E^{2}GraphRAG: Dual-Layer Entity–Event Indexing and Retrieval for Graph-Augmented Generation

Abstract

Hierarchical graph-based retrieval-augmented generation(RAG) methods represent knowledge at different granularities across each layer, forming a structure with more natural semantic connections. However, this approach still faces two challenges: the multi-level structure is complex and grows increasingly abstract in content, while direct retrieval matching often fails to adequately decompose queries based on community structures. In this work, we introduce a hierarchical entity–event knowledge indexing pipeline that injects event nodes extracted from entity-linked evidence, producing a compact yet semantically rich graph for RAG.We propose an event-based query decomposition strategy that couples dense retrieval–based query grounding with hierarchical graph retrieval, including local entity alignment and global event retrieval for coherent evidence organization.Extensive experiments on UltraDomain show that E^{2}GraphRAG consistently achieves win rates ranging from 50.0% to 99.2% against NaiveRAG, GraphRAG, LightRAG, and HiRAG. Ablations validate the contributions of E^{2}GraphRAG. Code is available at https://github.com/***.

1 Introduction

When applied to domain-specific or knowledge-intensive tasks, Retrieval-Augmented Generation (RAG) can significantly mitigate issues in large language models (LLMs), such as hallucinations and information latency (Lewis et al., 2020; Sudhi et al., 2024). Naive RAG divides the raw corpus into retrievable text chunks and, during querying, quickly indexes the most relevant chunks to generate answers (Zhang et al., 2025d; Lyu et al., 2025). Retrieving isolated fragments ignores the connections between them, failing to capture relationships of the same entity across different text chunks. While Graph-based RAG systems can help establish cross-chunk connections (Edge et al., 2024; Guo et al., 2024; Zhang et al., 2025a), they still suffer from certain limitations.

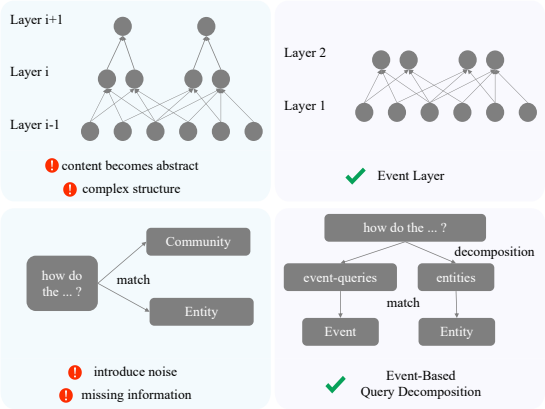

Figure 1: Hierarchical knowledge graph-based RAG.

GraphRAG (Edge et al., 2024) performs information fusion and deduplication by constructing a knowledge graph, aggregating information that refers to the same entity across different text chunks. It then employs the Leiden algorithm (Traag et al., 2019) to detect communities and generates a summary for each community. Building upon this, LightRAG (Guo et al., 2024) develops a more efficient retrieval-augmented generation system, with its seamless incremental update capability significantly enhanced. This approach does not sufficiently capture the deeper semantic relationships between entities. HiRAG (Huang et al., 2025) further leverages unsupervised clustering to construct multi-layer summary entities, using hierarchical knowledge (Sarrafzadeh and Lank, 2017) to strengthen semantic understanding and structural capturing, thereby bridging the "knowledge gap" between local and global knowledge, and shortening the distance between semantically similar entities. However, in complex and scenario-specific reasoning tasks, as illustrated in Figure 1, we demonstrate the challenges faced by former hierarchical Graph-based RAG methods:

Deep Hierarchical Complexity and content is abstract. Current hierarchical graph RAG meth-

ods construct a multi-layer structure through recursive summarization. This structure often leads to a dilution of semantic information density within community summaries. In many real-world applications, such as legal analysis, knowledge is more naturally organized into a clearer two-tier structure: concrete events, which reflect both the overall situation and contain dynamic developments (Guan et al., 2023), and their supporting details. **Inefficiency in Holistic Query Matching.** Existing graph RAG methods typically retrieve information by matching the entire user query against graph nodes or community summaries. This is suboptimal for complex queries involving multiple distinct aspects or events. For intricate queries, a single embedding vector often fails to capture nuances, resulting in the retrieval of irrelevant information or the omission of necessary information. Therefore, an explicit step is required to decompose user intent to match specific events.

To address these challenges, we propose a novel framework that reconceptualizes both graph structure and the retrieval process.

First, we replace the multi-layer hierarchical structure with a streamlined two-layer architecture: an event layer and an entity layer. Instead of clustering, we employ an iterative event extraction mechanism. This approach actively consolidates independent events from the original chunks containing the entities, encompassing both global summaries and narrative developments. This reduces structural complexity from k-layers to a fixed, interpretable two-layer depth.

Second, we introduce a query decomposition and event matching strategy. Rather than performing coarse retrieval based on the similarity of the entire query, we decompose the source query into event-based sub-queries. For complex problems, the system can precisely locate relevant events in addition to retrieving detailed entities. This effectively filters out noise—information semantically similar to the query but from irrelevant contexts—and prevents the omission of pertinent contexts.

Overall, our contributions can be summarized as follows:

- We propose a lightweight two-layer graph structure , replacing hierarchical community clustering with iterative event extraction. This causes higher-layer nodes to represent logically coherent events rather than abstract semantic summaries, thereby improving interpretability and indexing efficiency.
- We develop an event-based query decomposition mechanism. By decomposing complex queries to match specific events, we achieve higher retrieval precision than standard global and local matching, ensuring the context received by the LLM is strictly relevant to the sub-scenarios of the question.
- We conduct extensive experiments to evaluate the effectiveness of E^2GraphRAG. Compared with widely-used and advanced baseline methods, the experimental results demonstrate that E^2GraphRAG delivers significant improvements in answering complex questions.

2 Related Work

In this section, we briefly review several retrieval-augmented generation methods with graph structures.

Extracting knowledge graphs with LLMs. The significant advancements in LLMs have provided feasible solutions for the automatic construction of knowledge graphs. KGGen (Mo et al., 2025) significantly reduces the sparsity of extracted graphs through entity clustering and relation normalization, and introduces MINE, the first evaluation benchmark for text-to-knowledge graph extraction. SAC-KG (Chen et al., 2024) proposes an iterative framework comprising a generator, a verifier, and a pruner, enabling the construction of high-precision and scalable knowledge graphs in specialized domains. Tree-KG (Niu et al., 2025) presents a scalable tree-structured knowledge graph construction method, which performs hierarchical extraction by combining textbook structures with semantic operators, demonstrating excellent performance in structural alignment and domain adaptation. Furthermore, (Parović et al., 2025) directly leverages the implicit world knowledge within LLMs for graph construction. It introduces a prompt-based iterative generation pipeline, extracting domain-specific graphs from LLM parameters via schema guidance and chain-of-thought verification techniques. Although LLM-driven knowledge graph extraction methods show continuous progress in enhancing extraction coverage and structural consistency, the presence of redundant and noisy elements in the extracted knowledge graphs remains a noteworthy issue (Zheng et al., 2025).

Graph RAG. RAG enhances the performance of large language models on domain-specific

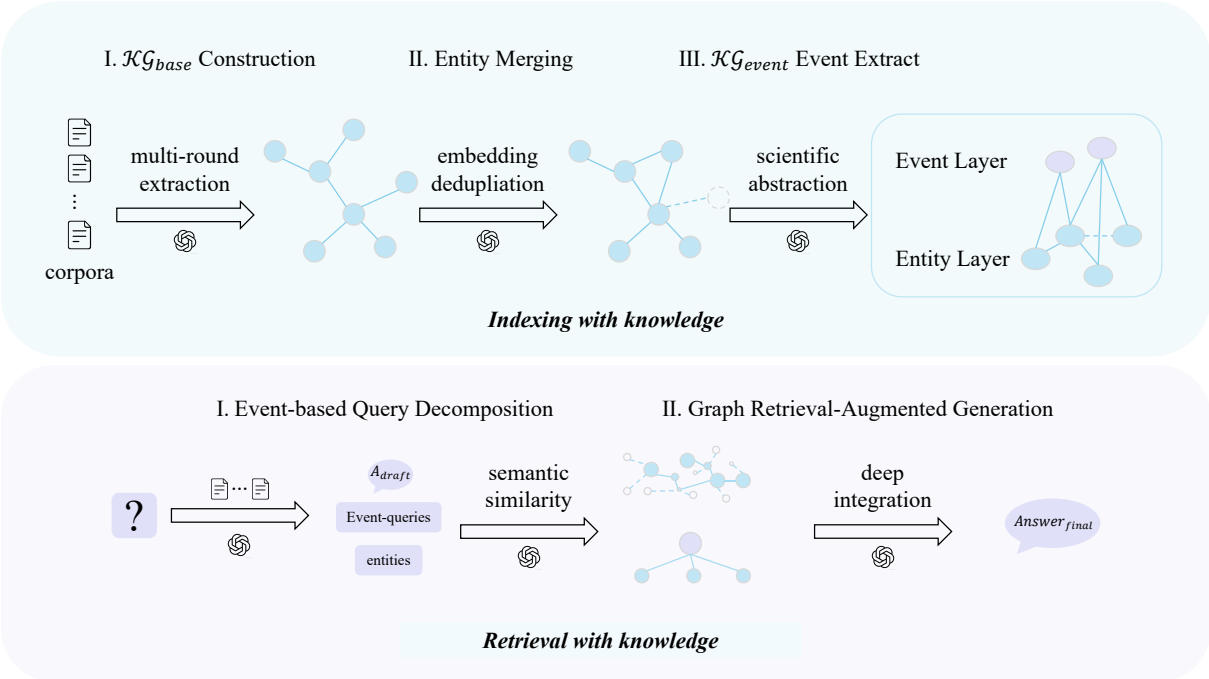

Figure 2: The overall architecture of the E^{2}GraphRAG framework.

tasks by incorporating external knowledge. Early RAG methods (Zhang et al., 2025d; Lyu et al., 2025) directly split documents into text chunks for vector retrieval, which overlooks semantic relationships both within and across chunks. To achieve better performance on complex reasoning tasks, GraphRAG (Edge et al., 2024), constructs a document-level knowledge graph and identifies community structures, these community summaries then provide global descriptions and insights over the entire corpus. LightRAG (Guo et al., 2024) eliminates the need for community construction by introducing a two-level retrieval mechanism. This approach effectively retrieves both detailed knowledge from specific chunks and higher-level conceptual summaries. HiRAG (Huang et al., 2025) further integrates hierarchical knowledge into GraphRAG, constructing a multi-level knowledge graph. During retrieval, it incorporates bridge-layer knowledge to connect global and local information, significantly enhancing the system's semantic understanding and structural capture capabilities. LeanRAG (Zhang et al., 2025c) proposes a semantic aggregation and hierarchical retrieval framework. By modeling relationships among aggregated entities and employing a lowest common ancestor-based retrieval strategy, it reduces information redundancy while strengthening cross-community reasoning. Despite these improvements, the deep and abstract semantic structures in existing methods often fail to capture the dynamic nature of events. Moreover, the retrieval process tends to overlook queries that may involve multiple interrelated events, resulting in retrieved contexts that are overly summarized and contain noise.

3 The Framework

As shown in the figure 2, RAG provides structured, multi-level contexts to the LLM through the Index and Retrieval modules, enabling the generation of comprehensive and coherent answers.

3.1 Indexing with Knowledge

During the construction of the OursGraph, we first follow the knowledge graph construction pipeline of LightRAG (Guo et al., 2024). This process takes a raw document collection $\mathcal{D}$ as input, segments the documents into a set of overlapping text chunks $\mathcal{C} = \{c_1, c_2, ..., c_m\}$. Subsequently, we employ an LLM together with carefully designed prompt templates to identify entities and extract relations:

$$\mathcal{F}_{extract} = c_i \rightarrow \{(h, r, t)\}, \quad (1)$$

where $h, t \in V_0$ are entities, and $r \in E_0$ represents a relation. To address redundancy in knowledge graph construction by LLMs (Zheng et al., 2025), we perform semantic-based entity fusion on the extracted $\mathcal{KG}_{base}$. Specifically, we generate vector representations for the descriptive text of each entity $v \in V_0$. For each entity type, we then calculate

the cosine similarity between entities:

$$\mathrm{sim}(v_i, v_j) = \frac{\mathbf{z}_{v_i} \cdot \mathbf{z}_{v_j}}{\|\mathbf{z}_{v_i}\| \, \|\mathbf{z}_{v_j}\|}, \tag{2}$$

if $\mathrm{sim}(v_i, v_j) > \theta_{merge}$, the entities are considered mergeable concepts. For each group of entities to be merged $\{v_1, v_2, \ldots, v_k\}$, we select the entity v_s with the most detailed description as the representative node. The descriptions and relations of all other entities in the group are then merged into v_s, thereby obtaining the knowledge graph $\mathcal{KG}_{base}$.

To capture the dynamic evolution and interdependencies within complex narratives, we introduce an Event layer to represent higher-level semantic units. This layer aims to discover and integrate structured event information from the graph:

$$E_{update}, E_{new} = \mathcal{U}(T_v, \mathcal{R}_{event}(v)), \tag{3}$$

here, for any non-event entity $v \in \mathcal{KG}_{base}$, T_v denotes the set of all chunks from which the edges connected to v originate. Text chunks are prioritized for processing based on their edge weights, ensuring the most relevant contexts are handled first. $\mathcal{R}_{event}$ and $\mathcal{U}$ represent ego-network event retrieval, and generative updating and extraction, respectively. E_{update} and E_{new} denote updated events and newly extracted events. After obtaining E_{update} and E_{new}, edges are added between these events and v in the graph.

Ego-Network Event Retrieval $\mathcal{R}_{event}$. To avoid extracting duplicate events, we retrieve events directly connected to the node v as well as those events connected to its neighboring nodes. This effectively captures a broader event landscape.**Generative Updating and Extraction $\mathcal{U}$.** The LLM analyzes the contextual text chunks and existing events to perform two tasks: If a text chunk provides more specific details for a known event, it updates the event's description and refines its title; It extracts new events not present in $\mathcal{R}_{event}$.

Finally, we perform summarization rewriting for lengthy descriptions of certain nodes and edges. This compresses the text while preserving factual accuracy, resulting in a coherent, deduplicated, and semantically aligned event knowledge graph, $\mathcal{KG}_{event}$.

3.2 Retrieval with Knowledge

$\mathcal{KG}_{event}$ is represented as an undirected graph $G = (V, E)$, where each node $n \in V$ stores information such as its description and entity type. In this section, we describe the query module of E^{2}GraphRAG. This module employs an event-based query decomposition strategy to retrieve knowledge from the knowledge graph for answer generation. The retrieval process can be formalized by a retriever $\mathcal{L}$, which retrieves a composite context comprising text chunks, entities, events, and paths:

$$A_{\text{draft}}, E_q, Q_{\text{sub}} = \Psi(q, c), \tag{4}$$

$$A_{\text{final}} = \mathcal{L}\big(q, A_{\text{draft}}, \Phi_{\text{context}}(E_q, Q_{\text{sub}})\big), \tag{5}$$

Event-based Query Decomposition Ψ. For a user-provided query q, we first perform a vector-based matching search over the text chunks. The chunks with high similarity to the query are concatenated to form a context c. This context, along with a prompt, is then fed into a large language model to generate an initial answer A_{draft}. This step ensures the answer is grounded in the most relevant textual evidence. Concurrently, the LLM is guided to extract content requiring further queries from the query q, the text chunks c, and the initial answer A_{draft}. This content includes a list of entity names $E_q = \{e_1, e_2, \ldots\}$ and a set of event-centric sub-queries $Q_{event} = \{q_{event1}, q_{event2}, \ldots\}$.

Event-based Context Assembly Φ_{context}. We compute the similarity between the entities in E_q and the non-event entity nodes within $\mathcal{KG}_{event}$, retrieving the most relevant entities as $E_{\text{entity}} = [e_1, \ldots, e_n]$. The sub-queries Q_{event} are used to match event-type nodes, which are saved as a list $E_{\text{events}} = [event_1, \ldots, event_m]$. The ordered lists E_{entity} and E_{events} represent the semantically most relevant nodes in the knowledge graph to the user's query.

Isolated entity descriptions often contain limited and incoherent information. Therefore, for the nodes in E_{events}, we retrieve their neighboring nodes to form a subgraph of the knowledge graph, denoted as KG_{sub}. For the nodes in E_{entity}, we sequentially generate paths in the order of their semantic relevance:

$$Paths = \{Path(e_1 \rightarrow e_2), \ldots\}, \tag{6}$$

where $Path(e_i \rightarrow e_j)$ represents a reasoning path between nodes e_i and e_j in the knowledge graph, thereby forming a coherent reasoning chain through this method. To avoid introducing irrelevant events, the aforementioned paths are searched for on a temporary graph, from which all event nodes have been removed.

Table 1: Win rates (%) of RAG, its two variants (for ablation study, Dec is "Decomposition"), and baseline methods.

	Agriculture		**Legal**		**CS**		**Mix**	
	NaiveRAG	**Ours**	NaiveRAG	**Ours**	NaiveRAG	**Ours**	NaiveRAG	**Ours**
Comprehensiveness	19.2%	80.8%	5.9%	94.1%	11.2%	88.8%	15.6%	84.4%
Diversity	3.3%	96.7%	2.1%	97.9%	0.8%	99.2%	9.8%	90.2%
Empowerment	10.0%	90.0%	3.0%	97.0%	7.4%	92.6%	13.9%	86.1%
Overall	10.8%	89.2%	3.8%	96.2%	7.4%	92.6%	13.5%	86.5%
	GraphRAG	**Ours**	GraphRAG	**Ours**	GraphRAG	**Ours**	GraphRAG	**Ours**
Comprehensiveness	46.3%	53.7%	40.4%	59.6%	40.5%	59.5%	46.5%	53.5%
Diversity	20.5%	79.5%	50.0%	50.0%	12.7%	87.3%	32.7%	67.3%
Empowerment	32.0%	68.0%	39.2%	60.8%	35.0%	65.0%	44.7%	55.3%
Overall	39.8%	60.2%	40.0%	60.0%	35.0%	65.0%	45.1%	54.9%
	LightRAG	**Ours**	LightRAG	**Ours**	LightRAG	**Ours**	LightRAG	**Ours**
Comprehensiveness	10.3%	89.7%	4.5%	95.5%	7.0%	93.0%	13.2%	86.8%
Diversity	2.9%	97.1%	1.6%	98.4%	0.8%	99.2%	2.8%	97.2%
Empowerment	6.2%	93.8%	2.8%	97.2%	3.7%	96.3%	11.6%	88.4%
Overall	6.2%	93.8%	2.8%	97.2%	3.7%	96.3%	10.4%	89.6%
	HiRAG	**Ours**	HiRAG	**Ours**	HiRAG	**Ours**	HiRAG	**Ours**
Comprehensiveness	40.5%	59.5%	24.4%	75.6%	36.2%	63.8%	42.6%	57.4%
Diversity	19.8%	80.2%	23.1%	76.9%	21.1%	78.9%	32.6%	67.4%
Empowerment	32.2%	67.8%	23.5%	76.5%	33.6%	66.4%	39.3%	60.7%
Overall	34.3%	65.7%	23.5%	76.5%	33.6%	66.4%	40.5%	59.5%
	w/o Index	**Ours**	w/o Index	**Ours**	w/o Index	**Ours**	w/o Index	**Ours**
Comprehensiveness	42.2%	57.8%	32.2%	67.8%	34.0%	66.0%	44.6%	55.4%
Diversity	19.7%	80.3%	17.4%	82.6%	11.5%	88.5%	20.2%	79.8%
Empowerment	38.9%	61.1%	31.4%	68.6%	33.2%	66.8%	42.1%	57.9%
Overall	40.2%	59.8%	31.4%	68.6%	34.0%	66.0%	42.6%	57.4%
	w/o Dec	**Ours**	w/o Dec	**Ours**	w/o Dec	**Ours**	w/o Dec	**Ours**
Comprehensiveness	41.1%	58.9%	30.7%	69.3%	34.0%	66.0%	41.9%	58.1%
Diversity	15.4%	84.6%	17.2%	82.8%	7.4%	92.6%	22.4%	77.6%
Empowerment	39.4%	60.6%	30.3%	69.7%	30.3%	69.7%	37.4%	62.6%
Overall	39.8%	60.2%	30.3%	69.7%	30.3%	69.7%	39.0%	61.0%

At this stage, the structured evidence $\Phi_{\text{context}} = \{KG_{\text{sub}}, Paths\}$ incorporates more granular knowledge. We then guide the LLM to perform deep integration, verification, and enhancement using the original query q, the initial answer A_{draft}, and Φ_{context}, ultimately outputting a comprehensive and logically rigorous final answer.

4 Experiments

In this section, we report the validation results of E^{2}GraphRAG on benchmark data. In Section 4.1, we describe how our experiments are conducted. Subsequently, in Section 4.2, we compare the answer generation performance of E^{2}GraphRAG with state-of-the-art methods. Finally, in Section 4.3, we explore the roles of event and two-stage mechanisms in E^{2}GraphRAG.

4.1 Experimental Setup

Baseline Methods. We compare E^{2}GraphRAG with the following advanced and widely used RAG methods: **NaiveRAG** (Gao et al., 2022, 2023) splits input documents into text chunks and retrieves relevant chunks via vector similarity, but does not model relationships between entities. **GraphRAG** (Edge et al., 2024) constructs a flat knowledge graph from documents, employs the Leiden algorithm to identify communities, and retrieves community summaries to generate community answers before selecting the most relevant information as the global context. **LightRAG** (Guo et al., 2024) adopts a two-level retrieval strategy

Table 2: Win rates (%) of RAG, and baseline methods.

	Agriculture		Legal		CS		Mix	
	GraphRAG	**Ours**	GraphRAG	**Ours**	GraphRAG	**Ours**	GraphRAG	**Ours**
Comprehensiveness	50.0%	50.0%	40.3%	59.7%	49.3%	50.7%	50.5%	49.5%
Diversity	18.9%	81.1%	48.4%	51.6%	17.4%	82.6%	36.6%	63.4%
Empowerment	33.5%	66.5%	33.9%	66.1%	42.0%	58.0%	47.3%	52.7%
Overall	43.9%	56.1%	37.9%	62.1%	44.9%	55.1%	49.5%	50.5%
	LightRAG	**Ours**	LightRAG	**Ours**	LightRAG	**Ours**	LightRAG	**Ours**
Comprehensiveness	43.8%	56.2%	15.6%	84.4%	39.1%	60.9%	23.6%	76.4%
Diversity	3.7%	96.3%	1.9%	98.1%	6.5%	93.5%	8.5%	91.5%
Empowerment	20.4%	79.6%	6.2%	93.8%	21.0%	79.0%	15.1%	84.9%
Overall	25.9%	74.1%	8.8%	91.2%	23.9%	76.1%	16.0%	84.0%
	HiRAG	**Ours**	HiRAG	**Ours**	HiRAG	**Ours**	HiRAG	**Ours**
Comprehensiveness	42.6%	57.4%	37.3%	62.7%	49.2%	50.8%	47.8%	52.2%
Diversity	29.4%	70.6%	27.6%	72.4%	26.2%	73.8%	44.1%	55.9%
Empowerment	38.2%	61.8%	32.8%	67.2%	41.3%	58.7%	48.5%	51.5%
Overall	40.4%	59.6%	34.3%	65.7%	42.9%	57.1%	47.8%	52.2%

to obtain both local entity descriptions and global relational knowledge, yet does not explicitly bridge these two layers. **HiRAG** (Huang et al., 2025) further enhances semantic connectivity and bridges knowledge gaps through hierarchical knowledge graph indexing and three-layer knowledge retrieval.

Datasets and Queries. We follow the experimental setup of LightRAG (Guo et al., 2024) and HiRAG (Huang et al., 2025), using four domain-specific datasets from the UltraDomain benchmark (Qian et al., 2024), which are designed for long-context, knowledge-intensive tasks. Mix contains a heterogeneous collection of documents from multiple domains. CS consists of technical documents related to computer science. Legal includes legal provisions and case documents. Agriculture covers documents related to agricultural science and practices. Following the methodology in LightRAG (Guo et al., 2024), we generate 125 questions for each dataset.

LLM. We employ GPT-4o-mini to perform knowledge graph construction, answer generation, and evaluation. Detailed parameter settings are provided in Appendix B. For embedding-based operations, such as vector search and semantic similarity matching, we utilize the Qwen3-embedding-0.6B model (Zhang et al., 2025b).

4.2 Overall Performance Comparison

Evaluation Details. Following the evaluation protocol established in recent Graph-based RAG studies (Edge et al., 2024; Guo et al., 2024), we use GPT-4o-mini as a powerful LLM judge to conduct a comprehensive evaluation by comparing the answers generated by two methods for each query (Zheng et al., 2023). The evaluation covers the following dimensions: (1) **Comprehensiveness**: How much detail does the answer provide to cover all aspects and details of the question? (2) **Empowerment**: How well does the answer help the reader understand and make informed judgments about the topic? (3) **Diversity**: How varied and rich is the answer in providing different perspectives and insights on the question? (4) **Overall**: a holistic assessment of the answer.

To ensure fairness, each pair of answers is compared twice, with their order swapped in the evaluation prompt, and the average win rate is then computed. The evaluation prompt is designed to output responses in a structured json format to ensure consistency in parsing.

Evaluation Results. Table 1 shows the win rates of E^{2}GraphRAG against each baseline method across the four datasets. The main observations are as follows:

E^{2}GraphRAG consistently outperforms all baseline methods across almost all evaluation dimensions and datasets. It achieves win rates exceeding 80% against NaiveRAG on the Agriculture, Mix, CS, and Legal datasets. Compared to baseline methods utilizing flat knowledge graphs, namely GraphRAG and LightRAG, E^{2}GraphRAG maintains a clear advantage with win rates generally above 50%. Even when compared to HiRAG, a

How have consumer preferences shifted in relation to local versus industrial honey production?

E²GraphRAG:
In recent years, ... reflecting a broader trend towards valuing ***local, sustainable, and ethically produced food***. This change is driven by a growing awareness of the environmental and health implications associated with industrial honey production, as well as a desire for transparency and quality in food sourcing...

The Rise of Local Honey Production...
Concerns with Industrial Honey Production...
The Role of Education and Awareness...

providing detailed insights

Conclusion. ... This shift is characterized by ***a desire for quality, transparency, and sustainability, as consumers increasingly seek to support local economies and environmentally friendly practices***...

HiRAG:
In recent years, ... reflects a growing desire for ***transparency*** in food systems, particularly in the context of honey production, where concerns about ***quality and ethical practices*** dominate consumer choices.

Emphasis on Local Sourcing...
Perception of Quality...
The Role of Education and Awareness...

Limited perspectives and insufficient discussion

Conclusion. The shift in consumer preferences towards local honey production is a multi-faceted trend driven by a desire for ***quality, transparency, and community engagement...***

"Overall Winner": {"Winner": *"Answer 1"*, "Explanation": "Answer 1 is the overall winner as it excels in all three criteria: it is comprehensive in covering various aspects of consumer preferences, diverse in presenting multiple perspectives, and empowering in educating readers about the implications of their choices. *Answer 2, while valuable, does not match the depth and breadth of insights provided in Answer 1.*"}

Figure 3: Case study, comparison between E²GraphRAG and the baseline method HiRAG.

baseline method based on hierarchical knowledge graphs, E²GraphRAG still demonstrates strong performance on the Agriculture, Mix, CS, and Legal datasets.

This indicates that the design of E²GraphRAG brings tangible improvements, enabling the generation of more comprehensive, empowering, and diverse answers. We attribute this improvement to our novel integration of semantic event extraction and event-based query decomposition. E²GraphRAG leverages entity scanning to dynamically generate and enrich event descriptions from multiple text chunks, effectively capturing temporal relationships and causal logic. During retrieval, it decomposes complex user queries into specific event-related and entity-centric sub-queries. This process significantly improves the signal-to-noise ratio within the retrieved context.

4.3 Ablation Study

To validate the contribution of each component in E²GraphRAG, we conducted two ablation experiments, as shown in table 1. These ablation experiments highlight the superiority of the proposed method in both indexing and retrieval.

w/o Index. A variant of E²GraphRAG that replaces the hierarchical knowledge graph with a flat knowledge graph $\mathcal{KG}_{base}$ underperforms the complete E²GraphRAG across all datasets, with win rates dropping by 5 to 30 percentage points. This fully demonstrates that Event nodes can provide refined and summarized content, thereby improving answer quality.

w/o Decomposition. Removing the two-stage retrieval module and directly feeding the retrieved text chunks, events, reasoning paths, and other information into the LLM to generate answers leads to an even more pronounced performance decline after losing the step-by-step reasoning process. While the LLM attempts to find the solution logic from the Paths, it struggles to extract additional details from the Events, resulting in reduced answer comprehensiveness.

5 Discussion

The experimental findings presented above demonstrate that our proposed method enhances the overall performance of RAG systems to a certain extent. Based on the obtained experimental data, we provide a concise discussion below.

Table 3: context tokens on the Agriculture dataset.

E²GraphRAG	GraphRAG	LightRAG	HiRAG
47954.4	1240157	29477.6	46574.6

5.1 Analysis of Context During Retrieval

As shown in the table 3, we calculate the average number of context tokens consumed by Graph-based methods when answering questions on the

Agriculture dataset. GraphRAG retrieves multiple communities, generates summaries for each, and then selects the most relevant one, leading to higher token consumption compared to E^{2}GraphRAG. Secondly, while the total Context Tokens input to the LLM by our method and the baseline HiRAG are comparable, our approach achieves a significant advantage of approximately 10% in win rate. This clearly reveals a fundamental difference in information density and relevance between the two methods: events may contain critical information absent in community summaries, and our retrieval strategy can effectively filter out redundancy and noise, selecting high-value information segments. These two aspects enable the LLM to acquire more supportive knowledge within a limited context capacity, thereby generating higher-quality answers.

5.2 Mitigating Length Bias in Evaluation

LLM-based evaluation of generated answers is susceptible to biases (Park et al., 2024), including those introduced by answer length. To mitigate the potential interference of length bias in assessment, we employ an LLM to expand answers from different methods to comparable lengths without introducing additional information. Following the evaluation protocol described in Section 4.1, we obtain the results presented in Table 2. These results demonstrate that our approach maintains its performance advantage even after controlling for answer length. This further confirms that the improvements achieved by E^{2}GraphRAG in answer generation are robust and not attributable to length-related artifacts.

5.3 Case Study

To more clearly demonstrate the performance of E^{2}GraphRAG in the answering process, as shown in Figure 3, we have selected a case example for analysis by comparing the answers generated by HiRAG and E^{2}GraphRAG. E^{2}GraphRAG was judged as the winner across all three evaluation dimensions because it provides a thorough analysis of the shifts in consumer preferences regarding honey production, incorporates various perspectives on sustainability and ethics, and empowers readers to make informed choices about their food sources. Its depth and breadth of information make it a more valuable resource. In contrast, the answer from HiRAG, while informative, lacks the depth and breadth found in the answer from E^{2}GraphRAG. The evaluation results of this case reflect the strengths of E^{2}GraphRAG, as it is capable of systematically extracting and integrating scattered events related to the query from the corpus, constructing a multi-dimensional perspective in its response, thereby demonstrating greater strength in providing informational depth and actionable insights. We provide detailed LLM decision analysis for this case in Appendix C.

6 Conclusions

In this work, we present E^{2}GraphRAG, a retrieval-augmented generation framework that incorporates events into hierarchical knowledge graphs. By introducing an iterative event extraction method, we successfully transform the knowledge graph from a topological map into a dynamic event semantic network. Moreover, our decomposed query matching mechanism bridges the gap between complex user intents and structured graph data, enabling precise event retrieval. Experimental results indicate that, compared with widely used state-of-the-art baselines, our method enhances the comprehensiveness, diversity, and empowerment of answers, providing a practical solution for domain-specific question answering.

7 Limitations

Although E^{2}GraphRAG performs well in the experiments of this study and provides a viable approach for multi-event reasoning in answer generation, we acknowledge its limitations. The graph construction process in E^{2}GraphRAG relies on LLM calls, which is constrained by the inherent capabilities of LLMs and incurs notable token consumption and time costs. Recent work (Zhuang et al., 2025) has significantly reduced token consumption by constructing relation-free hierarchical graphs. Furthermore, in many practical applications such as agriculture, there exists a substantial amount of multimodal data, including images and videos. We believe these limitations represent promising directions for future research.

References

Hanzhu Chen, Xu Shen, Qitan Lv, Jie Wang, Xiaoqi Ni, and Jieping Ye. 2024. SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4345–4360, Bangkok, Thailand. Association for Computational Linguistics.

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*.

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, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. 2023. What is event knowledge graph: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7569–7589.

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

Haoyu Huang, Yongfeng Huang, Junjie Yang, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. 2025. Retrieval-augmented generation with hierarchical knowledge.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Trans. Inf. Syst.*, 43(2).

Belinda Mo, Kyssen Yu, Joshua Kazdan, Joan Cabezas, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. 2025. Kggen: Extracting knowledge graphs from plain text with language models.

Songjie Niu, Kaisen Yang, Rui Zhao, Yichao Liu, Zonglin Li, Hongning Wang, and Wenguang Chen. 2025. Tree-KG: An expandable knowledge graph construction framework for knowledge-intensive domains. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18516–18529, Vienna, Austria. Association for Computational Linguistics.

Junsoo Park, Seungyeon Jwa, Ren Meiying, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1043–1067, Miami, Florida, USA. Association for Computational Linguistics.

Marinela Parović, Ze Li, and Jinhua Du. 2025. Generating domain-specific knowledge graphs from large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11558–11574, Vienna, Austria. Association for Computational Linguistics.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*.

Bahareh Sarrafzadeh and Edward Lank. 2017. Improving exploratory search experience through hierarchical knowledge graphs. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 145–154, New York, NY, USA. Association for Computing Machinery.

Viju Sudhi, Sinchana Ramakanth Bhat, Max Rudat, and Roman Teucher. 2024. Rag-ex: A generic framework for explaining retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2776–2780, New York, NY, USA. Association for Computing Machinery.

V. A. Traag, L. Waltman, and N. J. van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1).

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

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models.

Yaoze Zhang, Rong Wu, Pinlong Cai, Xiaoman Wang, Guohang Yan, Song Mao, Ding Wang, and Botian Shi. 2025c. Leanrag: Knowledge-graph-based generation with semantic aggregation and hierarchical retrieval.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025d. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, pages 1–46.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang,

Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Yilun Zheng, Dan Yang, Jie Li, Lin Shang, Lihui Chen, Jiahao Xu, and Sitao Luan. 2025. Less is more: Denoising knowledge graphs for retrieval augmented generation.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2025. Linearrag: Linear graph retrieval augmented generation on large-scale corpora.

Appendix

A Experimental Datasets

Dataset	Mix	CS	Legal	Agriculture
# of Documents	61	10	94	12
# of Tokens	623380	2195901	5007168	2019669

Table 4: Statistics of datasets.

Table 4 presents the statistical characteristics of the experimental datasets, where all documents were consistently tokenized using Byte Pair Encoding (BPE) tokenizer "cl100k_base".

B LLM parameter settings

Phase	Task	Model	Temp.	Max Tokens
Index	Event Extraction	gpt-4o-mini	0.1	4096
Retrieval	Event-based Query Decomposition	gpt-4o-mini	0.2	2048
	Answer Enrichment			4096

Table 5: LLM parameter settings by pipeline phase

Table 5 describes the parameter settings of the LLM used in different stages of E^{2}GraphRAG. For the indexing stage, we set a low temperature to ensure deterministic event extraction, which is crucial for maintaining consistency in knowledge graph construction. The max tokens limit varies according to task complexity, with higher token allocations assigned to context-intensive operations.

C A Simple Case

In Section 5.1, we analyze a case study. Figure 4 presents the json data obtained after event-based query decomposition. E^{2}GraphRAG leverages this data to subsequently retrieve supporting evidence, which is then compiled into a context for answer generation. Figure 5 displays the complete LLM decision process. As described in Section 4.1, we systematically alternate the order of compared answers to obtain a more comprehensive evaluation. The results in Figure 5 demonstrate that E^{2}GraphRAG achieves strong performance across all four evaluation dimensions in this specific case.

D Prompt Templates

For the construction of the entity layer, we adopt the prompt design utilized in LightRAG.

D.1 Prompt Templates for Event Extraction

As shown in Figure 6, we iteratively extract and update events based on the constructed entity layer and the segmented text chunks.

D.2 Prompt Templates for Event-based Query Decomposition

As shown in Figure 7, we decompose the user query into event-based sub-queries to acquire more precise and comprehensive local and global information.

D.3 Prompt Templates for Answer Enrich

As shown in Figure 8, we subsequently generate the final answer by leveraging the structured evidence.

Retrieval Information

```
"entities": ["local honey", "industrial honey", "community-supported agriculture"],
"sub_queries": ["consumer preferences for honey", "impact of local sourcing on honey production", "education on
sustainable honey practices"]
```

Figure 4: The entities and event-based queries in the case that require further retrieval.

LLM Decision

```
"forward": (Answer 1: E2GraphRAG)
"Comprehensiveness": "Winner": "Answer 1", "Explanation": "Answer 1 provides a thorough exploration of the shifts in
consumer preferences, detailing the rise of local honey production, concerns with industrial practices, and the role of
education and awareness. It covers multiple aspects of the topic, including environmental impacts, community engagement,
and the implications of consumer choices, making it more comprehensive than Answer 2.",
"Diversity": "Winner": "Answer 1", "Explanation": "Answer 1 presents a wider range of perspectives, discussing not only
consumer preferences but also the implications for local economies, bee health, and the sustainable agriculture movement.
It incorporates various dimensions of the issue, whereas Answer 2, while informative, is more focused on the consumer
perspective without as much exploration of broader implications.",
"Empowerment": "Winner": "Answer 1", "Explanation": "Answer 1 empowers readers by providing detailed insights into the
implications of their choices, emphasizing the importance of education and community engagement. It encourages informed
decision-making by highlighting the benefits of local honey and the drawbacks of industrial practices, whereas Answer 2,
while informative, does not delve as deeply into the implications of consumer choices.",
"Overall Winner": "Winner": "Answer 1", "Explanation": "Answer 1 is the overall winner as it excels in all three criteria:
it is comprehensive in covering various aspects of consumer preferences, diverse in presenting multiple perspectives, and
empowering in educating readers about the implications of their choices. Answer 2, while valuable, does not match the
depth and breadth of insights provided in Answer 1."
"reverse": (Answer 2: E2GraphRAG)
"Comprehensiveness": "Winner": "Answer 2", "Explanation": "Answer 2 provides a more detailed exploration of the shifts in
consumer preferences, covering various aspects such as the rise of local honey production, concerns with industrial
practices, and the role of education and awareness. It discusses the implications of these shifts on community engagement
and environmental sustainability, offering a broader context than Answer 1.",
"Diversity": "Winner": "Answer 2", "Explanation": "Answer 2 presents a richer variety of perspectives, including the
impact of the sustainable agriculture movement, community-supported agriculture, and the ethical implications of food
production. It highlights different dimensions of consumer preferences and the broader societal trends influencing these
choices, whereas Answer 1 is more focused on quality and transparency without as much emphasis on diverse perspectives.",
"Empowerment": "Winner": "Answer 2", "Explanation": "Answer 2 empowers readers by providing insights into the
implications of their food choices, emphasizing the importance of supporting local economies and understanding the
environmental impacts of industrial practices. It encourages informed decision-making through education and awareness,
while Answer 1, although informative, does not emphasize the broader implications of consumer choices as effectively.",
"Overall Winner": "Winner": "Answer 2", "Explanation": "Answer 2 is the overall winner as it excels in comprehensiveness,
diversity, and empowerment. It provides a thorough analysis of the shifts in consumer preferences regarding honey
production, incorporates various perspectives on sustainability and ethics, and empowers readers to make informed choices
about their food sources. Its depth and breadth of information make it a more valuable resource."
```

Figure 5: The LLM Decision during evaluation in the case.

Event Extraction

```
-Target activity-
You are an intelligent assistant that helps a human analyst to extract and manage a knowledge graph of events from a text
document.
-Goal-
Given a text document that is potentially relevant to this activity, and a list of existing events, perform two tasks:
1. **Update Existing Events**: For events already tracked, enrich their descriptions with new details and refine their
titles if the new information provides greater specificity.
2. **Extract New Events**: Identify significant events in the text that are not in the existing list.
-Definitions-
1. **Event Title**: A concise, standard-case phrase representing the event.
2. **Event Description**: A **comprehensive** summary of the event. It MUST include:
* **Who**: The specific named entities (persons, organizations, locations) involved.
* **When**: All specific dates, times, or durations mentioned (e.g., "on 2023-01-05").
* **What/Why**: The specific action taken, the cause, and the detailed outcome or impact.
* **Quantitative Data**: Any monetary values, percentages, or counts associated with the event.
3. **Event Date**: The specific date or date range of the event in ISO-8601 format (YYYY-MM-DD). If unknown, use "NONE".
-Steps-
1. Analyze the input text and the list of existing events.
2. For each **existing event** found in the text:
* Assess if the new text provides additional details.
* Update the description to merge the old information with the new information.
* Format
({"updated_event"}{tuple_delimiter}<original_title>{tuple_delimiter}<new_title>{tuple_delimiter}<new_description>
{tuple_delimiter}<event_date>)
3. Identify **new events** in the text that do not overlap with existing events:
* Ensure the event is distinct and significant.
* Generate a descriptive title and a comprehensive description.
* Extract the source entity (the entity reporting or primarily associated with the event).
* Format:
("new_event"{tuple_delimiter}<title>{tuple_delimiter}<description>{tuple_delimiter}<event_date>{tuple_delimiter}<source>)
4. Return output in English as a single list of all updated and new events. Use **record_delimiter** as the list delimiter.
5. When finished, output completion_delimiter
-Real Data-
Existing Events:
existing_events
Text:
input_text
Output:
```

Figure 6: The prompt template for extracting and updating event information.

Event-based Query Decomposition

```
—Role—
You are a helpful assistant responding to questions about a dataset by synthesizing perspectives from **retrieved text
chunks**.
—User Question—
question
—Goal—
1) Generate a response of the target length and format that responds to the user's question, summarize all the **evidence
from text chunks provided below**.
Note that the **evidence items** provided below are ranked in the **descending order of importance**.
If you don't know the answer or if the provided **context** does not contain sufficient information to provide an answer,
just say so. Do not make anything up.
The final response should remove all irrelevant information from the **retrieved data** and merge the cleaned information
into a comprehensive answer that provides explanations of all the key points and implications appropriate for the response
length and format.
The response shall preserve the original meaning and use of modal verbs such as "shall", "may" or "will".
The response should also preserve all the data references previously included in the **context**, but do not mention the
**mechanics of the retrieval process**.
**Do not list more than 3 record ids in a single reference**. Instead, list the top 3 most relevant record ids and add
"+more" to indicate that there are more.
For example:
"Person X is the owner of Company Y and subject to many allegations of wrongdoing [Data: Chunks (C_12, C_3), +more]. He is
also CEO of company X [Data: Chunks (C 1)]"
where C_12 represent the id (not the index) of the relevant data record.
Do not include information where the supporting evidence for it is not provided.
Limit your response length to 2000 words.
2) A JSON block (only JSON) containing two fields: "entities" (list of specific entity names found in the query or answer)
and "sub_queries" (list of event-centered sub-query strings) that you extracted. Example:
{{"entities": ["entity1","entity2"], "sub_queries": ["event query 1","event query 2"]}}
—Target response length and format—
multiple paragraphs.
Add sections and commentary to the response as appropriate for the length and format. Style the response in markdown.
—Retrieved Context—
evidence
—Goal—
Generate a response of the target length and format that responds to the user's question, summarize all the **evidence
from text chunks provided below**.
Note that the **evidence items** provided below are ranked in the **descending order of importance**.
If you don't know the answer or if the provided **context** does not contain sufficient information to provide an answer,
just say so. Do not make anything up.
The final response should remove all irrelevant information from the **retrieved data** and merge the cleaned information
into a comprehensive answer that provides explanations of all the key points and implications appropriate for the response
length and format.
The response shall preserve the original meaning and use of modal verbs such as "shall", "may" or "will".
The response should also preserve all the data references previously included in the **context**, but do not mention the
**mechanics of the retrieval process**.
**Do not list more than 3 record ids in a single reference**. Instead, list the top 3 most relevant record ids and add
"+more" to indicate that there are more.
For example:
"Person X is the owner of Company Y and subject to many allegations of wrongdoing [Data: Chunks (C_12, C_3), +more]. He is
also CEO of company X [Data: Chunks (C 1)]"
where C_12 represent the id (not the index) of the relevant data record.
Do not include information where the supporting evidence for it is not provided.
Limit your response length to 2000 words.
—Target response length and format—
multiple paragraphs.
Add sections and commentary to the response as appropriate for the length and format. Style the response in markdown.
Answer now, then the JSON.
```

Figure 7: The prompt template for decomposing queries into event-based sub-queries.

Answer Enrich

```
—Role—
You are an expert editor and analyst. Your task is to refine and enrich a preliminary answer using new structured evidence.
—User Question—
question
—Inputs—
1. **Original Answer**: A preliminary answer generated from text chunks.
2. **Additional Evidence**: A set of specific Events and Reasoning Paths (connections between entities) retrieved from a
Knowledge Graph.
—Goal—
Generate a refined response that synthesizes the "Original Answer" with the "Additional Evidence".
Note that the **Additional Evidence** provided below is ranked in the **descending order of importance**.
**Instructions:**
1. **Integrate & Verify**: Seamlessly weave the Evidence into the narrative to explain *how* and *why* things happened. If
the "Additional Evidence" conflicts with the "Original Answer", prioritize the Evidence (Events and Paths) as it is more
structured and precise. Correct any hallucinations.
2. **Clean & Merge**: Remove all irrelevant information and merge the cleaned information into an answer that provides
explanations of all the key points and implications.
3. **Tone & Style**: The response shall preserve the original meaning and use of modal verbs such as "shall", "may" or
"will". Style the response in markdown.
4. **Citations**: The response should preserve and add data references. **Do not list more than 3 record ids in a single
reference**. Instead, list the top 3 most relevant record ids and add "+more".
**Citation Example:**
"Person X is the owner of Company Y [Data: Events (Event 12, Event 45), Paths (Path 1), +more]. He is also CEO of company
X [Data: Events (Event 99)]"
(Where Event 12, Path 1, etc., are the IDs found in the evidence below).
Do not include information where the supporting evidence for it is not provided.
—Target response length and format—
multiple paragraphs.
Limit your response length to 2000 words.
Add sections and commentary to the response as appropriate for the length and format.
—Data—
[Original Answer]
old_answer
[Additional Evidence]
evidence
—Goal—
Generate a refined response that synthesizes the "Original Answer" with the "Additional Evidence".
Note that the **Additional Evidence** provided below is ranked in the **descending order of importance**.
**Instructions:**
1. **Integrate & Verify**: Seamlessly weave the Evidence into the narrative to explain *how* and *why* things happened. If
the "Additional Evidence" conflicts with the "Original Answer", prioritize the Evidence (Events and Paths) as it is more
structured and precise. Correct any hallucinations.
2. **Clean & Merge**: Remove all irrelevant information and merge the cleaned information into an answer that provides
explanations of all the key points and implications.
3. **Tone & Style**: The response shall preserve the original meaning and use of modal verbs such as "shall", "may" or
"will". Style the response in markdown.
4. **Citations**: The response should preserve and add data references. **Do not list more than 3 record ids in a single
reference**. Instead, list the top 3 most relevant record ids and add "+more".
**Citation Example:**
"Person X is the owner of Company Y [Data: Events (Event 12, Event 45), Paths (Path 1), +more]. He is also CEO of company
X [Data: Events (Event 99)]"
(Where Event 12, Path 1, etc., are the IDs found in the evidence below).
Do not include information where the supporting evidence for it is not provided.
—Target response length and format—
multiple paragraphs.
Limit your response length to 2000 words.
Add sections and commentary to the response as appropriate for the length and format.
—Final Answer—
```

Figure 8: The prompt template for generating answers using events and paths.