

KG-Infused RAG: Augmenting Corpus-Based RAG with External Knowledge Graphs

Dingjun Wu¹, Yukun Yan^{1,*}, Zhenghao Liu^{2,*}, Zhiyuan Liu¹, Maosong Sun¹
¹Tsinghua University ²Northeastern University

Abstract

Retrieval-Augmented Generation (RAG) improves factual accuracy by grounding responses in external knowledge. However, existing methods typically rely on a single source, either unstructured text or structured knowledge. Moreover, they lack cognitively inspired mechanisms for activating relevant knowledge. To address these issues, we propose **KG-Infused RAG**, a framework that integrates KGs into RAG systems to implement *spreading activation*, a cognitive process that enables concept association and inference. KG-Infused RAG retrieves KG facts, expands the query accordingly, and enhances generation by combining corpus passages with structured facts, enabling interpretable, multi-source retrieval grounded in semantic structure. We further improve KG-Infused RAG via preference learning on sampled key stages in the pipeline. Experiments on five QA benchmarks show that KG-Infused RAG consistently outperforms vanilla RAG (by 3.8% to 13.8%). Additionally, when integrated into Self-RAG, KG-Infused RAG brings further performance gains, demonstrating its effectiveness and versatility as a plug-and-play enhancement module for corpus-based RAG methods.¹

1 Introduction

Large language models (LLMs) have shown strong performance in question answering tasks [1, 10, 30, 5], yet they remain susceptible to factual errors and hallucinations [19, 21, 8]. Retrieval-Augmented Generation (RAG) mitigates these issues by grounding generation in external sources such as text corpus or knowledge graphs (KGs) [17, 25, 2, 3, 28]. However, existing RAG methods typically rely on a single retrieval source: corpus-based methods retrieve passages [8, 22, 25, 2], while KG-based methods extract facts from an external graph [3, 28]. This single-source constraint limits their ability to integrate complementary evidence from both unstructured and structured knowledge. Recent hybrid approaches attempt to bridge this gap by constructing ad-hoc KGs from text using LLMs [6, 11, 32], but they often suffer from high computation costs and potential factual inaccuracies due to imperfect entity and relation extraction.

More fundamentally, current RAG methods lack the ability to retrieve knowledge by activating semantically related concepts based on prior knowledge structures. This cognitive process, known as **spreading activation** [4], originates from cognitive psychology and **explains how humans access relevant information by traversing semantic associations**, even when it is not explicitly mentioned. In RAG, this mechanism can be implemented through the use of KGs, where a seed concept (e.g., an entity in the question) activates connected nodes (e.g., related entities or facts), guiding the retrieval toward semantically relevant evidence. This enables richer knowledge coverage, improves recall of long-tail or implicit facts, and provides a more interpretable reasoning path compared to relying solely on surface-form matching or parametric memory in LLMs.

*Corresponding authors: yanyk.thu@gmail.com, liuzhenghao@mail.neu.edu.cn.

¹Our code and data are available at <https://github.com/thunlp/KG-Infused-RAG>

To efficiently implement spreading activation, we leverage existing human-curated KGs (e.g., Wikidata), which provide high-quality, structured, and long-tail factual knowledge. Unlike prior work that constructs ad-hoc KGs [6, 11, 32] from text using LLMs—often prone to factual errors and computational costs—those curated KGs offer a stable and cost-effective foundation for semantically guided retrieval. Using KGs as external memory networks enable RAG systems to traverse meaningful concept associations, improving both recall and interpretability of the retrieved evidence.

Based on this insight, we introduce **Knowledge Graph Infused Retrieval-Augmented Generation (KG-Infused RAG)**, a framework that integrates KGs into the RAG pipeline to implement spreading activation. KG-Infused RAG consists of three stages: (1) retrieving relevant KG facts via spreading activation; (2) expanding the query with these facts to improve corpus retrieval; (3) generating answers conditioned on passages enriched with KG facts. This design enables multi-source retrieval that combines KG facts and corpus passages, leveraging KG structure to guide spreading activation. By aligning retrieval with human-like spreading activation, KG-Infused RAG supports more accurate, interpretable, and fact-grounded QA. Moreover, it can be integrated into corpus-based RAG methods (e.g., Self-RAG) to further boost performance.

To further improve KG-Infused RAG, we apply preference learning on sampled key pipeline stages for targeted tuning. Experiments on multiple benchmarks demonstrate consistent performance gains and show that KG-Infused RAG significantly enhances retrieval quality compared to vanilla RAG.

Our main contributions are summarized as follows:

1. We introduce KG-Infused RAG, a framework that incorporates human-curated knowledge graphs (KGs) into RAG to emulate spreading activation for semantically guided retrieval and generation.
2. KG-Infused RAG supports multi-source retrieval and can be integrated into corpus-based RAG methods (e.g., Self-RAG) as a plug-and-play module.
3. We develop a data sampling method to improve preference learning and model performance.
4. Experiments on five QA benchmarks show that KG-Infused RAG consistently improves performance, with absolute gains of 3.8% to 13.8% over vanilla RAG and substantial improvements when combined with Self-RAG.

2 Related Work

2.1 Multi-Turn Retrieval in Retrieval-Augmented Generation

Single-turn retrieval followed by generation often yields insufficient evidence for complex questions, such as multi-hop QA requiring LLMs to perform multi-step retrieval and reasoning [8, 25].

Query Rewriting and Expansion. Query rewriting reformulates an original query into a clearer or more effective one, especially when it is ambiguous or complex [18, 20]. Self-Ask [22] and IRCOT [25] enhance multi-hop reasoning by decomposing complex queries into sub-questions for iterative retrieval. In parallel, query expansion enriches the original query with informative content, such as keywords or pseudo-documents, to improve retrieval [26, 14, 16]. HyDE [7] generates hypothetical documents from the query to improve retrieval relevance. In contrast to these methods that rely solely on an LLM’s parametric knowledge, KG-Infused RAG incorporates factual signals from a KG during query expansion. Grounded in relevant facts, KG-Infused RAG enables more targeted and evidence-grounded retrieval.

Evidence Aggregation and Enhancement. Recent RAG methods extend single-turn retrieval to multi-turn, collecting richer and more targeted evidence [8]. IRCOT [25] interleaves retrieval with chain-of-thought reasoning steps, while FLARE [15] and Self-RAG [2] dynamically decide when and what to retrieve based on generation confidence or intermediate outputs. These methods enable incremental evidence accumulation during generation. Similarly, KG-Infused RAG gathers additional evidence through multi-turn spreading activation and KG-based query expansion, followed by integration and refinement of retrieved information.

2.2 Knowledge Graphs Augmented Question Answering

Knowledge graphs (KGs) provide structured representations of entities and their relations, offering signals beyond plain text. This structured knowledge helps reduce hallucinations, improves reasoning

enabling more effective retrieval and integration of KG facts and textual passages. The overall process is illustrated in Figure 1, consisting of three main modules:

- 1) **KG-Guided Spreading Activation.** Starting from query-relevant entities, we iteratively expand semantically related triples to build a task-specific subgraph \mathcal{G}_q , which is then summarized for downstream use.
 - 2) **KG-Based Query Expansion (KG-Based QE).** The LLM leverages both the original query and the expanded KG subgraph to generate an expanded query, improving retrieval over the corpus \mathcal{D}_c .
 - 3) **KG-Augmented Answer Generation (KG-Aug Gen).** The LLM generates the answer by integrating retrieved passages and KG summaries, ensuring fact-grounded and interpretable generation.
- Details of all prompts used in the framework are in Appendix D.1.

3.3 Preparation: KG Preprocessing

We prepare our KG \mathcal{G} by extracting and preprocessing triples from Wikidata5M [27], a large-scale knowledge graph dataset derived from Wikidata and Wikipedia. Wikidata5M spans multiple domains, making it suitable for open-domain QA tasks. The KG used in KG-Infused RAG is defined as:

$$\mathcal{G} = \{\langle e, r, d \rangle \mid e \in \mathcal{E}, r \in \mathcal{R}, d \in \mathcal{D}_e\} \quad (2)$$

where \mathcal{E} , \mathcal{R} , and \mathcal{D}_e denote the set of entities, relations, and entity descriptions, respectively. We preprocess Wikidata5M to meet our requirements for the KG, resulting in Wikidata5M-KG with approximately ~21M triples. The preprocessing details and statistics are in Appendix B.

3.4 KG-Guided Spreading Activation

This stage constructs a query-specific subgraph \mathcal{G}_q by simulating spreading activation over the KG, starting from query-relevant entities and propagating through related facts. The retrieved structured knowledge complements corpus evidence and supports downstream retrieval and generation.

Seed Entities Initialization. We first retrieve the top- k_e entities most relevant to the query q as the starting points for spreading activation over the KG. Each entity e_i is associated with a textual description $d_i \in \mathcal{D}_e$. The similarity between the query and each entity is computed via inner product:

$$\text{sim}(q, d_i) = q \cdot d_i \quad (3)$$

We then select the top- k_e entities whose descriptions have the highest similarity scores:

$$E_q^0 = \{e_i \in \mathcal{E} \mid d_i \in \text{TopK}_{k_e}(\mathcal{D}_e; \text{sim}(q, d_i))\} \quad (4)$$

Here, $\text{TopK}_{k_e}(\mathcal{D}_e; \text{sim})$ denotes the set of k_e descriptions in \mathcal{D}_e with the highest similarity to q , and e_i is the entity associated with d_i . The resulting entity set $E_q^0 = \{e_1, e_2, \dots, e_{k_e}\}$ initializes the spreading activation process over the KG.

Iterative Spreading Activation. Starting from the seed entities E_q^0 , we perform iterative spreading activation over \mathcal{G} , where each round i expands from the currently activated entities E_q^i to retrieve new query-relevant triples. Each round consists of:

1. **Triple Selection.** For each entity in the current set E_q^i , we retrieve its 1-hop neighbors from \mathcal{G} . An LLM is prompted to select a subset of triples relevant to the query q . The selected triples for round i are denoted as $\{(e, r, e')_i\}$, where e , r , and e' denote the head entity, relation, and tail entity in a triple respectively.
2. **Activation Memory Construction and Updating.** We maintain an **Activation Memory** $\mathcal{M} = \{\mathcal{G}_q^{act}, E_q^{act}\}$ to track the multi-round activation process, where the subscript *act* denotes the activation stage. Here, \mathcal{G}_q^{act} is the accumulated subgraph of all query-relevant triples retrieved so far, and E_q^{act} is the set of entities activated up to the current round. This memory helps aggregate knowledge while preventing redundant reactivation of entities. At each round i , given the newly retrieved triples \mathcal{G}_q^i and the associated entity E_q^i , we update the memory as follows:

$$\mathcal{G}_q^{act} \leftarrow \mathcal{G}_q^{act} \cup \mathcal{G}_q^i \quad \text{and} \quad E_q^{act} \leftarrow E_q^{act} \cup E_q^i \quad (5)$$

where \cup denote the set union operation.

3. Next-Round Activation Entities. At each round i , we determine the next-round entity set E_q^{i+1} from the current triples $\mathcal{G}_q^i = \{(e, r, e')_i\}$ by extracting their tail entities $\{e'\}$ as candidates for further activation. To prevent revisiting entities, we exclude those already in the activated entity set E_q^{act} . Formally,

$$E_q^{i+1} = \left(\bigcup_{(e, r, e') \in \mathcal{G}_q^i} \{e'\} \right) \setminus E_q^{act} \quad (6)$$

where the set difference \setminus excludes previously activated entities, ensuring E_q^{i+1} contains only newly activated entities to guide the next activation round.

The activation process terminates when either the predefined maximum number of activation rounds k is reached or no new entities remain, i.e., $E_q^{i+1} = \emptyset$. This iterative expansion progressively retrieves query-relevant triples from \mathcal{G} , covering paths from 1-hop to k -hop neighborhoods.

Expanded Subgraph Summarization. We prompt the LLM to summarize the expanded KG subgraph \mathcal{G}_q^{act} , accumulated in the Activation Memory \mathcal{M} , into a natural language summary $\mathcal{S}_{\mathcal{G},q}^{act}$. This summarization serves two purposes: **(1)** It condenses discrete graph-structured facts into a coherent natural language narrative, revealing semantic paths and concept interactions underlying the query. **(2)** It converts structured knowledge into natural language, making it more accessible and usable for the LLM in subsequent stages.

3.5 KG-Based Query Expansion

The goal of this stage is to generate an expanded query q' that complements the original query q by incorporating structured knowledge retrieved from the KG. The expansion broadens retrieval coverage and enhances the relevance of corpus evidence.

We prompt the LLM with both the original query q and the KG subgraph summary $\mathcal{S}_{\mathcal{G},q}^{act}$ constructed via spreading activation. Conditioned on the prompt, the model performs associative reasoning to transform the query—either simplifying it based on retrieved facts (e.g., replacing intermediate entities), or extending it with new, KG-grounded content inferred through LLM knowledge. Formally,

$$q' = \text{LLM}_{\text{prompt}}(q, \mathcal{S}_{\mathcal{G},q}^{act}) \quad (7)$$

We then perform dual-query retrieval over the corpus using both q and q' , and merge the results:

$$\mathcal{D}_{c,(q,q')} = \mathcal{D}_{c,q} \cup \mathcal{D}_{c,q'} \quad (8)$$

where $\mathcal{D}_{c,q}$ and $\mathcal{D}_{c,q'}$ denote the passage sets retrieved by q and q' , respectively.

3.6 KG-Augmented Answer Generation

This stage integrates corpus and KG evidence to generate the final answer. We enhance retrieved passages with structured facts to support more informed and accurate reasoning.

Passage Note Construction. We first prompt the LLM to summarize the retrieved passages $\mathcal{D}_{c,(q,q')}$, producing a query-focused passage note $\mathcal{S}_{\mathcal{D},q}$ that distills key information relevant to answering q :

$$\mathcal{S}_{\mathcal{D},q} = \text{LLM}_{\text{prompt}}(q, \mathcal{D}_{c,(q,q')}) \quad (9)$$

KG-Guided Knowledge Augmentation (KG-Guided KA). To incorporate structured knowledge, we prompt the LLM to augment the passage note $\mathcal{S}_{\mathcal{D},q}$ with the KG subgraph summary $\mathcal{S}_{\mathcal{G},q}^{act}$, yielding a fact-enhanced note $\mathcal{S}_q^{\text{final}}$:

$$\mathcal{S}_q^{\text{final}} = \text{LLM}_{\text{prompt}}(\mathcal{S}_{\mathcal{D},q}, \mathcal{S}_{\mathcal{G},q}^{act}) \quad (10)$$

Answer Generation. Finally, the LLM generates the answer conditioned on the original query q and the enriched note $\mathcal{S}_q^{\text{final}}$:

$$\text{answer} = \text{LLM}_{\text{prompt}}(q, \mathcal{S}_q^{\text{final}}) \quad (11)$$

This design enables the model to generate answers grounded not only in textual evidence but also in structured KG facts, facilitating more comprehensive, accurate, and interpretable reasoning.

3.7 Training: Data Construction and DPO Optimization

To improve the instruction-following ability at key stages of KG-Infused RAG’s pipeline, we construct a simple preference-based training dataset by sampling a single key stage from the pipeline and train the generator used in multiple stages using Direct Preference Optimization (DPO) [23].

DPO Data Construction. We use an advanced LLM, GPT-4o-mini, to construct a training dataset for DPO based solely on outputs from a single stage—KG-guided knowledge augmentation. Specifically, we sample examples from the 2WikiMultiHopQA [12] training set and collect intermediate outputs from this stage to form the dataset D_{KA} . For each input x , we generate multiple candidate outputs using diverse decoding strategies, and prompt GPT-4o-mini to identify the best and worst outputs. We construct preference-labeled triples (x, y^-, y^+) . The resulting dataset is denoted as:

$$D_{DPO} = \{(x, y^-, y^+) \mid (x, y^-, y^+) \in D_{KA}, y^+ \succ y^-\} \quad (12)$$

DPO Training. To train the model to prefer higher-quality knowledge-augmented outputs, we apply the DPO objective. Given a preference-labeled triple (x, y^-, y^+) from D_{DPO} , the DPO loss encourages the policy model π_θ to assign higher likelihood to the preferred output y^+ , while maintaining consistency with a fixed reference model π_{ref} . The training objective is defined as:

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(x, y^+, y^-) \sim D} \left[\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} - \log \frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)} \right) \right) \right] \quad (13)$$

where π_θ denotes the trainable model, and π_{ref} is a fixed reference model. Additional training details are provided in Appendix E.

4 Experiment Setup

4.1 Datasets & Evaluation

Multi-Hop QA Task. To evaluate our method in complex QA scenarios, we conduct experiments on four challenging multi-hop QA datasets: HotpotQA [31], 2WikiMultiHopQA (2WikiMQA) [12], MuSiQue [24], and Bamboogle [22]. For HotpotQA, 2WikiMQA, and MuSiQue, we use the subsets provided by Trivedi et al. [25]. These datasets require models to perform complex multi-hop reasoning across multiple pieces of evidence, making them ideal for evaluating our method’s ability to integrate and reason over knowledge. We report F1 Score (F1), Exact Match (EM), Accuracy (Acc), and their average (Avg) as evaluation metrics. The summary of datasets and evaluation is in Appendix C.1.

Commonsense QA Task. To assess the generalizability of our method to simpler QA settings, we evaluate it on the StrategyQA dataset [9], a commonsense QA task requiring binary Yes/No answers. Unlike multi-hop QA tasks that demand explicit retrieval and integration of multiple evidence pieces, StrategyQA emphasizes implicit reasoning based on implicit commonsense knowledge. From the RAG perspective, it poses a lower retrieval challenge. We randomly sample 500 examples from the test set for evaluation, with accuracy as the evaluation metric.

4.2 Baselines

Standard Baselines. We compare our method against three standard baselines: (1) No-Retrieval (NoR) directly feeds the input query into an LLM without retrieving any external knowledge. (2) Vanilla RAG retrieves the top- k_p passages from the corpus based on the raw query q . (3) Vanilla Query Expansion (Vanilla QE) prompts an LLM to generate an expanded query q' without incorporating KG facts. Both the original and expanded queries are used to retrieve top- $k_p/2$ passages each. For a fair comparison, both Vanilla RAG and Vanilla QE adopt our passage note construction step to form passage notes $\mathcal{S}_{\mathcal{D}, q}$, which are then used by the LLM to generate answers.

Plug-in Baselines. To evaluate the generalizability of our method as a plug-and-play module, we integrate it into two strong adaptive RAG methods: Self-RAG [2] and Self-Ask [22].

For the Self-RAG baseline, since all evaluation datasets are short-form generation tasks, we follow the official setting where multiple passages are retrieved only once for generation. **To apply KG-Infused RAG as a plug-in to Self-RAG**, we first perform KG-Based QE after KG-Guided Spreading

Table 1: **Overall results (%)** on multiple datasets. **Bold** numbers denote the best-performed results. “w/ n -Round” denotes KG-Infused RAG with n activation rounds.

Method	Multi-hop															Short-form	
	HotpotQA				2WikiMQA				MuSiQue				Bamboogle				StrategyQA
	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	
Qwen2.5-7B																	
NoR	19.8	25.8	18.0	21.2	23.4	26.8	22.0	24.1	4.0	11.6	3.4	6.3	10.4	16.6	8.0	11.7	64.4
Vanilla RAG	35.0	41.9	31.0	36.0	30.6	33.5	27.2	30.4	5.2	11.5	3.6	6.8	22.4	30.1	21.6	24.7	68.6
Vanilla QE	35.0	41.8	30.8	35.9	28.6	31.5	25.4	28.5	5.4	11.1	2.2	6.2	20.8	28.6	18.4	22.6	72.2
KG-Infused RAG (w/ 2-Round)	39.0	44.7	34.4	39.4	44.8	44.5	36.0	41.8	10.8	16.7	7.6	11.7	26.4	34.1	24.0	28.2	68.2
+DPO (w/ 2-Round)	37.2	43.9	31.0	37.4	45.6	45.8	35.8	42.4	12.8	19.2	8.6	13.5	36.8	41.8	31.2	36.6	72.8
Δ KG-Infused RAG \rightarrow Vanilla RAG	4.0 \uparrow	2.8 \uparrow	3.4 \uparrow	3.4 \uparrow	14.2 \uparrow	11.0 \uparrow	8.8 \uparrow	11.4 \uparrow	5.6 \uparrow	5.2 \uparrow	4.0 \uparrow	4.9 \uparrow	4.0 \uparrow	4.0 \uparrow	2.4 \uparrow	3.5 \uparrow	-0.4 \downarrow
Δ KG-Infused RAG+DPO \rightarrow Vanilla RAG	2.2 \uparrow	2.0 \uparrow	0.0 \uparrow	1.4 \uparrow	15.0 \uparrow	12.3 \uparrow	8.6 \uparrow	12.0 \uparrow	7.6 \uparrow	7.7 \uparrow	5.0 \uparrow	6.7 \uparrow	14.4 \uparrow	11.7 \uparrow	9.6 \uparrow	11.9 \uparrow	4.2 \uparrow
LLaMA3.1-8B																	
NoR	22.2	27.0	20.6	23.3	28.0	29.3	24.4	27.2	3.6	8.8	3.0	5.1	11.2	16.4	8.0	11.9	68.4
Vanilla RAG	32.2	39.0	28.8	33.4	30.6	31.6	25.0	29.1	3.6	9.6	2.0	5.1	20.0	24.8	18.4	21.1	66.2
Vanilla QE	32.6	38.8	28.8	33.4	26.6	28.6	21.6	25.6	5.2	11.0	3.4	6.5	18.4	23.3	15.2	19.0	68.0
KG-Infused RAG (w/ 2-Round)	34.4	39.8	30.2	34.8	37.0	36.0	27.4	33.5	12.0	16.9	7.4	12.1	24.0	28.5	22.4	25.0	67.0
+DPO (w/ 2-Round)	36.6	43.8	31.0	37.1	45.0	46.8	37.0	42.9	12.2	18.5	8.8	13.2	31.2	33.6	24.8	29.9	69.4
Δ KG-Infused RAG \rightarrow Vanilla RAG	2.2 \uparrow	0.8 \uparrow	1.4 \uparrow	1.4 \uparrow	6.4 \uparrow	4.4 \uparrow	2.4 \uparrow	4.4 \uparrow	8.4 \uparrow	7.3 \uparrow	5.4 \uparrow	7.0 \uparrow	4.0 \uparrow	3.7 \uparrow	4.0 \uparrow	3.9 \uparrow	0.8 \uparrow
Δ KG-Infused RAG+DPO \rightarrow Vanilla RAG	4.4 \uparrow	4.8 \uparrow	2.2 \uparrow	3.7 \uparrow	14.4 \uparrow	15.2 \uparrow	12.0 \uparrow	13.8 \uparrow	8.6 \uparrow	8.9 \uparrow	6.8 \uparrow	8.1 \uparrow	11.2 \uparrow	8.8 \uparrow	6.4 \uparrow	8.8 \uparrow	3.2 \uparrow

Activation to retrieve new passages, replacing the originals. We then apply KG-Aug Gen to each retrieved passage individually. The subsequent generation process remains unchanged.

The details of the Self-Ask baseline and its combination with KG-Infused RAG are in Appendix F.1.

4.3 Resources and Model Components

KG & Corpus. To ensure consistency, we use the same KG and text corpus across all datasets and baselines. For the KG \mathcal{G} , we utilize Wikidata5M-KG, an open-domain KG derived from Wikidata entities aligned with Wikipedia. Preprocessing details are provided in Section 3.3. For the corpus \mathcal{D}_c , we adopt Wikipedia-2018, a large-scale open-domain corpus containing 21,015,324 passages².

Retriever and Generator. We adopt Contriever-MS MARCO [13] as the retriever for both entity and passage retrieval across all experiments. For generation, Qwen2.5-7B [30] and LLaMA3.1-8B [10] serve as base generators in our method. For Self-RAG, we employ selfrag-llama2-7b³ as the generator, which is additionally fine-tuned following the original Self-RAG paper. For the Self-Ask baseline, Qwen2.5-7B is selected for its superior instruction-following capability in this setting. The DPO training dataset is constructed using gpt-4o-mini-0718 from the GPT-4o family [1].

4.3.1 Method Configuration

Passage Retrieval. For all baselines and our method, the default number of retrieved passages k_p is set to 6. In both Vanilla-QE and KG-Infused RAG, the original query q and the expanded query q' each retrieve $k_p/2 = 3$ passages from the corpus. Further details on passage retrieval configuration can be found in Appendix C.2.

Initial Entities Retrieval. For KG-Infused RAG, we begin by retrieving the top k_e entities from KG, where k_e is set to 3 by default.

Other Implementation Details. We set the maximum number of activation rounds to 6. Because each round may extract numerous triples, potentially introducing noise that degrades model performance, we introduce two hyperparameters to control the expansion scope:

- MAX_ENTITIES_PER_ROUND: Limits the number of new entities $\{e'\}$ introduced in each activation round, preventing excessive expansion and potential noise.
- MAX_TRIPLES_PER_ENTITY: Restricts the number of triples retrieved per entity, reducing noise from entities with an excessive number of triples (e.g., countries or regions).

²<https://github.com/facebookresearch/DPR>

³https://huggingface.co/selfrag/selfrag_llama2_7b

Table 2: **Plug-in experimental results (%) of Self-RAG.** KG-Infused RAG as a plug-in module is implemented using the DPO-trained LLaMA3.1-8B.

Method	HotpotQA				2WikiMQA				MuSiQue				Bamboogle				StrategyQA
	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc
Self-RAG (selfrag-llama2-7b)	26.2	33.7	24.0	28.0	23.0	27.3	22.0	24.1	4.4	10.4	3.8	6.2	8.8	17.3	8.0	11.4	58.0
+ KG-Infused RAG (w/ 2-Round)	26.6	32.1	22.8	27.2	35.2	39.6	30.0	34.9	9.4	15.8	8.8	11.3	27.2	32.8	24.8	28.3	63.8
+ KG-Infused RAG (w/ 3-Round)	25.8	32.7	21.6	26.7	38.8	41.0	31.4	37.1	10.8	17.4	9.0	12.4	25.6	31.9	22.4	26.6	63.4

5 Results and Analysis

5.1 Main Results

Table 1 presents the overall performance of our method across five datasets.

Performance on Complex QA Tasks. Our method consistently outperforms all standard baselines on four multi-hop datasets, using both Qwen2.5-7B and LLaMA3.1-8B backbones. For example, with two rounds of activation, KG-Infused RAG based on the DPO-trained LLaMA3.1-8B improves the Avg metric over Vanilla RAG by 3.8% to 13.8%. These consistent gains demonstrate the effectiveness of activation-guided KG retrieval in supporting multi-step reasoning and answer generation. In contrast, Vanilla QE does not show clear improvements over Vanilla RAG, suggesting that relying solely on LLM-based query expansion, without external structured knowledge, offers limited benefits for complex reasoning tasks. Overall, these results indicate that KG-based query expansion and knowledge augmentation play a critical role in enhancing multi-hop QA. By guiding retrieval with KG facts, our method better captures relevant context and supports more accurate reasoning.

Performance on Commonsense QA Task. On StrategyQA, which requires implicit commonsense reasoning and binary decisions, our method still brings positive improvements over baselines. This demonstrates the generality of our activation-based retrieval, which remains effective even when explicit multi-hop reasoning is less critical.

Effect of DPO Training. We further apply DPO to enhance the LLM’s instruction-following ability in key stages of our pipeline. Although the training data comes solely from the KG-guided KA stage, we observe consistent performance gains on both multi-hop and commonsense QA tasks. In most cases, models perform better after lightweight DPO tuning. Notably, LLaMA3.1-8B shows greater improvement than Qwen2.5-7B, with Avg gains ranging from 1.1% to 9.4% across five datasets. These results suggest that even minimal preference tuning can effectively improve the model’s ability to incorporate KG facts, leading to further enhancements in our overall framework.

5.2 Plug-in Results

We apply KG-Infused RAG as a plug-in module to the existing corpus-based RAG method Self-RAG. As shown in Table 2, integrating KG-Infused RAG leads to performance improvements on four datasets, with Avg gains ranging from 5.1% to 16.9% under the 2-round activation setting. Although the Avg score on HotpotQA slightly drops by 0.8%, the method remains effective overall.

These results validate the flexibility and generalizability of our activation-based KG retrieval module, which can enhance existing RAG pipelines. We also apply KG-Infused RAG to Self-Ask, with results presented in Appendix F.1.

5.3 Ablation Study

To evaluate the contributions of two key components in KG-Infused RAG: KG-Based QE and KG-Aug Gen, we conduct an ablation study, with results shown in Table 3.

Adding either component to Vanilla RAG yields consistent improvements across all datasets, confirming their individual effectiveness. Specifically, KG-Based QE improves retrieval by activating structured knowledge to enrich queries, while KG-Aug Gen enhances factual grounding by injecting relevant triples during generation. Combining both (“+ Both”) yields the strongest or competitive results in most cases, demonstrating their complementary strengths. However, in some cases, using

Table 3: **Ablation study results (%)**. Vanilla RAG results are based on LLaMA3.1-8B, while all other variants are based on the DPO-trained LLaMA3.1-8B.

Method	HotpotQA				2WikiMQA				MuSiQue				Bamboogle				StrategyQA
	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc
Vanilla RAG (LLaMA3.1-8B)	32.2	39.0	28.8	33.4	30.6	31.6	25.0	29.1	3.6	9.6	2.0	5.1	20.0	24.8	18.4	21.1	66.2
+ KG-Based QE (w/ 2-Round)	36.2	42.9	31.0	36.7	34.6	36.9	28.8	33.4	9.6	15.8	6.2	10.5	25.6	30.2	22.4	26.1	69.0
+ KG-Aug Gen (w/ 2-Round)	36.0	41.9	29.8	35.9	42.6	45.8	35.4	41.3	10.8	16.2	7.0	11.3	32.8	38.5	29.6	33.6	69.0
+ Both (w/ 2-Round)	36.6	43.8	31.0	37.1	45.0	46.8	37.0	42.9	12.2	18.5	8.8	13.2	31.2	33.6	24.8	29.9	69.4

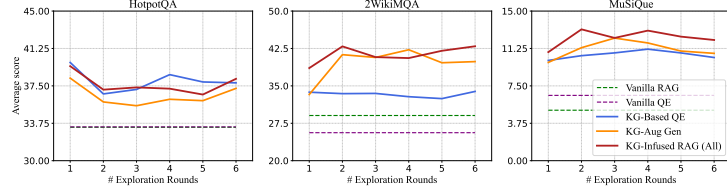


Figure 2: **Impact of the number of activation rounds on KG-Infused RAG**. Vanilla RAG and Vanilla QE use LLaMA3.1-8B, while others use the DPO-trained version.

only KG-Based QE or KG-Aug Gen yields better performance than combining both. This suggests that the effectiveness of KG activation depends on the task: some questions benefit more from KG-enhanced retrieval, while others are better supported by KG-informed answer generation.

5.4 Analysis

Evaluation of KG-Based QE: From a Retrieval Perspective. We evaluate the effectiveness of KG-Based QE from a retrieval perspective by comparing Vanilla RAG with and without KG-based activation. For Vanilla RAG, the top-6 passages are retrieved using only the raw query q , while for Vanilla RAG + KG-Based QE, 3 passages are retrieved using q and 3 using the expanded query q' enriched with KG facts.

Table 4: **Retrieval performance (%)** on four datasets.

Method	Hotpot	2Wiki	MuSiQue	Bambo
	R@6	R@6	R@6	R@6
Vanilla RAG	41.0	34.4	13.4	20.8
+ KG-Based QE (1-Round)	45.0	40.0	19.4	24.8
+ KG-Based QE (2-Round)	47.0	41.0	21.8	24.8

Table 4 shows that KG-Based QE significantly improves retrieval performance, with gains ranging from 4.0% to 8.4% across four datasets. These results confirm that activation-guided query expansion enables more relevant passage retrieval, laying a stronger foundation for downstream reasoning.

Impact of activation rounds. We investigate how the number of activation rounds affects performance. We set a maximum number of rounds to 6 and evaluate performance as rounds increase. Here, KG-Based QE and KG-Aug Gen refer to adding each component individually on top of Vanilla RAG. As shown in Figure 2, 2WikiMQA and MuSiQue see clear gains in the second activation round, but performance fluctuates slightly afterwards. HotpotQA peaks at the first activation, followed by a slight drop. These results suggest that more activation rounds do not always improve outcomes. Limiting the number of activation rounds to one or two is often both efficient and effective.

Furthermore, we observe that for 2WikiMQA and MuSiQue, KG-Aug Gen contributes more to KG-Infused RAG than KG-Based QE. In contrast, on HotpotQA, KG-Based QE alone outperforms the full method. This indicates that KG-Infused RAG does not always outperform its individual components. In some cases, using a single module yields better performance.

6 Conclusion

In this work, we identify key limitations of existing RAG systems: single-source retrieval and the absence of a human-like spreading activation process. To address these issues, we propose **KG-Infused RAG**, a framework that incorporates human-curated KGs to guide query expansion and knowledge augmentation via spreading activation. Extensive experiments show that KG-Infused RAG consistently improves QA performance and can be integrated as a plug-in to enhance corpus-based RAG methods like Self-RAG.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=hSyW5go0v8>.
- [3] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In Bhavana Dalvi Mishra, Greg Durrett, Peter Jansen, Danilo Neves Ribeiro, and Jason Wei, editors, *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106, Toronto, Canada, June 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlrse-1.7. URL <https://aclanthology.org/2023.nlrse-1.7/>.
- [4] Allan M Collins and Elizabeth F Loftus. A spreading-activation theory of semantic processing. *Psychological review*, 82(6):407, 1975.
- [5] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- [6] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.
- [7] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.99. URL <https://aclanthology.org/2023.acl-long.99/>.
- [8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.
- [9] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 04 2021. ISSN 2307-387X. doi: 10.1162/tac1_a_00370. URL https://doi.org/10.1162/tac1_a_00370.
- [10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [11] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024.
- [12] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.coling-main.580>.
- [13] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=jKNipXi7b0>.
- [14] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*, 2023.
- [15] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL <https://aclanthology.org/2023.emnlp-main.495/>.

- [16] Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. Corpus-steered query expansion with large language models. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401, St. Julian’s, Malta, March 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.eacl-short.34/>.
- [17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.
- [18] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.322. URL <https://aclanthology.org/2023.emnlp-main.322/>.
- [19] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL <https://aclanthology.org/2023.acl-long.546/>.
- [20] Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. RaFe: Ranking feedback improves query rewriting for RAG. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 884–901, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.49. URL <https://aclanthology.org/2024.findings-emnlp.49/>.
- [21] Sewon Min, Kalpesh Krishna, Xinxu Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL <https://aclanthology.org/2023.emnlp-main.741/>.
- [22] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL <https://aclanthology.org/2023.findings-emnlp.378/>.
- [23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.
- [24] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.
- [25] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL <https://aclanthology.org/2023.acl-long.557/>.
- [26] Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.585. URL <https://aclanthology.org/2023.emnlp-main.585/>.

- [27] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021. doi: 10.1162/tacl_a_00360.
- [28] Yilin Wen, Zifeng Wang, and Jimeng Sun. MindMap: Knowledge graph prompting sparks graph of thoughts in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10370–10388, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.558. URL <https://aclanthology.org/2024.acl-long.558/>.
- [29] Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2905–2909, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3661370. URL <https://doi.org/10.1145/3626772.3661370>.
- [30] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [31] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [32] Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. *arXiv preprint arXiv:2502.06864*, 2025.

A Illustrations of Retrieval and Activation Processes

A.1 Comparison of Retrieval Processes

Figure 3 illustrates the differences in retrieval strategies between our KG-Infused RAG framework and existing RAG methods.

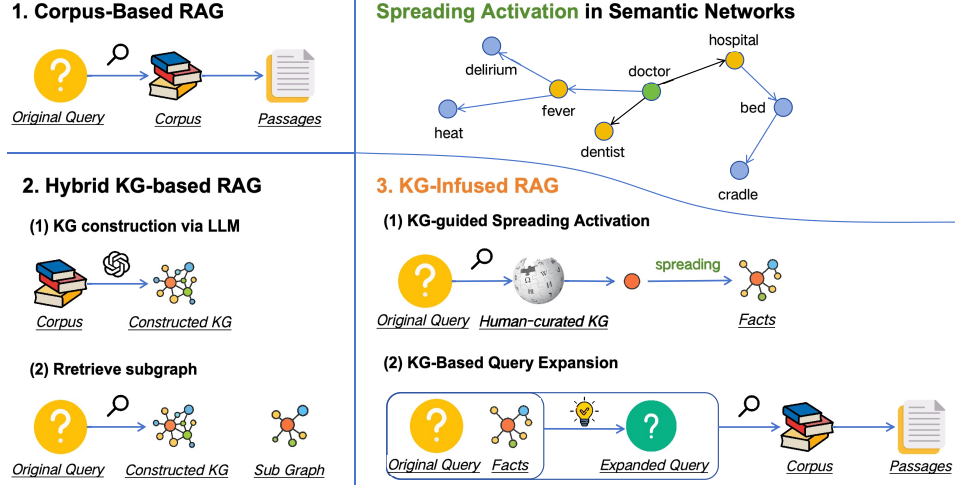


Figure 3: Comparison of retrieval processes: KG-Infused RAG vs. Existing RAG Methods.

A.2 A Case of the Accumulated Subgraph from Spreading Activation

Figure 4 illustrates a case of the accumulated subgraph \mathcal{G}_q^{act} in Wikidata5M-KG.

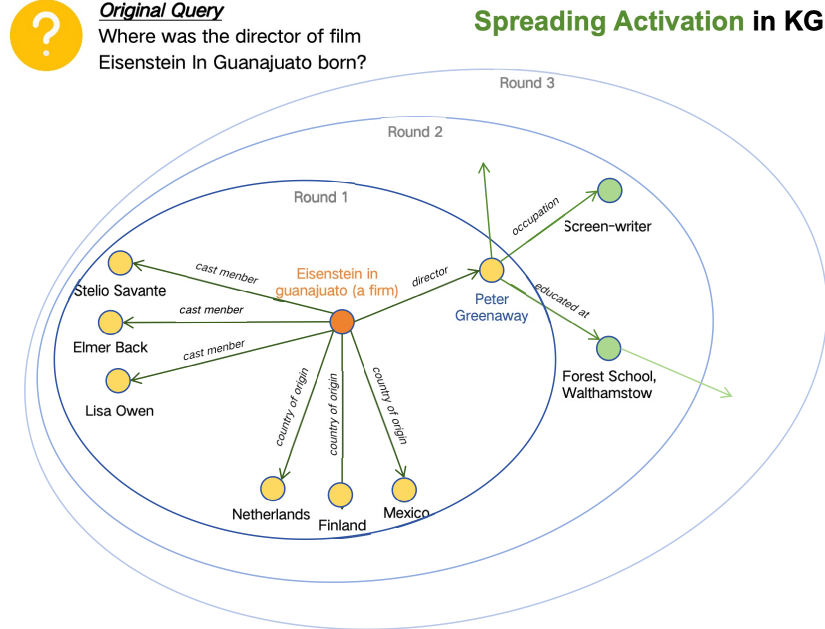


Figure 4: An example of the accumulated subgraph \mathcal{G}_q^{act} from spreading activation in Wikidata5M-KG. This case, taken from the MuSiQue dataset, illustrates the resulting subgraph \mathcal{G}_q^{act} constructed through KG-guided spreading activation. Due to space limitations, we only show part of the accumulated subgraph, and some activated entities are omitted from the figure.

B KG Preprocessing

B.1 Preprocessing Details

To ensure the effectiveness of the KG-Guided Spreading Activation, we enforce two constraints on the preprocessed \mathcal{G} :

- **Entity Description Completeness.** Each entity e must have a textual description d , which can be vectorized for similarity computation with the query.
- **Entity Triples Completeness.** Each entity e must appear as the head entity in at least one triple to enable spreading activation starting from it.

We filter Wikidata5M by removing 5.65% of entities that lack descriptions or associated triples, resulting in a refined KG, denoted as Wikidata5M-KG. Compared to constructing task-specific KGs via LLMs, using Wikidata5M-KG ensures higher factual accuracy and significantly reduces preprocessing overhead.

B.2 Statistics of KG

The statistics of Wikidata5M-KG are shown in Table 5. During preprocessing, some entities and relations were removed for failing to meet two constraints: **Entity Description Completeness** and **Entity Triples Completeness**. Specifically, 5.65% of entities and 2.17% of relations in the original Wikidata5M were discarded. As a result of these removals, only 1.72% of the triples were lost, indicating that the overall triple loss from the original dataset is minimal.

Table 5: **Statistics of Wikidata5M dataset and Wikidata5M-KG.**

Dataset/KG	# Entity	# Relation	# Triple
Wikidata5M	4944931	828	21354359
Wikidata5M-KG	4665331	810	20987217

C Experimental Setup Details

C.1 Datasets & Evaluation

We summarize all datasets and the evaluation details in Table 6.

Table 6: **Summary of the properties of datasets used in experiments.**

Property	HotpotQA	2WikiMQA	MuSiQue	Bamboogle	StrategyQA
# Samples	500	500	500	125	500
Eval Metric	Acc,F1,EM	Acc,F1,EM	Acc,F1,EM	Acc,F1,EM	Acc

C.2 Passage Retrieval

Table 7 summarizes the number of retrieved passages used by each method. Notably, to mitigate duplication between passages retrieved by q and q' , we adopt the following strategy: (1) retrieve the top $k_p/2$ passages using q and the top k_p passages using q' ; (2) select the top $k_p/2$ passages retrieved using q' that do not overlap with those retrieved using q ; (3) merge both sets to obtain k_p unique passages for downstream processing.

Table 7: **Overview of passage retrieval configurations for all methods.**

Method	# Passages from q	# Passages from q'	# Total
Vanilla RAG	6	/	6
Vanilla QE	3	3	6
KG-Infused RAG	3	3	6
Self-Ask	6	0	6
Self-RAG	6	0	6
Self-Ask+KG-Infused RAG	3	3	6
Self-RAG+KG-Infused RAG	3	3	6

D Prompt Details

D.1 Prompts for Inference

Prompts enclosed in a black frame represent the base prompts used across different baselines or methods in the experiments. Prompts enclosed in a [cyan frame](#) indicate prompts specifically designed for or used within KG-Infused RAG.

Table 8: **Prompt of the answer generation stage (without retrieval).**

Prompt of the Answer Generation Stage (without retrieval)

Instructions

Only give me the answer and do not output any other words.

Question: {question}

Answer:

Table 9: **Prompt of the answer generation stage (with retrieval).**

Prompt of the Answer Generation Stage (with retrieval)

Instructions

Answer the question based on the given passages. Only give me the answer and do not output any other words.

Passages:
{passages}

Question: {question}

Answer:

Table 10: **Prompt of the vanilla query expansion.**

Prompt of the Vanilla Query Expansion

Instructions

Generate a new short query that is distinct from but closely related to the original question. This new query should aim to retrieve additional passages that fill in gaps or provide complementary knowledge necessary to thoroughly address the original question. Ensure the new query is relevant, precise, and broadens the scope of information tied to the original question. Only give me the new short query and do not output any other words.

Original Question:
{question}

New Query:

Table 11: **Prompt of the passage note construction stage.**

Prompt of the Passage Note Construction Stage
<p>Instructions</p> <p>Based on the provided document content, write a note. The note should integrate all relevant information from the original text that can help answer the specified question and form a coherent paragraph. Please ensure that the note includes all original text information useful for answering the question. Based on the provided document content, write a note. The note should integrate all relevant information from the original text that can help answer the specified question and form a coherent paragraph. Please ensure that the note includes all original text information useful for answering the question.</p> <p>-----</p> <p>Question to be answered: {question}</p> <p>Document content: {passages}</p> <p>Note:</p>

Table 12: **Prompt of the triples selection stage.**

Prompt of the Triples Selection Stage
<p>Instructions</p> <p>Given a question and a set of retrieved entity triples, select only the triples that are relevant to the question.</p> <p>Information:</p> <ol style="list-style-type: none"> 1. Each triple is in the form of <subject, predicate, object>. 2. The objects in the selected triples will be further explored in the next steps to gather additional relevant triples information. <p>Rules:</p> <ol style="list-style-type: none"> 1. Only select triples from the retrieved set. Do not generate new triples. 2. A triple is relevant if it contains information about entities or relationships that are important for answering the question, either directly or indirectly. <ul style="list-style-type: none"> – For example, if the question asks about a specific person, include triples about that person’s name, occupation, relationships, etc. – If the question asks about an event or entity, include related background information that can help answer the question. 3. Output triples exactly as they appear in angle brackets (<...>). <p>-----</p> <p>Question: {question}</p> <p>Retrieved Entity Triples: {triples}</p> <p>Selected Triples:</p>

Table 13: **Prompt of the triples update stage.**

Prompt of the Triples Update Stage
<p>Instructions</p> <p>Given a question, a set of previously selected entity triples that are relevant to the question, and a new set of retrieved entity triples, select only the triples from the new set of retrieved entity triples that expand or enhance the information provided by the previously selected triples to help address the question.</p> <p>Information:</p> <ol style="list-style-type: none"> 1. Each triple is in the form of <subject, predicate, object>. 2. The objects in the selected triples will be further explored in the next steps to gather additional relevant triples information. <p>Rules:</p> <ol style="list-style-type: none"> 1. Only select triples from the new set of retrieved entity triples. Do not include duplicates of the previously selected triples or generate new triples. 2. A triple is considered relevant if it: <ul style="list-style-type: none"> – Provides new information that complements or builds upon the entities, relationships, or concepts in the previously selected triples, and – Helps to better address or provide context for answering the question. 3. Do not include triples that are unrelated to the question or do not expand on the previously selected triples. 4. Output triples exactly as they appear in angle brackets (<...>). <p>-----</p> <p>Question: {question}</p> <p>Previously Selected Triples: {previous_selected_triples}</p> <p>New Retrieved Entity Triples: {new_retrieved_triples}</p> <p>Selected Triples:</p>

Table 14: **Prompt of the triples (subgraph) summary stage.**

Prompt of the Triples (Subgraph) Summary Stage
<p>Instructions</p> <p>Given a question and a set of retrieved entity triples, write a summary that captures the key information from the triples. If the triples do not provide enough information to directly answer the question, still summarize the information provided in the triples, even if it does not directly relate to the question. Focus on presenting all available details, regardless of their direct relevance to the query, in a concise and informative way.</p> <p>-----</p> <p>Question: {question}</p> <p>Selected Triples: {selected_triples}</p> <p>Summary:</p>

Table 15: Prompt of the KG-based query expansion stage.

Prompt of the KG-Based Query Expansion Stage
<p>Instructions</p> <p>Generate a new short query that is distinct from but closely related to the original question. This new query should leverage both the original question and the provided paragraph to retrieve additional passages that fill in gaps or provide complementary knowledge necessary to thoroughly address the original question. Ensure the new query is relevant, precise, and broadens the scope of information tied to the original question. Only give me the new short query and do not output any other words.</p> <p>-----</p> <p>Original Question: {question}</p> <p>Related Paragraph: {triples_summary}</p> <p>New Query:</p>

Table 16: Prompt of the KG-guided knowledge augmentation stage.

Prompt of the KG-Guided Knowledge Augmentation Stage
<p>Instructions</p> <p>You are an expert in text enhancement and fact integration. Given a question, a retrieved passage, and relevant factual information, your task is to improve the passage by seamlessly incorporating useful details from the factual information. Ensure that the enhanced passage remains coherent, well-structured, and directly relevant to answering the question. Preserve the original meaning while making the passage more informative. Avoid introducing unrelated content.</p> <p>-----</p> <p>Question: {question}</p> <p>Retrieved Passage: {passage}</p> <p>Relevant Factual Information: {triples_summary}</p> <p>Enhanced passage:</p>

D.2 Prompts for DPO Data Collection

Table 17: Prompt of the KG-guided knowledge augmentation stage for DPO.

Prompt of the KG-Guided Knowledge Augmentation Stage for DPO

Instructions

Task: You will receive a list of enhanced passage outputs generated based on a given question, a retrieved passage, and relevant factual information (triples summary).

Your task is to evaluate and compare the outputs to identify the best and worst ones.

Rules:

1. Focus only on the final enhanced passage. Ignore any prefatory comments, explanations, or formatting differences that do not affect content.
2. The quality of an enhanced passage is determined by:
 - Integration: How well the factual information has been integrated into the passage.
 - Coherence: The passage should be logically structured, readable, and maintain a natural flow.
 - Relevance: The enhanced passage should directly support answering the question.
 - Accuracy: Factual information should be incorporated correctly without hallucination or distortion.
 - Preservation: The original passage’s meaning should be preserved and enhanced, not changed incorrectly.
3. If two outputs have substantially the same informational content (even if wording differs slightly), they are considered of equal quality.
4. If all outputs are of similar quality, or if no significant difference can be determined, use the same `_id` for both best and worst.

Input:

Question:

{question}

Retrieved Passage:

{passage}

Relevant Factual Information:

{facts}

Enhanced Passage Outputs:

{output}

Output format:

Output the result as a JSON object:

```
json {{"best_id": <_id of the highest-quality output>, "worst_id": <_id of the lowest-quality output>}}
```

Important:

Do not include any explanations, just the JSON output.

D.3 Prompts of Self-ASK

Table 18: Prompt of Self-Ask.

Prompt of Self-Ask
<p>Instructions</p> <p>Instruction: Answer through sequential questioning. Follow these rules:</p> <ol style="list-style-type: none">1. Generate ONLY ONE new follow-up question per step2. Each follow-up MUST use information from previous answers3. NEVER repeat any form of follow-up question4. When sufficient data is collected, give the final answer. <p>Format Template:</p> <p>Follow up: [New specific question based on last answer]</p> <p>Intermediate answer: [Concise fact from response] ... (repeat until conclusion)</p> <p>So the final answer is: [Answer of the Original Question, only give me the answer and do not output any other words.]</p> <p>-----</p> <p>Few-shot demonstrations</p> <p>Question: Who lived longer, Muhammad Ali or Alan Turing?</p> <p>Are follow up questions needed here: Yes.</p> <p>Follow up: How old was Muhammad Ali when he died?</p> <p>Intermediate answer: Muhammad Ali was 74 years old when he died.</p> <p>Follow up: How old was Alan Turing when he died?</p> <p>Intermediate answer: Alan Turing was 41 years old when he died.</p> <p>So the final answer is: Muhammad Ali</p> <p>Question: When was the founder of craigslist born?</p> <p>Are follow up questions needed here: Yes.</p> <p>Follow up: Who was the founder of craigslist?</p> <p>Intermediate answer: Craigslist was founded by Craig Newmark.</p> <p>Follow up: When was Craig Newmark born?</p> <p>Intermediate answer: Craig Newmark was born on December 6, 1952.</p> <p>So the final answer is: December 6, 1952</p> <p>Question: Are both the directors of Jaws and Casino Royale from the same country?</p> <p>Are follow up questions needed here: Yes.</p> <p>Follow up: Who is the director of Jaws?</p> <p>Intermediate answer: The director of Jaws is Steven Spielberg.</p> <p>Follow up: Where is Steven Spielberg from?</p> <p>Intermediate answer: The United States.</p> <p>Follow up: Who is the director of Casino Royale?</p> <p>Intermediate answer: The director of Casino Royale is Martin Campbell.</p> <p>Follow up: Where is Martin Campbell from?</p> <p>Intermediate answer: New Zealand.</p> <p>So the final answer is: No</p> <p>Question: Who was the maternal grandfather of George Washington?</p> <p>Are follow up questions needed here: Yes.</p> <p>Follow up: Who was the mother of George Washington? Intermediate answer: The mother of George Washington was Mary Ball Washington.</p> <p>Follow up: Who was the father of Mary Ball Washington?</p> <p>Intermediate answer: The father of Mary Ball Washington was Joseph Ball.</p> <p>So the final answer is: Joseph Ball</p> <p>-----</p> <p>Question: {question}</p>

Table 19: **Statistics of DPO training data.** D_{KA}^1 and D_{KA}^2 denote the sampled examples from the first and second rounds of knowledge augmentation, respectively. $D_{KA} = D_{KA}^1 \cup D_{KA}^2$.

	D_{KA}^1	D_{KA}^2	D_{KA}
# Sample for LLaMA3.1-8B	1449	1460	2909
# Sample for Qwen2.5-7B	1202	1259	2461

E DPO Training Details

Training Data. We construct the DPO training set using outputs from the knowledge augmentation stage of KG-Infused RAG with two rounds of spreading activation, based on the 2WikiMultiHopQA training set, sampling 1500 examples from each round to obtain over 3000 input instances. For each input x , we generate 6 candidate outputs by sampling the LLM with different decoding configurations, varying both the temperature and top- p parameters. Specifically, the sampling settings used are:

$$(\text{temperature}, \text{top-}p) = \{(0.0, 1.0), (0.3, 0.95), (0.5, 0.9), (0.7, 0.9), (0.9, 0.8), (1.0, 0.7)\}.$$

We then prompt GPT-4o-mini to identify the best and worst outputs among the candidates for each input. After filtering low-confidence or ambiguous judgments, we obtain 2909 preference-labeled examples for LLaMA3.1-8B and 2461 for Qwen2.5-7B, which are used to train their respective DPO models.

Hyper-parameters for Training. During DPO training, we perform full parameter fine-tuning on 4xA800 GPUs, training the model for one epoch. The detailed hyper-parameters are shown in Table 20.

Table 20: **DPO training hyper-parameters** for LLaMA3.1-8B and Qwen2.5-7B.

Parameter	LLaMA/Qwen
<i>General</i>	
Max sequence length	8000
Batch size	4
Learning rate	5e-7
Training epochs	1
Optimizer	AdamW
AdamW β_1	0.9
AdamW β_2	0.999
AdamW ϵ	1e-8
Weight decay	0.0
Warmup ratio	0.0
<i>DPO-Specific</i>	
DPO β	0.1
Ref model mixup α	0.6
Ref model sync steps	512

F Additional Experimental Results

F.1 Experimental Result of Self-Ask

Applying KG-Infused RAG as a Plug-in to Self-Ask. For the Self-Ask baseline, to ensure consistent settings and reduce API costs, we replace the original web-based search engine with our local corpus retriever and substitute the GPT-3 generator with an open-source LLM. The Self-Ask baseline generates one sub-question (“follow up” question) iteratively based on few-shot demonstrations and the previous reasoning process, then retrieves passages for the sub-question and generates a corresponding answer. This continues until the model determines it can produce the final answer.

Table 21: **Plug-in experimental results (%) of Self-Ask.** KG-Infused RAG as a plug-in module is implemented using the DPO-trained LLaMA3.1-8B.

Method	HotpotQA				2WikiMQA				MuSiQue				Bamboogle				StrategyQA
	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc	F1	EM	Avg	Acc
Self-Ask	36.6	42.6	35.4	38.2	39.2	42.4	33.6	38.4	11.8	16.7	7.8	12.1	32.8	40.1	29.6	34.2	65.6
+ KG-Infused RAG (w/ 1-Round)	34.2	40.3	31.8	35.4	44.8	45.9	35.2	42.0	13.2	18.9	10.8	14.3	32.0	38.5	28.0	32.8	69.6
+ KG-Infused RAG (w/ 2-Round)	34.8	41.1	32.0	36.0	41.8	43.1	33.0	39.3	12.6	17.8	9.0	13.1	30.4	37.0	26.4	31.3	69.2
+ KG-Infused RAG (w/ 3-Round)	35.6	41.6	33.0	36.7	40.6	41.3	30.8	37.6	12.4	18.0	9.2	13.2	33.6	40.5	30.4	34.8	68.2

To incorporate KG-Infused RAG as a plug-in module, we enhance the sub-question answering stage. Specifically, we first perform KG-Based QE on each sub-question to retrieve new passages, followed by KG-Aug Gen to enhance the retrieved passages with relevant KG triples. Answers are then generated based on the fact-enhanced notes.

Results of Self-Ask combined with KG-Infused RAG. KG-Infused RAG is compatible with various corpus-based RAG frameworks. Here, we integrate it into Self-Ask and evaluate its effectiveness.

As shown in Table 21, combining KG-Infused RAG with Self-Ask yields moderate gains on four datasets, while a slight performance drop is observed on HotpotQA. Compared to Self-RAG, the overall improvements are smaller and less consistent. This difference is likely due to Self-Ask’s step-by-step decomposition of complex questions into simpler single-hop sub-questions (see prompt in Table 18 and examples in Table 24). This decomposition enables effective retrieval using only the corpus, thereby reducing reliance on external structured knowledge. In such cases, the marginal benefit of KG-based augmentation diminishes, and injecting external facts may introduce irrelevant information that interferes with reasoning.

G Limitation

KG Data Modalities. This work focuses on activating structured knowledge in the form of triples from KGs, which we integrate into a corpus-based RAG pipeline. However, KGs often contain heterogeneous forms of information beyond triples, such as long-form text, images, and tables. Our current framework is designed specifically for triple-based activation and does not yet account for these heterogeneous modalities. Effectively leveraging such diverse KG content in the activation process is an important direction for future research, potentially enabling broader applicability and richer factual grounding.

Triple Utilization. Our method leverages KG triples in two main ways: query expansion and knowledge augmentation of retrieved passages. While effective, we believe KG facts can be further exploited through other strategies to better align with corpus retrieval and enhance downstream reasoning. Investigating more flexible or task-specific integration mechanisms remains an open and valuable area of future work.

H Case Study

H.1 Case I of Main Results: Direct Answer Retrieval via KG Activation

Table 22 presents a case study comparing the answer generation process of KG-Infused RAG and two baselines. This example illustrates how, through two rounds of KG-guided activation, KG-Infused RAG directly acquires answer-relevant triples from the KG, providing sufficient structured context for the subsequent stages.

KG-Infused RAG begins by retrieving key query-relevant triples from the KG during the KG-Guided Spreading Activation stage. These triples explicitly capture the relationship between the manufacturer of the AAM-N-4 Oriole and its successor company. Leveraging these structured facts, KG-Infused RAG performs KG-based query expansion, generating a targeted query that focuses on pivotal events and transitions in the history of Martin Marietta after it succeeded the Martin Company. In the subsequent KG-augmented generation stage, KG-Infused RAG integrates the retrieved facts into a fact-enhanced note, which clearly states the corporate succession and related developments, ultimately supporting an accurate and grounded final answer.

Table 22: **An example from the MuSiQue dataset.** Blue text indicates segments that positively support answering the question, while red text marks segments that are either uninformative or provide incorrect information. The ✗ denotes a wrong answer, and ✓ represents a correct answer.

Question: Which company followed the company that made AAM-N-4 Oriole?

Process of Vanilla RAG:

Passage note: The company that followed the Glenn L. Martin Company, which developed the AAM-N-4 Oriole, is not directly related to the information provided in the document. ... However, if we look at the information provided in the document about the Oriole record label, we can see that it was acquired by CBS in 1964...

Process of Vanilla QE:

New query: What companies developed and produced the AAM-N-4 Oriole’s successors or related air-to-air missiles in the United States?

Passage note: The company that followed the Glenn L. Martin Company, which developed the AAM-N-4 Oriole, was the Ryan Aeronautical Company. The Ryan Aeronautical Company developed the AAM-A-1 Firebird, an early American air-to-air missile, which was the first air-to-air missile program developed for the United States Air Force. The AAM-A-1 project began in 1946, a year before the development of the AAM-N-4 Oriole started in 1947. The Ryan Aeronautical Company’s AAM-A-1 Firebird was extensively tested in the late 1940s but...

Process of KG-Infused RAG:

Retrieved query-relevant triples (After 2-round KG-guided Spreading Activation):

`<rtv-n-16 oriole | manufacturer | martin company>`
`<rtv-n-16 oriole | instance of | Air-to-air missiles>`
`<martin company | followed by | Martin Marietta>`

Triples summary: ...Based on the provided information, it appears that Martin Marietta is the company that followed the company that made the AAM-N-4 Oriole, as Martin Marietta is the successor to the Martin Company.

New query: What were the key events and milestones in the history of Martin Marietta after it succeeded the Martin Company, particularly in relation to its involvement in the development and production of air-to-air missiles?

Passages note (after query expansion): ...The Glenn L. Martin Company developed the AAM-N-4 Oriole, an early American air-to-air missile, for the United States Navy. In 1961, the Martin Company merged with American-Marietta Corporation to form Martin Marietta Corporation, a leading company in chemicals, aerospace, and electronics. Martin Marietta continued to evolve and eventually merged with Lockheed Corporation in 1995 to form Lockheed Martin Corporation, becoming the world’s largest defense contractor.

Fact-enhanced note: The company that followed the Glenn L. Martin Company, which developed the AAM-N-4 Oriole, an early American air-to-air missile for the United States Navy, is Martin Marietta Corporation. The Martin Company, a renowned manufacturer of the AAM-N-4 Oriole, an instance of Air-to-air missiles, merged with American-Marietta Corporation in 1961 to form Martin Marietta Corporation, a leading company in chemicals, aerospace, and electronics. Martin Marietta continued to evolve and eventually merged with Lockheed Corporation in 1995 to form Lockheed Martin Corporation, becoming the world’s largest defense contractor.

Vanilla RAG: CBS ✗

Vanilla QE: Not mentioned ✗

KG-Infused RAG: Martin Marietta Corporation. ✓

Golden Answer: Martin Marietta

In contrast, the answer generation processes of the two baselines suffer from the lack of such structured knowledge input. Vanilla RAG mistakenly associates the missile “AAM-N-4 Oriole” with the unrelated “Oriole Records” label due to misleading surface-form overlaps in the corpus, leading to an erroneous inference that CBS followed the missile manufacturer. Meanwhile, Vanilla QE misinterprets the earlier development date of Ryan Aeronautical’s missile project as indicating that Ryan “followed” the Glenn L. Martin Company, incorrectly equating temporal precedence with corporate succession.

In summary, KG-Infused RAG benefits significantly from its activation-based use of KG triples, which provide high-precision, structured context that guides the downstream stages toward factual correctness. Without access to such KG-derived evidence, the baselines rely solely on corpus-based retrieval, which often fails to retrieve passages from the corpus that are truly effective for answering the question, leading to factually incorrect answers.

H.2 Case II of Main Results: KG-Guided Query Simplification via Intermediate Nodes

Table 23: **Examples from the 2WikiMQA dataset.** We present the original multi-hop queries and the corresponding simplified queries generated through KG-based Query Expansion (2-round). **Blue text** indicates the simplified or key-focused portions in the expanded query compared to the raw query.

Raw Query: Where was the director of film Eisenstein In Guanajuato born?
Expanded Query: What is Peter Greenaway ’s birthplace?
Raw Query: What is the place of birth of Elijah Blue Allman’s father?
Expanded Query: What is the place of birth of Gregg Allman ?
Raw Query: What is the date of birth of George Frederick, Count Of Erbach-Breuberg’s father?
Expanded Query: What are the dates of birth and death of George Albert I, Count of Erbach-Schonberg ’s siblings?
Raw Query: Which film whose director is younger, Many Tanks Mr. Atkins or Do Musafir?
Expanded Query: What are the ages of the directors of the 2004 film Musafir and the 2013 film Musafir, and how do they compare to the age of R. William Neill ?

Even when the answer node itself is not present in the KG—making it impossible to directly retrieve the answer via multi-round KG-guided activation—the retrieved multi-hop intermediate nodes can still play a crucial role. Specifically, KG-guided Spreading Activation can still identify intermediate entities and relations that form partial reasoning chains, providing useful cues for query reformulation and retrieval. In such cases, KG-Based QE transforms the original complex question into one or more simplified single-hop sub-questions that focus on intermediate concepts surfaced by the KG. These simplified queries improve corpus retrieval by aligning better with the surface forms and local context found in text passages. Table 23 presents several examples of simplified single-hop questions generated by KG-Based QE, demonstrating how intermediate KG nodes help bridge the gap between the original query and corpus evidence.

H.3 Case of Self-Ask

Table 24: **Examples from HotpotQA and 2WikiMQA.** Blue text indicates segments that positively support answering the question, while red text marks segments that are either uninformative or provide incorrect information. The ✗ denotes a wrong answer, and ✓ represents a correct answer.

Question 1 (from HotpotQA): Emily Beecham is best known for her role in a television series whose second season premiered on what date?

Reasoning Process of Self-Ask:

{ Prompt of Self-Ask, see in Table 18 }

Question: Emily Beecham is best known for her role in a television series whose second season premiered on what date?

Are follow up questions needed here: **Follow up:** What is the name of the television series in which Emily Beecham starred?

Intermediate answer: Into the Badlands.**Follow up:** When did the second season of Into the Badlands premiere?

Intermediate answer: March 19, 2017.

So the final answer is: March 19, 2017 ✓

Reasoning Process of Self-Ask + KG-Infused RAG:

{ Prompt of Self-Ask, see in Table 18 }

Question: Emily Beecham is best known for her role in a television series whose second season premiered on what date?

Are follow up questions needed here: **Follow up:** What is the name of the television series in which Emily Beecham starred?

Intermediate answer: Afterlife.**Follow up:** When did the second season of Afterlife premiere?

Intermediate answer: 16 September 2006.

So the final answer is: 16 September 2006 ✗

Golden Answer: March 19, 2017

Question 2 (from 2WikiMQA): Who is the child of the director of film Mukhyamantri (1996 Film)?

Reasoning Process of Self-Ask:

{ Prompt of Self-Ask, see in Table 18 }

Question: Where was the place of death of the director of film Happy Go Lucky (1936 Film)?

Are follow up questions needed here: Yes.

Follow up: Who was the director of the film Happy Go Lucky (1936 Film)?

Intermediate answer: The passage does not provide information about a 1936 film titled “Happy Go Lucky.” The director for the 1943 film is given as Curtis Bernhardt, but there is no information about a 1936 version. **Therefore, the answer to the question as posed cannot be determined from the given information.**

So the final answer is: Information not sufficient to determine the place of death of the director of a 1936 film titled “Happy Go Lucky.” ✗

Reasoning Process of Self-Ask + KG-Infused RAG:

{ Prompt of Self-Ask, see in Table 18 }

Question: Where was the place of death of the director of film Happy Go Lucky (1936 Film)?

Are follow up questions needed here: Yes.

Follow up: Who was the director of the film Happy Go Lucky (1936 Film)?

Intermediate answer: Aubrey Scotto.**Follow up:** Where did Aubrey Scotto die?

Intermediate answer: Los Angeles.

So the final answer is: Los Angeles. ✓

Golden Answer: Los Angeles
