilizing the retrieved information $\psi(q; \hat{\mathcal{D}})$, our LightRAG employs a general-purpose LLM to generate answers based on the collected data. This data comprises concatenated values V from relevant entities and relations, produced by the profiling function $\mathrm{P}(\cdot)$. It includes names, descriptions of entities and relations, and excerpts from the original text.

Context Integration and Answer Generation. By unifying the query with this multi-source text, the LLM generates informative answers tailored to the user's needs, ensuring alignment with the query's intent. This approach streamlines the answer generation process by integrating both context and query into the LLM model, as illustrated in detailed examples (Appendix 7.2).

3.4 Complexity Analysis of the LightRAG Framework

In this section, we analyze the complexity of our proposed LightRAG framework, which can be divided into two main parts. The first part is the graph-based Index phase. During this phase, we use the large language model (LLM) to extract entities and relationships from each chunk of text. As a result, the LLM needs to be called $\frac{\text{total tokens}}{\text{chunk size}}$ times. Importantly, there is no additional overhead involved in this process, making our approach highly efficient in managing updates to new text.

The second part of the process involves the graph-based retrieval phase. For each query, we first utilize the large language model (LLM) to generate relevant keywords. Similar to current Retrieval-Augmented Generation (RAG) systems Gao et al. (2023; 2022); Chan et al. (2024), our retrieval mechanism relies on vector-based search. However, instead of retrieving chunks as in conventional RAG, we concentrate on retrieving entities and relationships. This approach markedly reduces retrieval overhead compared to the community-based traversal method used in GraphRAG.

4 Evaluation

We conduct empirical evaluations on benchmark data to assess the effectiveness of the proposed LightRAG framework by addressing the following research questions: • **(RQ1)**: How does LightRAG compare to existing RAG baseline methods in terms of generation performance? • **(RQ2)**: How do dual-level retrieval and graph-based indexing enhance the generation quality of LightRAG? • **(RQ3)**: What specific advantages does LightRAG demonstrate through case examples in various scenarios? • **(RQ4)**: What are the costs associated with LightRAG, as well as its adaptability to data changes?

4.1 Experimental Settings

Evaluation Datasets. To conduct a comprehensive analysis of LightRAG, we selected four datasets from the UltraDomain benchmark (Qian et al., 2024). The UltraDomain data is sourced from 428 college textbooks and encompasses 18 distinct domains, including agriculture, social sciences, and humanities. From these, we chose the Agriculture, CS, Legal, and Mix datasets. Each dataset contains between 600,000 and 5,000,000 tokens, with detailed information provided in Table 4. Below is a specific introduction to the four domains utilized in our experiments:

- **Agriculture**: This domain focuses on agricultural practices, covering a range of topics including beekeeping, hive management, crop production, and disease prevention.
- **CS**: This domain focuses on computer science and encompasses key areas of data science and software engineering. It particularly highlights machine learning and big data processing, featuring content on recommendation systems, classification algorithms, and real-time analytics using Spark.

- **Legal**: This domain centers on corporate legal practices, addressing corporate restructuring, legal agreements, regulatory compliance, and governance, with a focus on the legal and financial sectors.
- **Mixed**: This domain presents a rich variety of literary, biographical, and philosophical texts, spanning a broad spectrum of disciplines, including cultural, historical, and philosophical studies.

Question Generation. To evaluate the effectiveness of RAG systems for high-level sensemaking tasks, we consolidate all text content from each dataset as context and adopt the generation method outlined in Edge et al. (2024). Specifically, we instruct an LLM to generate five RAG users, along with five tasks for each user. Each generated user is accompanied by a textual description detailing their expertise and traits that motivate their question-raising activities. Each user task is also described, emphasizing one of the user's potential intentions when interacting with RAG systems. For each user-task combination, the LLM generates five questions that require an understanding of the entire corpus. In total, this process results in 125 questions for each dataset.

Baselines. LightRAG is compared against the following state-of-the-art methods across all datasets:

- **Naive RAG** (Gao et al., 2023): This model serves as a standard baseline in existing RAG systems. It segments raw texts into chunks and stores them in a vector database using text embeddings. For queries, Naive RAG generates vectorized representations to directly retrieve text chunks based on the highest similarity in their representations, ensuring efficient and straightforward matching.
- **RQ-RAG** (Chan et al., 2024): This approach leverages the LLM to decompose the input query into multiple sub-queries. These sub-queries are designed to enhance search accuracy by utilizing explicit techniques such as rewriting, decomposition, and disambiguation.
- **HyDE** (Gao et al., 2022): This method utilizes the LLM to generate a hypothetical document based on the input query. This generated document is then employed to retrieve relevant text chunks, which are subsequently used to formulate the final answer.
- **GraphRAG** (Edge et al., 2024): This is a graph-enhanced RAG system that utilizes an LLM to extract entities and relationships from the text, representing them as nodes and edges. It generates corresponding descriptions for these elements, aggregates nodes into communities, and produces a community report to capture global information. When handling high-level queries, GraphRAG retrieves more comprehensive information by traversing these communities.

Implementation and Evaluation Details. In our experiments, we utilize the *nano vector database* for vector data management and access. For all LLM-based operations in LightRAG, we default to using GPT-4o-mini. To ensure consistency, the chunk size is set to 1200 across all datasets. Additionally, the gleaning parameter is fixed at 1 for both GraphRAG and LightRAG.

Defining ground truth for many RAG queries, particularly those involving complex high-level semantics, poses significant challenges. To address this, we build on existing work (Edge et al., 2024) and adopt an LLM-based multi-dimensional comparison method. We employ a robust LLM, specifically GPT-4o-mini, to rank each baseline against our LightRAG. The evaluation prompt we used is detailed in Appendix 7.3.4. In total, we utilize four evaluation dimensions, including:

i) **Comprehensiveness**: How thoroughly does the answer address all aspects and details of the question? ii) **Diversity**: How varied and rich is the answer in offering different perspectives and insights related to the question? iii) **Empowerment**: How effectively does the answer enable the reader to understand the topic and make informed judgments? iv) **Overall**: This dimension assesses the cumulative performance across the three preceding criteria to identify the best overall answer.

The LLM directly compares two answers for each dimension and selects the superior response for each criterion. After identifying the winning answer for the three dimensions, the LLM combines the results to determine the overall better answer. To ensure a fair evaluation and mitigate the potential bias that could arise from the order in which the answers are presented in the prompt, we alternate the placement of each answer. We calculate win rates accordingly, ultimately leading to the final results.

4.2 COMPARISON OF LIGHTRAG WITH EXISTING RAG METHODS (RQ1)

We compare LightRAG against each baseline across various evaluation dimensions and datasets. The results are presented in Table 1. Based on these findings, we draw the following conclusions:

Table 1: Win rates (%) of baselines v.s. LightRAG across four datasets and four evaluation dimensions.

	Agriculture		**CS**		**Legal**		**Mix**	
	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**
Comprehensiveness	32.4%	67.6%	38.4%	61.6%	16.4%	83.6%	38.8%	61.2%
Diversity	23.6%	76.4%	38.0%	62.0%	13.6%	86.4%	32.4%	67.6%
Empowerment	32.4%	67.6%	38.8%	61.2%	16.4%	83.6%	42.8%	57.2%
Overall	32.4%	67.6%	38.8%	61.2%	15.2%	84.8%	40.0%	60.0%
	RQ-RAG	**LightRAG**	RQ-RAG	**LightRAG**	RQ-RAG	**LightRAG**	RQ-RAG	**LightRAG**
Comprehensiveness	31.6%	68.4%	38.8%	61.2%	15.2%	84.8%	39.2%	60.8%
Diversity	29.2%	70.8%	39.2%	60.8%	11.6%	88.4%	30.8%	69.2%
Empowerment	31.6%	68.4%	36.4%	63.6%	15.2%	84.8%	42.4%	57.6%
Overall	32.4%	67.6%	38.0%	62.0%	14.4%	85.6%	40.0%	60.0%
	HyDE	**LightRAG**	HyDE	**LightRAG**	HyDE	**LightRAG**	HyDE	**LightRAG**
Comprehensiveness	26.0%	74.0%	41.6%	58.4%	26.8%	73.2%	40.4%	59.6%
Diversity	24.0%	76.0%	38.8%	61.2%	20.0%	80.0%	32.4%	67.6%
Empowerment	25.2%	74.8%	40.8%	59.2%	26.0%	74.0%	46.0%	54.0%
Overall	24.8%	75.2%	41.6%	58.4%	26.4%	73.6%	42.4%	57.6%
	GraphRAG	**LightRAG**	GraphRAG	**LightRAG**	GraphRAG	**LightRAG**	GraphRAG	**LightRAG**
Comprehensiveness	45.6%	54.4%	48.4%	51.6%	48.4%	51.6%	50.4%	49.6%
Diversity	22.8%	77.2%	40.8%	59.2%	26.4%	73.6%	36.0%	64.0%
Empowerment	41.2%	58.8%	45.2%	54.8%	43.6%	56.4%	50.8%	49.2%
Overall	45.2%	54.8%	48.0%	52.0%	47.2%	52.8%	50.4%	49.6%

The Superiority of Graph-enhanced RAG Systems in Large-Scale Corpora When handling large token counts and complex queries that require a thorough understanding of the dataset's context, graph-based RAG systems like LightRAG and GraphRAG consistently outperform purely chunk-based retrieval methods such as NaiveRAG, HyDE, and RQRAG. This performance gap becomes particularly pronounced as the dataset size increases. For instance, in the largest dataset (Legal), the disparity widens significantly, with baseline methods achieving only about 20% win rates compared to the dominance of LightRAG. This trend underscores the advantages of graph-enhanced RAG systems in capturing complex semantic dependencies within large-scale corpora, facilitating a more comprehensive understanding of knowledge and leading to improved generalization performance.

Enhancing Response Diversity with LightRAG: Compared to various baselines, LightRAG demonstrates a significant advantage in the Diversity metric, particularly within the larger Legal dataset. Its consistent lead in this area underscores LightRAG's effectiveness in generating a wider range of responses, especially in scenarios where diverse content is essential. We attribute this advantage to LightRAG's dual-level retrieval paradigm, which facilitates comprehensive information retrieval from both low-level and high-level dimensions. This approach effectively leverages graph-based text indexing to consistently capture the full context in response to queries.

LightRAG's Superiority over GraphRAG: While both LightRAG and GraphRAG use graph-based retrieval mechanisms, LightRAG consistently outperforms GraphRAG, particularly in larger datasets with complex language contexts. In the Agriculture, CS, and Legal datasets—each containing millions of tokens—LightRAG shows a clear advantage, significantly surpassing GraphRAG and highlighting its strength in comprehensive information understanding within diverse environments. **Enhanced Response Variety**: By integrating low-level retrieval of specific entities with high-level retrieval of broader topics, LightRAG boosts response diversity. This dual-level mechanism effectively addresses both detailed and abstract queries, ensuring a thorough grasp of information. **Complex Query Handling**: This approach is especially valuable in scenarios requiring diverse perspectives. By accessing both specific details and overarching themes, LightRAG adeptly responds to complex queries involving interconnected topics, providing contextually relevant answers.

4.3 Ablation Studies (RQ2)

We also conduct ablation studies to evaluate the impact of our dual-level retrieval paradigm and the effectiveness of our graph-based text indexing in LightRAG. The results are presented in Table 2.

Effectiveness of Dual-level Retrieval Paradigm. We begin by analyzing the effects of low-level and high-level retrieval paradigms. We compare two ablated models—each omitting one module—against LightRAG across four datasets. Here are our key observations for the different variants:

Table 2: Performance of ablated versions of LightRAG, using NaiveRAG as reference.

	Agriculture		**CS**		**Legal**		**Mix**	
	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**	NaiveRAG	**LightRAG**
Comprehensiveness	32.4%	67.6%	38.4%	61.6%	16.4%	83.6%	38.8%	61.2%
Diversity	23.6%	76.4%	38.0%	62.0%	13.6%	86.4%	32.4%	67.6%
Empowerment	32.4%	67.6%	38.8%	61.2%	16.4%	83.6%	42.8%	57.2%
Overall	32.4%	67.6%	38.8%	61.2%	15.2%	84.8%	40.0%	60.0%
	NaiveRAG	**-High**	NaiveRAG	**-High**	NaiveRAG	**-High**	NaiveRAG	**-High**
Comprehensiveness	34.8%	65.2%	42.8%	57.2%	23.6%	76.4%	40.4%	59.6%
Diversity	27.2%	72.8%	36.8%	63.2%	16.8%	83.2%	36.0%	64.0%
Empowerment	36.0%	64.0%	42.4%	57.6%	22.8%	77.2%	47.6%	52.4%
Overall	35.2%	64.8%	44.0%	56.0%	22.0%	78.0%	42.4%	57.6%
	NaiveRAG	**-Low**	NaiveRAG	**-Low**	NaiveRAG	**-Low**	NaiveRAG	**-Low**
Comprehensiveness	36.0%	64.0%	43.2%	56.8%	19.2%	80.8%	36.0%	64.0%
Diversity	28.0%	72.0%	39.6%	60.4%	13.6%	86.4%	33.2%	66.8%
Empowerment	34.8%	65.2%	42.8%	57.2%	16.4%	83.6%	35.2%	64.8%
Overall	34.8%	65.2%	43.6%	56.4%	18.8%	81.2%	35.2%	64.8%
	NaiveRAG	**-Origin**	NaiveRAG	**-Origin**	NaiveRAG	**-Origin**	NaiveRAG	**-Origin**
Comprehensiveness	24.8%	75.2%	39.2%	60.8%	16.4%	83.6%	44.4%	55.6%
Diversity	26.4%	73.6%	44.8%	55.2%	14.4%	85.6%	25.6%	74.4%
Empowerment	32.0%	68.0%	43.2%	56.8%	17.2%	82.8%	45.2%	54.8%
Overall	25.6%	74.4%	39.2%	60.8%	15.6%	84.4%	44.4%	55.6%

- **Low-level-only Retrieval**: The -High variant removes high-order retrieval, leading to a significant performance decline across nearly all datasets and metrics. This drop is mainly due to its emphasis on the specific information, which focuses excessively on entities and their immediate neighbors. While this approach enables deeper exploration of directly related entities, it struggles to gather information for complex queries that demand comprehensive insights.
- **High-level-only Retrieval**: The -Low variant prioritizes capturing a broader range of content by leveraging entity-wise relationships rather than focusing on specific entities. This approach offers a significant advantage in comprehensiveness, allowing it to gather more extensive and varied information. However, the trade-off is a reduced depth in examining specific entities, which can limit its ability to provide highly detailed insights. Consequently, this high-level-only retrieval method may struggle with tasks that require precise, detailed answers.
- **Hybrid Mode**: The hybrid mode, or the full version of LightRAG, combines the strengths of both low-level and high-level retrieval methods. It retrieves a broader set of relationships while simultaneously conducting an in-depth exploration of specific entities. This dual-level approach ensures both breadth in the retrieval process and depth in the analysis, providing a comprehensive view of the data. As a result, LightRAG achieves balanced performance across multiple dimensions.

Semantic Graph Excels in RAG. We eliminated the use of original text in our retrieval process. Surprisingly, the resulting variant, -Origin, does not exhibit significant performance declines across all four datasets. In some cases, this variant even shows improvements (*e.g.* in Agriculture and Mix). We attribute this phenomenon to the effective extraction of key information during the graph-based indexing process, which provides sufficient context for answering queries. Additionally, the original text often contains irrelevant information that can introduce noise in the response.

4.4 CASE STUDY (RQ3)

To provide a clear comparison between baseline methods and our LightRAG, we present specific case examples in Table 3, which includes responses to a machine learning question from both the competitive baseline, GraphRAG, and our LightRAG framework. In this instance, LightRAG outperforms in all evaluation dimensions assessed by the LLM judge, including comprehensiveness, diversity, empowerment, and overall quality. Our key observations are as follows:

i) Comprehensiveness. Notably, LightRAG covers a broader range of machine learning metrics, showcasing its comprehensiveness and ability to effectively discover relevant information. This highlights the strength of our graph-based indexing paradigm, which excels in precise entity and relation extraction as well as LLM profiling. **ii) Both Diversity and Empowerment**. Furthermore, LightRAG not only offers a more diverse array of information but also delivers more empowering content. This success is due to LightRAG's hierarchical retrieval paradigm, which combines in-depth

Table 3: Case Study: Comparison Between LightRAG and the Baseline Method GraphRAG.

Query	Which methods can normalize the feature values for effective machine learning?
GraphRAG	**Key Metrics for Evaluating Movie Recommendation Systems**: Evaluating the effectiveness of movie recommendation systems is crucial for ensuring user satisfaction and improving the overall performance of these systems. Several metrics are widely recognized for their ability to provide insights into how well a recommendation system performs. Below are the most informative metrics: 1. **Precision** [...] 2. **Recall** [...] 3. **F1 Score** [...] 4. **Mean Average Precision (MAP)** [...] 5. **Root Mean Squared Error (RMSE)** [...] 6. **User Satisfaction Metrics** [...] **Conclusion**: In summary, metrics such as precision, recall, F1 score, MAP, RMSE, and user satisfaction are critical for evaluating the performance of movie recommendation systems. Each metric provides unique insights that can help developers and researchers refine their algorithms and enhance user experiences. By focusing on these metrics, stakeholders can ensure that their recommendation systems effectively meet user preferences and expectations.
LightRAG	**Key Metrics for Evaluating Movie Recommendation Systems**: Evaluating movie recommendation systems involves multiple metrics that assess the effectiveness and accuracy of the recommendations provided to users. Below are some of the most informative metrics commonly used in this context, derived from the data and concepts surrounding recommendation systems. 1. **Mean Average Precision at K (MAPK)** [...] 2. **Precision and Recall** [...] 3. **Root Mean Squared Error (RMSE) and Mean Squared Error (MSE)** [...] 4. **Area Under the Curve (AUC)** [...] 5. **F-Measure** [...] 6. **User Engagement Metrics** [...] **Conclusion**: A comprehensive evaluation of movie recommendation systems employs a range of metrics that capture both the accuracy of the recommendations and user satisfaction. By effectively leveraging these metrics, system designers can optimize recommendation engines to deliver more personalized and relevant experiences.
LLM Decision	**Comprehensiveness**: "Winner": "Answer 2 (LightRAG)", "Explanation": "Answer 2 provides a broader array of metrics including MAPK, AUC, and user engagement metrics alongside precision, recall, RMSE, and F-Measure. This gives a fuller view of how recommendation systems can be evaluated. Answer 1, while detailed, is more focused on traditional metrics and lacks the inclusion of engagement metrics which are crucial for understanding user interaction." **Diversity**: "Winner": "Answer 2 (LightRAG)", "Explanation": "Answer 2 not only covers a wide variety of metrics but also includes nuanced explanations of how some metrics interrelate and differ from one another, like the inclusion of both RMSE and MSE, as well as the details behind AUC. In contrast, Answer 1 sticks primarily to standard metrics without much exploration of potential nuances." **Empowerment**: "Winner": "Answer 2 (LightRAG)", "Explanation": "Answer 2 empowers the reader more effectively by detailing how each metric functions and its importance in evaluating recommendation systems. By providing context such as the trade-offs between precision and recall and emphasizing user engagement metrics, it enables readers to make more informed judgments and understand the implications of different metrics. Answer 1 is more straightforward but lacks the depth of insight regarding why these metrics matter." **Overall Winner**: "Winner": "Answer 2 (LightRAG)", "Explanation": "While Answer 1 is more direct and systematic, Answer 2 excels in comprehensiveness, diversity, and empowerment. It provides a richer exploration of the topic, including insights into user engagement and nuanced differences between metrics. This depth and breadth make it more informative for readers seeking to thoroughly understand the evaluation of movie recommendation systems."

explorations of related entities through low-level retrieval to enhance empowerment with broader explorations via high-level retrieval to improve answer diversity. Together, these approaches capture a comprehensive global perspective of the knowledge domain, contributing to better RAG performance.

4.5 Model Cost and Adaptability Analysis (RQ4)

We compare the cost of our LightRAG with that of the top-performing baseline, GraphRAG, from two key perspectives. First, we examine the number of tokens and API calls during the indexing and retrieval processes. Second, we analyze these metrics in relation to handling data changes in dynamic environments. The results of this evaluation on the legal dataset are presented in Table 2. In this context, T_{extract} represents the token overhead for entity and relationship extraction, C_{max} denotes the maximum number of tokens allowed per API call, and C_{extract} indicates the number of API calls required for extraction.

Figure 2: Comparison of Cost in Terms of Tokens and API Calls for GraphRAG and LightRAG on the Legal Dataset.

Phase	Retrieval Phase		Incremental Text Update	
Model	GraphRAG	Ours	GraphRAG	Ours
Tokens	$610 \times 1{,}000$	< 100	$1{,}399 \times 2 \times 5{,}000$ $+T_{\text{extract}}$	T_{extract}
API Calls	$\frac{610 \times 1{,}000}{C_{\text{max}}}$	1	$1{,}399 \times 2 + C_{\text{extract}}$	C_{extract}

In the retrieval phase, GraphRAG generates 1,399 communities, with 610 level-2 communities actively utilized for retrieval in this experiment. Each community report averages 1,000 tokens, resulting in a total token consumption of 610,000 tokens (610 communities × 1,000 tokens per community). Additionally, GraphRAG's requirement to traverse each community individually leads to hundreds of API calls, significantly increasing retrieval overhead. In contrast, LightRAG optimizes this process by using fewer than 100 tokens for keyword generation and retrieval, requiring only a single API call for the entire process. This efficiency is achieved through our retrieval mechanism, which seamlessly integrates graph structures and vectorized representations for information retrieval, thereby eliminating the need to process large volumes of information upfront.

In the incremental data update phase, designed to address changes in dynamic real-world scenarios, both models exhibit similar overhead for entity and relationship extraction. However, GraphRAG shows significant inefficiency in managing newly added data. When a new dataset of the same size as the legal dataset is introduced, GraphRAG must dismantle its existing community structure to incorporate new entities and relationships, followed by complete regeneration. This process incurs a substantial token cost of approximately 5,000 tokens per community report. Given 1,399 communities, GraphRAG would require around 1,399 × 2 × 5,000 tokens to reconstruct both the original and new community reports—an exorbitant expense that underscores its inefficiency. In contrast, LightRAG seamlessly integrates newly extracted entities and relationships into the existing graph without the need for full reconstruction. This approach results in significantly lower overhead during incremental updates, demonstrating its superior efficiency and cost-effectiveness.

5 Related Work

5.1 Retrieval-Augmented Generation with LLMs

Retrieval-Augmented Generation (RAG) systems enhance LLM inputs by retrieving relevant information from external sources, grounding responses in factual, domain-specific knowledge Ram et al. (2023); Fan et al. (2024). Current RAG approaches Gao et al. (2022; 2023); Chan et al. (2024); Yu et al. (2024) typically embed queries in a vector space to find the nearest context vectors. However, many of these methods rely on fragmented text chunks and only retrieve the top-k contexts, limiting their ability to capture comprehensive global information needed for effective responses.

Although recent studies Edge et al. (2024) have explored using graph structures for knowledge representation, two key limitations persist. First, these approaches often lack the capability for dynamic updates and expansions of the knowledge graph, making it difficult to incorporate new information effectively. In contrast, our proposed model, LightRAG, addresses these challenges by enabling the RAG system to quickly adapt to new information, ensuring the model's timeliness and accuracy. Additionally, existing methods often rely on brute-force searches for each generated community, which are inefficient for large-scale queries. Our LightRAG framework overcomes this limitation by facilitating rapid retrieval of relevant information from the graph through our proposed dual-level retrieval paradigm, significantly enhancing both retrieval efficiency and response speed.

5.2 Large Language Model for Graphs

Graphs are a powerful framework for representing complex relationships and find applications in numerous fields. As Large Language Models (LLMs) continue to evolve, researchers have increasingly focused on enhancing their capability to interpret graph-structured data. This body of work can be divided into three primary categories: i) **GNNs as Prefix** where Graph Neural Networks (GNNs) are utilized as the initial processing layer for graph data, generating structure-aware tokens that LLMs can use during inference. Notable examples include GraphGPT Tang et al. (2024) and LLaGA Chen et al. (2024). ii) **LLMs as Prefix** involves LLMs processing graph data enriched with textual information to produce node embeddings or labels, ultimately refining the training process for GNNs, as demonstrated in systems like GALM Xie et al. (2023) and OFA Liu et al. (2024). iii) **LLMs-Graphs Integration** focuses on achieving a seamless interaction between LLMs and graph data, employing techniques such as fusion training and GNN alignment, and developing LLM-based agents capable of engaging with graph information directly Li et al. (2023); Brannon et al. (2023).

6 Conclusion

This work introduces an advancement in Retrieval-Augmented Generation (RAG) through the integration of a graph-based indexing approach that enhances both efficiency and comprehension in information retrieval. LightRAG utilizes a comprehensive knowledge graph to facilitate rapid and relevant document retrieval, enabling a deeper understanding of complex queries. Its dual-level retrieval paradigm allows for the extraction of both specific and abstract information, catering to diverse user needs. Furthermore, LightRAG's seamless incremental update capability ensures that the system remains current and responsive to new information, thereby maintaining its effectiveness over time. Overall, LightRAG excels in both efficiency and effectiveness, significantly improving the speed and quality of information retrieval and generation while reducing costs for LLM inference.