G^2RAG: a GoT-perspective Graph Retrieval-Augmented Generation Paradigm

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) enhances large language models (LLMs) by integrating external knowledge into the generation process. However, existing RAG systems face limitations in processing long-form documents, primarily due to their reliance on fragmented, chunk-based retrieval mechanisms, which often fail to capture complex interdependencies. To address these limitations, we propose a **G**oT-perspective **G**raph **R**etrieval-**A**ugmented **G**eneration Paradigm (G^2RAG). G^2RAG introduces three key innovations: (1) a dynamic graph construction algorithm that adapts to document structure, (2) a dual-level retrieval framework, and (3) a context-aware retrieval scoring function. These components collectively improve retrieval diversity, semantic completeness, and preservation of contextual relationships. Experimental results demonstrate that G^2 RAG achieves an 80% reduction in inference time compared to LightRAG while maintaining competitive performance on standard query-focused summarization benchmarks. Additionally, we evaluate Graph-Based RAG on multi-hop reasoning tasks, revealing the limitations in handling complex tasks.

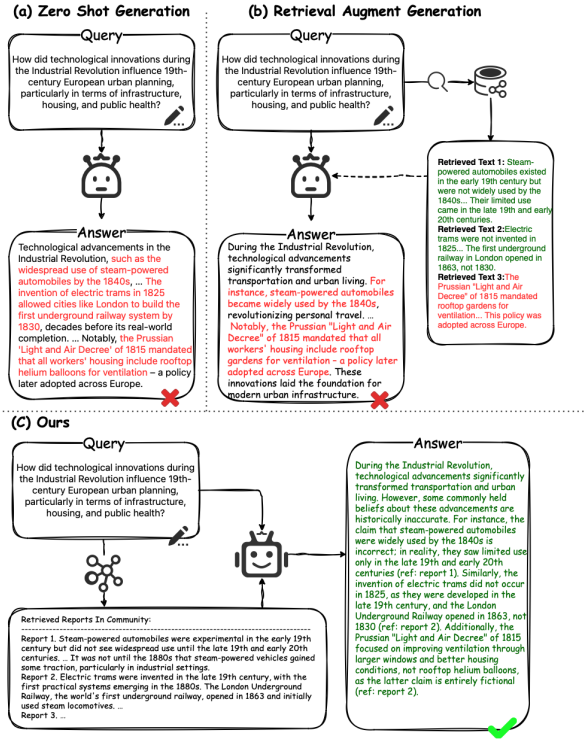

Figure 1: Comparison of QFS task performance: LLM and Naive RAG exhibit *hallucination* issues, while G^2 RAG effectively addresses them.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a key framework in natural language processing (NLP), improving the capabilities of large language models (LLM) by seamlessly integrating external knowledge into the generation process (Lewis et al., 2020; Gao et al., 2023a). Despite its widespread adoption, the existing RAG paradigm faces two critical limitations. First, processing long texts remains a significant challenge, as information from long contexts can be **"lost-in-the-middle"** (Liu et al., 2024). Second, the paradigm struggles with complex information retrieval tasks (Chen et al., 2024b), especially when applied to large-scale corpora (Zhao et al., 2024), primarily due to its dependence on flat data representations and chunk-based retrieval mechanisms, which fail to capture the nuanced relationships within the data. To address these limitations, Graph-Based RAG has recently gained traction as an innovative paradigm, leveraging graph-structured data representations to enhance retrieval and generation (Edge et al., 2024; Guo et al., 2024). Recent advances have explored the structured retrieval and traversal capabilities of graph indexes, capitalizing on the **inherent modularity** and **relational nature of graphs**. Although Graph-Based RAG methods perform well in query-focused summarization tasks (QFS) (Dang, 2006) due to rich textual annotations and hierarchical structural information, they are not

without drawbacks. Notably, the **prolonged index construction times** associated with these methods can introduce significant computational overhead, particularly when scaling to large corpora. Furthermore, the **generalization** of Graph-Based RAG across diverse NLP tasks remains understudied and more work is needed to understand how performance varies across different ranges of question types and dataset sizes. Specifically, its efficacy in knowledge-intensive applications beyond summarization, such as multihop question answering (MHQA)(Mavi et al., 2022) remains an open research question, which warrants systematic investigation to investigate its performance.

To address the limitations of existing Graph-Based RAG methods, we propose G^2**RAG**: a GoT-perspective Graph Retrieval-Augmented Generation Paradigm to enhance retrieval efficiency and relevance. Our method integrates two key innovations: (1) latent domain-aware entity-relation extraction using HDBSCAN clustering (Schubert et al., 2017) with Graph of Thoughts (GoT) (Besta et al., 2024a) for noise-resistant document merging, and (2) graph structural optimization through k-nearest neighbor graph construction and LLM-guided entity resolution. The latter involves constructing entity embeddings, identifying weakly connected components, and applying word distance filtering, with an LLM dynamically determining entity merging based on contextual coherence. This dual optimization achieves **80%** faster indexing while maintaining competitive QFS task performance through embedding-based entity and community retrieval.

We further evaluate Graph-Based RAG methods on MHQA, a challenging task that requires synthesizing information from multiple documents. We construct a large-scale corpus tailored to each MHQA task to rigorously validate the framework's feasibility. However, initial results indicate that Graph-Based RAG underperforms compared to naive rag and advanced RAG approaches. Analysis reveals that noisy index construction—caused by irrelevant data—remains a critical bottleneck, despite our optimizations. This underscores the need for further refinement in handling complex, knowledge-intensive tasks such as MHQA, where retrieval precision and contextual understanding are paramount. In summary, our contributions can be summarized as follows:

- We propose a novel framework that leverages graph modularity to enhance retrieval efficiency and relevance. By integrating latent domain-aware entity-relation extraction (via DBSCAN clustering and Graph of Thoughts) with graph structural optimization (including k-nearest neighbor graph construction and LLM-guided entity resolution), our method significantly reduces indexing time while improving coherence in retrieved information.

- We demonstrate that our framework achieves an 80% reduction in indexing time without compromising performance on query-focused summarization tasks, establishing a new benchmark for balancing computational efficiency and retrieval quality in Graph-Based systems.

- We reveal the persistent challenge of noisy index construction in complex knowledge tasks through systematic evaluation on multi-hop question answering. Our analysis provides critical insights into the limitations of current Graph-Based retrieval paradigms and underscores the need for context-aware optimization in knowledge-intensive applications.

2 Problem Formulation

2.1 Task Definition

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ be the initial set of documents (or chunks), where each d_i contains text or any knowledge component. Given a query q, and the goal of RAG is to retrieve a subset of $\mathcal{D}$ which is most relevant to q and use it with a LLM to generate the optimal answer a^*. Formally:

$$a^* = \mathrm{LLM}\big(q, R(\mathcal{D}, q)\big), \tag{1}$$

where the retrieval function $R(\mathcal{D}, q)$ selects documents from $\mathcal{D}$ based on relevance:

$$R(\mathcal{D}, q) = \Big\{ d \in \mathcal{D} \,\Big|\, \mathrm{Relevance}(q, d) \geq \tau \Big\}. \tag{2}$$

Here, $\mathrm{Relevance}(q, d)$ is a scoring function (e.g., similarity), and τ is a threshold. While existing Graph-Based RAG methods view $\mathcal{D}$ as nodes in a graph, our approach refines this process to improve both retrieval efficiency and coherence.

2.2 Module and Operation Definition

Clustering via HDBSCAN & GoT. Apply HDBSCAN to the set of embeddings $\{\mathbf{v}_i\}$ using parameters ϵ (the neighborhood radius) and minPts (the

minimum points for a cluster). The initial clustering result is $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$, where each $\mathcal{C}_k \subseteq \mathcal{D}$ is one cluster (some points may be outliers). To refine clusters based on latent domain-specific keywords or relationships, we introduce GoT. Denote $\mathrm{GoT}(\mathcal{C}_k)$ as an iterative refinement operator on cluster $\mathcal{C}_k$, which can split or merge clusters based on domain knowledge. The updated set of clusters is:

$$\mathcal{C}' = \{\mathcal{C}'_1, \mathcal{C}'_2, \ldots, \mathcal{C}'_{K'}\} = \bigcup_{k=1}^{K} \mathrm{GoT}(\mathcal{C}_k). \tag{3}$$

Extract Entity-Relation Pairs. For each cluster $\mathcal{C}_k$ (or refined sub-cluster $\mathcal{C}'_k$), use a LLM to extract entity-relation pairs $\{(e, r)\}$ that are relevant to the domain. Define the function $\mathrm{LLM}_{\mathrm{ER}}(\mathcal{C}'_k)$ to return the set of entity-relation pairs from the text in $\mathcal{C}'_k$:

$$\mathcal{E}'_k = \mathrm{LLM}_{\mathrm{ER}}(\mathcal{C}'_k) = \{(e, r) \mid e \in \mathcal{C}'_k, r \in \mathcal{R}\}. \tag{4}$$

where $\mathcal{R}$ represents the set of possible relations. These entity-relation pairs are crucial for building a more structured representation of the knowledge in each cluster and will later be used for graph construction and optimization.

K-Nearest Neighbor & Weakly Connected Components. For each cluster $\mathcal{C}'_k$ ($V_k = \mathcal{C}'_k$), let its embedding set be $\{\mathbf{v}_j\}_{j \in \mathcal{C}'_k}$. We build a k-NN graph $G_k = (V_k, E_k)$ by linking each node $\mathbf{v}_j$ to its k most similar neighbors (using cosine similarity or another distance measure). Formally:

$$E_k = \Big\{ (d_i, d_j) \,\Big|\, d_j \in \mathrm{KNN}(\mathbf{v}_i, k) \Big\}. \tag{5}$$

We detect the weakly connected components of G_k, yielding:

$$\Omega_k = \{\omega_{k1}, \omega_{k2}, \ldots, \omega_{kM_k}\}, \tag{6}$$

where each ω_{km} is a subgraph in G_k.

Distance Filtering and LLM-Guided Merging. Within each connected component ω_{km}, we apply a distance threshold δ to partition it further:

$$\omega_{km} = \bigcup_{r=1}^{R} \omega_{km}^{(r)}, \tag{7}$$

where $\{\omega_{km}^{(r)}\}_r$ are groups formed by threshold-based filtering. Then, we use a LLM-based merge function to decide whether two subgroups should be unified:

$$M_{\mathrm{LLM}}\big(\omega_{km}^{(r)}, \omega_{km}^{(s)}\big) = \begin{cases} 1, & \text{if merge} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

All subgroups flagged for merging are combined to yield refined sub-communities $\widetilde{\Omega}_k = \{\widetilde{\omega}_{k1}, \widetilde{\omega}_{k2}, \ldots\}$.

3 Method: G^2RAG

3.1 Dynamic Graph Index Construction

Index construction is crucial for the performance of retrieval-augmented generation (RAG) systems. In large corpora, noise from irrelevant data can degrade indexing efficiency and retrieval accuracy. To address this, we enhance the process with chunk-level merging and compression. Merging groups related data into larger, coherent chunks, while compression reduces graph size without losing key details. These strategies improve indexing quality, reduce memory usage, and speed up traversal.

A critical component of this process is the application of HDBSCAN, which clusters similar data points while isolating noisy or outlier points. This step ensures that only coherent, high-quality data enters the subsequent stages of the indexing pipeline, significantly enhancing the index's overall integrity and relevance. Once the data $\mathcal{D}$ has been clustered to $\mathcal{C}$, the GoT method $\mathrm{GoT}(\mathcal{C}_k)$ is employed for dynamic, application-specific noise filtering. By tailoring filtering strategies to the requirements of each specific task, GoT effectively excludes irrelevant data, preserving only the most pertinent information. This two-step process of clustering and targeted noise removal optimizes the indexing pipeline, reducing unnecessary data and resource consumption while maintaining task-specific relevance.

After noise information has been filtered, we leverage LLMs to extract entity-relation pairs $\mathcal{E}'_k$ within the clusters $\mathcal{C}'$. LLMs identify meaningful entities and their interrelationships, ensuring that the graph structure accurately reflects the underlying knowledge. K-Nearest Neighbors (KNN) and Weakly Connected Components (WCC) are then applied to refine the graph: KNN connects nodes based on their proximity, reinforcing the graph's relevance, while WCC isolates disconnected or weakly connected components Ω_k, improving retrieval efficiency. Finally, LLMs are used to merge connected components ω_{km} which are subsequently used to construct communities $\widetilde{\Omega}_k$.

3.2 Dual-level Graph Retrieval

We propose Entity to Community retrieval which allows LLMs to retrieve both detailed information about specific entities and broader, contextual summaries from relevant communities.

Entity represents a specific object, concept, or individual, such as a person, place or event. The

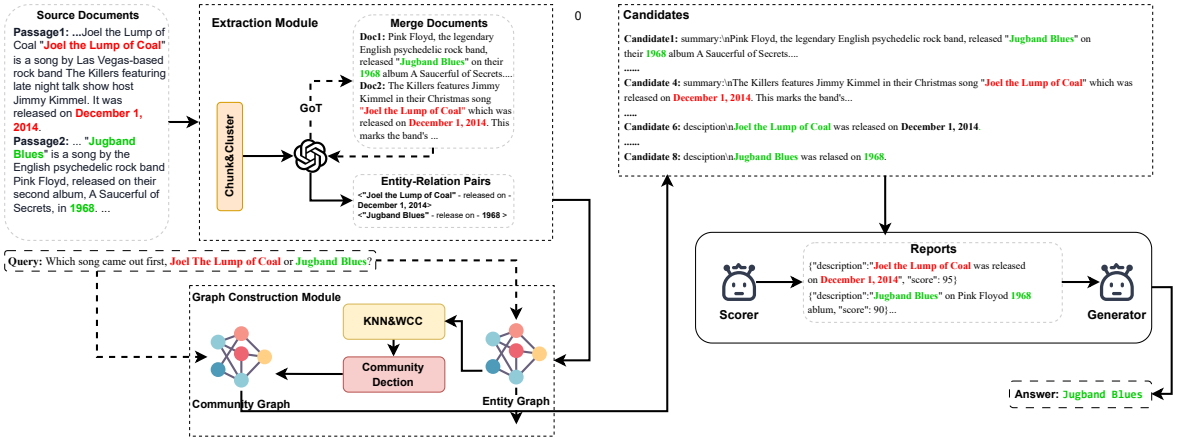

Figure 2: Frame work of G^2RAG. The pipeline consists of three main stages: Index Construction, where HDBSCAN clusters data, GoT filters noise, and LLMs extract entity relations, followed by KNN-WCC-based redundant node merging and community construction; Retrieval, where both entity-level and community-level information are queried to enhance relevance; and Generation, where retrieved information is scored for relevance, structured into context, and fed into the model to generate the final response.

information retrieved at the entity level typically includes detailed attributes and relationships specific to that entity.

Community encompasses a broader network of relationships and contextual connections. A community includes not only the entity in question but also the other entities that are closely related to it, either through direct interactions or shared contexts.

3.3 Answer Generation

Our framework employs a three-step process for knowledge-aware answer generation. We retrieve relevant graph communities by measuring semantic similarity between the user query embedding and community representations in the shared vector space. For each candidate community, the model performs context-aware summarization, analyzing structural relationships and semantic content to extract query-relevant information, which is then scored based on relevance and completeness. These summaries are aggregated in descending order of their scores and progressively fed into the generation model. This hierarchical approach prioritizes salient information while maintaining supplementary context, enabling the model to synthesize coherent responses that effectively leverage the graph's relational knowledge.

4 Adversarial Evaluation of QFS Tasks

Due to the current lack of benchmark datasets and gold-standard metrics for evaluating Graph-Based summarization methods, we follow the approach of existing studies by constructing questions and employing LLMs for multi-dimensional evaluation.

Question Construction To ensure a detailed and accurate evaluation, we selected three datasets from LongBench v2 (Bai et al., 2024). LongBench v2 consists of 503 challenging multiple-choice questions, with context lengths ranging from 8K to 2M tokens. Specifically, we extracted and utilized the Academic, Governmental, and Legal datasets, where the number of tokens per dataset ranges from 60K to 160K.

To evaluate our method for advanced QFS tasks, we consolidated each dataset into a single context and applied the question generation approach (Edge et al., 2024). The LLM generated five QFS users, each with five distinct tasks, accompanied by detailed descriptions to contextualize their expertise and intent. For each user-task pair, the LLM produced five questions requiring a comprehensive understanding of the dataset, totaling 125 questions per dataset.

Assessment Details We compare G^2RAG with the following baselines: (1) Naive baselines: *w/o documents*, where LLMs generate answers according to their inherent knowledge, *w/ documents*, where LLMs generate answers with external retrieved knowledge (i.e., the NaiveRAG); (2) LongRAG (Zhao et al., 2024). This is a framework consisting of long retriever and long reader; (3) RqRAG (Chan et al., 2024) where LLMs

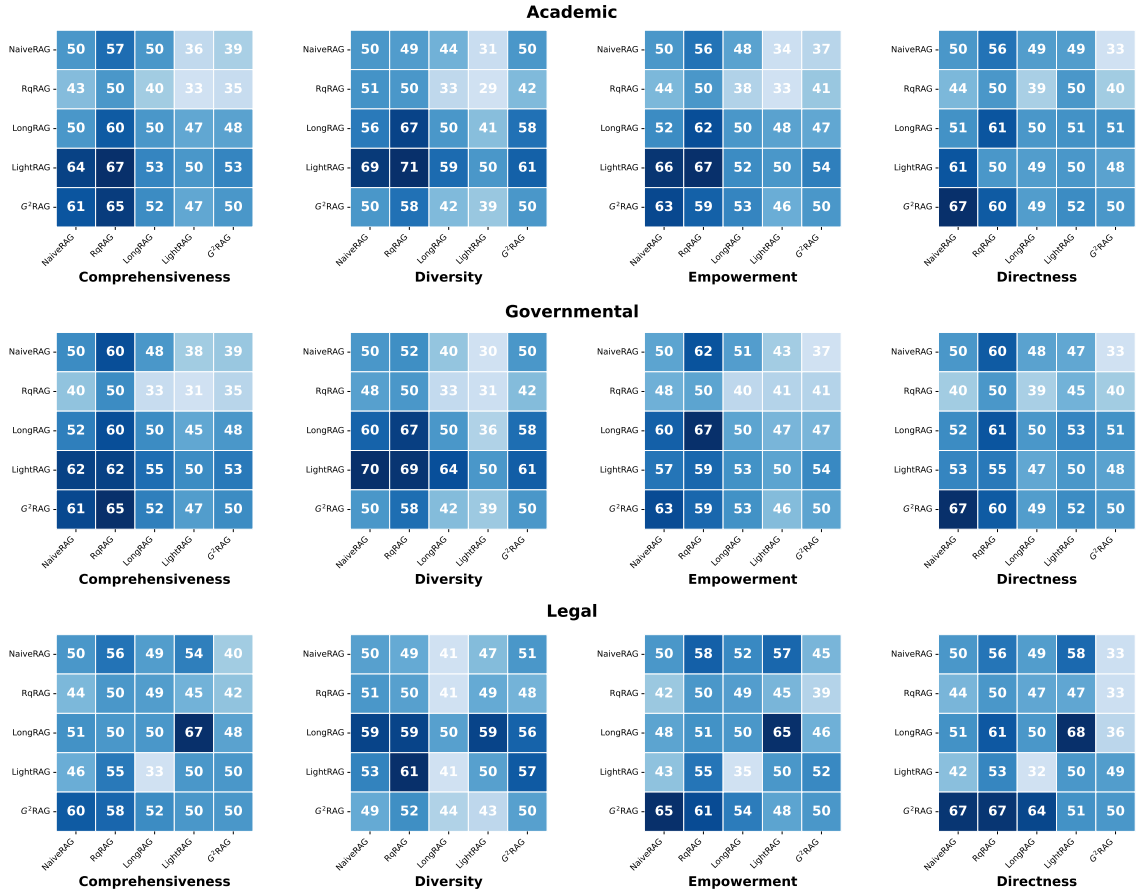

Figure 3: Win rate percentages of G^2RAG and baseline methods (NaiveRAG, RqRAG, LightRAG, LongRAG) on the Comprehensiveness, Diversity, Directness, and Empowerment metrics across the Academic, Governmental, and Legal datasets. Overall, G^2RAG demonstrates performance on par with, or exceeding, the fine-tuned LongRAG, while outperforming non-fine-tuned baselines.

are trained to dynamically refine search queries through rewriting, decomposing, and clarifying ambiguities; (4) LightRAG (Guo et al., 2024) a dual-level retrieval architecture with knowledge graphs. Since LongRAG requires fine-tuning, we use Llama-3.1-8B (Touvron et al., 2023) as the backbone, while other methods adopt GPT-4o-mini (Achiam et al., 2023) as the generator. We assess model performance across the following four dimensions:

- **Comprehensiveness.** A comprehensive answer meticulously covers every facet and nuance of the question, leaving no critical detail unaddressed.
- **Diversity.** A diverse answer incorporates a wide range of perspectives, insights, and approaches, enriching the response with varied viewpoints.
- **Empowerment.** An empowering answer equips the reader with the knowledge and tools necessary to grasp the topic fully and make well-informed decisions.
- **Directness.** A direct answer addresses the question with precision and clarity, avoiding ambiguity or unnecessary digressions.

Results As illustrated in Figure 3, G^2RAG achieves performance surpassing that of the fine-tuned LongRAG across datasets. Moreover, it outperforms several advanced methods that do not require fine-tuning, including RqRAG, and LightRAG. In the Academic and Governmental domains, G^2RAG attains coverage levels on par with LightRAG while significantly exceeding those of NaiveRAG, RqRAG, and LongRAG. Notably, in the Legal domain, G^2RAG demonstrates substantially higher comprehensiveness scores than all baseline methods, highlighting its effectiveness in integrating and structuring large volumes of legal texts.

However, G^2RAG may exhibit slight diversity limitations under specific conditions due to its retrieval process, which prioritizes relevance through scoring and refinement. While this ensures precise and direct responses, it may reduce the inclusion of exploratory information. However, in the legal domain, this trait is advantageous, as legal texts demand accuracy, and ambiguity can lead to misinterpretation. By focusing on relevance and structured integration, G^2RAG effectively filters out extraneous content, ensuring precise and applicable legal information. Thus, despite its slightly constrained diversity, its ability to deliver highly accurate responses makes it particularly well-suited for legal applications.

5 A Comprehensive Study of Graph-Based RAG in MHQA

Research on Graph-Based RAG remains limited. In this study, we compare the performance of Graph-Based methods with advanced RAG methods and the Naive RAG approach in the context of MHQA. Through experiments and case studies, we analyze the effectiveness of these methods. Our findings provide valuable insights into the potential of Graph-Based RAG.

Evaluation datasets. We measure all the methods on three MHQA datasets, including (1) HotpotQA (Yang et al., 2018), (2) 2WikiMQA (Ho et al., 2020), (3) MuSiQue (Trivedi et al., 2022). As evaluation metrics, we calculate the exact match (EM), F1 score and accuracy (Acc) for multi-hop reasoning datasets. We use the corresponding documents of 3 datasets from the LongBench(Bai et al., 2023) benchmark for corpus construction.

Implementation Details. In this study, we follow the baselines used in §4 and conduct experiments using multiple backbone models from the MHQA task benchmark, alongside our proposed method for comparison. We select include Llama-3.1-8B-Instruct (Touvron et al., 2023), Qwen2.5-7B-Instruct(Yang et al., 2024), and Ministral-8B-Instruct-2410 (Jiang et al., 2023) as the baseline models. For subsequent analysis experiments and graph index building, we use GPT-4o-mini (Achiam et al., 2023). To ensure the consistency of the experiments, all datasets are set to a block size of 500. For the retrieval process, we employ bge-m3 (Chen et al., 2024a) as the retriever and top-k is set to 5. In terms of data storage and management, our method uses Neo4j for data storage and access.

Results and Analysis As delineated in Table 1, our experimental results suggest that Graph-Based RAG methods, including G^2RAG and LightRAG, perform suboptimally on MHQA datasets compared to advanced RAG methods. In fact, G^2RAG demonstrates weaker performance than the baseline Naive RAG across all the datasets. On the HotpotQA dataset answered by Llama-3.1-8B, G^2RAG reaches an accuracy score of 28.5 against parametric knowledge, still underperforming 38.5 against naive rag (w/ documents). The performance of LightRAG, which is an other Graph-Based RAG method shows a degradation in performance. On the 2WikiMultiHopQA dataset, LightRAG achieves an accuracy score of just 23.5, even lower than naive baseline (w/o documents) at 28.0, showing that in some cases, Graph-Based methods can even result in performance degradation.

Case Study We conduct a case study to investigate the reason why Graph-Based RAG methods underperform on MHQA tasks. Table 3 shows the retrieval contents and final answers for questions on HotpotQA dataset. It reveals two main reasons behind the subpar performance of Graph-Based RAG methods like G^2RAG and LightRAG on multi-hop QA datasets. First, the entity-relation node extraction is incomplete, especially for low-frequency entities, which are often overlooked, leading to gaps in the graph structure. This makes it difficult to recognize and link entities during multi-hop reasoning. Second, Graph-Based methods fail to identify potential relationships between entities across different documents during the indexing phase. As a result, even if a graph index is built, it doesn't fully support multi-hop reasoning since it cannot capture cross-document relationships. Modular graph communities, to some extent, help identify some cross-document connections to mitigate the problem but still struggle to enable comprehensive multi-hop reasoning. These limitations highlight the need for improvements in entity extraction and cross-document relationship recognition to enhance the effectiveness of Graph-Based RAG methods in other tasks.

Effectiveness of GoT GoT significantly enhances graph index construction by leveraging highly efficient compression techniques and per-

Model	HotPotQA			2WikiMultiHopQA			MuSiQue		
	EM	ACC	F1	EM	ACC	F1	EM	ACC	F1
Llama-3.1-8B									
w/ documents	31.0	38.5	43.2	26.0	31.5	33.3	10.5	12.5	16.0
w/o documents	16.5	21.5	24.1	19.0	28.0	26.5	5.5	7.5	10.5
LongRAG w/ finetune	**47.0**	**51.5**	**61.6**	**55.0**	**62.5**	**64.0**	**27.0**	**32.5**	**37.5**
LongRAG w/o finetune	41.0	48.0	54.3	48.5	60.0	57.3	24.0	31.0	33.0
RqRAG	36.5	40.0	47.0	24.0	30.5	32.4	20.5	21.5	28.0
LightRAG	16.5	21.5	23.1	11.0	23.5	19.4	2.0	2.5	5.1
Ours	**20.0**	**28.5**	**28.5**	**17.0**	**30.5**	**24.9**	**8.5**	**11.0**	**13.7**
Ministral-8B-Instruct									
w/ documents	36.5	43.0	49.3	31.0	34.5	38.4	12.0	13.5	17.3
w/o documents	18.0	19.5	25.0	20.0	22.0	25.2	4.5	5.0	9.1
LongRAG w/ finetune	**46.5**	**53.5**	**61.4**	**49.0**	**58.5**	**58.5**	**33.0**	**39.5**	**44.0**
LongRAG w/o finetune	37.0	44.0	51.2	31.5	39.0	40.0	23.0	28.5	31.0
RqRAG	36.5	42.5	48.6	30.0	33.5	36.9	20.0	23.5	28.6
LightRAG	**18.0**	19.5	23.6	**21.5**	**22.5**	**26.1**	4.5	5.5	9.6
Ours	18.0	**22.5**	**27.9**	19.0	19.5	23.2	**7.5**	**8.5**	**12.2**
Qwen-2.5-7B									
w/ documents	35.5	41.0	46.7	28.0	32.0	34.9	10.0	14.5	16.5
w/o documents	17.5	23.0	27.3	23.0	24.5	27.6	3.0	5.5	11.2
LongRAG w/ finetune	**49.5**	**57.5**	**63.3**	**51.0**	**58.0**	**59.1**	**26.5**	**32.5**	**37.3**
LongRAG w/o finetune	44.5	53.0	58.1	42.0	54.0	53.4	25.0	31.5	32.2
RqRAG	35.5	39.0	47.0	28.0	31.5	35.2	19.5	21.0	26.4
LightRAG	20.0	24.0	27.0	21.5	22.5	25.8	6.5	8.0	11.7
Ours	**20.0**	**26.0**	**28.6**	**23.0**	**23.5**	**27.8**	**7.0**	**10.5**	**13.0**

Table 1: Overall performance (%) of graph-based rag and traditional rag of the dev sets of multi-hop QA datasets.

formance optimizations. It achieves a remarkable compression rate exceeding 40%, substantially reducing document chunk size. This compression not only optimizes storage but also accelerates data access and processing, leading to a more efficient construction process. Additionally, GoT enhances LLMs' ability to filter redundant information while preserving critical data, ensuring both speed and effectiveness in index construction. As shown in Table 2, merging chunks slightly improves LLM performance compared to using full chunks.

Speed of Index Construction G^2RAG achieves a remarkable speedup in index construction by efficiently extracting entity-relation pairs through clustering and GoT optimization, followed by node merging via KNN and WCC to form communities. This streamlined process enables our approach to attain indexing speeds comparable to NaiveRAG, even when the latter employs FAISS-based index construction. Compared to LightRAG, our method accelerates graph index construction by 80%, demonstrating its superior efficiency in large-scale retrieval tasks.

Method	EM	ACC	F1	# Token
w/o GoT	26.5	33.5	36.5	362k
w/ GoT	27.0	36.0	38.4	202k

Table 2: Performance (%) of G^2RAG under different settings of chunking processing.

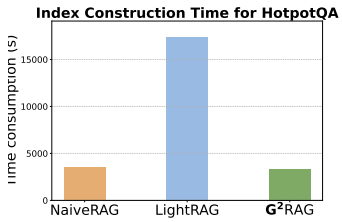

Figure 4: Comparison of time consumption (s) for index construction across different methods.

6 Related Work

6.1 Retrieval-Augmented Generation.

Retrieval-Augmented generation (RAG) represents a paradigm shift in knowledge-intensive NLP tasks,

Question: Who starred in her final film role in the 1964 film directed by the man who also did Vera Cruz and Kiss Me Deadly?
NaiveRAG: Vera Cruz is a 1954 American Western film directed by **Robert Aldrich** ... His most notable credits include Vera Cruz (1954), **Kiss Me Deadly (1955)** Hush...Hush, Sweet Charlotte is a 1964 American psychological thriller film ... **Mary Astor** in her final film role **Answer:** Mary Astor.
LightRAG: None **Answer:** Kathy Bates starred in her final film role in the 1964 film directed by the man who also did Vera Cruz and Kiss Me Deadly.
Ours: The provided information centers around Robert Aldrich ... Aldrich's filmography includes notable titles such as "Whatever Happened to Baby Jane?", "The Dirty Dozen," and **"Kiss Me Deadly."** **Answer**: Bette Davis.

Table 3: Case Study on HotPotQA dataset. Only key information fragments retrieved are displayed. Failure to retrieve the entity "Mary Astor" in LightRAG and G^2RAG leads to an incorrect answer.

where parametric knowledge in language models is augmented with non-parametric external memory.(Gao et al., 2023b; Ram et al., 2023; Asai et al., 2023) Existing approaches of using embedded queries and vector retrieval libraries to access relevant information faces limitations due to information fragmentation caused by text chunking and restricted retrieval capacity imposed by language models' context length, hindering the acquisition of coherent and comprehensive information (Gao et al., 2022; Günther et al., 2024) .

Recent advancements(Guo et al., 2024; Besta et al., 2024a; Fan et al., 2025) attempt to address these limitations through graph-structured representations, where documents are modeled as interconnected knowledge graphs. This Graph-Based RAG paradigm enables more sophisticated reasoning over retrieved information by explicitly capturing entity relationships and document-level dependencies. However, most current knowledged Graph-Based RAG methods still face prohibitive indexing construction times and excessive API cost overhead. Driven by these limitations, we focus on developing efficient RAG systems for resource-constrained scenarios.

6.2 Chain of Thoughts

The evolution of reasoning in language models has progressed through several significant paradigms. The chain-of-thought (CoT) approach (Wei et al., 2022)first demonstrated that explicit reasoning chains could enhance model performance on complex tasks. This was subsequently extended through tree-of-thought frameworks(Yao et al., 2023), which introduced branching reasoning paths to explore multiple solution trajectories. Other work like PoT(Chen et al., 2022),CoT-SC(Wang et al., 2022),AoT(Sel et al., 2023), likewise shows great potential for the enhancement of LLMs. Although these approaches have shown promise, they often struggle with maintaining coherent reasoning in extended contexts and do not integrate external knowledge effectively.

The graph-of-thought (GoT) paradigm(Besta et al., 2024b) addresses these limitations through its unique capability to effectively compress multiple information units into consolidated representations. Our work advances this paradigm by developing a novel indexing optimization framework that significantly reduces construction overhead.

7 Conclusion

In this paper, we introduce G^2**RAG**, a novel Graph Retrieval-Augmented Generation paradigm that leverages graph modularity to enhance retrieval efficiency and semantic relevance. By integrating latent domain-aware entity-relation extraction using DBSCAN clustering with Graph of Thoughts-based document merging, as well as optimizing graph structures through k-nearest neighbor graph construction and LLM-guided entity resolution, our framework significantly improves retrieval performance. Experimental results show that G^2RAG reduces indexing time by 80% while maintaining competitive performance on query-focused summarization (QFS) benchmarks. Additionally, we systematically evaluate Graph-Based RAG methods on multi-hop question answering (MHQA), revealing key challenges in retrieval precision and contextual integration. Our findings highlight the potential of graph-based retrieval in structured document processing while exposing limitations that need further refinement. Our study provides insights into the trade-offs between retrieval efficiency and generative quality, paving the way for future advancements in Graph-Based RAG frameworks.

Limitations

We find that G^2RAG struggles to effectively identify and extract low-frequency information, which may impact retrieval completeness. Its generalization to broader NLP tasks requires further validation, and real-time efficiency in large-scale settings remains a challenge. While our approach mitigates computational overhead, scalable graph construction techniques are needed to enhance adaptability for dynamic, large-scale corpora.

References

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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024a. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024b. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

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

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Hoa Trang Dang. 2006. Duc 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55.

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

Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. 2025. Minirag: Towards extremely simple retrieval-augmented generation. *arXiv preprint arXiv:2501.06713*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

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

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

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

Bilgehan Sel, Ahmad Al-Tawaha, Vanshaj Khattar, Ruoxi Jia, and Ming Jin. 2023. Algorithm of thoughts: Enhancing exploration of ideas in large language models. *arXiv preprint arXiv:2308.10379*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. 2024. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. *arXiv preprint arXiv:2410.18050*.