TH-RAG : Topic-Based Hierarchical Knowledge Graphs for Robust Multi-hop Reasoning in GraphRAG Systems

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) enhances large language models (LLMs) by enabling them to incorporate external knowledge at inference time. While graph-based RAG methods have shown promise in multi-hop reasoning by leveraging structured representations such as triplets, they often struggle with semantic sparsity, noisy or inconsistent triplet extraction, and a lack of higher-level abstraction, which together hinder coherent and efficient reasoning. We propose **TH-RAG**, a novel graph-based RAG framework that constructs **a three-level hierarchical Knowledge Graph (KG)** composed of entities, subtopics, and topics. TH-RAG maintains high connectivity by semantically organizing triplets through **Triplet Extraction with Topic**. With **Topic-based Hierarchical Graph Traversal**, TH-RAG finds related entities through topic and subtopics. Finally, a **Query-Based Filtering** selects only the most relevant triplets and sentence chunks. Experimental results on both open-domain and multi-hop QA benchmarks demonstrate that TH-RAG consistently outperforms existing strong baselines in terms of accuracy and robustness. To support further research, we release our code at: https://anonymous.4open.science/r/KGRAG-2C8D

1 Introduction

In recent years, large language models (LLMs) have demonstrated outstanding performance across various natural language processing tasks (Achiam et al., 2023; Yang et al., 2025a; Matarazzo and Torlone, 2025), owing to their extended context windows and strong document understanding capabilities (Team et al., 2023; Guo et al., 2025). However, integrating new knowledge into LLMs typically requires iterative fine-tuning, which incurs significant computational costs, consumes time, and introduces the risk of catastrophic forgetting (Kirkpatrick et al., 2017; Luo et al., 2023).

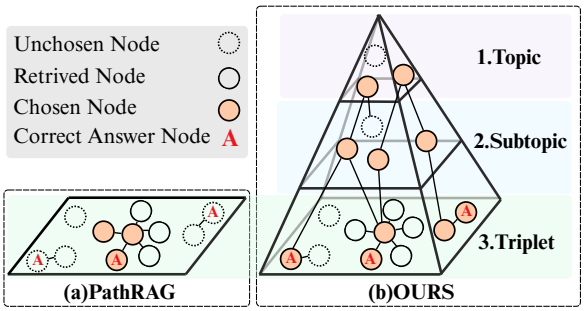

Figure 1: Simple example of TH-RAG compared to PathRAG. TH-RAG can retrieve almost all information in corpus effeciently, since use hierarchical graph structure.

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2024) and graph-based RAG (Han et al., 2024) offers a promising alternative to overcome these challenges. RAG leverages sparse or dense retrieval mechanisms (Robertson and Zaragoza, 2009; Karpukhin et al., 2020) to fetch relevant information from external corpora and generates responses based on the retrieved content (Soudani et al., 2024; Balaguer et al., 2024). Graph-based RAG further extends this paradigm by structuring the retrieval database as Knowledge Graphs (KGs) (Yang et al., 2025b; Kamra et al., 2024). his approach brings two key advantages: (1) it improves multi-hop reasoning over dispersed information compared to standard RAG, and (2) it facilitates the understanding of documents by capturing their logical structure and semantic relationships (Peng et al., 2024; Wu et al., 2024).

Recent graph-based RAG methods(Edge et al., 2024; Guo et al., 2024) have focused on constructing KGs directly from domain-specific corpora by extracting triplets (subject–relation–object). This fine-grained representation improves the precision of reasoning by structuring information at the semantic level. However, these methods often assume that sufficient connectivity exists among triplets within a chunk, which is rarely true in practice

(Han et al., 2024; Zhu et al., 2025c).

Moreover, triplet-based graphs generated by large language models are frequently fragmented, which significantly hinders effective reasoning over the graph. To address this, several studies have proposed techniques such as graph clustering, community detection, or node merging based on summarization (Edge et al., 2024; Xu et al., 2025; Wang et al., 2025). While these methods attempt to restore coherence, they often introduce additional computational overhead and may distort semantics or hallucinate facts, thereby compromising the integrity of the information.

To overcome these challenges, we propose **TH-RAG**, a novel graph-based RAG framework that constructs a **three-level hierarchical KG composed of Triplets, Subtopics, and Topics**. This semantic hierarchy enhances graph connectivity, facilitates integration across fragmented information, and supports efficient multi-hop reasoning. TH-RAG operates in three stages: (1) **Hierarchical KG Construction**, where an LLM extracts Triplets, Subtopics, and Topics simultaneously to build a semantically structured graph; (2) **Topic-based Graph Traversal**, which begins from the most relevant Topic node and recursively explores related Subtopics and Entities to retrieve candidate Triplets; and (3) **Query-based Retrieval & Filtering**, where cosine similarity is computed between the query and each edge of candidate triplet, and the most relevant information is selected as the final context for answer generation.

Experimental results show that TH-RAG outperforms existing graph-based RAG methods across both general-purpose (abroad-type) and domain-specific (specific-type) QA benchmarks. Notably, TH-RAG achieves accuracy improvements of 6.9 and 1.3 percentage points over current state-of-the-art methods on the MultiHopRAG and HotpotQA datasets, respectively. Additional ablation studies validate the effectiveness of our hierarchical graph design and retrieval strategy, demonstrating that TH-RAG offers a promising and scalable approach for enabling more robust multihop reasoning in triplet-based graph RAG systems.

2 Related Works

2.1 RAG and the Graph-based RAG

Early RAG methods(Gao et al., 2024) utilized dense retrievers such as DPR(Karpukhin et al., 2020; Lewis et al., 2020) to chunk long documents into smaller units and retrieve relevant chunks based on similarity to a given query(Sharma, 2025; Hu and Lu, 2024; Gao et al., 2023). Since then, the RAG framework has evolved through integration with techniques such as reranking (Chen et al., 2024), query expansion (Jagerman et al., 2023; Wang et al., 2023; Chan et al., 2024), and fusion-in-decoder (Izacard and Grave, 2021).

However, similarity-based retrieval alone often struggles with capturing logical dependencies or supporting multi-hop reasoning (Zhao et al., 2024; Wu et al., 2024). To address this limitation, graph-based RAG approaches have been proposed (Peng et al., 2024; Zhang et al., 2025b).

Initial graph-based RAG systems (Sun et al., 2023; Ma et al., 2024) relied on pre-constructed KGs such as Freebase (Bollacker et al., 2008) or Wikidata (Vrandečić and Krötzsch, 2014). More recent work has shifted toward constructing graphs directly from the corpus to improve adaptability to domain-specific settings (Zhu et al., 2024; Chen and Bertozzi, 2023).

2.2 Triplet-based Graph-based RAG

Triplet-based Graph-based RAG focuses on extracting triplets from within document chunks to build structured representations at the entity level (Han et al., 2024; Zhu et al., 2025c).

These triplet-based graphs are used to support structured multi-hop reasoning over the document content. Recent studies propose various enhancements to this paradigm, such as improving triplet connectivity (Luo et al., 2025), enabling lightweight reasoning (Luo et al., 2024; Böckling et al., 2025), or focusing on explicit path-based retrieval (Chen et al., 2025).

Edge et al. (2024)demonstrated promising results by constructing a triplet-based KG and applying community detection (Traag et al., 2019) to enhance semantic grouping and retrieval. Guo et al. (2024); Abane et al. (2024) proposed a more efficient and simplified usage of such graphs, relying on coarse graph structure for lightweight retrieval.

Subsequent studies (Liang et al., 2024; Jimenez Gutierrez et al., 2024; Gutiérrez et al., 2025) have explored various strategies to enhance graph connectivity and utility (Panda et al., 2024; Zhao et al., 2025). Some methods(Xu et al., 2025; Wang et al., 2025) adopt clustering techniques such as HNSW (Malkov, 2018) to group similar entities and reduce graph sparsity, while others employ explicit graph traversal strategies—such as path

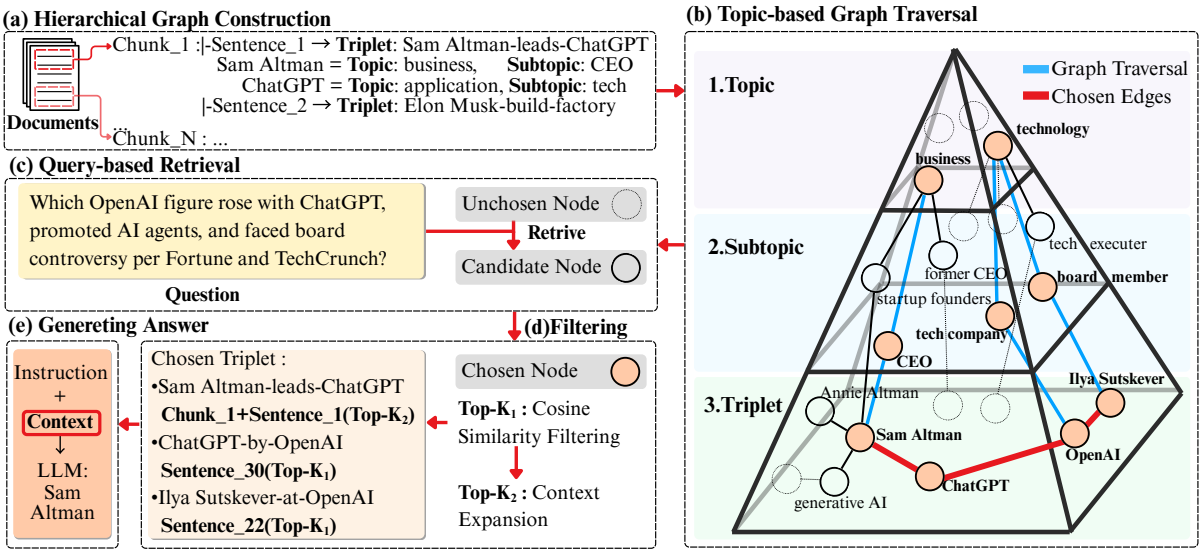

Figure 2: Overview of TH-RAG Framework. The framework consists of three stages: (a) **Hierarchical Graph Construction** – Documents are processed into triplet extraction with topic to build a hierarchical graph. (b) **Topic-based Graph Traversal** – The graph is navigated from topic to subtopic to triplet, guided by LLM-based relevance to the query. (c-d) **Query-based Retrieval & Filtering** – Triplets linked to selected subtopics are expanded by retrieving 1-hop neighboring entities. Retrieved triplets are filtered by cosine similarity (Top-K_1), and context is expanded from relevant chunks (Top-K_2).

finding or reasoning over entity connections—to support multi-hop question answering (Luo et al., 2024; Han et al., 2025b; Chen et al., 2025; Böckling et al., 2025). In addition, several hybrid approaches (Zhu et al., 2025a; Sarmah et al., 2024) have been proposed that combine graph-based structures with traditional chunk-level retrieval.

3 Method

We now describe the architecture and implementation details of TH-RAG. Each component of the framework corresponds to the stages illustrated in Figure 2 with simple example.

3.1 Hierarchical Graph Construction

Following prior graph-based RAG approaches, TH-RAG constructs a KG from a corpus by extracting triplets (subject–relation–object) using an LLM.

As discussed in Section 1, existing methods for addressing graph fragmentation face two key limitations: increased computational cost and impaired information fidelity.

To improve efficiency and minimize information distortion,, we propose **Triplet Extraction with Topic**, a method that augments each extracted triplet with subtopic and topic annotations to form a hierarchically structured graph. Each entity is connected to one or more subtopics, and each subtopic to one or more topics, creating a semantic containment hierarchy:

- **Entities** represent factual units.
- **Subtopics** cluster semantically related entities.
- **Topics** abstract groups of subtopics into higher-level categories.

To ensure semantic grounding, we instruct the LLM to extract only corpus-relevant subtopics and topics (see Table 11). The output example can be found on Appendix D.1

Then We define the resulting graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where:

- V consists of three disjoint sets of nodes:
  - E: entity nodes,
  - ST: subtopic nodes,
  - T: topic nodes.
- $\mathcal{E}$ consists of typed directed edges:
  - $E_{triplet}$: entity-to-entity relations (i.e., (h, r, t) where $h, t \in V$),
  - E_{sub}: subtopic–entity links (i.e., (s, e) where $s \in ST, e \in E$),
  - E_{top}: topic–subtopic links (i.e., (t, s) where $t \in T, s \in ST$).

Each edge in $E_{triplet}$ also stores its source sentence as an attribute for sentence-level retrieval.

Each entity is connected via at least two edges, and each subtopic is linked to both its entities and

parent topic. This structure improves connectivity and prevents node isolation.

Each triplet also stores its source sentence as an edge attribute. As these sentences are directly extracted from the corpus, hallucination risk is minimized. These annotations support sentence-level retrieval in later stages. Moreover, graph updates require only one LLM call per chunk, making the method highly scalable.

The overall process of graph construction is summarized in Algorithm 1.

Algorithm 1 Hierarchical Graph Construction

Input: Corpus chunk C
Output: Hierarchical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with triplets, subtopics, and topics
$TRIP, ST, T = LLM(C_i)$
for each triplet $(s, r, o) \in TRIP$ **do**
    Attach source sentence S as an edge for r
    Connect s and o with edge r
    Connect s and o to each st
    Connect st to each tp
end for

3.2 Topic-based Graph Traversal

To leverage the hierarchical structure of our graph, we design a two-step LLM-guided traversal strategy : **Topic-based Graph Traversal**.

Step 1: Topic Selection. Given a query, the LLM selects N_T relevant topics from all available topic nodes. Since topics are core keywords that represent the entire corpus, this step can be interpreted as the first step in setting the scope of responses for LLM.

Step 2: Subtopic Selection. For each selected topic, the LLM chooses N_{ST}subtopics from its connected subtopic nodes, based on semantic relevance to the query. In practice, N_T and N_{ST} are bounded to small values, enabling our traversal method to scale with minimal LLM calls.

While the entire list of candidates is provided in each step, the extended context capacity of modern LLMs (Hurst et al., 2024) ensures that this selection process remains efficient. Typically, N_T and N_{ST} are small values, requiring just one LLM call for topic selection and N_T calls for subtopic selection.

This approach offers greater robustness compared to methods that extract entities from query (Guo et al., 2024) or implicitly infer topics and subtopics. By providing the LLM with explicit lists of candidate topics, subtopics, and entities as context, it selects the most relevant ones based on the query, reducing ambiguity and increasing reliability.

Ultimately, this process can be viewed as a hierarchical graph traversal that progressively narrows down the search space within a large corpus to efficiently locate the answer (the visualization of results can be found in Figure 4.

Algorithm 2 Topic-based Graph Traversal & Query-based Retrieval

Input: Query q, graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, parameters N_T, N_{ST}, K_1, K_2
Output: Final context set
Extract top-N_T topics relevant to q:
    $T_{selected} = LLM(T_{list}, q)$
for each $t_i \in T_{chosen}$ **do**
    Extract top-N_{ST} subtopics relevant to t_i and q:
        $S_i = LLM(ST_{list}^{(t_i)}, q)$
        $ST_{selected} \leftarrow ST_{selected} \cup S_i$
end for
Retrieve all entities under $ST_{selected}$
For each entity e, collect its 1-hop neighbors
Compute similarity between q and all sentences
Select top-K_1 sentences as primary context
Select top-K_2 sentences and include their source chunks as extended context

3.3 Query-based Retrieval

From each selected subtopic, we collect the connected entity nodes, which act as anchors for context retrieval. For each entity, we explore its 1-hop neighbors within the graph, collecting all associated edges, since each edge is annotated with its original source sentence. This process yields a set of candidate evidence sentences directly grounded in the source corpus.

These edge-level sentences form the basis for our filtering mechanism, enabling precise and faithful sentence-level evidence retrieval. To reduce redundancy and improve relevance, we apply a **two-stage filtering strategy**:

- **Cosine Similarity Filtering:** We compute the cosine similarity between the query and each candidate sentence. The top-K_1 most relevant sentences are selected as the primary context for generation.
- **Context Expansion:** We further select a subset of K_2 sentences ($K_2 \ll K_1$) and retrieve their full source chunks. This expansion pro-

vides additional contextual cues around high-confidence sentence.

As noted by Han et al. (2025a), answer entities or key supporting sentences are sometimes omitted during the graph construction process. To mitigate this risk, we adopt targeted context expansion around the most relevant sentences.

This pipeline combines hierarchical graph traversal and semantic filtering to enable scalable, accurate, and context-aware retrieval.

The overall process of topic-based graph traversal and query-based retrieval is summarized in Algorithm 2.

3.4 Multi-hop Reasoning Robustness

While triplet-based graphs theoretically enable structured reasoning, practical limitations arise when constructing them using LLMs. In natural language, semantic relations do not always conform to a clean subject–relation–object pattern. A single sentence may express reflexive relations (e.g., `(the committee, reorganized, itself)`), symmetric interactions (e.g., `(Alice, collaborates_with, Bob)` and `(Bob, collaborates_with, Alice)`), or implicit structures with missing arguments (e.g., `(Tesla, founded, −)` inferred from "Tesla was founded in 2003"). LLMs often overlook such subtle or implicit connections, leading to incomplete or oversimplified triplet representations. As a result, node-centric graph reasoning like Guo et al. (2024) can become brittle, especially when crucial edges are omitted due to these structural ambiguities.

This limitation becomes particularly evident in cases like the following:

Query: Who collaborated with Marie Curie on research related to radioactivity?

Corpus Sentence: Marie Curie and Pierre Curie conducted groundbreaking research on radioactivity together.

Here, an LLM-based triplet extractor may only produce `(Marie Curie, conducted, research)`, and fail to encode the co-reference to Pierre Curie. Despite the sentence clearly implying collaboration, the triplet graph lacks a direct edge connecting the two entities. Consequently, graph traversal mechanisms alone would be insufficient to reach the correct answer node.

To mitigate this issue, TH-RAG attaches the original source sentence as an edge attribute for every triplet. This allows the retrieval mechanism to operate at the sentence level rather than relying solely on the triplet graph structure. By preserving the full semantic context of each relation, this design reduces noise during retrieval and minimizes distortion of the original corpus semantics.

	Agri	CS	Legal	Mix	Hotpot	MultiHop
Tokens	1.9M	2M	4.7M	602K	1.2M	991K
Passages	12	10	94	61	9,827	435
# QA	125	125	125	125	1,000	1,000

Table 1: **Document statistics** for our experimental datasets. Agri, Hotpot, and MultiHop refer to Agriculture, HotpotQA, and MultiHopRAG, respectively.

Taken together, TH-RAG's design enables robust multi-hop reasoning by integrating three complementary strategies: (1) a semantically grounded hierarchical graph that improves connectivity across fragmented information, (2) LLM-guided hierarchical graph traversal that efficiently focuses retrieval on relevant subregions of the graph, and (3) sentence-level evidence filtering and targeted context expansion, enabling the retrieval of relevant information even when it is not structurally captured in the triplet graph. This holistic approach allows TH-RAG to retrieve and reason over information that is semantically dispersed but contextually relevant, leading to more accurate and complete answers on complex multi-hop queries.

4 Experiments

To evaluate the effectiveness and robustness of TH-RAG, we design experiments to answer the following research questions:

- **RQ1:** Is our method effective for QA datasets of multi-hop reasoning and abroad domain?
- **RQ2:** How well does our method perform, especially in terms of mitigating graph fragmentation?
- **RQ3:** How efficient is our method in terms of resource usage and scalability?
- **RQ4:** What are the core components of our method and the optimal hyperparameters?

4.1 Datasets

In our experiments, we used two types of data sets. One is an **open-domain QA** dataset, such as **UltraDomain** (Qian et al., 2025), which does not have specific evidence and requires answering open-ended questions based on abroad knowledge. Following prior studies, we used three domain-specific datasets (Agriculture, CS, and Legal) and one mixed-domain corpus.

	Agriculture								CS							
	Comprehensive		Diversity		Empowerment		Overall		Comprehensive		Diversity		Empowerment		Overall	
Win Rate	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline
vs Naive	**84.2%**	15.8%	**88.3%**	11.7%	**87.5%**	12.5%	**86.7%**	13.3%	**86.9%**	13.1%	**91.0%**	9.0%	**86.9%**	13.1%	**86.9%**	13.1%
vs GraphRAG G	**87.0%**	13.0%	**91.1%**	8.9%	**88.6%**	11.4%	**88.6%**	11.4%	**78.7%**	21.3%	**74.6%**	25.4%	**78.7%**	21.3%	**78.7%**	21.3%
vs GraphRAG L	**88.7%**	11.3%	**90.3%**	9.7%	**89.5%**	10.5%	**89.5%**	10.5%	**84.6%**	15.4%	**86.2%**	13.8%	**87.0%**	13.0%	**86.2%**	13.8%
vs LightRAG	**87.1%**	12.9%	**91.9%**	8.1%	**89.5%**	10.5%	**88.7%**	11.3%	**80.7%**	19.3%	**80.7%**	19.3%	**81.5%**	18.5%	**81.5%**	18.5%
vs PathRAG	**80.7%**	19.3%	**92.4%**	7.6%	**85.7%**	14.3%	**85.7%**	14.3%	**78.4%**	21.6%	**86.4%**	13.6%	**81.6%**	18.4%	**81.6%**	18.4%
vs HypergraphRAG	**52.3%**	47.7%	**60.4%**	39.6%	**51.4%**	48.6%	**52.3%**	47.7%	49.1%	**50.9%**	46.4%	**53.6%**	**50.9%**	49.1%	49.1%	**50.9%**
	Legal								Mix							
	Comprehensive		Diversity		Empowerment		Overall		Comprehensive		Diversity		Empowerment		Overall	
Win Rate	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline	TH-RAG	Baseline
vs Naive	**86.2%**	13.8%	**89.4%**	10.6%	**90.2%**	9.8%	**90.2%**	9.8%	**91.9%**	8.1%	**95.5%**	4.5%	**93.7%**	6.3%	**93.7%**	6.3%
vs GraphRAG G	**79.8%**	20.2%	**69.4%**	30.6%	**81.5%**	18.5%	**81.5%**	18.5%	**84.5%**	15.5%	**84.1%**	15.9%	**90.1%**	9.9%	**90.1%**	9.9%
vs GraphRAG L	**89.4%**	10.6%	**87.0%**	13.0%	**90.2%**	9.8%	**90.2%**	9.8%	**96.5%**	3.5%	**98.3%**	1.7%	**96.5%**	3.5%	**96.5%**	3.5%
vs LightRAG	**85.5%**	14.5%	**85.5%**	14.5%	**89.5%**	10.5%	**89.5%**	10.5%	**91.3%**	8.7%	**95.7%**	4.3%	**92.2%**	7.8%	**92.2%**	7.8%
vs PathRAG	**86.4%**	13.6%	**84.0%**	16.0%	**86.4%**	13.6%	**86.4%**	13.6%	**90.5%**	9.5%	**97.4%**	2.6%	**92.2%**	7.8%	**92.2%**	7.8%
vs HypergraphRAG	**50.9%**	49.1%	43.8%	**56.2%**	**50.9%**	49.1%	**50.9%**	49.1%	**57.9%**	42.1%	**63.2%**	36.8%	**57.0%**	43.0%	**57.9%**	42.1%

Table 2: **Main Results on UltraDomain**, specially for Agriculture, CS, Legal and Mix domains. Metrics using 1vs1 win rate, as llm-as-a-judge. We exclude GraphRAG-G from our evaluation, as its use of global community detection and summarization spans numerous chunks, making the comparison less meaningful in our setting.

	Answer								Retrieval							
	MultiHopRAG				HotpotQA				MultiHopRAG				HotpotQA			
	F1	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	Recall	F1	Rec@5	NDCG@5	Recall	F1	Rec@5	NDCG@5
Naive	0.501	0.475	0.599	0.604	0.584	0.612	0.590	0.509	0.330	0.210	0.337	0.375	0.394	0.143	0.342	0.352
GraphRAG-G	0.526	0.501	0.618	0.653	0.393	0.410	0.402	0.343	-	-	-	-	-	-	-	-
GraphRAG-L	0.469	0.451	0.536	0.535	0.668	0.696	0.678	0.595	0.267	0.239	0.267	0.412	0.830	**0.479**	0.833	0.794
LightRAG	0.464	0.448	0.527	0.526	0.496	0.519	0.507	0.439	0.072	0.039	0.061	0.082	0.323	0.129	0.282	0.217
PathRAG	0.468	0.453	0.523	0.525	0.551	0.578	0.562	0.488	0.203	0.113	0.182	0.265	0.818	0.326	0.808	**0.805**
HyperGraphRAG	0.526	0.503	0.619	0.621	**0.674**	**0.703**	0.683	0.599	**0.426**	**0.283**	**0.402**	0.460	**0.848**	0.382	**0.848**	0.763
TH-RAG	**0.712**	**0.711**	**0.720**	**0.722**	0.671	0.692	**0.685**	**0.612**	0.392	0.249	0.393	**0.522**	0.781	0.304	0.781	0.743

Table 3: **Main results on HotpotQA and MultiHopRAG**. **Bold** indicates the best result, and underline indicates the second-best.

The other is a **answer-specific QA** dataset, such as **HotpotQA** (Yang et al., 2018) and **MultiHopRAG** (Tang and Yang, 2024), which has concrete multi-hop evidence that must be retrieved to generate answers. More detailed explanations about data sets are provided in Appendix C.1

We randomly selected 1,000 QA pairs along with their corresponding passages from both MultiHopRAG and HotpotQA to construct the corpus for evaluation. A detailed description of these datasets is provided in the Table 1.

4.2 Metrics

We used two evaluation approaches: For the **Ultradomain** dataset, we applied the **LLM-as-a-judge** method (Zheng et al., 2023), comparing answers pairwise as in Guo et al. (2024). For **MultiHopRAG** and **HotpotQA**, we used traditional metrics—**F1**, **Recall**, **Precision**, and **Accuracy**—along with retrieval metrics like **recall**, **F1**, **recall@5**, and **NDCG@5**.

A detailed description of these metrics is provided in the Appendix C.4.

4.3 Baselines

We compared TH-RAG against several representative baseline methods categorized into three groups: (1) a basic retrieval form, **NaiveRAG** (Gao et al., 2024); (2) triplet-based graph baselines, including **GraphRAG** (Edge et al., 2024) and **LightRAG** (Guo et al., 2024); and (3) current state-of-the-art methods, **PathRAG** (Chen et al., 2025) and **HyperGraphRAG** (Luo et al., 2025). For GraphRAG, we implemented both the local and global retrieval methods. We refer to the local version as **GraphRAG-L** and the global version as **GraphRAG-G** throughout our experiments. A detailed explanation of baselines can be found in the Appendix C.3.

4.4 Implementation Details

We used the following hyperparameters and implementation settings: K_1 and K_2 were fixed at 30 and 5, respectively. N_T and N_{ST} were determined through prompt-based selection, with values ranging from 5–10 and 10–25, respectively. The exact numbers varied depending on the LLM's output. Additonal implementation details are in Appendix B, and used prompts are in Appendix A

5 Results

5.1 Main Results (RQ1)

On the **Ultradomain** dataset, TH-RAG outperforms all baselines except HyperGraphRAG across all four domains (in Table 2). Notably, when compared to PathRAG—a widely regarded - state-of-the-art method — TH-RAG achieves an average win rate of 86.48%. While HyperGraphRAG shows slightly better results in the CS domain, TH-RAG outperforms it in all other domains.

Particularly in the mixed-domain setting, TH-RAG demonstrates a more substantial performance

	LightRAG	TH-RAG
# Nodes	20,914	**50,162**
# Topic Nodes	-	**531**
# Subtopic Nodes	-	**15,142**
# Entity Nodes	20,914	**30,248**
# Edges	24,707	**94,507**
# Topic-subtopic edges	-	**20,675**
# Subtopic-entity edges	-	**34,017**
# Entity-Entity edges	**24,707**	21,906
# Subgraphs	8,805	**3**
% of Biggest Subgraph	56.11%	**99.98%**

Table 4: Constructed graph statistics comparison with LightRAG way on Legal dataset. PathRAG use same graph structure with LightRAG, so #subgraphs and % of Biggest Subgraph means PathRAG has a great weakness. % is calculated on Nodes.

gap, suggesting that our method is more robust in handling diverse, open-domain questions. On the **specific-type** datasets, TH-RAG consistently achieves higher scores than all baselines across most evaluation metrics (in Table 3). In MultiHopRAG, it surpasses GraphRAG-G and HyperGraphRAG by 6.9% and 10.1%, respectively, showing clear superiority in multi-hop reasoning.

However, when examining retrieval performance, TH-RAG does not achieve state-of-the-art-level results, especially in HotpotQA, where it lags behind across several retrieval metrics. Nevertheless, its answer generation performance remains superior. This discrepancy highlights known issues (Tang and Yang, 2024) with the HotpotQA dataset—some questions can be answered using single-document evidence, even when multiple supporting facts are provided—making retrieval metrics less indicative of final answer quality.

In MultiHopRAG, TH-RAG demonstrates strong performance not only in answer quality but also in retrieval. An interesting observation is that **NaiveRAG** performs relatively well on specific-type datasets, indicating that entity missing during graph construction can critically impact performance in fact-based QA (Edge et al., 2024; Han et al., 2025a).

5.2 Graph Fragmentation and Robustness Analysis (RQ2)

To assess the impact of TH-RAG's hierarchical structure on mitigating graph fragmentation, we compare the structural properties of graphs constructed by TH-RAG and a representative triplet-based method, LightRAG.

Compared to LightRAG, TH-RAG substantially reduces the number of disconnected subgraphs and achieves a much higher largest-connected-component ratio. These improvements highlight the effectiveness of our Topic–Subtopic–Entity hierarchy in enhancing global graph connectivity. Summary statistics are presented in Table 4. Results on other datasets are provided in Appendix C.2, and visualization results are included in Appendix D.2.

	TH-RAG	HyperGraphRAG
Comparison	**0.625**	0.538
Temporal	**0.509**	0.197
Inference	**0.954**	0.938
Null	**0.786**	0.664

Table 5: Comparison on HyperGraphRAG by question type of MultiHopRAG.

Figure 1 further illustrates the benefit of reduced fragmentation through a qualitative comparison with PathRAG. In conventional triplet-based methods, answer-relevant entities often appear in separate subgraphs, making reasoning paths incomplete or unreachable—especially for methods like PathRAG that depend heavily on connectivity. In contrast, TH-RAG does not rely on direct entity–entity connections. Instead, it accesses relevant information by navigating through topic-based hierarchical graph traversal and retrieving sentence-level evidence, enabling robust reasoning even in partially disconnected entities.

Furthermore, the number of topic nodes remains small, and the average Topic-to-Subtopic ratio is approximately *1:30*. This ensures that the Topic and Subtopic selection process remains token-efficient and computationally lightweight during inference.

We also provide a comparison of question-type-level performance on MultiHopRAG in Table 5. While TH-RAG performs comparably to HyperGraphRAG on *Inference*, *Comparison*, and *Null* types, it significantly outperforms on *Temporal* questions. This suggests that TH-RAG's sentence-based retrieval and topic-aware traversal are better at capturing temporally grounded relations compared to HyperGraphRAG, leading to high robustness of TH-RAG.

5.3 Efficiency Analysis (RQ3)

We next evaluate the efficiency of TH-RAG in terms of token usage and LLM call overhead, focusing on two key stages: indexing and retrieval. We compare TH-RAG against **GraphRAG-L**, **HyperGraphRAG**, and **LightRAG**—three strong baselines known for either high performance or retrieval efficiency (Table 6).

In the indexing phase, TH-RAG demonstrates

	Light	Hyper	Local	TH-RAG
Indexing Call	5,978	2,772	4,354	**902**
Indexing Token	8M	20.3M	15M	**2.3M**
Querying Time	2.66s	9.78s	**0.77s**	3.54s
Context Token	25K	20K	13.6K	**7.4K**

Table 6: Efficiency comparison of representative methods on MultiHopRAG. Token counts include both prompt and context. Light, Hyper, and Local refer to LightRAG, HyperGraphRAG, and GraphRAG-L, respectively.

	Accuracy	F1	Recall	Precision
Original	**0.722**	**0.712**	**0.72**	**0.71**
w/o chunks	0.580	0.576	0.577	0.577
w/o Triplets	0.692	0.68	0.691	0.678
w/o Traversal	0.624	0.62	0.622	0.62

Table 7: Ablatioin study on key components of TH-RAG. W/o Traversal means we don't apply graph-travesal, using only filtering by all sentences.

remarkable efficiency. It requires only **32.5%** of the LLM calls used by HyperGraphRAG (902 vs. 2,772) and just **29%** of the tokens consumed by LightRAG (2.3M vs. 8M). This reduction is primarily attributed to our graph construction method and prompt-based topic/subtopic annotation, which eliminate the need for costly iterative clustering or summarization at the entity level.

During retrieval, TH-RAG incurs slightly higher latency compared to GraphRAG-L due to the $(N_T + 1)$ LLM calls needed for topic and subtopic selection. Nevertheless, its total token usage remains low—only **54%** of that required by HyperGraphRAG (7.4K vs. 13.6K). Since the number of topic nodes rarely exceeds 1,000, the retrieval time complexity remains $O(N_T + 1)$, making TH-RAG scalable even for large corpora. Overall, TH-RAG achieves a favorable balance between computational efficiency and retrieval quality.

5.4 Ablation and Hyperparameter Analysis (RQ4)

We conduct ablation studies to evaluate the contribution of each component in TH-RAG. As shown in Table 7, removing chunks in context leads to significant performance degradation. Disabling triplet usage or bypassing the topic–subtopic traversal (e.g., applying filtering over all sentences) also results in noticeable accuracy drops.

These results confirm that TH-RAG's strength lies in its ability to semantically scope the graph through topic and subtopic selection, enabling it to isolate focused subgraphs that are rich in relevant information. This targeted traversal leads to the extraction of high-quality chunks grounded in the original corpus, enabling more robust and reliable multi-hop reasoning.

	K_1=5	10	30	50
K_2=1	0.536	0.565	0.622	0.630
3	0.662	0.679	0.697	0.702
5	0.697	0.685	0.72	0.719
10	x	0.714	**0.743**	0.726

Table 8: Ablation on K1 & K2 on MultiHopRAG with accuracy.

We also evaluate the impact of varying the hyperparameters K_1 and K_2, which control the number of retrieved sentences and the number of expanded chunks, respectively (Table 8). While our main experiments adopt $K_1 = 30$ and $K_2 = 5$ for cost efficiency, increasing K_2 to 10 leads to slightly better performance, indicating a trade-off between answer quality and token cost (Joren et al., 2024).

Interestingly, increasing both K_1 and K_2 beyond a certain point (e.g., $K_1 = 50$ and $K_2 = 10$) degrades performance—likely due to *context rot* or *lost-in-the-middle* effects, as noted in recent studies (Zhang et al., 2025a; Hsieh et al., 2024). This underscores the importance of careful context engineering (Mei et al., 2025) and hyperparameter tuning in retrieval-augmented systems.

6 Conclusion

We proposed TH-RAG, a novel graph-based RAG framework designed to address two central challenges of prior methods: graph fragmentation and difficulty in multi-hop reasoning. TH-RAG constructs a three-level hierarchical knowledge graph—composed of topics, subtopics, and entities—that semantically organizes information extracted from unstructured text.

Leveraging this structure, TH-RAG performs topic-guided graph traversal to retrieve focused subgraphs relevant to the query. Each edge in the graph stores its original sentence, allowing the retrieval process to operate directly on sentence-level evidence grounded in the source corpus. This design improves both semantic fidelity and reasoning robustness, especially in cases where traditional triplet-based graphs may omit key relationships.

Through this integrated approach, TH-RAG achieves strong performance across both general and multi-hop QA tasks, while maintaining scalability and reducing graph fragmentation. Our results suggest TH-RAG provides a reliable and extensible foundation for graph-based retrieval in LLM-augmented systems.

Limitations

TH-RAG introduces a hierarchical KG built from LLM-extracted topics, subtopics, and triplets. However, the current approach has several limitations that suggest avenues for future improvement. First, the topic and subtopic normalization remains imperfect. Due to inconsistencies in LLM outputs, semantically similar concepts are often assigned to different topic or subtopic labels, unnecessarily inflating the graph structure (e.g. sports <-> sport, film director <-> director). To address this, future work could explore embedding-based clustering techniques to group semantically equivalent nodes (Chang et al., 2025; Liu et al., 2025b). Additionally, incorporating conversational history or memory-based context into the topic extraction step may help the LLM produce more consistent and coherent topic assignments. Second, this work deliberately omits widely-used RAG techniques such as query expansion, and context reranking, in order to isolate the effectiveness of our hierarchical graph structure in its most basic and efficient form. However, given the demonstrated effectiveness of these techniques in recent literature (Gao et al., 2024; Sharma, 2025), integrating them in a way that aligns with our topic-based hierarchy could further enhance performance. Lastly, future directions include enabling the LLM to directly interact with the graph structure for more explicit reasoning over graphs (Han et al., 2025b; Ma et al., 2024), potentially unlocking stronger multi-hop capabilities and interpretability.

References

Amar Abane, Anis Bekri, and Abdella Battou. 2024. Fastrag: Retrieval augmented generation for semi-structured data. *arXiv preprint arXiv:2411.13773*.

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

Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O Nunes, Rafael Padilha, and 1 others. 2024. Rag vs fine-tuning: pipelines, tradeoffs, and a case study on agriculture. *arXiv preprint arXiv:2401.08406*.

Martin Böckling, Heiko Paulheim, and Andreea Iana. 2025. Walk&retrieve: Simple yet effective zero-shot retrieval-augmented generation via knowledge graph walks. *arXiv preprint arXiv:2505.16849*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

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. In *First Conference on Language Modeling*.

Chia-Hsuan Chang, Jui-Tse Tsai, Yi-Hang Tsai, and San-Yih Hwang. 2025. Lita: An efficient llm-assisted iterative topic augmentation framework. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 449–460. Springer.

Bohan Chen and Andrea L Bertozzi. 2023. Autokg: Efficient automated knowledge graph generation for language models. In *2023 IEEE International Conference on Big Data (BigData)*, pages 3117–3126. IEEE.

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

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

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

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Haoyu Han, Harry Shomer, Yu Wang, Yongjia Lei, Kai Guo, Zhigang Hua, Bo Long, Hui Liu, and Jiliang Tang. 2025a. Rag vs. graphrag: A systematic evaluation and key insights. *arXiv preprint arXiv:2502.11371*.

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

Haoyu Han, Yaochen Xie, Hui Liu, Xianfeng Tang, Sreyashi Nag, William Headden, Yang Li, Chen Luo, Shuiwang Ji, Qi He, and 1 others. 2025b. Reasoning with graphs: Structuring implicit knowledge to enhance llms reasoning. *arXiv preprint arXiv:2501.07845*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Yucheng Hu and Yuxing Lu. 2024. Rag and rau: A survey on retrieval-augmented language model in natural language processing. *Available at SSRN 5015182*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *EACL 2021-16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880. Association for Computational Linguistics.

Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. 2014. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. *PLOS ONE*, 9:1–12.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. 2024. Sufficient context: A new lens on retrieval augmented generation systems. In *The Thirteenth International Conference on Learning Representations*.

Vikas Kamra, Lakshya Gupta, Dhruv Arora, and Ashwin Kumar Yadav. 2024. Enhancing document retrieval using ai and graph-based rag techniques. In *2024 5th International Conference on Communication, Computing Industry 6.0 (C2I6)*, pages 1–7.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

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

Xun Liang, Simin Niu, Sensen Zhang, Shichao Song, Hanyu Wang, Jiawei Yang, Feiyu Xiong, Bo Tang, Chenyang Xi, and 1 others. 2024. Empowering large language models to set up a knowledge retrieval indexer via self-learning. *arXiv preprint arXiv:2405.16933*.

Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025a. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. *arXiv preprint arXiv:2502.12442*.

Jianghan Liu, Ziyu Shang, Wenjun Ke, Peng Wang, Zhizhao Luo, Jiajun Liu, Guozheng Li, and Yining Li. 2025b. LLM-guided semantic-aware clustering for topic modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18420–18435, Vienna, Austria. Association for Computational Linguistics.

Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and 1 others. 2025. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *arXiv preprint arXiv:2503.21322*.

L Luo, YF Li, G Haffari, and S Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *ICLR 2024: The Twelfth International Conference on Learning Representations*. ICLR.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2024. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. *arXiv preprint arXiv:2407.10805*.

Yu A Malkov. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

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

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, and 1 others. 2025. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*.

Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and Prathosh Ap. 2024. Holmes: Hyper-relational knowledge graphs for multi-hop question answering using llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13263–13282.

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

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

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Chaitanya Sharma. 2025. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers. *arXiv preprint arXiv:2506.00054*.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024. Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pages 12–22.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423.

Shu Wang, Yixiang Fang, Yingli Zhou, Xilin Liu, and Yuchi Ma. 2025. Archrag: Attributed community-based hierarchical retrieval-augmented generation. *arXiv preprint arXiv:2502.09891*.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and 1 others. 2024. Retrieval-augmented generation for natural language processing: A survey. *CoRR*.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wenli Yang, Lilian Some, Michael Bain, and Byeong Kang. 2025b. A comprehensive survey on integrating large language models with knowledge-based methods. *Knowledge-Based Systems*, 318:113503.

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*.

Junhao Zhang, Richong Zhang, Fanshuang Kong, Ziyang Miao, Yanhan Ye, and Yaowei Zheng. 2025a. Lost-in-the-middle in long-text generation: Synthetic dataset, evaluation framework, and mitigation. *arXiv preprint arXiv:2503.06868*.

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

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. 2024. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He, and Xiang Li. 2025. Eˆ 2graphrag: Streamlining graph-based rag for high efficiency and effectiveness. *arXiv preprint arXiv:2505.24226*.

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

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025a. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924, Albuquerque, New Mexico. Association for Computational Linguistics.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. 2025b. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8912–8924.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2024. Llms for knowledge graph construction and reasoning: recent capabilities and future opportunities. *World Wide Web*, 27(5).

Zulun Zhu, Tiancheng Huang, Kai Wang, Junda Ye, Xinghe Chen, and Siqiang Luo. 2025c. Graph-based approaches and functionalities in retrieval-augmented generation: A comprehensive survey. *arXiv preprint arXiv:2504.10499*.

A Prompts

A.1 Answer Generation Prompt

Instruction description
—Role— You are a helpful assistant responding to user query **—Goal—** Generate a concise response based on the following information and follow Response Rules. Do not include information not provided by following Information **—Target response length and format—** Multiple Paragraphs **—Information—** {{context}} **—Response Rules—** - Use markdown formatting with appropriate section headings - Please respond in the same language as the user's question. - If you don't know the answer, just say so. - Do not make anything up. Do not include information not provided by the Information. **—Query—** {{question}}

Table 9: Answer Generation Prompt for Ultradomain. This prompts is used when we need long, comprehensive response.

A.2 Short Answer Generation Prompt

Instruction description

—Role—
You are a multi-hop retrieval-augmented assistant.

—Goal—
Read the Information passages and generate the correct answer to the Query. Use only the given Information; if it is insufficient, reply with "Insufficient information.". If you need to answer like yes or no, use "Yes" or "No" only.
—Target response length and format—
- One-word or minimal-phrase answer (max 5 words).

—Response Rules—
- Answer must be short and concise.
- Answer language must match the Query language.
- Do NOT add or invent facts beyond the Information.

—Information—
{{context}}

—Query—
{{question}}

Table 10: Short Answer Generation Prompt used for HotpotQA and MultiHopRAG. This prompt is used when we need short, concise response.

A.3 Triplet Extraction with Topic Prompt

Instruction description

—Role—
You are a highly skilled information extraction system designed to process factual information accurately and clearly.

—Goal—
Extract factual (subject, relation, object) triples from the document and classify the subject and object into a subtopic and a main topic.

—Instructions—
1. Read the entire document below and extract all factual (subject, relation, object) triples. Each triple must be grounded in a specific sentence from the document.
2. Paraphrasing is acceptable only if the relation is clearly implied by the sentence.
3. Resolve all pronouns such as "it", "he", "she", "they", etc. using the surrounding context. Replace all pronouns in the triple with their correct referents.
- Do not include any unresolved or ambiguous pronouns in the triples.
- Be specific and use full entity names instead of pronouns wherever applicable.
4. For each subject and object:
- Assign a Subtopic (a specific category such as "Electronic Musician", "Sound Label", etc.)
- Assign a Main topic (a broader category such as "Music", "Art", etc.)
- Ensure the subtopic and main topic reflect both the entity and the overall context of the document.
5. Return only valid JSON in the specified format. Do not include markdown, comments, or any other text.
6. Ensure that the JSON is well-formed and valid.

—Examples—
{{example}}

—Input Document—
{{document}}

Table 11: Triplet Extraction with Topic Prompt

A.4 Topic Selection Prompt

Instruction description

—Goal—

Given the user's question, choose all topics from the supplied list that are directly relevant to answering the question. Select between {min_topics} and {max_topics} topics. Choose exhaustively but do NOT invent new topics. Return the chosen topics exactly as they appear in the list. Always return at least {min_topics} topics.

—Instructions—

1. The list of allowed topics will be provided in the placeholder {TOPIC_LIST}.
2. Read the user question provided in the placeholder {question}.
3. Identify every topic from {TOPIC_LIST} that is pertinent to the question.
4. Output only valid JSON. Do not include markdown, comments, or extra text.
5. Output JSON format: { "topics": ["TopicLabel1", "TopicLabel2", ...]}
6. You MUST ONLY choose from the list provided below. Do not invent or rephrase any subtopics.
7. If you cannot find any relevant topics, just find the most relevant {min_topics} topics.

—Question—

{{question}}

—Allowed Topics—

{{TOPIC_LIST}}

Table 12: Topic Selelction Prompt

A.5 Subtopic Selection Prompt

Instruction description

—Goal—

Given the user's question, choose all topics from the supplied list that are directly relevant to answering the question. For the given topic {TOPIC_LABEL}, choose every subtopic from the list below that is helpful for answering the user's question. Select {min_subtopics} to {max_subtopics} subtopics. Do NOT invent new subtopics. Always return at least {min_subtopics} subtopics, unless case of list is shorter than {min_subtopics}.

—Instructions—

1. Consider only the subtopics provided in {SUBTOPIC_LIST}.
2. Read the user's question provided in {question}.
3. Output your selection as valid JSON without markdown, comments, or extra text.
4. Preserve the original order of {SUBTOPIC_LIST} when listing the chosen subtopics.
5. Output JSON Format: {"subtopics": ["SubLbl1", "SubLbl2", ...]}
6. You MUST ONLY choose from the list provided below. Do not invent or rephrase any subtopics.
7. If you cannot find any relevant topics, just find the most relevant {min_subtopics} topics.

—Question—

{{question}}

—Allowed Subtopics for {{TOPIC_LABEL}}—

{{SUBTOPIC_LIST}}

Table 13: Subtopic Selection Prompt

Category	Agriculture	CS	Legal	Mix	HotpotQA	MultiHopRAG
Nodes	44,588	45,921	50,162	19,806	50,256	26,250
Topic Nodes	1,568	531	424	401	374	446
Subtopic Nodes	12,280	15,142	14,319	5,993	9,188	7,921
Entity Nodes	30,740	30,248	35,419	13,412	40,694	17,883
Edges	76,946	76,598	94,507	31,580	87,757	42,857
Topic-Subtopic	18,424	20,675	19,212	7,436	12,672	9,825
Subtopic-Entity	35,219	34,017	41,268	14,312	43,843	19,680
Entity-Entity	23,303	21,906	34,027	9,832	31,242	13,352

Table 14: Detailed graph statistics of datasets.

B Implementation Details

Our implementation details on experiments are as follows:

- NaiveRAG and TH-RAG used Faiss as the vector DB for retrieval.
- For similarity calculation with the query, we did not use Faiss's built-in L2-distance or inner product but implemented cosine similarity.
- Answer generation prompts were unified across all methods, and the rest of the settings were based on the default values of the respective baselines.
- We fixed the chunk size at 1200 and overlap at 100 for all methods. The temperature during answer generation was set to 0, and gleaning was also set to 0.
- Including graph construction and answer generation, we used *gpt-4o-mini* when needed, and for sentences and chunks embedding, we used *text-embedding-small-3* for all methods.

C Datasets and Baselines Details

C.1 Datasets

- **Ultradomain**: A collection of 20 domain-specific datasets, consisting of long-form passages that make it ideal for abroad-type evaluation. We generated a total of 125 questions, following the same methodology used in Edge et al. (2024); Guo et al. (2024).
- **HotpotQA**: A Wikipedia-based QA dataset that requires multi-hop reasoning across two to four steps. Each question comes with context that contains relevant information. HotpotQA has evaluation settings: Distractor and FullWiki. We conducted evaluations only on the setting, where 8 out of 10 paragraphs are irrelevant, making it suitable for evaluating the ability to retrieve accurate information.
- **MultihHopRAG**: A QA dataset based on English news articles, requiring multi-hop reasoning across 2–4 documents. The question types include **Inference**, **Comparison**, **Temporal**, and **Null**.

C.2 Statistics of our mehtods

Table 14 presents graph statistics of TH-RAG across the entire dataset.

C.3 Baselines

- **NaiveRAG**: The most basic version, where chunks with high similarity are retrieved and used. We used top-7 similar chunks, for fair comparison with other methods on context length.
- **GraphRAG**: One of the first successful applications of KG construction for RAG. It includes **Global** and **Local** configurations. While former one is closer to original paper's method and use global community summarization, later one uses more detail and samller commulity to generate answer. We evaluated both versions, denoted as **GraphRAG-G** and **GraphRAG-L**.

- **LightRAG**: An efficient version of GraphRAG that improves retrieval efficiency. Since it is known for simlpe and effecient, we compare our effeciency with this baseline.
- **PathRAG**: A method specialized for multi-hop reasoning, based on LightRAG. It retrieves only the necessary information by connecting entities and pruning path to answer.
- **HyperGraphRAG**: state-of-the-art method that extends traditional triplet structures to use hyperedges for connecting multiple entities in a graph.

There exist other strong baselines, such as Gutiérrez et al. (2025); Zhu et al. (2025b); Zhao et al. (2025), as well as chunk-to-graph approaches like Sarthi et al. (2024); Liu et al. (2025a). However, we excluded the former because they do not operate on fixed-length chunks, and the latter because they are not based on triplet-style graph construction.

C.4 Metrics

We used two evaluation approaches depending on the dataset type.

For the **Ultradomain** dataset, we followed previous studies and used the LLM-as-a-judge method (Zheng et al., 2023). Similar to Guo et al. (2024), answers were compared 1vs1 in three dimensions, and the overall win rates were computed. This approach was adopted due to the longer answer nature of this dataset.

For the MultiHopRAG and HotpotQA datasets, we adopted traditional evaluation metrics—F1, Recall, Precision, and Accuracy—as the answers are typically short and fact-based. While LLM-as-a-judge has demonstrated strong alignment with human evaluation, it may introduce bias. Therefore, we employed quantitative metrics to provide a more objective assessment of our method on these datasets. For both HotpotQA and MultiHopRAG, we followed the official evaluation protocol of HotpotQA. Accuracy is determined by whether the predicted answer contains the gold answer.

For retrieval evaluation, we additionally used Recall, F1, Recall@5, and NDCG@5 to ensure a fair and comprehensive comparison. All methods either generate answers from specific chunks or indicate the chunk IDs from which their context is derived; we consider these as the predicted chunks. For the gold chunks, we use those that contain the supporting evidence for each query, treating them as ground truth for retrieval evaluation.

D Examples

D.1 Triplet Extraction Example

Example Input and Output Format

—Input—

Moscow State University **Lomonosov Moscow State University** is a coeducational and public research university. ... MSU **was renamed after** Lomonosov in 1940 and was then known as "Lomonosov University". It also houses the tallest educational building in the world. ...

—Output—

"triple": ["Lomonosov Moscow State University", "was renamed after", "Mikhail Lomonosov"],
"sentence": "MSU was renamed after Lomonosov in 1940 and was then known as 'Lomonosov University'.",
"subject": { "subtopic": "University", "main_topic": "Education" },
"object": { "subtopic": "Person", "main_topic": "Biography" }

Table 15: Example Input and Output Format for Triplet Extraction with Topic. We divide entity, subtopic and topic for graph structure corruption.

D.2 Example of Constructed KG and Retrieval Result

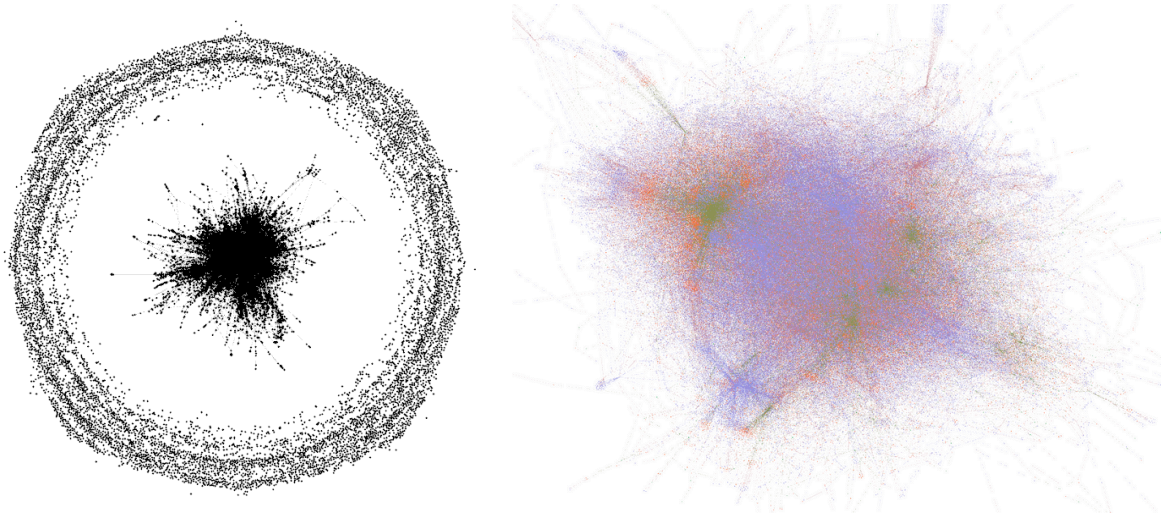

Figure 3: Comparison of Knowledge Graph (KG) structures between LightRAG (left) and TH-RAG (right). In the TH-RAG visualization, green nodes represent topics, red nodes represent subtopics, and purple nodes represent entities. The graphs are visualized using the Force Atlas 2 layout algorithm (Jacomy et al., 2014).

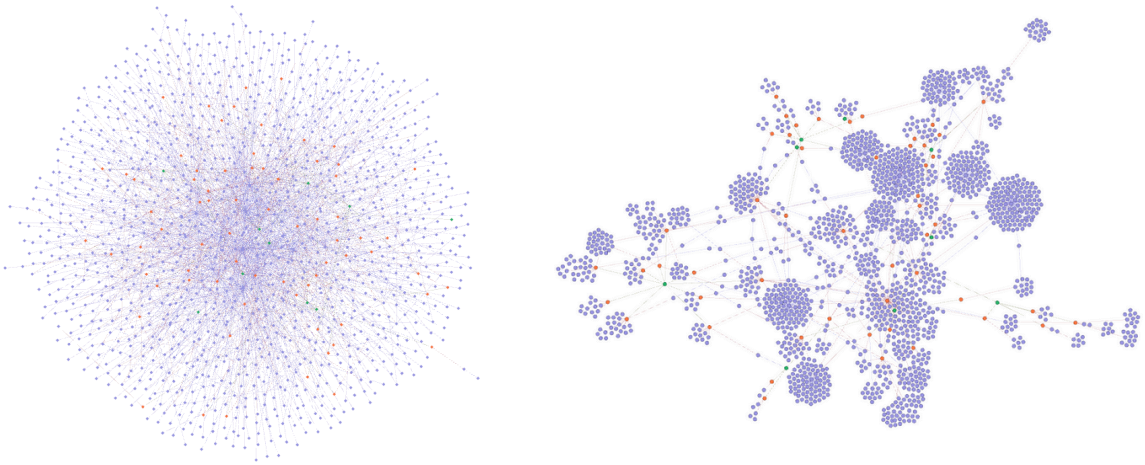

Figure 4: Retrieved subgraphs for different questions using TH-RAG