Knowledge Graph-Guided Retrieval Augmented Generation

Xiangrong Zhu[♣] **Yuexiang Xie**[♡] **Yi Liu**[♣] **Yaliang Li**[♡] **Wei Hu**[♣]
[♣] State Key Laboratory for Novel Software Technology, Nanjing University, China
[♡] Alibaba Group
{xrzhu, yiliu07}.nju@gmail.com, whu@nju.edu.cn
{yuexiang.xyx, yaliang.li}@alibaba-inc.com

Abstract

Retrieval-augmented generation (RAG) has emerged as a promising technology for addressing hallucination issues in the responses generated by large language models (LLMs). Existing studies on RAG primarily focus on applying semantic-based approaches to retrieve isolated relevant chunks, which ignore their intrinsic relationships. In this paper, we propose a novel Knowledge Graph-Guided Retrieval Augmented Generation (KG^{2}RAG) framework that utilizes knowledge graphs (KGs) to provide fact-level relationships between chunks, improving the diversity and coherence of the retrieved results. Specifically, after performing a semantic-based retrieval to provide seed chunks, KG^{2}RAG employs a KG-guided chunk expansion process and a KG-based chunk organization process to deliver relevant and important knowledge in well-organized paragraphs. Extensive experiments conducted on the HotpotQA dataset and its variants demonstrate the advantages of KG^{2}RAG compared to existing RAG-based approaches, in terms of both response quality and retrieval quality.

1 Introduction

Recently, large language models (LLMs) (Li et al., 2024; Ren et al., 2024; Touvron et al., 2023; Brown et al., 2020) have achieved remarkable success across a broad range of real-world tasks, including question answering (Sen et al., 2023), writing assistance (Calamo et al., 2023), code generation (Cheng et al., 2024), and many others (Kaddour et al., 2023; Wu et al., 2023). However, hallucinations (Xu et al., 2024b; Liu et al., 2024a) in the generated responses becomes a critical challenge, which often results from containing outdated information or lacking domain-specific knowledge. Retrieval-augmented generation (RAG) (Gao et al., 2023; Fan et al., 2024) has emerged as a feasible solution to mitigate hallucinations by retrieving relevant knowledge from provided documents and incorporating it into the prompts of LLMs for response generation.

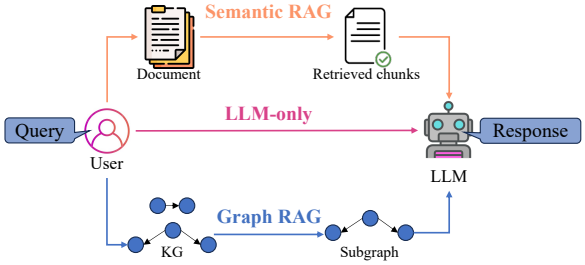


Figure 1: A comparison among LLM-only, Semantic RAG, and Graph RAG paradigms.

Existing studies in RAG (Lewis et al., 2020; Yu, 2022; Purwar and Sundar, 2023; Gao et al., 2023; Ziletti and D'Ambrosi, 2024), as shown in Fig. 1, employ keyword-based or semantic-based approaches to retrieve documents or chunks having the highest similarities to user queries. However, these retrieved chunks can be homogeneous and redundant, which fails to provide the intrinsic relationships among these chunks and cannot further activate the reasoning abilities of LLMs. Furthermore, the retrieved chunks are often directly concatenated in the order of their similarity scores and fed into LLMs as part of the prompts. Such a practice can lead to isolated pieces of information, limiting the utility of LLMs in generating comprehensive and reliable responses.

Knowledge graphs (KGs) (Auer et al., 2007; Ji et al., 2022), as structured abstractions of real-world entities and their relations, can be expected to effectively supplement existing semantic-based RAG approaches by integrating structured factual knowledge. Knowledge within a KG, represented in the form of triplets (*head entity*, *relation*, *tail entity*), is naturally linked through overlapping entities. A simplified workflow for utilizing KGs in RAG is shown in Fig. 1, where relevant triplets are retrieved to augment the context for response gen-

eration in LLMs, providing fact-level relationships among chunks and highlighting important facts that may be missed by semantic-based approaches.

Shed light by such insights, in this paper, we propose a novel **K**nowledge **G**raph-**G**uided **R**etrieval **A**ugmented **G**eneration framework, called KG^{2}RAG. Specifically, we first perform chunking and KG-chunk association during the offline processing of the provided documents, establishing linkages between chunks and a specific KG to capture the fact-level relationships among these chunks. Based on the chunks and the KG, KG^{2}RAG employs KG-enhanced chunk retrieval, which consists of a semantic-based retrieval and graph-guided expansion. The semantic-based retrieval prepares several seed chunks using embedding and ranking techniques (Nussbaum et al., 2024; Li and Li, 2024). These seed chunks are then used to extract a relevant subgraph from the association KG, onto which we can apply graph traversal algorithms to include the chunks containing overlapped or related entities and triplets. Such a design of graph-guided expansion provides a greater diversity of retrieved chunks and a comprehensive knowledge network.

After that, we incorporate a post-processing stage named KG-based context organization in KG^{2}RAG. On one hand, the KG-based context organization serves as a filter to retain the most relevant information contained in the subgraph, thereby enhancing the informativeness of the retrieved chunks. On the other hand, it serves as an arranger to organize the chunks into internally coherent paragraphs with the knowledge graph as a skeleton. These semantically coherent and well-organized chunks are fed into the LLMs along with user queries for response generation.

We conduct a series of experiments on the widely-used HotpotQA (Yang et al., 2018) dataset and its newly constructed variants to mitigate the impacts of prior knowledge on LLMs. We adopt a distractor and a fullwiki setting, comparing KG^{2}RAG with several RAG-based approaches. The experimental results demonstrate that KG^{2}RAG consistently outperforms baselines in terms of both response quality and retrieval quality. Moreover, we conduct an ablation study to highlight the effectiveness of different modules in KG^{2}RAG. The constructed dataset and source code are released at `https://github.com/nju-websoft/KG2RAG` to further promote the development and application of KGs in RAG.

2 Methodology

An overview of the workflow of KG^{2}RAG is illustrated in Fig. 2. In the following subsections, we provide more details following the workflow of KG^{2}RAG, including document offline processing (Sec. 2.1), KG-enhanced chunk retrieval (Sec. 2.2), and KG-based context organization (Sec. 2.3).

2.1 Document Offline Processing

Following the existing studies in RAG (Lewis et al., 2020; Gao et al., 2023; Fan et al., 2024), all documents are first split into n chunks based on the structure of sentences and paragraphs given a predefined chunk size, which can be given as $\mathcal{D} = \{c_1, \ldots, c_n\}$. These chunks can be further processed, for example, by adding relevant context (Jiang et al., 2023; Eibich et al., 2024), extracting meta-information (Mombaerts et al., 2024) (e.g., title, abstract), and generating corresponding questions (Ma et al., 2023; Wang et al., 2024b). Since these chunk-enhancing techniques are orthogonal to the proposed method in this paper, we recommend referring to the original paper for more details. Hereafter, we continue to denote the processed chunks as $\mathcal{D} = \{c_1, \ldots, c_n\}$.

To capture the rich fact-level relationships among these chunks, we associate them with a KG, which can be implemented via the following approaches. In cases where a KG is available, such as in WebQSP (Yih et al., 2016) and CWQ (Talmor and Berant, 2018), the chunk-KG association can be performed through entity and relation recognition and linkage algorithms (Zhao et al., 2023; Tian et al., 2024). Another approach involves directly extracting multiple entities and relations from the chunks to form subgraphs, which can be used to combine into a complete graph. In this paper, to avoid reliance on existing KGs, we adopt the latter approach, implementing it by providing appropriate prompts (refer to Fig. 3) to LLMs.

After this process, we provide linkages between chunks and a specific KG, which can be given as

$$\mathcal{G} = \{(h, r, t, c) \mid c \in \mathcal{D}\}, \tag{1}$$

where h, r, and t denote the head entity, relation, and tail entity, respectively, and c denotes the chunk that derives the triplets. Note that the chunk-KG association process is query-independent, which implies that it can be performed offline, only needs to be constructed once for all documents, and supports incremental updates for new documents. As

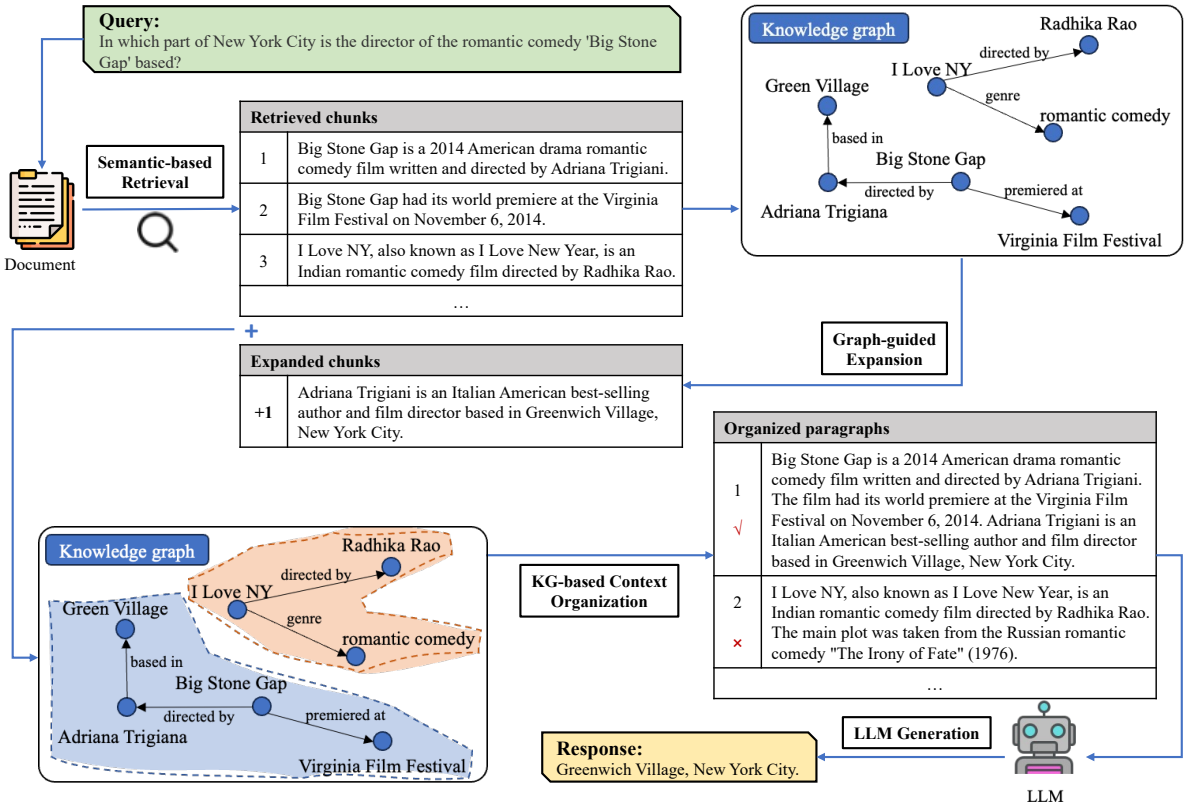


Figure 2: Workflow of the proposed KG^{2}RAG.

the document offline processing aligns with what vanilla RAG does, KG^{2}RAG naturally supports adding new documents to or removing documents from the existing knowledge base and KG efficiently.

2.2 KG-enhanced Chunk Retrieval

Given the chunks $\mathcal{D}$ and the associated KG $\mathcal{G}$, the proposed KG^{2}RAG suggests a two-stage retrieval process, including semantic-based retrieval and graph-guided expansion.

Semantic-based Retrieval During the semantic-based retrieval process, the semantic similarities between a user query q and all the chunks can be measured as

$$\mathcal{S} = \{s(q, c) \,|\, c \in \mathcal{D}\}, \tag{2}$$

where the similarity function $s(\cdot)$ employs an embedding model (Nussbaum et al., 2024; Li and Li, 2024) to transfer the query and chunks into high-dimensional representations, followed by computing their cosine similarity.

The chunks with the top-k highest similarities to the query are selected as the retrieved chunks, denoted by $\mathcal{D}_q$. These retrieved chunks can be integrated into the prompts as context and fed into LLMs for RAG. As discussed in Sec. 1, relying solely on semantic-based retrieval may result in isolated chunks, missing crucial factual knowledge and the intrinsic connections among the chunks. To tackle this, we regard the retrieved chunks $\mathcal{D}_q$ as seed chunks, and propose a graph-guided expansion process.

Graph-guided Expansion During communication and thinking processes, people often connect one event to others as these events involve the same entities, such as persons and places. For example, *Capitol Hill, Washington, D.C.* connects our impressions of *Barack Obama*, *Donald Trump*, and *Joe Biden*, as they all delivered their presidential inaugural speeches there in 2013, 2017, and 2021, respectively. Shed light by such insights, KG^{2}RAG suggests linking one chunk to other chunks through the overlapping or connected entities that they contain for retrieved chunk expansion.

Specifically, given the retrieved chunks $\mathcal{D}_q \subseteq \mathcal{D}$ and the KG $\mathcal{G} = \{(h, r, t, c) \,|\, c \in \mathcal{D}\}$, we first get the relevant subgraph of $\mathcal{D}_q$ as follows:

$$\mathcal{G}_q^0 = \{(h, r, t, c) \,|\, c \in \mathcal{D}_q\} \subseteq \mathcal{G}. \tag{3}$$

After that, we traverse the m-hop neighborhood of $\mathcal{G}_q$ to get the expanded subgraph $\mathcal{G}_q^m$, which can

Prompt for Triplet Extraction

Instruction:
Extract informative triplets directly from the text following the examples. Do not add any extra words, line breaks, or spaces.
Example 1:
Text: Scott Derrickson (born July 16, 1966) is an American director, screenwriter and producer.
Triplets:
<Scott Derrickson, born in, 1966>,
<Scott Derrickson, nationality, America>,
<Scott Derrickson, occupation, director>,
<Scott Derrickson, occupation, screenwriter>,
<Scott Derrickson, occupation, producer>
Example 2:
Text: A Kiss for Corliss is a 1949 American comedy film directed by Richard Wallace and written by Howard Dimsdale.
Triplets:
<A Kiss for Corliss, year, 1949>,
<A Kiss for Corliss, country, America>,
<A Kiss for Corliss, genre, comedy film>,
<A Kiss for Corliss, director, Richard Wallace>,
<A Kiss for Corliss, writer, Howard Dimsdale>

Target Text: <target text>
Triplets:

Figure 3: The prompt for triplet extraction.

be given as

$$\mathcal{G}_q^m = \text{traverse}(\mathcal{G}, \mathcal{G}_q^0, m), \tag{4}$$

where traverse$(\cdot)$ can be implemented with the breadth-first search (BFS) algorithm, serving as a function that captures all entities in $\mathcal{G}_q^0$, corresponding m-hop neighboring entities, and all edges linking these entities to form an expanded subgraph.

Given the expanded subgraph $\mathcal{G}_q^m$, we can readout all the chunks associated with the graph (i.e., containing facts corresponding to the triplets in this graph) as follows:

$$\mathcal{D}_q^m = \{c \,|\, (h, r, t, c) \in \mathcal{G}_m^q\} \subseteq \mathcal{D}, \tag{5}$$

where $\mathcal{D}_q^m$ is referred to as the expanded chunks.

Discussions Several semantic-based and context-based approaches can also achieve chunk expansion. For example, one can increase the value of k in the aforementioned similarity-based retrieval process, or apply a context window expansion (Jiang et al., 2023) (i.e., when a chunk is retrieved, the chunks within the context window are also recalled together). Different from these approaches, the proposed graph-guided expansion gathers chunks that contain the same or related entities or triplets, without requiring these expanded chunks to have high semantic similarity to the query or to be located around the retrieved chunks. Such a design of graph-guided expansion helps prevent redundancy and excessive homogeneity among the retrieved and expanded chunks, leading to greater diversity and the development of a more comprehensive knowledge network. We provide some empirical evidence to further confirm the effectiveness of the proposed graph-guided expansion in Sec. 3.3.

2.3 KG-based Context Organization

After the KG-enhanced chunk retrieval, KG^{2}RAG incorporates a post-processing stage before response generation of LLMs, motivated by the following two considerations.

Firstly, the number of expanded chunks through the graph-guided expansion is tied to the triplets contained in the expanded subgraph, which can be too large, potentially exceeding the context length and introducing noise that may obscure helpful information. Secondly, inspired by human reading habits and previous studies (Li, 2023; Liu et al., 2024b), providing semantically coherent and well-organized materials as context makes positive impacts on the understanding and generation performance of LLMs. As a result, we propose a KG-based context organization module in KG^{2}RAG, which serves as both a filter and an arranger to meet these requirements.

Serving as a Filter Specifically, we first calculate the semantic similarities between the expanded chunks with the user query, according to Eq. (2). Based on these similarities, the expanded subgraph $\mathcal{G}_q^m$ can be transformed into an undirected weighted graph as follows:

$$\mathcal{U}_q^m = \{(h \leftrightarrow t, \text{rel} : r, \text{src} : c, \text{weight} : s(q, c)) \,|\, (h, r, t, c) \in \mathcal{G}_q^m\}, \tag{6}$$

where $h \leftrightarrow t$ represents an undirected edge, attached with the corresponding relation and the source chunk as meta information. We reuse the semantic similarities calculated in Sec. 2.2 to save computing resources.

Due to the cohesive nature of knowledge, $\mathcal{U}_q^m$ can naturally be divided into p connected components, denoted by $\mathcal{B}_i, 1 \leq i \leq p$, where nodes within each connected component $\mathcal{B}_i$ represent entities from the KG. Note that multiple edges may connect a pair of nodes due to redundant knowledge, which promotes us to generate the maximum spanning tree (MST) of each connected component

for filtering. This can be formulated as

$$\mathcal{T}_i = \text{MST}(\mathcal{B}_i). \tag{7}$$

Through such a filtering process, we retain only the most relevant linking information between entities and eliminate redundant edges, thereby enhancing the informativeness of the retrieved chunks.

Serving as an Arranger With the KG-based context organization module, we aim to integrate the retrieved chunks into intrinsically related and self-consistent paragraphs with the KG as the skeleton.

To achieve this, we provide two representations for each generated MST $\mathcal{T}_i$, including a text representation and a triplet representation. For the text representation, we pick the edge with the highest weight as the root, and concatenate all the chunks linked to the edges using a depth-first search (DFS) algorithm to form a coherent paragraph. For the triplet representation, we concatenate all the edges in the form of $< h, r, t >$ within the MST.

We calculate the relevance scores between MSTs and the user query based on their triplet representations using a cross-encoder reranking function (Xiao et al., 2023):

$$R(q, \mathcal{T}_i) = C(q, \text{conc}(\mathcal{T}_i)), \tag{8}$$

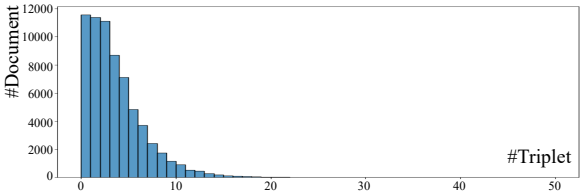
where $C(\cdot)$ is the cross-encoder reranking function and conc$(\cdot)$ is used to obtain the triplet representations. We use triplet representations instead of text representations because triplets provide a concise and structured refinement of the key information associated with the corresponding chunks, allowing relevance matching to focus on key information.

After computing the relevance scores, we sort the MSTs $\{\mathcal{T}_i \mid 1 \leq i \leq p\}$ according to their relevance $\{R(q, \mathcal{T}_i)\}$ to the user query q in descending order. Then, we include their text representations in order until the top-k constraint on the number of chunks has been reached. Finally, these selected chunks are fed into the LLMs along with the user query for response generation.

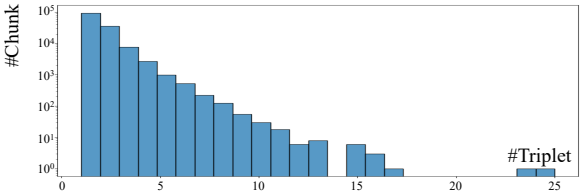
3 Experiments

3.1 Experiment Setup

Datasets We conduct experiments on the benchmark dataset HotpotQA (Yang et al., 2018), where each query can be associated with several materials (e.g., relevant content in Wikipedia) to help in response generation. The HotpotQA dataset consists of two settings, named **HotpotQA-Dist** and



(a) Distribution of documents according to triplet number.



(b) Distribution of chunks according to triplet number.

Figure 4: Statistics of triplet extraction.

HotpotQA-Full. In the distractor setting, a total of ten documents are provided as supporting materials, including all useful knowledge as well as some irrelevant content. In the fullwiki setting, it is required to identify useful knowledge from the entire 66,581 documents extracted from Wikipedia.

For the KG-chunk association, we provide a manual prompt to Llama-3 (Dubey et al., 2024) for extracting entities and relations from the 66,581 documents of HotpotQA, resulting in a total of 211,356 triplets consisting of 98,226 entities and 19,813 relations. Each triplet in the constructed KG is linked to its source chunk. We record the number of triplets extracted from each chunk and document, and plot the corresponding distributions of chunks and documents in Fig. 4, which shows a long-tail phenomenon.

Furthermore, to alleviate the dependence on prior knowledge during the generation process (i.e., the training corpus of LLMs may contain Wikipedia content) and to better demonstrate the effects of RAG, we construct variants of HotpotQA. Specifically, for each entity, we randomly replace it with another entity in the same category, and then update the queries, triplets, and documents accordingly. For example, the entity *Family Guy* can be replaced with *Rick and Morty*, and all instances of *Family Guy* contained in queries, triplets, and documents would be updated to *Rick and Morty*. Therefore, LLMs have to identify and extract relevant content from the documents rather than relying on prior knowledge about *Family Guy* from training data to correctly answer the queries. Note there might generate lots of new triplets such as (*Rick and Morty*, *language*, *French*), as the original tail entity can be also transformed from *En-*

Methods	Hotpot-Dist			Hotpot-Full			Shuffle-Hotpot-Dist			Shuffle-Hotpot-Full		
	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
LLM-only	0.237	0.259	0.234	0.237	0.259	0.234	0.158	0.175	0.158	0.158	0.175	0.158
Semantic RAG	0.617	0.646	0.643	0.528	0.558	0.535	0.508	0.533	0.524	0.422	0.449	0.433
+ Rerank	0.652	0.685	0.665	0.587	0.613	0.603	0.532	0.560	0.546	0.447	0.476	0.456
Hybrid RAG	0.653	0.676	0.655	0.551	0.582	0.558	0.520	0.548	0.534	0.443	0.473	0.446
LightRAG	0.293	0.288	0.480	0.261	0.259	0.364	0.285	0.284	0.404	0.202	0.199	0.293
GraphRAG	0.400	0.408	0.491	0.169	0.157	0.429	0.351	0.365	0.401	0.163	0.155	0.362
KG^{2}RAG	**0.663**	**0.690**	**0.683**	**0.631**	**0.665**	**0.643**	**0.545**	**0.572**	**0.566**	**0.507**	**0.539**	**0.512**

Table 1: Comparisons in terms of response quality between KG^{2}RAG and baselines.

Methods	Hotpot-Dist			Hotpot-Full			Shuffle-Hotpot-Dist			Shuffle-Hotpot-Full		
	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Semantic RAG	0.343	0.206	0.894	0.300	0.178	0.790	0.321	0.201	0.837	0.268	0.167	0.708
+ Rerank	0.357	0.224	**0.932**	0.306	0.197	0.833	0339	0.213	**0.886**	0.286	0.179	0.754
Hybrid RAG	0.354	0.222	0.921	0.302	0.189	0.795	0.334	0.210	0.837	0.279	0.174	0.739
LightRAG	0.234	0.150	0.638	0.132	0.083	0.340	0.227	0.148	0.535	0.116	0.073	0.295
GraphRAG	0.255	0.167	0.594	0.180	0.113	0.470	0.210	0.138	0.482	0.199	0.126	0.510
KG^{2}RAG	**0.436**	**0.301**	0.908	**0.310**	**0.203**	**0.838**	**0.405**	**0.279**	0.840	**0.305**	**0.193**	**0.790**

Table 2: Comparisons in terms of retrieval quality between KG^{2}RAG and baselines.

glish to *French*. The produced variant datasets are denoted by **Shuffle-HotpotQA-Dist** and **Shuffle-HotpotQA-Full**, respectively.

Evaluation Metrics We compare KG^{2}RAG with existing RAG-based methods in terms of response quality and retrieval quality, which can be influenced by both the retrieved chunks and context organization. For retrieval quality, we use the evaluation script provided by HotpotQA to measure the F1 score, precision, and recall between the retrieved chunks and referenced facts. For response quality, we adopt the F1 score, precision, and recall as metrics, comparing the generated responses against ground truth answers.

Baselines In the experiments, we compare KG^{2}RAG with the following baseline methods:

- *LLM-only*, which directly instructs LLMs to generate responses to user queries without any additional retrieval mechanisms.
- *Semantic RAG* (Jiang et al., 2023), which employs a semantic-based approach to retrieve relevant chunks. These chunks are concatenated into the prompt and fed into the LLMs for response generation. For more details, please refer to Sec. 2.2.
- *Hybrid RAG* (Gao et al., 2021), which combines a semantic-based retrieval method with a keyword-based retrieval method (e.g., BM25 (Askari et al., 2023)) for chunk retrieval. The retrieved chunks are subsequently merged through a cross-encoder reranker.
- *GraphRAG* (Edge et al., 2024), which constructs a graph-based index with an LLM. GraphRAG derives a knowledge graph from the source documents and pre-generates community summaries for clustered entities. Given a query, it generates partial responses with each related community summary and aggregates them into the final answer.
- *LightRAG* (Guo et al., 2024), which acts as a lightweight version of GraphRAG. LightRAG extracts entities and relations from the source documents and generates a short description of each entity for retrieval. The retrieved information is unified with the query and fed into the LLM for generation.

For KG^{2}RAG and all baseline methods, we use LLaMA3-8B (Dubey et al., 2024) as the LLM for KG construction and response generation, mxbai-embed-large (Li and Li, 2024) as the embedding model, and bge-reranker-large (Xiao et al., 2023) as the cross-encoder reranker for both Hybrid RAG and KG^{2}RAG. The value of k is set to 10 unless otherwise specified.

3.2 Comparisons and Analyses

Response Quality The comparisons in terms of response quality between KG^{2}RAG and the

	Response Quality			Retrieval Quality			
	F1	Precision	Recall	F1	Precision	Recall	#Avg.
KG^{2}RAG	0.663	0.690	0.683	0.436	0.301	0.908	8.11
w/o organization	0.660	0.678	0.679	0.259	0.153	0.963	16.76
w/o expansion	0.626	0.653	0.645	0.473	0.341	0.842	4.41

Table 3: Experimental results of an ablation study conducted on HotpotQA in the distractor setting.

	Response Quality			Retrieval Quality			
	F1	Precision	Recall	F1	Precision	Recall	#Avg.
KG^{2}RAG	0.545	0.572	0.566	0.405	0.279	0.840	8.09
w/o organization	0.538	0.563	0.560	0.182	0.102	0.962	24.56
w/o expansion	0.474	0.503	0.485	0.511	0.458	0.656	3.82

Table 4: Experimental results of an ablation study conducted on Shuffle-HotpotQA in the distractor setting.

baselines are shown in Table 1. From the table, we can observe that all methods utilizing RAG achieve significant improvements compared to the LLM-only approach, exceeding 29.1% improvements in F1 scores on the original HotpotQA and 26.4% improvements in F1 scores on Shuffle-HotpotQA. Among these RAG-based methods, KG^{2}RAG achieves consistent outperformance, especially in the fullwiki setting and on the Shuffle-HotpotQA dataset.

In the fullwiki setting, a large pool of candidate documents (thousands of times more than in the distractor setting) is provided to LLMs, necessitating high-quality retrieval results and effective context organization. In such a challenging setup, our proposed method KG^{2}RAG achieves at least 8% improvements compared to baselines, demonstrating that KG^{2}RAG enhances chunk retrieval through KG-guided approaches that surpass semantic-based and keyword-based methods. Besides, on the Shuffle-HotpotQA dataset, where LLMs should rely more on RAG rather than prior knowledge, our proposed method achieves at least 2.5% and 6.4% improvements in the distractor and fullwiki settings, respectively.

Retrieval Quality The experimental results are shown in Table 2, which demonstrate that KG^{2}RAG strikes a favorable balance between retrieval precision and recall, highlighting the effectiveness of KG-guided expansion and context organization. In the distractor setting, where irrelevant chunks are limited, our proposed method achieves similar performance in recall but significantly better performance in precision (more than 7.9% and 6.9% on HotpotQA and Shuffle-HotpotQA, respectively). In the fullwiki setting where identifying relevant chunks is more challenging, our proposed method achieves consistent improvements in both precision and recall compared to other RAG-based methods. These results further confirm the effectiveness of KG^{2}RAG in providing high-quality retrieval results with the help of KG.

3.3 Further Discussions

Ablation Study We conduct an ablation study to demonstrate the contributions of different modules in KG^{2}RAG, including KG-guided expansion and KG-based context organization. The experimental results on the HotpotQA and Shuffle-HotpotQA datasets in the distractor setting are shown in Tables 3 and 4, where we also report the average number of retrieved chunks.

From these results, we can observe that using only KG-guided expansion without KG-based context organization (denoted by "w/o organization" in the table), KG^{2}RAG achieves similar performance in terms of answer quality but significantly worse retrieval quality. The reason is that, without the KG-based context organization module, the number of retrieved chunks can be noticeably larger, potentially containing irrelevant chunks that do not contribute positively to performance but consume additional tokens. These findings confirm the contribution of the KG-based context organization module in effectively selecting and organizing retrieved chunks to preserve relevant information.

With only the KG-based context organization module (denoted by "w/o expansion" in the table), KG^{2}RAG achieves high retrieval precision and F1 score with a significantly smaller number of chunks, but fails to provide better responses, as

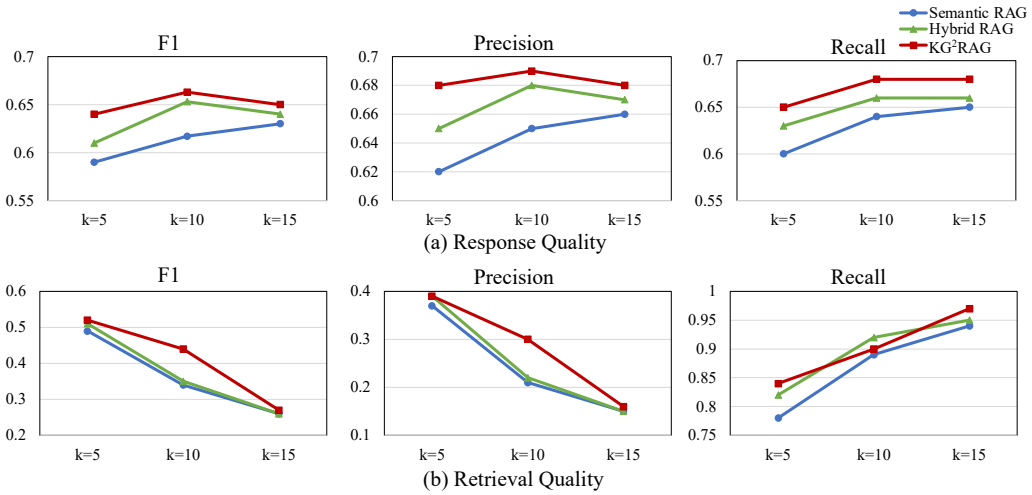


Figure 5: Experimental results with varying top-k on HotpotQA in distractor setting.

	Response Quality			Retrieval Quality			
	F1	Precision	Recall	F1	Precision	Recall	#Avg.
$m = 1$	0.663	0.690	0.683	0.436	0.301	0.908	8.11
$m = 2$	0.656	0.681	0.674	0.420	0.291	0.917	8.53
$m = 3$	0.658	0.678	0.675	0.421	0.284	0.924	8.19

Table 5: Experimental results on HotpotQA in the distractor setting with varying m.

some necessary chunks may not be retrieved using only semantic-based approaches. These results confirm the importance of the KG-guided expansion module in successfully leveraging KG to capture fact-level relationships between chunks and retrieve key information that might be missed by semantic-based approaches.

Performance w.r.t. Varying k We conduct experiments with varying top-k values on HotpotQA in the distractor setting. The experimental results are shown in Fig. 5. From these figures, we can observe that KG^{2}RAG maintains superior performance compared to baselines with different k. When k is set to a suitable value (e.g., 5 or 10), KG^{2}RAG ensures the efficient retrieval of high-quality chunks, thereby providing coherent and contextually consistent contexts for generating high-quality responses.

However, when k is set to a too large value (e.g., 15), although the retrieval recall significantly improves, the quality of the generated responses does not increase proportionally, which indicates simply increasing the number of chunks cannot always result in a better retrieval recall ratio and response quality. KG^{2}RAG exhibits the least sensitivity to the hyperparameter k compared to baselines, which makes the RAG process robust.

Performance w.r.t. Varying m In KG^{2}RAG, m serves as the hyperparameter for graph expansion, balancing the trade-off between retrieval precision and recall. We set the m-hop value to 1 in the previous experiments. To further explore the effects of m, we conduct experiments with varying m on HotpotQA dataset. The results are shown in Table 5. These results indicate that setting $m = 1$ is appropriate for the experiments, and KG^{2}RAG shows low sensitivity to the hyperparameter m.

Robustness Analysis To further confirm the robustness of KG^{2}RAG with quality-limited KGs, we randomly drop 5% or 10% of the triplets from the constructed KG, and show the experimental results in Table 6. The results demonstrate that KG^{2}RAG maintains robust performance even with quality limitations and outperforms the baselines.

4 Related Work

Retrieval-augmented Generation To address the issues of hallucinations (Xu et al., 2024b; Liu et al., 2024a) due to a lack of corresponding knowledge or containing outdated knowledge, retrieval-augmented generation (RAG) (Gao et al., 2023; Fan et al., 2024) has been proposed for retrieving relevant chunks from a pool of candidate documents to assist LLM generation.

	Response Quality			Retrieval Quality		
	F1	Precision	Recall	F1	Precision	Recall
Hybrid RAG	0.653	0.676	0.655	0.354	0.222	0.921
KG^{2}RAG	0.663	0.690	0.683	0.436	0.301	0.908
-5%	0.662	0.681	0.676	0.434	0.306	0.898
-10%	0.654	0.688	0.682	0.432	0.305	0.890

Table 6: Experimental results on HotpotQA in the distractor setting with triplets dropped.

In a typical RAG system (Lewis et al., 2020), the documents are first segmented into chunks based on lengths and structures, and then encoded with an embedding model (Nussbaum et al., 2024; Li and Li, 2024) and indexed for efficient retrieval. Inspired by the idea of sliding windows (Jiao, 2006), sentence window retrieval (Jiang et al., 2023; Eibich et al., 2024) fetches the neighboring chunks around the retrieved chunks and concatenates them into a single larger chunk for context enrichment. However, sentence window retrieval only considers the physical proximity of text chunks within the same document. Different from existing studies, KG^{2}RAG performs retrieval expansion based on factual associations among chunks that may be across multiple documents.

Reranking (Ampazis, 2024; Glass et al., 2022) is a critical technique in information retrieval (Grems, 1962; Kuo et al., 2024). In RAG systems, feeding the retrieved chunks along with the queries into a deep learning-based cross-encoder (Xiao et al., 2023) can measure the semantic relevance more precisely, thereby enhancing both the retrieval and generation quality. KG^{2}RAG organizes the retrieved chunks into paragraphs with KGs as the skeleton, allowing a fine-grained measurement of paragraph-level relevance to queries.

LLMs with Knowledge Graph LLM (Li et al., 2024; Ren et al., 2024) is one of the most representative achievements of contemporary artificial intelligence (AI). KGs (Ji et al., 2022), as graph-structured relational databases, serve as a crucial data infrastructure for AI applications. Research indicates that LLMs have the potential to address tasks related to KGs, such as knowledge graph completion (Liu et al., 2024c) and knowledge graph question answering (Sen et al., 2023).

Recently, the research community begins to explore how KGs can be used to enhance the generation capability of LLMs (Wang et al., 2024a; Edge et al., 2024; Xu et al., 2024a). For example, KGP (Wang et al., 2024a) constructs a document KG consisting of page and passage nodes, and links passage nodes with TF-IDF. The document KG is employed for retrieval expansion. The document KG constructed by KGP is based on sentence-level text similarity, which essentially functions similarly to simply expanding the context window. GraphRAG (Edge et al., 2024) targets at query-focused summarization tasks. GraphRAG extracts KGs automatically from the document base with an LLM and analyzes the semantic structure of the dataset before querying, by splitting the KG from different level and detecting linked nodes hierarchically. Different from previous studies, KG^{2}RAG aims to enhance RAG with the fact-level structure and factual knowledge of KGs.

5 Conclusion

In this paper, we propose KG^{2}RAG, a novel framework designed to enhance the performance of RAG through the integration of KGs. We introduce linkages between chunks and a specific KG, which help in providing fact-level relationships among these chunks. Consequently, KG^{2}RAG suggests performing the KG-guided chunk expansion and the KG-based context organization based on seed chunks retrieved by semantic-based retrieval approaches. Through these processes, the retrieved chunks become diverse, intrinsically related, and self-consistent, forming well-organized paragraphs that can be fed into LLMs for high-quality response generation. We compare KG^{2}RAG with existing RAG-based approaches, demonstrating its superior performance in both response quality and retrieval quality. An ablation study is also conducted to further confirm the contributions of KG-guided chunk expansion and KG-based context organization, indicating that these two modules collaboratively enhance the effectiveness of KG^{2}RAG.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62272219).

Limitations

Retrieval-augmented generation (RAG) is a systematic engineering framework that can be refined from multiple perspectives, including query rewriting (Xiao et al., 2023), retrieval optimization (Eibich et al., 2024), multi-turn dialogue (Yao et al., 2023) and so on (Gao et al., 2023). KG^{2}RAG only focuses on the part of retrieval optimization and aims to perform KG-guided retrieval expansion and KG-based context organization to enhance RAG with the structured factual knowledge from KGs, without optimizing other modules. However, the proposed KG^{2}RAG is orthogonal and compatible with the aforementioned modules. In the future, we will develop KG^{2}RAG into a plug-and-play tool that can be easily integrated with other approaches, thereby better facilitating the research community.

References

Nicholas Ampazis. 2024. Improving RAG quality for large language models with topic-enhanced reranking. In *AIAI*, volume 712, pages 74–87, Corfu, Greece. Springer.

Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the BM25 score as text improves BERT-based re-rankers. *CoRR*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A nucleus for a web of open data. In *ISWC*, volume 4825, pages 722–735, Busan, Korea. Springer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*, Virtual. Curran Associates.

Marco Calamo, Francesca De Luzi, Mattia Macrì, Tommaso Mencattini, and Massimo Mecella. 2023. CICERO: A GPT2-based writing assistant to investigate the effectiveness of specialized LLMs' applications in e-justice. In *ECAI*, volume 372, pages 3196–3203, Kraków, Poland. IOS.

Wei Cheng, Yuhan Wu, and Wei Hu. 2024. Dataflow-guided retrieval augmentation for repository-level code completion. In *ACL*, pages 7957–7977, Bangkok, Thailand. ACL.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The Llama 3 herd of models. *CoRR*.

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. *CoRR*.

Matous Eibich, Shivay Nagpal, and Alexander Fred-Ojala. 2024. ARAGOG: Advanced RAG output grading. *CoRR*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In *KDD*, pages 6491–6501, Barcelona, Spain. ACM.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement lexical retrieval model with semantic residual embeddings. In *ECIR*, volume 12656, pages 146–160, Glasgow, UK. Springer.

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

Michael R. Glass, Gaetano Rossiello, Md. Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *NAACL*, pages 2701–2715, Seattle, WA, USA. ACL.

Mandalay Grems. 1962. A survey of languages and systems for information retrieval. *Commun. ACM*, 5(1):43–46.

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

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *EMNLP*, pages 7969–7992, Singapore. ACL.

Yishan Jiao. 2006. Maintaining stream statistics over multiscale sliding windows. *ACM Trans. Database Syst.*, 31(4):1305–1334.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pages 1601–1611, Vancouver, Canada. ACL.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *CoRR*.

Tzu-Lin Kuo, Tzu-Wei Chiu, Tzung-Sheng Lin, Sheng-Yang Wu, Chao-Wei Huang, and Yun-Nung Chen. 2024. A survey of generative information retrieval. *CoRR*.

Patrick S. H. 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 *NeurIPS*, Virtual. Curran Associates.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Yiguan Lin, Bin Xu, Ren Bowen, Chong Feng, Yang Gao, and Heyan Huang. 2024. Fundamental capabilities of large language models and their applications in domain scenarios: a survey. In *ACL*, pages 11116–11141, Bangkok, Thailand. ACL.

Xianming Li and Jing Li. 2024. AoE: Angle-optimized embeddings for semantic textual similarity. In *ACL*, pages 1825–1839, Bangkok, Thailand. ACL.

Yinheng Li. 2023. A practical survey on zero-shot prompt design for in-context learning. In *RANLP*, pages 641–647, Varna, Bulgaria. INCOMA.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *CoRR*.

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

Yang Liu, Xiaobin Tian, Zequn Sun, and Wei Hu. 2024c. Finetuning generative large language models with discrimination instructions for knowledge graph completion. In *ISWC*, Baltimore, MD, USA. Springer.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *CoRR*.

Laurent Mombaerts, Terry Ding, Adi Banerjee, Florian Felice, Jonathan Taws, and Tarik Borogovac. 2024. Meta knowledge for retrieval augmented large language models. *CoRR*.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *CoRR*.

Anupam Purwar and Rahul Sundar. 2023. Keyword augmented retrieval: Novel framework for information retrieval integrated with speech interface. In *AIML-Systems*, pages 58:1–58:5, Bangalore, India. ACM.

Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh V. Chawla, and Chao Huang. 2024. A survey of large language models for graphs. In *KDD*, pages 6616–6626, Barcelona, Spain. ACM.

Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. In *NLRSE*, pages 1–8, Toronto, Canada. ACL.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL*, pages 641–651, New Orleans, Louisiana, USA. ACL.

Xiaobin Tian, Zequn Sun, and Wei Hu. 2024. Generating explanations to understand and repair embedding-based entity alignment. In *ICDE*, pages 2205–2217, Utrecht, Netherlands. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa F. Siu, Ruiyi Zhang, and Tyler Derr. 2024a. Knowledge graph prompting for multi-document question answering. In *AAAI*, pages 19206–19214, Vancouver, Canada. AAAI.

Yujing Wang, Hainan Zhang, Liang Pang, Binghui Guo, Hongwei Zheng, and Zhiming Zheng. 2024b. MaFeRw: Query rewriting with multi-aspect feedbacks for retrieval-augmented large language models. *CoRR*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged resources to advance general Chinese embedding. *CoRR*.

Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024a. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *SIGIR*, pages 2905–2909, Washington DC, USA. ACM.

Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *CoRR*.

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. In *EMNLP*, pages 2369–2380, Brussels, Belgium. ACL.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, Kigali, Rwanda. OpenReview.net.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *ACL*, Berlin, Germany. ACL.

Wenhao Yu. 2022. Retrieval-augmented generation across heterogeneous knowledge. In *NAACL*, pages 52–58, Seattle, WA, USA. ACL.

Wenzheng Zhao, Yuanning Cui, and Wei Hu. 2023. Improving continual relation extraction by distinguishing analogous semantics. In *ACL*, pages 1162–1175, Toronto, Canada. ACL.

Angelo Ziletti and Leonardo D'Ambrosi. 2024. Retrieval augmented text-to-SQL generation for epidemiological question answering using electronic health records. In *NAACL*, pages 47–53, Mexico City, Mexico. ACL.

A Additional Experimental Results

A.1 Results on More Datasets

We conduct additional experiments on two different datasets to confirm the effectiveness and generality of KG^{2}RAG in various scenarios.

As shown in Table 7, KG^{2}RAG maintains superiority on the widely-used MuSiQue dataset (Trivedi et al., 2022) in response F1 score, response exact match (EM) rate, and retrieval F1 score.

Methods	Response F1	Response EM	Retrieval F1
LLM-only	0.075	0.025	-
Semantic RAG	0.367	0.248	0.365
+ Rerank	0.380	0.249	0.372
Hybrid RAG	0.380	0.250	0.364
LightRAG	0.248	0.170	0.289
GraphRAG	0.231	0.156	0.273
KG^{2}RAG	**0.419**	**0.303**	**0.451**

Table 7: Comparison results on MuSiQue.

Also, we conduct experiments on the typical long-context dataset TriviaQA (Joshi et al., 2017). On average, each document in this dataset contains 2,895 words. For comparison, documents in HotpotQA have an average of 917 words. The experimental results are shown in Table 8, which confirms the effectiveness of KG^{2}RAG in a typical long-context setting.

Methods	Response F1	Response Prec.	Response Recall
LLM-only	0.182	0.303	0.144
Semantic RAG	0.259	0.413	0.211
+ Rerank	0.265	0.409	0.235
Hybrid RAG	0.262	0.415	0.229
LightRAG	0.118	0.157	0.237
GraphRAG	0.127	0.193	0.225
KG^{2}RAG	**0.273**	**0.416**	**0.240**

Table 8: Comparison results on TriviaQA.

A.2 Efficiency Analysis

We compare the KG construction cost of KG^{2}RAG with two other KG-enhanced RAG approaches: LightRAG (Guo et al., 2024) and GraphRAG (Edge et al., 2024). The results, as summarized in Table 9, demonstrate that KG^{2}RAG is more efficient in terms of token cost, the number of LLM calls, and time cost.

We calculate the average retrieval time and generation time of KG^{2}RAG compared to LightRAG and GraphRAG. The results in Table 10 indicate that KG^{2}RAG requires less time for both retrieval and response generation than LightRAG and GraphRAG, and is very close to Semantic RAG. Note that KG^{2}RAG might need a lower time for response generation using a condensed and informative context as input.

	#Input tokens	#Output tokens	#LLM calls	Extraction time
LightRAG	1,269	381	1	3s
GraphRAG	2,791	629	5	6s
KG^{2}RAG	561	22	1	1s

Table 9: Comparison of average LLM and time cost per chunk during KG construction.

Method	Avg. retrieval time	Avg. generation time
Semantic RAG	21ms	2,500ms
LightRAG	40ms	5,600ms
GraphRAG	42ms	5,500ms
KG^{2}RAG	25ms	2,300ms

Table 10: Comparison of average retrieval and generation time per query.