CoTKR: Chain-of-Thought Enhanced Knowledge Rewriting for Complex Knowledge Graph Question Answering

Yike Wu[1,2,*,†] , **Yi Huang**[3,*,†], **Nan Hu**[1,2,*,†],
Yuncheng Hua[4], **Guilin Qi**[1,2], **Jiaoyan Chen**[5], **Jeff Z. Pan**[6†]
[1]Southeast University, Nanjing, Jiangsu, China
[2]Key Laboratory of New Generation Artificial Intelligence Technology and
Its Interdisciplinary Applications (Southeast University), Ministry of Education
[3]China Mobile Research Institute, Beijing, China [4]Monash University, Melbourne, Australia
[5]University of Manchester, Manchester, UK [6]University of Edinburgh, Edinburgh, UK
{yike.wu,nanhu}@seu.edu.cn, huangyi@chinamobile.com

Abstract

Recent studies have explored the use of Large Language Models (LLMs) with Retrieval Augmented Generation (RAG) for Knowledge Graph Question Answering (KGQA). They typically require rewriting retrieved subgraphs into natural language formats comprehensible to LLMs. However, when tackling complex questions, the knowledge rewritten by existing methods may include irrelevant information, omit crucial details, or fail to align with the question's semantics. To address them, we propose a novel rewriting method CoTKR, **C**hain-**o**f-**T**hought Enhanced **K**nowledge **R**ewriting, for generating reasoning traces and corresponding knowledge in an interleaved manner, thereby mitigating the limitations of single-step knowledge rewriting. Additionally, to bridge the preference gap between the knowledge rewriter and the question answering (QA) model, we propose a training strategy PAQAF, **P**reference **A**lignment from **Q**uestion **A**nswering **F**eedback, for leveraging feedback from the QA model to further optimize the knowledge rewriter. We conduct experiments using various LLMs across several KGQA benchmarks. Experimental results demonstrate that, compared with previous knowledge rewriting methods, CoTKR generates the most beneficial knowledge representation for QA models, which significantly improves the performance of LLMs in KGQA [1].

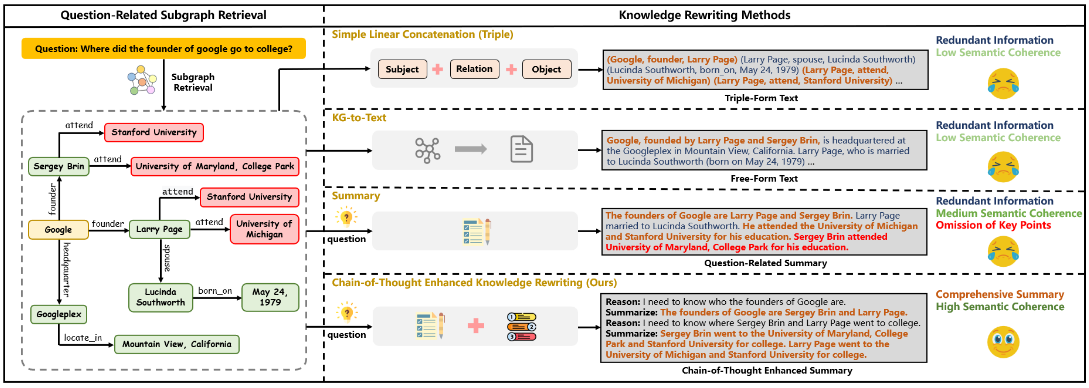
1 Introduction

Large Language Models (LLMs) have achieved remarkable performance across various natural language processing tasks, marking a significant milestone (Sanh et al., 2022; Brown et al., 2020; Zhang et al., 2022; Azaria et al., 2024; Chen, 2024). Despite their superior performance in zero-shot scenarios (Wei et al., 2022a; Kojima et al., 2022), they still encounter factual errors, known as "hallucinations" (Ji et al., 2023b), especially in knowledge-intensive tasks like question answering (QA) (Hu et al., 2023; Tan et al., 2023; Li et al., 2023; He et al., 2023). This issue arises due to the intrinsic limitations of LLMs, including factual inaccuracies and outdated knowledge (Pan et al., 2023). To address this challenge, a substantial of work (Ma et al., 2023a; Trivedi et al., 2023; Wu et al., 2023b) retrieves task-relevant knowledge from external sources as context, thereby enhancing the capabilities of LLMs in downstream tasks, known as Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2023; Huang et al., 2023).

Recent work (Wu et al., 2023b; Baek et al., 2023; Sen et al., 2023; Wang et al., 2024) under the RAG paradigm explores the use of Knowledge Graphs (KGs) (Pan et al., 2017b,a) as an information source to enhance the capabilities of LLMs in Question Answering (QA). Unlike typical QA tasks, a key challenge in KGQA under this paradigm lies in transforming question-related subgraphs into natural language that LLMs can understand while preserving the structural information (Ko et al., 2024; Ding et al., 2024; Wu et al., 2023b). This process is referred to as Knowledge Rewriting (KR) in this study. As illustrated in Figure 1, this paper summarizes the commonly used knowledge rewriting methods in existing work. Most prior studies (Baek et al., 2023; Sen et al., 2023; Wang et al., 2023a) employ simple linear concatenation method (Triple), which concatenates the subject, relation, and object of each triple to form triple-form text. Additionally, considering that LLMs are pretrained on text corpora and struggle with structured triple-form text, some efforts (Wu et al., 2023b; Bian et al., 2021; Chen et al., 2022) focus on converting triples into natural language through KG-to-Text. Furthermore, given that retrieved subgraphs often contain redundant information irrelevant to the question, other studies (Ko et al., 2024; Dern-

[*] Equal contribution. †Corresponding author.

[1] Our code is available at https://github.com/wuyike2000/CoTKR.



(Google, founder, Larry Page) (Larry Page, spouse, Lucinda Southworth) (Lucinda Southworth, born_on, May 24, 1979) (Larry Page, attend, University of Michigan) (Larry Page, attend, Stanford University) ...

Google, founded by Larry Page and Sergey Brin, is headquartered at the Googleplex in Mountain View, California. Larry Page, who is married to Lucinda Southworth (born on May 24, 1979) ...

The founders of Google are Larry Page and Sergey Brin. Larry Page married to Lucinda Southworth. He attended the University of Michigan and Stanford University for his education. Sergey Brin attended University of Maryland, College Park for his education.

Reason: I need to know who the founders of Google are.
Summarize: The founders of Google are Sergey Brin and Larry Page.
Reason: I need to know where Sergey Brin and Larry Page went to college.
Summarize: Sergey Brin went to the University of Maryland, College Park and Stanford University for college. Larry Page went to the University of Michigan and Stanford University for college.

Figure 1: The commonly used knowledge rewriting methods in existing work.

bach et al., 2024) aim to extract question-relevant knowledge from the triples to generate summary pertinent to the question.

Although these strategies are effective, they exhibit several limitations: **(1) Redundancy or omissions.** As illustrated in Figure 1, knowledge generated by Triple and KG-to-Text are verbose, containing excessive irrelevant information. Summary provides a question-related summary but attempts to organize all relevant knowledge in one step. Given the extensive knowledge necessary to address complex questions, this method may not encapsulate all critical information, potentially resulting in the omission of key points. **(2) Semantic mismatch.** The three existing methods shown in Figure 1 ignore the semantics of the question and lack a logical organization that aligns with the question's reasoning path.

To this end, we propose **C**hain-**o**f-**T**hought Enhanced **K**nowledge **R**ewriting, CoTKR. Inspired by ReAct (Yao et al., 2023), the core of our method involves generating reasoning traces and corresponding knowledge in an interleaved manner. As shown in Figure 1, we alternate the following two operations: (1) **Reasoning**: decomposing the question to identify the knowledge required for inference; (2) **Summarization**: summarizing the relevant knowledge from retrieved triples, informed by the reasoning step's output. By integrating Chain-of-Thought (CoT) (Wei et al., 2022b) with knowledge rewriting, CoTKR filters out irrelevant information and extracts question-related knowledge. Moreover, it generates a well-organized knowledge representation[2] semantically aligned with the question. Unlike traditional CoT applications in QA, our framework employs the knowledge rewriter to first summarize knowledge, which then serves as contextual information to enhance QA performance. This strategy offers superior robustness. Although the summary might be inaccurate, it still contributes valuable information, potentially leading to correct answers. However, applying CoT to QA requires more precise reasoning chains, which are significantly affected by the error propagation (Wang et al., 2023a; Yao et al., 2023). To train knowledge rewriters based on LLMs, we design a training framework for CoTKR. In the first stage, inspired by previous work (Ma et al., 2023a; Wu et al., 2023b; Ko et al., 2024), we use knowledge representations generated by ChatGPT to guide the supervised fine-tuning of the knowledge rewriter, enabling it to initially master the capability of knowledge rewriting. In the second stage, we introduce **P**reference **A**lignment from **Q**uestion **A**nswering **F**eedback (PAQAF) to bridge the preference gap between the knowledge rewriter and the QA model. This method evaluates the quality of different knowledge representations based on the corresponding responses from the QA model. Subsequently, it constructs preference pairs, and fine-tunes LLMs through direct preference optimization (DPO) (Rafailov et al., 2023).

We conduct experiments on GrailQA (Gu et al., 2021) and GraphQuestions (Su et al., 2016), comparing commonly used knowledge rewriting methods in existing work. Contrary to previous findings (Dai et al., 2024; Baek et al., 2023), which suggest that LLMs perform better with knowledge in triple-form rather than in natural language, our findings demonstrate that LLMs can significantly benefit from knowledge represented in carefully crafted natural language. This indicates that our method could substantially enhance the performance of LLMs in KGQA.

[2] In this paper, "knowledge representation" refers to the natural language form of question-related knowledge.

The main contributions of this paper are:

- We propose CoTKR, a **C**hain-**o**f-**T**hought Enhanced **K**nowledge **R**ewriting method to improve the quality of knowledge representation through the application of CoT. This method generates reasoning traces and corresponding knowledge in an interleaved manner, thereby producing well-organized knowledge representations that are coherent with the question's semantics.

- We propose a training strategy PAQAF, **P**reference **A**lignment from **Q**uestion **A**nswering **F**eedback, to bridge the preference gap between the knowledge rewriter and the QA model. It assesses the quality of different knowledge representations by evaluating corresponding responses from the QA model. Then, it constructs preference pairs and employs DPO to optimize the knowledge rewriter.

- We conduct experiments on two KGQA benchmarks. Compared with other knowledge rewriting methods, CoTKR can generate the most beneficial knowledge representation for the QA model and further enhance the performance of LLMs in KGQA. Additionally, considering privacy and cost issues, we evaluate the performance of open-source and closed-source LLMs as the foundational models for knowledge rewriting and QA.

2 Related Work

2.1 KG-Augmented LLMs for KGQA

To mitigate hallucination in LLMs, existing work (Wu et al., 2023b; Baek et al., 2023; Sen et al., 2023; Wang et al., 2024) attempts to enhance LLMs with KGs in the RAG paradigm. The naïve approach involves retrieving question-related triples from KGs as contextual information for QA (Baek et al., 2023; Sen et al., 2023). Although this method has proven effective, there remains ample room for improvement. Some studies (Wang et al., 2024, 2023a) integrate Chains-of-Thought (CoT) (Wei et al., 2022b) with RAG to tackle complex questions. Keqing(Wang et al., 2024) decomposes complex questions using predefined templates, retrieves candidate entities from KG, reasons through sub-questions, and ultimately generates answers with clear reasoning paths. KD-CoT (Wang et al., 2023a) validates and adjusts reasoning traces in CoT through interactions with external knowledge, thereby addressing issues of hallucinations and error propagation. Furthermore, alternative efforts (Wu et al., 2023b; Ko et al., 2024) address the limitations of LLMs in processing structured knowledge and the noise in retrieved triples by post-processing these triples into natural language or summaries pertinent to the questions.

This paper focuses on optimizing the knowledge representation under the RAG paradigm for KGQA. Unlike previous work that transforms triples into the natural language in one step, we adopt CoT to summarize relevant knowledge step-by-step, ensuring comprehensiveness and semantic coherence in the generated knowledge.

2.2 Preference Alignment for LLMs on Question Answering

LLMs have the potential to generate content that contains gender discrimination, unethical elements, and racial biases, inconsistent with human values (Wu et al., 2023a; Ray, 2023). To address this issue, Preference Alignment (PA) (Ji et al., 2023a; Wang et al., 2023b) aims to fine-tune LLMs to align with human preferences. Existing QA work based on LLMs uses PA to bridge the gap between model preferences and those of humans or the QA tasks. KnowPAT (Zhang et al., 2023) trains LLMs on a knowledge preference set to align their knowledge biases with human preferences, selecting better factual knowledge as context. BGM (Ke et al., 2024) utilizes downstream task metrics as rewards to optimize the bridging model between retrievers and QA models. Rewrite-Retrieve-Read (Ma et al., 2023b) employs QA evaluation metrics as reward signals, fine-tuning the query rewriting module. EFSUM (Ko et al., 2024) constructs preference pairs sampled from LLMs and fine-tunes the knowledge summarizer using the Direct Preference Optimization (DPO) (Rafailov et al., 2023) algorithm.

Our study innovatively employs responses from the QA model to evaluate the quality of knowledge representations. We then construct preference pairs from these evaluations and optimize the knowledge rewriter using DPO.

3 Preliminaries

Knowledge Graph (KG) is a structured collection of triples in the form of (s, r, o), where s, r, o represent the subject, the relation, and the object. This

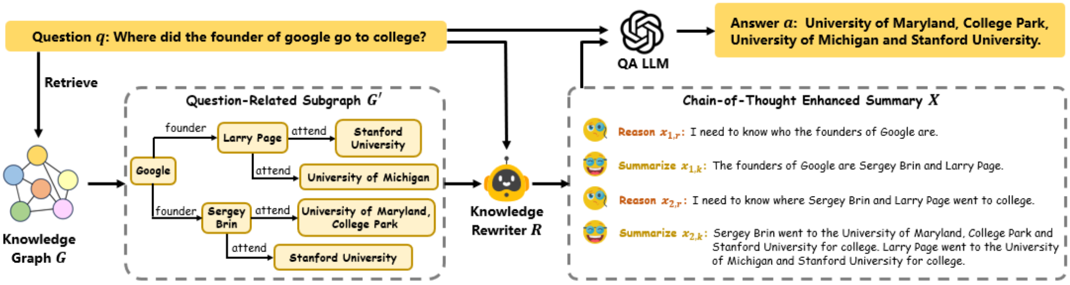

Question q: Where did the founder of google go to college?
Retrieve
Knowledge Graph G
Question-Related Subgraph G'
Google
founder
Larry Page
attend
Stanford University
attend
University of Michigan
founder
Sergey Brin
attend
University of Maryland, College Park
attend
Stanford University
Knowledge Rewriter R
QA LLM
Answer a: University of Maryland, College Park, University of Michigan and Stanford University.
Chain-of-Thought Enhanced Summary X
Reason $x_{1,r}$: I need to know who the founders of Google are.
Summarize $x_{1,k}$: The founders of Google are Sergey Brin and Larry Page.
Reason $x_{2,r}$: I need to know where Sergey Brin and Larry Page went to college.
Summarize $x_{2,k}$: Sergey Brin went to the University of Maryland, College Park and Stanford University for college. Larry Page went to the University of Michigan and Stanford University for college.

Figure 2: Illustration of our KGQA framework. CoTKR generates reasoning traces and corresponding knowledge in an interleaved manner.

collection is denoted by $G = \{(s, r, o) \mid s, o \in E, r \in R\}$, where E represents the set of entities and R represents the set of relations.

Knowledge Graph Question Answering (KGQA) aims to answer natural language questions by utilizing a set of facts within KGs. Following previous work (Saxena et al., 2020; Jiang et al., 2023b), we assume that the subject entity of the question is given. Given a question q and a subject entity e, the objective is to generate a response a using the relevant facts in the KG G that accurately addresses the question.

Knowledge Rewriting (KR) for KGQA aims to transform question-related triples into natural language that can be consumed by LLMs. Given a question q and a subgraph $G' = \{(s, r, o) \mid s, o \in E, r \in R\}$ retrieved from KG G, the task is to generate a natural language sequence X that provides contextual information to answer the question.

4 Methods

4.1 Chain-of-Thought Enhanced Knowledge Rewriting

The architecture of our QA framework is depicted in Figure 2. Initially, our framework retrieves a question-related subgraph from the KG, which is subsequently transformed into contextual knowledge using CoTKR. This contextual knowledge, along with the question, prompts the QA model to generate an answer. The core of this framework is the knowledge rewriter. Briefly, it alternatively conducts the following two operations: **Reasoning**: decomposing the question and generating a reasoning trace based on generated knowledge representation and pointing out the specific knowledge needed for the current step; **Summarization**: summarizing the relevant knowledge based on the current reasoning trace from the subgraph.

Assume we have the reasoning traces at step $t-1$ as $x_{t-1,r}$ and the summarized knowledge at step $t-1$ as $x_{t-1,k}$. The corresponding knowledge representation, i.e., X_{t-1} is represented as:

$$X_{t-1} = [x_{1,r}, x_{1,k}, ..., x_{t-1,r}, x_{t-1,k}]. \quad (1)$$

For knowledge rewriting at step t, given the question q, the subgraph G', and the previously generated content X_{t-1}, the knowledge rewriter R first generates the reasoning trace $x_{t,r}$:

$$x_{t,r} = R(q, G', X_{t-1}) \quad (2)$$

Subsequently, based on the question q, the subgraph G', the previously generated content X_{t-1}, and the reasoning trace at step t $x_{t,r}$, CoTKR summarizes relevant knowledge $x_{t,k}$:

$$x_{t,k} = R(q, G', X_{t-1}, x_{t,r}) \quad (3)$$

$x_{t,r}$ and $x_{t,k}$ are attached to X_{t-1} for the knowledge representation at step t. Note in step 1, X_0 is initialized to None.

4.2 Training Framework for CoTKR

Figure 3 illustrates the training framework of CoTKR.

4.2.1 Supervised Fine-tuning with Knowledge Distilled from ChatGPT

This stage enables open-source LLMs to initially acquire the knowledge rewriting capability through supervised fine-tuning. This primarily comprises two steps: reference knowledge representation generation and supervised fine-tuning.

Reference Knowledge Representation Generation. Inspired by previous work (Ma et al., 2023a; Wu et al., 2023b; Ko et al., 2024), we employ ChatGPT as the data generator to construct training corpora. We verbalize the question-related subgraph, G', through simple linear concatenation and

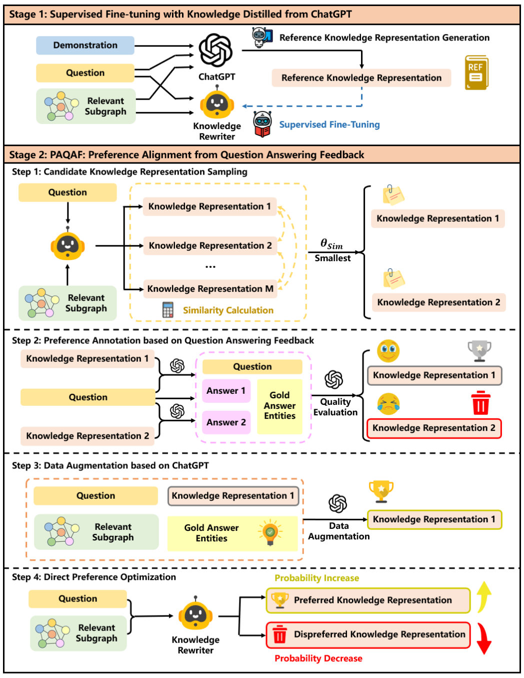

Stage 1: Supervised Fine-tuning with Knowledge Distilled from ChatGPT
Demonstration
Question
Relevant Subgraph
ChatGPT
Knowledge Rewriter
Reference Knowledge Representation Generation
Reference Knowledge Representation
REF
Supervised Fine-Tuning
Stage 2: PAQAF: Preference Alignment from Question Answering Feedback
Step 1: Candidate Knowledge Representation Sampling
Question
Relevant Subgraph
Knowledge Representation 1
Knowledge Representation 2
...
Knowledge Representation M
Similarity Calculation
θ_{Sim}
Smallest
Knowledge Representation 1
Knowledge Representation 2
Step 2: Preference Annotation based on Question Answering Feedback
Knowledge Representation 1
Question
Knowledge Representation 2
Question
Answer 1
Answer 2
Gold Answer Entities
Quality Evaluation
Knowledge Representation 1
Knowledge Representation 2
Step 3: Data Augmentation based on ChatGPT
Question
Knowledge Representation 1
Relevant Subgraph
Gold Answer Entities
Data Augmentation
Knowledge Representation 1
Step 4: Direct Preference Optimization
Question
Relevant Subgraph
Knowledge Rewriter
Probability Increase
Preferred Knowledge Representation
Dispreferred Knowledge Representation
Probability Decrease

Figure 3: Our training framework for CoTKR.

combine it with the question, q, to form the input prompt x. Subsequently, ChatGPT generates the reference knowledge representation k based on several examples (i.e., demonstrations) and the provided input x. Finally, we construct the training dataset $D_T = \{(x_1, k_1), (x_2, k_2), ..., (x_T, k_T)\}$.

Supervised Fine-tuning. For every pair of input and output (x_i, k_i) in the training dataset D_T, our knowledge rewriter R_θ is trained to generate k_i based on x_i using the following objective:

$$L_{SFT} = -\frac{1}{T}\sum_{i=1}^{T} \log p_\theta(k_i|x_i) \tag{4}$$

where θ represents the parameters of the knowledge rewriter R_θ and $p_\theta(k_i|x_i)$ signifies the probability that R_θ generates k_i, given the input x_i.

4.2.2 Preference Alignment from Question Answering Feedback (PAQAF)

In this stage, Preference Alignment (PA) is employed to bridge the preference gap between the knowledge rewriter and the QA model. This stage includes four steps: candidate knowledge representation sampling, preference annotation based on QA feedback, data augmentation based on ChatGPT, and direct preference optimization (DPO).

Candidate Knowledge Representation Sampling. We input the question q and the corresponding subgraph G', then sample M candidate knowledge representations, $k_1, k_2, ..., k_M$, from the knowledge rewriter R_θ.

Preference Annotation based on Question Answering Feedback. Among the candidate knowledge representations, we select the two, k_1 and k_2, with the greatest semantic difference (i.e., the lowest similarity) to facilitate faster convergence during training. Utilizing standard evaluation methods for assessing these representations is suboptimal, as they fail to align with the preferences of QA models. Inspired by the findings in previous work(Wu et al., 2023b; Ko et al., 2024; Zhang et al., 2023), we posit that better knowledge representations generally lead to better performance on QA. Consequently, we adopt k_1 and k_2 as contextual knowledge, prompting the QA model Q to answer the question q, generating answers a_1 and a_2, respectively. Subsequently, we prompt ChatGPT to assess the quality of a_1 and a_2 from the perspectives of accuracy and relevance. This evaluation aims to identify the preferred knowledge representation k^+ and the dispreferred knowledge representation k^-. Details of the evaluation prompt are provided in Appendix A.5.

Data Augmentation based on ChatGPT. ChatGPT is able to produce higher-quality knowledge representations, compared with open-source LLMs. Therefore, in order to improve the quality of preferred knowledge representation and enhance the diversity of the training data, we employ ChatGPT to paraphrase k^+. In addition to the question q, the retrieved subgraph G', and the preferred knowledge representation k^+, we also provide the answer entity e. This allows ChatGPT to organize relevant knowledge around e, ensuring that the rewritten knowledge covers key evidence. We concatenate the question q and the textualized subgraph G' using a prompt template as the input x, and use the paraphrased knowledge representation k^{++} and k^- as the preferred pair. Finally, we construct the preference dataset $P_N = (x_1, k_1^{++}, k_1^-), (x_2, k_2^{++}, k_2^-), ..., (x_N, k_N^{++}, k_N^-)$. The prompt for knowledge augmentation is in Appendix A.6.

Direct Preference Optimization. We employ Direct Preference Optimization (DPO) on our knowledge rewriter, R_θ, to develop a preference-tuned version, R_{θ^*}. It minimizes the following objective:

$$L_{DPO}(\theta^*;\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log\sigma[r(x_i,k_i^{++}) - r(x_i,k_i^{-})] \tag{5}$$

$$r(x_i,k_i) = \frac{p_{\theta^*}(k_i|x_i)}{p_\theta(k_i|x_i)} \tag{6}$$

Considering the varying preferences of different QA models, CoTKR is specifically trained for each QA model. Through the two stages of training, CoTKR tends to generate more favorable knowledge representation k^{++} for each QA model, while avoiding unhelpful knowledge representation k^{-}.

5 Experiments

5.1 Datasets

GrailQA (Gu et al., 2021) is a challenging, large-scale multi-hop KGQA benchmark that consists of 64,331 questions (44,337 train, 6,763 dev, 13,231 test). The training and dev sets provide annotated SPARQL query and answer entities, while the test set comprises only the questions. For evaluation convenience, the dev set is used for testing.

GraphQuestions (Su et al., 2016) is a characteristic-rich dataset for factoid question answering based on Freebase. It comprises 5,166 questions (2,771 train, 2,395 test). For each question, the dataset provides corresponding SPARQL query and answer entities.

5.2 Large Language Models

In this experiment, Llama-2 (7B) (Touvron et al., 2023b), Llama-3 (8B) (AI@Meta, 2024), and ChatGPT [3] are employed for knowledge rewriting, while ChatGPT and Mistral (7B) (Jiang et al., 2023a) are used for QA tasks. The details of these LLMs are provided in Appendix A.3.

5.3 Baselines

We compare **CoTKR** (without PAQAF) and **CoTKR+PAQAF** (**CoTKR+PA** for shortness) with other knowledge rewriting methods in KGQA:

Simple linear concatenation (Triple) (Baek et al., 2023; Sen et al., 2023) concatenates the subject, predicate, and object of a triple to generate triple-form text. This method does not require additional models for knowledge rewriting.

KG-to-Text (Wu et al., 2023b) transforms facts into the free-form text for each relation path with a KG-to-Text model, addressing the limitations of LLMs in understanding structured triple-form text.

Summary (Ko et al., 2024) converts triples into a question-relevant summary, alleviating the issue of redundant contextual knowledge.

To ensure a fair comparison, we employ the same corpus generation method for both the baselines and our method. All baselines undergo supervised fine-tuning without preference alignment.

5.4 Retrieval Methods

The retrieval module is not the focus of our research. Therefore, we adopt three commonly used retrieval methods. For detailed implementation, please refer to Appendix A.4.

2-Hop. We retain 30 triples from the 2-hop subgraph of the head entity, prioritizing those with higher semantic similarity to the question.

BM25. We follow the processing method in DecAF (Yu et al., 2023), simply linearizing the 1-hop subgraph of the topic entity as the article. We take the top 30 triples corresponding to the candidate documents as the retrieval results.

Ground Truth Subgraph (GS). We modify the SPARQL queries from the datasets to obtain the ground truth subgraphs. These subgraphs represent the results of an ideal retriever.

5.5 Evaluation Metrics

Following previous work on generative KGQA (Wu et al., 2023b; Baek et al., 2023; Ko et al., 2024), we adopt **Accuracy (Acc)** as one of our evaluation metrics. It measures whether the model's response includes at least one answer entity. For a dataset comprising N questions, **Acc** is calculated as follows:

$$Acc = \frac{\sum_{i=1}^{N} Acc_i}{N} \tag{7}$$

$$Acc_i = \begin{cases} 1, & \text{if at least one answer entity} \\ & \text{appears in the response} \\ 0, & \text{if no answer entity appears in} \\ & \text{the response} \end{cases} \tag{8}$$

In order to comprehensively assess the performance of KGQA, we employ **Recall** to evaluate the proportion of correct answer entities present in the model's response. For a dataset containing N questions, **Recall** is calculated as follows:

$$Recall = \frac{\sum_{i=1}^{N} Recall_i}{N} \tag{9}$$

[3] https://api.openai.com/

$$Recall_i = \frac{N_{appear}}{N_{total}} \quad (10)$$

where N_{appear} refers to the number of answer entities contained in the model's responses, and N_{total} refers to the total number of the answer entities. Additionally, we consider utilizing **Exact Match (EM)** as an evaluation metric. Given that the responses generated by LLMs consist of multiple paragraphs, while the corresponding answers are entities, we adjust the traditional EM metric. Our modified EM metric assesses whether all answer entities are included in the model's responses. For a dataset consisting of N questions, **EM** is calculated as follows:

$$EM = \frac{\sum_{i=1}^{N} EM_i}{N} \quad (11)$$

$$EM_i = \begin{cases} 1, & \text{if all answer entities} \\ & \text{appear in the response} \\ 0, & \text{other cases} \end{cases} \quad (12)$$

5.6 Main Results

To comprehensively evaluate various knowledge rewriting methods, we employ the widely used 2-Hop retrieval method. Table 1 presents the overall results. We observe that: **(1) Our method outperforms the baselines across most evaluation metrics and LLMs, confirming the effectiveness of our knowledge rewriting strategy.** This also demonstrates the broad practical applicability of CoTKR, effective for both open-source LLMs requiring fine-tuning and closed-source LLMs using ICL. Integrating question-related knowledge significantly improves QA performance compared with direct question answering, underscoring the efficacy of the RAG paradigm in KGQA. KG-to-Text exhibits the weakest performance, indicating that mere conversion of triples into text may result in loss of information inherent in the subgraph. Summary outperforms KG-to-Text but generally lags behind CoTKR/CoTKR+PA, suggesting that filtering out irrelevant knowledge is effective but not adequate. **(2) CoTKR+PA matches or even surpasses the performance of ChatGPT as the knowledge rewriter, proving the effectiveness of our training framework and the preference alignment.** CoTKR+PA outperforms CoTKR, indicating that preference alignment can bridge the preference gap between the knowledge rewriter and the QA model, thereby enhancing the quality of knowledge representation. **(3) A well-crafted knowledge representation is crucial for LLM used in KGQA.** Although Triple does not require an additional knowledge rewriting module, it provides a strong baseline and, in some cases, outperforms KG-to-Text and Summary. Conversely, CoTKR+PA consistently surpasses Triple. This indicates that Triple is simple yet effective and explains its widespread use in existing work. On the other hand, it demonstrates that a carefully designed knowledge representation can effectively enhance the performance of KGQA.

5.7 Impact of Retrieval Methods

To investigate the impact of retrieval methods, we select Llama-3 as the knowledge rewriter and ChatGPT as the QA model. According to the results shown in Figure 4, we have the following observations: **(1) 2-Hop retrieval method may be insufficient for more challenging questions, but it is suitable for simpler ones.** Both BM25 and 2-Hop perform similarly on GrailQA, but 2-Hop shows a significant advantage over BM25 on GraphQuestions. This is likely because GrailQA is a more complex benchmark with a larger question-related subgraph, making 2-hop subgraphs often inadequate. Conversely, for GraphQuestions, a 2-hop subgraph usually provides precise context for most questions. **(2) The design of a high-quality retriever remains an open problem.** GS significantly outperforms BM25 and 2-Hop, indicating that retrieval noise substantially affects KGQA performance. **(3) CoTKR consistently outperforms all baselines across various retrieval methods, demonstrating its robustness and practicality.**

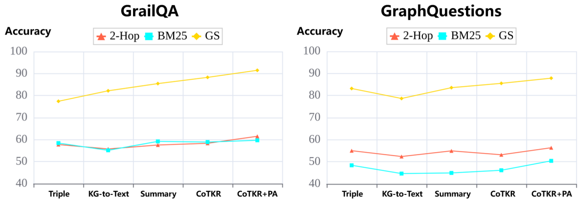


Figure 4: KGQA results using different knowledge rewriters and different retrieval methods.

5.8 Comparison with Triple Method

Several studies (Dai et al., 2024; Baek et al., 2023) suggest that LLMs can better comprehend triple-form text compared with natural language. However, our results show the contrary. Therefore, we delve deeper into this issue by comparing knowledge rewriting methods that use triple-form text

KR LLMs	Methods	GrailQA			GraphQuestions		
		Acc	Recall	EM	Acc	Recall	EM
		ChatGPT as QA model					
None	No Knowledge	28.91	22.81	20.14	35.87	25.76	22.09
	Triple	57.76	49.67	44.73	55.03	46.65	41.63
Llama-2	KG-to-Text	54.75	47.35	42.44	49.73	40.00	33.74
	Summary	58.14	51.38	46.38	52.94	44.70	38.41
	CoTKR	58.64	52.33	47.88	51.36	45.20	39.96
	CoTKR+PA	**59.25**	**53.52**	**49.64**	**56.78**	**47.99**	**42.46**
Llama-3	KG-to-Text	55.76	48.41	43.90	52.40	45.06	39.83
	Summary	57.55	51.06	46.80	54.95	46.86	40.75
	CoTKR	58.33	52.55	48.65	53.19	47.23	43.17
	CoTKR+PA	**61.51**	**56.08**	**52.67**	**56.37**	**49.31**	**45.26**
ChatGPT	KG-to-Text	56.32	49.05	44.73	53.53	45.59	41.17
	Summary	58.54	51.81	47.29	**55.62**	**48.93**	**44.97**
	CoTKR	**59.87**	**53.19**	**49.02**	54.28	48.18	44.68
		Mistral as QA model					
None	No Knowledge	29.44	23.13	20.30	38.20	26.92	22.13
	Triple	54.47	47.78	43.25	51.32	45.97	41.67
Llama-2	KG-to-Text	49.49	42.91	38.41	44.59	37.98	32.82
	Summary	54.10	47.79	43.15	49.85	42.33	36.45
	CoTKR	56.75	51.10	46.71	50.19	43.73	38.54
	CoTKR+PA	**58.15**	**52.98**	**49.13**	**55.07**	**47.02**	**41.71**
Llama-3	KG-to-Text	50.64	44.32	40.13	49.06	43.04	38.25
	Summary	53.84	47.71	43.49	52.03	44.30	38.50
	CoTKR	56.47	51.33	47.36	52.65	46.48	42.21
	CoTKR+PA	**59.31**	**54.13**	**50.24**	**54.82**	**47.76**	**43.09**
ChatGPT	KG-to-Text	51.04	44.87	40.97	49.14	43.04	38.83
	Summary	54.44	48.16	43.97	52.28	47.10	43.30
	CoTKR	**57.28**	**51.14**	**47.09**	**52.82**	**47.13**	**43.55**

Table 1: The overall results of CoTKR and the baselines on GrailQA and GraphQuestions using 2-Hop as retrieval method. For each combination of a knowledge rewriter (KR) LLM and a QA model, the best and second-best results are highlighted in bold and underlined, respectively.

as input (i.e., KG-to-Text, Summary, CoTKR, CoTKR+PA) with Triple. We use Accuracy as the criterion to evaluate the correctness of responses. For each method, we consider three scenarios: **(1) Incorrect→Correct:** Triple provides a wrong answer, but the comparative method answers correctly. **(2) Correct->Incorrect:** Triple answers correctly, but the comparative method answers incorrectly. **(3) No change:** both Triple and the comparative method answer correctly or incorrectly. We adopt Llama-3 as the knowledge rewriter and ChatGPT as the QA model, with 2-Hop as the retrieval method. Then we calculate the proportions of three distinct cases within GrailQA. From the results shown in Figure 5, we draw the following conclusions: **(1) KG-to-Text and Summary have a predominantly negative impact, partially validating the conclusions of prior studies.** Triple provides a strong baseline, and the adoption of KG-to-Text and Summary leads to more incorrect answers. This indicates that LLMs can understand triple-form text effectively, and using simple knowledge rewriting methods leads to loss of information. **(2) Well-designed knowledge representations substantially benefit the question-answering model.** The knowledge representations rewritten by CoTKR/CoTKR+PA generally enhance the QA model's performance. This reflects that the suboptimal knowledge representations in previous work are key contributors to performance degradation. Our method generates comprehensive and semantically coherent knowledge representations, thereby improving the efficacy of KGQA.

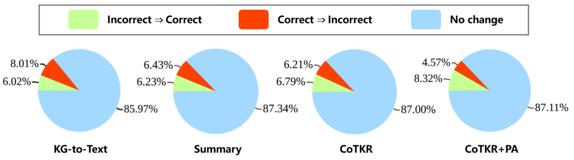


Figure 5: The comparative results on GrailQA. We use 2-Hop as the retrieval method.

5.9 Effectiveness of Data Augmentation

KR LLMs	Methods	ChatGPT			Mistral		
		Acc	Recall	EM	Acc	Recall	EM
Llama-2	CoTKR	51.36	45.20	39.96	50.19	43.73	38.54
	CoTKR+PA*	54.57	47.40	**42.88**	52.99	45.69	41.09
	CoTKR+PA	**56.78**	**47.99**	42.46	**55.07**	**47.02**	**41.71**
Llama-3	CoTKR	53.19	47.23	43.17	52.65	46.48	42.21
	CoTKR+PA*	54.70	48.79	44.72	53.24	47.64	**43.63**
	CoTKR+PA	**56.37**	**49.31**	**45.26**	**54.82**	**47.76**	43.09

Table 2: Ablation study results for data augmentation on GraphQuestions, employing the 2-Hop retrieval method.

To evaluate the effectiveness of data augmentation, we compare the performance of three variants: CoTKR, CoTKR+PA, and CoTKR+PA* (using supervised fine-tuning and preference alignment

Question: what is the unit of area that the measurement system that have an electric field strength units of volt per metre have?
Answer: Square kilometer, Square meter

KR Methods	KR Results
Summary	The International System of Units includes various measurement systems for different physical quantities such as length, area, volume, conductance, resistance, current, force, and pressure. The measurement system with an electric field strength unit of Volt per meter does not directly relate to a specific area unit within the International System of Units.
CoTKR	Reason: I need to know the unit of area associated with the measurement system that has an electric field strength unit of Volt per meter. Summarize: The International System of Units includes various units for different physical quantities such as length, area, volume, etc. Reason: I need to find the specific unit of area in the International System of Units that corresponds to the electric field strength unit of Volt per meter. Summarize: The International System of Units includes units like Square kilometer, Square meter, etc. for measuring area.
CoTKR+PA	Reason: I need to know which measurement system includes the unit for electric field strength in Volt per meter. Knowledge: The International System of Units contains the unit for electric field strength in Volt per meter. Reason: I need to identify the unit of area within the International System of Units. Knowledge: The International System of Units includes the units Square kilometer and Square meter as area units.

Table 3: An example of knowledge rewriting results for different methods.

without data augmentation). The experimental results are shown in Table 2, from which we can draw two conclusions: **(1) CoTKR+PA* generally outperforms CoTKR+PA, indicating that PAQAF does not solely rely on data augmentation based on ChatGPT. (2) CoTKR+PA performs best in most scenarios, proving that data augmentation enhances the preference alignment.**

5.10 Case Study

In this section, we compare Summary with CoTKR through an example. (Please refer to Appendix B for the full comparison result.) As illustrated in Table 3, Summary struggles to extract useful information when faced with an abundance of triples. In contrast, CoTKR, leveraging CoT reasoning, effectively emphasizes the key evidence (i.e., Square meter) in the second rewriting step. Furthermore, after preference alignment, CoTKR+PA is capable of generating more natural reasoning steps, significantly enhancing its applicability to KGQA.

6 Conclusion

In this paper, we propose **C**hain-**o**f-**T**hought Enhanced **K**nowledge **R**ewriting, CoTKR, for higher quality knowledge representation of triples in KG augmented QA. To bridge the preference gap between the knowledge rewriter and the QA model, we propose **P**reference **A**lignment from **Q**uestion **A**nswering **F**eedback, PAQAF. Experimental results demonstrate that, compared with existing knowledge rewriting methods, CoTKR can generate the most beneficial knowledge representation for QA models. In future work, we will go beyond KGQA to explore knowledge representations for other kinds of structured data for RAG.

7 Limitations

We acknowledge the limitations of this work. (1) This study is limited to KGQA and does not explore broader application scenarios. Therefore, we did not design experiments to explore whether CoTKR is effective for all or most RAG scenarios. In future work, we aim to expand the range of data sources. We intend to design a knowledge rewriting method that can be applied to not only KGs but also tables, textual data, and other formats. This enhancement will allow the QA framework to access knowledge from a wider range of sources, thus improving its practicality. Furthermore, we plan to investigate a knowledge representation beneficial for various downstream tasks, such as fact verification and dialogue generation. (2) The training framework for CoTKR depends on the powerful capabilities of closed-source LLMs. However, these models have inherent limitations, and the training data they generate contains noise, which constrains the performance ceiling of CoTKR.

8 Ethical Considerations

We explore optimizing knowledge representations for KGQA on public benchmarks, avoiding any potential harm to any individuals or groups. To promote transparency and facilitate replication of our research, we provide the technical details necessary for reproducing our results and release both the source code and the collected data. Our code and data are available for academic research, commercial use, and other applications.

It is important to acknowledge the potential risks and ethical considerations associated with LLMs. In this study, we construct the training data using ChatGPT and implement our knowledge rewriters based on LLMs. Due to the inherent limitations of LLMs, including factual inaccuracies, racial discrimination, and gender bias, our knowledge rewriters might generate incorrect content or inadvertently reflect prevalent societal biases.

Acknowledgments

This work is partially supported by National Nature Science Foundation of China under No. U21A20488, by the project "Key Laboratory of rich-media Digital Publishing Content Organization and Knowledge Service Open Fund-Research on Knowledge-enhanced Training Techinques of Large Language Model" No. ZD2024-04/01, and by the EPSRC project OntoEm (EP/Y017706/1). We thank the Big Data Computing Center of Southeast University for providing the facility support on the numerical calculations in this paper. ChatGPT was used to enhance the readability of some of the text and improve the language of this paper, after the content was first added manually. All material was then checked manually before submission.

References

AI@Meta. 2024. Llama 3 model card.

Amos Azaria, Rina Azoulay, and Shulamit Reches. 2024. Chatgpt is a remarkable tool - for experts. *Data Intell.*, 6(1):240–296.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106, Toronto, Canada. Association for Computational Linguistics.

Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. In *AAAI*, pages 12574–12582. AAAI Press.

Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. 2007. Freebase: A shared database of structured general human knowledge. In *AAAI*, pages 1962–1963. AAAI Press.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250. ACM.

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*.

Huajun Chen. 2024. Large knowledge model: Perspectives and challenges. *Data Intelligence*, 6(3):587–620.

Zhuo Chen, Yufeng Huang, Jiaoyan Chen, Yuxia Geng, Yin Fang, Jeff Z. Pan, Ningyu Zhang, and Wen Zhang. 2022. Lako: Knowledge-driven visual question answering via late knowledge-to-text injection. In *IJCKG*, pages 20–29. ACM.

Xinbang Dai, Yuncheng Hua, Tongtong Wu, Yang Sheng, and Guilin Qi. 2024. Counter-intuitive: Large language models can better understand knowledge graphs than we thought. *CoRR*, abs/2402.11541.

Stefan Dernbach, Khushbu Agarwal, Alejandro Zuniga, Michael Henry, and Sutanay Choudhury. 2024. Glam: Fine-tuning large language models for domain knowledge graph alignment via neighborhood partitioning and generative subgraph encoding. In *AAAI Spring Symposia*, pages 82–89. AAAI Press.

Wentao Ding, Jinmao Li, Liangchuan Luo, and Yuzhong Qu. 2024. Enhancing complex question answering over knowledge graphs through evidence pattern retrieval. In *WWW*, pages 2106–2115. ACM.

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*, abs/2312.10997.

Yu Gu, Sue Kase, Michelle Vanni, Brian M. Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond I.I.D.: three levels of generalization for question answering on knowledge bases. In *WWW*, pages 3477–3488. ACM / IW3C2.

Jie He, Simon Chi Lok U, Víctor Gutiérrez-Basulto, and Jeff Z. Pan. 2023. BUCA: A Binary Classification Approach to Unsupervised Commonsense Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

Nan Hu, Yike Wu, Guilin Qi, Dehai Min, Jiaoyan Chen, Jeff Z. Pan, and Zafar Ali. 2023. An empirical study of pre-trained language models in simple knowledge graph question answering. *World Wide Web (WWW)*, 26(5):2855–2886.

Wenyu Huang, Mirella Lapata, Pavlos Vougiouklis, Nikos Papasarantopoulos, and Jeff Z. Pan. 2023. Retrieval augmented generation with rich answer encoding. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1012–1025.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. 2023a. AI alignment: A comprehensive survey. *CoRR*, abs/2310.19852.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023b. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

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

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *ICLR*. OpenReview.net.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. *CoRR*, abs/2401.06954.

SungHo Ko, Hyunjin Cho, Hyungjoo Chae, Jinyoung Yeo, and Dongha Lee. 2024. Evidence-focused fact summarization for knowledge-augmented zero-shot question answering. *CoRR*, abs/2403.02966.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

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*.

Linhan Li, Huaping Zhang, Chunjin Li, Haowen You, and Wenyao Cui. 2023. Evaluation on chatgpt for chinese language understanding. *Data Intell.*, 5(4):885–903.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Frassetto Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *SIGIR*, pages 2356–2362. ACM.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *EMNLP*, pages 2511–2522. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting in retrieval-augmented large language models. In *EMNLP*, pages 5303–5315. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023b. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *EMNLP*, pages 12076–12100. Association for Computational Linguistics.

Jeff Z. Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, ussa Biswas, Gerard de Melo, Angela Bonifati, Edlira Vakaj, Mauro Dragoni, and amien Graux. 2023. Large language models and knowledge graphs: Opportunities and challenges. *Transactions on Graph Data and Knowledge*.

J.Z. Pan, D. Calvanese, T. Eiter, I. Horrocks, M. Kifer, F. Lin, and Y. Zhao, editors. 2017a. *Reasoning Web: Logical Foundation of Knowledge Graph Construction and Querying Answering*. Springer.

J.Z. Pan, G. Vetere, J.M. Gomez-Perez, and H. Wu, editors. 2017b. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Partha Pratim Ray. 2023. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. internet of things and cyber-physical systems, 3, 121–154. *URL https://doi. org/10.1016/j. iotcps*, 3.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *ICLR*. OpenReview.net.

Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *ACL*, pages 4498–4507. Association for Computational Linguistics.

Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 1–8, Toronto, Canada. Association for Computational Linguistics.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of GPT-4: reliably evaluating large language models on sequence to sequence tasks. In *EMNLP*, pages 8776–8788. Association for Computational Linguistics.

Yu Su, Huan Sun, Brian M. Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for QA evaluation. In *EMNLP*, pages 562–572. The Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL-HLT*, pages 641–651. Association for Computational Linguistics.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions. *CoRR*, abs/2303.07992.

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. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *ACL (1)*, pages 10014–10037. Association for Computational Linguistics.

Chaojie Wang, Yishi Xu, Zhong Peng, Chenxi Zhang, Bo Chen, Xinrun Wang, Lei Feng, and Bo An. 2024. keqing: knowledge-based question answering is a nature chain-of-thought mentor of LLM. *CoRR*, abs/2401.00426.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023a. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *CoRR*, abs/2308.13259.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Aligning large language models with human: A survey. *CoRR*, abs/2307.12966.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *ICLR*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Xiaodong Wu, Ran Duan, and Jianbing Ni. 2023a. Unveiling security, privacy, and ethical concerns of chatgpt. *CoRR*, abs/2307.14192.

Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. 2023b. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *CoRR*, abs/2309.11206.

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*. OpenReview.net.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2023. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. In *ICLR*. OpenReview.net.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Yichi Zhang, Zhuo Chen, Yin Fang, Lei Cheng, Yanxi Lu, Fangming Li, Wen Zhang, and Huajun Chen. 2023. Knowledgeable preference alignment for llms in domain-specific question answering. *CoRR*, abs/2311.06503.

A Experimental Details

A.1 Data Construction Details

To construct the supervised fine-tuning dataset, we set the temperature to 0 and adopt GPT-3.5 Turbo. We concatenate question q and its related subgraph G' using a prompt template to form the input x. ChatGPT generates candidate knowledge representation k based on 3 examples as demonstrations and input x under ICL paradigm. Given that, the objective of k is to augment the performance of the QA model, we evaluate the quality of k by examining the QA model's results. We utilize k as contextual knowledge for the QA model to generate the answer a. If the answer a encompasses all the answer entities, knowledge k is considered helpful for answering the question, and (x, k) is used as an input-output pair for supervised training.

For the construction of the preference dataset, we sample knowledge representations from the knowledge rewriter after supervised fine-tuning. We set the temperature to 1 to foster greater diversity. During preference annotation, given that the knowledge rewriter aims to generate contextual knowledge beneficial for QA, we first assess the quality of knowledge representation based on the number of answer entities they contain. This naïve strategy is robust and cost-effective, avoiding additional API calls for evaluations using ChatGPT, thus saving time and reducing costs. We label the candidate with the highest number of answer entities as the preferred knowledge representation k^+ and the one with the fewest as the dispreferred k^-. If the number of answer entities is the same, we select the two, k_1 and k_2, with the greatest semantic difference by using all-MiniLM-L6-v2[4] as the encoder. This selection process ensures a significant semantic gap between the two chosen representations, facilitating more rapid model convergence during training. Subsequently, k_1 and k_2 serve as contextual knowledge to prompt the QA model, yielding answers a_1 and a_2. We annotate the preferred knowledge representation k^+ and the dispreferred knowledge representation k^- by evaluating the quality of a_1 and a_2 using ChatGPT. Finally, ChatGPT is used to paraphrase the preferred knowledge representation k^+ into an enhanced version k^{++}, forming a preference pair k^{++} and k^- for direct preference optimization (DPO).

A.2 Datasets

GrailQA[5] (Gu et al., 2021) is a challenging, large-scale multi-hop KGQA benchmark. It is an English dataset that utilizes Freebase (Bollacker et al., 2007, 2008) as KG. It spans 86 domains, such as Sports, Location, and Computer Video Games, and comprises 64,331 questions (44,337 train, 6,763 dev, 13,231 test). This dataset features a large number of entities and relations, complex logical forms, and noise in entity mentions within the questions. The training and dev sets provide annotated SPARQL queries and answer entities, while the test set comprises only the questions. For evaluation convenience, the dev set is used for testing.

GraphQuestions[6] (Su et al., 2016) is a characteristic-rich dataset for factoid question answering based on Freebase across 70 domains, like People, Astronomy, and Medicine. This English dataset focuses on the following question characteristics: structure complexity, function, commonness, paraphrasing, and answer cardinality. It comprises 5,166 questions (2,771 train, 2,395 test), with nearly half requiring multi-hop reasoning. For each question, the dataset provides corresponding SPARQL query and answer entities.

A.3 Large Language Models

Llama-2[7] (Touvron et al., 2023b), an updated version of Llama-1 (Touvron et al., 2023a), is developed using a training corpus comprising 2 trillion tokens and features a context length twice that of Llama-1. To better accomplish the knowledge rewriting task, we select Llama-2-7B-Chat[8].

Llama-3[9] (AI@Meta, 2024) is the latest model in the Llama series. It is renowned for its mastery of language nuances, contextual comprehension,

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[5] This dataset is distributed under the CC BY-SA 4.0 license and our utilization complies with the terms specified in the license.

[6] This dataset is licensed under the Creative Commons Attribution 4.0 and our usage aligns with the intended purposes outlined in this license.

[7] The license of Llama-2 is available at https://ai.meta.com/resources/models-and-libraries/llama-downloads/.

[8] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[9] The license of Llama-3 is available at https://llama.meta.com/llama3/license.

and proficiency in executing complex tasks such as translation and generating dialogues. We choose Llama-3-8B-Instruct[10] for knowledge rewriting.

Mistral[11] (Jiang et al., 2023a) is an open-source LLM developed by Mistral AI. We select the latest instruction-tuned version, Mistral-7B-Instruct-v0.3[12], as our QA model.

ChatGPT[13], developed by OpenAI, is a milestone in the era of LLMs. Its robust capabilities in natural language understanding and generation facilitate superior performance across various tasks. We leverage GPT-3.5-turbo via the API[14] for knowledge rewriting and question answering.

All the LLMs above are general-domain models. Regarding language support, Llama-2, Llama-3, and Mistral only support English, while ChatGPT is multilingual. In this study, the use of these LLMs complies with their respective licenses or terms.

KG-to-Text Prompt

[Instruction]
Your task is to transform a knowledge graph to a sentence or multiple sentences. The knowledge graph is: {triples}. The sentence is:

Summary Prompt

[Instruction]
Your task is to summarize the relevant knowledge that is helpful to answer the question from the following triples.
Triples: {triples}
Question: {question}
Knowledge:

CoTKR Prompt

[Instruction]
Your task is to summarize the relevant information that is helpful to answer the question from the following triples. Please think step by step and iteratively generate the reasoning chain and the corresponding knowledge.
Triples: {triples}
Question: {question}

Table 4: Prompts for Knowledge Rewriting Methods.

Prompt for Question Answering with Triple/KG-to-Text/Summary Knowledge

[Instruction]
Your task is to answer the question based on the knowledge that might be relevant. Try to use the original words from the given knowledge to answer the question. But if it is not useful, just ignore it and generate your own guess.
Knowledge: {knowledge}
Question: {question}
Answer:

Prompt for Question Answering with CoTKR/CoTKR+PA Knowledge

[Instruction]
Your task is to answer the question based on the reasoning chain that might be relevant. Try to use the original words from the given knowledge to answer the question. But if it is not useful, just ignore it and generate your own guess.
Knowledge: {knowledge}
Question: {question}
Answer:

Prompt for Question Answering without Context

Question: {question}
Answer:

Table 5: Prompts for Question Answering.

A.4 Retrieval Methods Details

2-Hop subgraph is a naïve question-related context. Most KBQA studies under RAG paradigms consider triples within the N-hop subgraph of the head entity as contextual knowledge (Baek et al., 2023; Sen et al., 2023; Wang et al., 2024; Ko et al., 2024). To retrieve the 2-hop subgraph around the head entity, we execute SPARQL queries on Freebase. Given the large size of the 2-hop subgraph, we use all-MiniLM-L6-v2[15] to encode all 1-hop and 2-hop relations of the head entity and the question, excluding meaningless relations, such as "common.topic.webpage". Then, we select semantically similar relation paths based on cosine similarity. Finally, we sample the corresponding triples from KG based on these relation paths. In this experiment, we select the top 30 triples as our retrieval results. However, the small size of the 2-hop subgraphs for some entities may result in fewer than 30 triples being retrieved.

BM25 (Robertson and Zaragoza, 2009) is a retrieval method based on TF-IDF scores of sparse

[10] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[11] It is under the Apache 2 License.

[12] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[13] The terms of use for ChatGPT are available at https://openai.com/policies/terms-of-use/.

[14] https://api.openai.com/

[15] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Preference Annotation Prompt

[Instruction]
Your task is to evaluate the quality of two responses to the question based on predefined criteria. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Be as objective as possible.
[Criteria]
For this evaluation, you should primarily consider the following criteria:
Accuracy: The response should contain as many answer entities as possible, and use the original words of the answer entities.
Relevance: The response should be to the point of the question.
Question: {question}
Answer: {answer}
Response A: {response A}
Response B: {response B}
[Evaluation Rule]
Begin your evaluation by comparing the two responses and provide a short explanation. Then output only the single character: "A" if Response A is better, "B" if Response B is better, and "C" for a tie. At the end, repeat just the letter again by itself on a new line.

Table 6: Preference Annotation Prompt.

word matching between input questions and passages. This method is commonly used for integrating multimodal data sources or text-based QA (Wang et al., 2023a; Yu et al., 2023). We follow the processing method in DecAF (Yu et al., 2023), simply linearizing the 1-hop subgraph of the topic entity as the article. We use the BM25 implemented by Pyserini (Lin et al., 2021) and collect the triples corresponding to the candidate articles as the retrieval result. Specifically, we initially retrieve 10 candidate articles, each containing up to 10 triples. Subsequently, we remove any redundant triples or those containing meaningless relations. Given the limited context length of LLMs, we select the top 30 triples as the context information for question answering. After filtering, the number of triples in the candidate articles may be less than 30, thus resulting in the retrieval subgraphs for some questions containing fewer than 30 triples.

Ground Truth Subgraph (GS) refers to a subgraph consisting of the triples necessary for answering a question. In this experiment, we modify the SPARQL queries provided in the datasets and execute them on Freebase to obtain the ground truth subgraphs. We use this subgraph to represent the results of an ideal retriever, aiming to explore the performance upper bound of different knowledge rewriting strategies for the QA model.

Data Augmentation Prompt

[Instruction]
You are a knowledge graph summarizer for Question Answering. I will give you "Question", "Triple", "Answer" and "Knowledge". Your task is to paraphrase the original "Knowledge" into a more helpful representation format for Question Answering. The "Paraphrased Knowledge" should contain the original words of all the answer entities.
Question: {question}
Triple: {triples}
Knowledge: {knowledge}
Paraphrased Knowledge:

Table 7: Data Augmentation Prompt.

A.5 Implementation Details

We utilize LoRA (Hu et al., 2022) to achieve parameter-efficient fine-tuning. For supervised fine-tuning and DPO, the batch size, learning rate, lora rank, lora alpha, and lora dropout are set to 128, 1e-4, 64, 128, and 0.05, respectively. In supervised fine-tuning, we train for 10 epochs and save the best model based on validation set results. In DPO, we observe that more training steps may lead to decreased model performance. Consequently, we train for 1 to 2 epochs on GraphQuestions using approximately 2,000 training samples, and for 5 to 20 steps on GrailQA using 640 to 2560 training samples. During inference, the temperature is set to 0 for ChatGPT and 0.01 for open-source LLMs, ensuring relatively stable output. All the parameter settings mentioned above are the optimal results after multiple trials.

Our implementation utilizes PyTorch[16], Transformers[17], DeepSpeed[18], Datasets[19], PEFT[20], and TRL[21]. We use Datasets for data preprocessing. Both training and inference are based on PyTorch and Transformers. Supervised fine-tuning and DPO are implemented using PEFT, TRL, and DeepSpeed. Experiments are conducted on 4 NVIDIA A100-SXM4-40GB GPUs, with each training or inference session completed within one day. Due to

[16] https://pytorch.org/
[17] https://huggingface.co/docs/transformers/en/index
[18] https://www.deepspeed.ai/
[19] https://huggingface.co/docs/datasets/en/index
[20] https://huggingface.co/docs/peft/en/index
[21] https://huggingface.co/docs/trl/en/index

Question: what is the unit of area that the measurement system that have an electric field strength units of volt per metre have?
Answer: Square kilometer, Square meter

KR Methods	KR Result	QA Response
Triple	(International System of Units, electric_field_strength_units, Volt per metre) (International System of Units, area_units, Square meter) (International System of Units, area_units, Square kilometer) (Volt per metre, strength_in_volts_per_meter, 1.0) ...	Square meter.
KG-to-Text	The International System of Units includes various units for different physical quantities such as length, area, potential, substance, conductance, volume, resistance, current, force, and pressure. For example, the unit for electric field strength in volts per meter is 1.0. Additionally, it covers a wide range of units including square kilometers, square meters ...	The unit of area in the International System of Units that has an electric field strength unit of volt per meter is square meters (m²).
Summary	The International System of Units includes various measurement systems for different physical quantities such as length, area, volume, conductance, resistance, current, force, and pressure. The measurement system with an electric field strength unit of Volt per meter does not directly relate to a specific area unit within the International System of Units.	There is no direct relationship between the measurement system with an electric field strength unit of Volt per meter and any specific area unit within the International System of Units.
CoTKR	Reason: I need to know the unit of area associated with the measurement system that has an electric field strength unit of Volt per meter. Summarize: The International System of Units includes various units for different physical quantities such as length, area, volume, etc. Reason: I need to find the specific unit of area in the International System of Units that corresponds to the electric field strength unit of Volt per meter. Summarize: The International System of Units includes units like Square kilometer, Square meter, etc. for measuring area.	The unit of area in the measurement system that has an electric field strength unit of Volt per meter is Square meter (m²).
CoTKR+PA	Reason: I need to know which measurement system includes the unit for electric field strength in Volt per meter. Knowledge: The International System of Units contains the unit for electric field strength in Volt per meter. Reason: I need to identify the unit of area within the International System of Units. Knowledge: The International System of Units includes the units Square kilometer and Square meter as area units.	The unit of area within the International System of Units that the measurement system having an electric field strength unit of volt per metre has is either Square kilometer or Square meter.

Table 8: An example of knowledge rewriting results for different methods. We use Llama-3 and Mistral as the Knowledge Rewriting (KR) backbone and the QA model, respectively.

the high computational costs of LLMs, we conduct each experiment once and then report the results.

All the prompts involved in this experiment are as follows. Table 4 shows the prompts for different knowledge rewriting methods. Table 5 shows the prompts for question answering. Table 6 shows the preference annotation prompt.

A.6 Prompt for Data Augmentation

Table 7 shows the data augmentation prompt.

B Case Study

In this section, we compare different knowledge rewriting strategies through an example. As illustrated in Table 8, the knowledge generated by both Triple and KG-to-Text contains excessive redundant information. This redundancy complicates the process for the QA model, making it challenging to extract relevant knowledge. Summary struggles to extract useful information when faced with an abundance of triples. In contrast, CoTKR and CoTKR+PA summarize the most pertinent knowledge in the rewriting step, thereby enabling the QA model to provide a concise and accurate answer. Furthermore, after preference alignment, our knowledge rewriter is capable of generating more natural reasoning steps, significantly enhancing its applicability to KGQA.

C Additional Experimental Results

C.1 Experiments on ComplexWebQuestions

To evaluate the robustness of CoTKR, we conduct our experiments on ComplexWebQuestions(Talmor and Berant, 2018). We utilize ChatGPT as the knowledge rewriter, Mistral as the question-answering model, and 2-Hop as the retrieval method. The experimental results, presented in Table 9, demonstrate the effectiveness of CoTKR.

Methods	Acc	Recall	EM
No Knowledge	36.82	31.30	27.62
Triple	39.29	33.94	30.86
KG-to-Text	35.96	31.53	28.96
Summary	38.53	33.87	30.89
CoTKR	**40.70**	**35.74**	**32.72**

Table 9: Experiments on ComplexWebQuestions use ChatGPT as the knowledge rewriter, Mistral as the QA model, and 2-Hop for retrieval.

C.2 Knowledge Rewriter with GPT-4

To assess the applicability of CoTKR to GPT-4, we further conduct the experiments with GPT-4 as the knowledge rewriter, Mistral as the question-answering model, and 2 hop as the retrieval method on 1,000 test questions from GrailQA. The detailed results are presented in Table 10. The findings show that CoTKR outperforms other approaches, with CoTKR utilizing GPT-4 achieving the highest performance. This suggests that employing a more advanced LLM backbone, such as GPT-4, leads to superior outcomes.

Methods	Acc	Recall	EM
No Knowledge	29.10	21.87	18.80
Triple	54.30	47.15	42.40
KG-to-Text	53.20	45.42	40.60
Summary	56.00	48.63	43.60
CoTKR	**57.50**	**52.16**	**48.20**

Table 10: Experiments on 1,000 GrailQA test questions use GPT-4 for rewriting, Mistral for QA, and 2-Hop as the retrieval method.

C.3 Time Analysis

We conduct experiments to analyze the average time cost of the knowledge rewriting methods discussed in this paper. We adopt Llama-3 as the knowledge rewriter, Mistral as the question-answering model, and 2-Hop as the retrieval method. The experiments are conducted on GraphQuestions, utilizing one A100-SXM4-40GB GPU. The average runtime for each question by different methods is shown in Table 11 (unit: seconds). The average runtime of each question for all methods is within an acceptable range (i.e., less than 1.5 seconds). Although our method is the most time-consuming, it exhibits a clear advantage in performance.

C.4 GPT-4-score as Evaluation Metrics

Given the powerful natural language understanding and generation capabilities of closed-source large models, many existing works employ ChatGPT as an evaluator to provide high-quality evaluation results (Sottana et al., 2023; Liu et al., 2023; Min et al., 2023). In our approach, we use GPT-4 as the evaluator to assess whether all answer entities are present in the responses. We refer to this evaluation metric as **GPT-4-score**. Compared to EM, this metric is more flexible, as LLMs are capable of recognizing synonyms of answer entities. We use it to evaluate the first 300 questions from GrailQA using Llama-3 as knowledge rewriter, ChatGPT as question-answering model, and 2 hop as retrieval method. We provide the prompt for **GPT-4-score** in Table 12.

Process	No Knowledge	Triple	KG-to-Text	Summary	CoTKR
Rewrite	0	0	1.0828	0.4615	0.9835
Answer	0.3787	0.5998	0.2701	0.2502	0.4414
Rewrite+Answer	0.3787	0.5998	1.3529	0.7117	1.4249

Table 11: Time analysis on GraphQuestions (seconds) using Llama-3 as the knowledge rewriter, Mistral for question answering, and 2-Hop for retrieval.

GPT-4-score Prompt

[Instruction]
Your task is to evaluate the quality of the response to the question. You should consider whether all the answer entities appear in the response.
Question: {question}
Answer: {answer}
Response: {response}
Begin your evaluation by comparing the response and the answer and provide a short explanation. Then output only the single number: "1" if all the answer entities appear in the response, and "0" if not. At the end, repeat just the number again by itself on a new line.

Table 12: GPT-4-score Prompt.

The experimental results are shown in Table 13. The results indicate that our implemented GPT-4-score is effective and consistent with the outcomes reflected by other evaluation metrics. Furthermore, this also demonstrates that CoTKR possesses significant advantages compared to other knowledge rewriting methods.

C.5 Qualitative Analysis on Data Augmentation

To clearly demonstrate the importance of data augmentation, we perform a qualitative analysis. The results reveal that data augmentation enhances performance in three key areas: reducing redundant reasoning steps, supplementing critical infor-

Methods	GPT-4-score
No Knowledge	0.2767
Triple	0.4700
KG-to-Text	0.4749
Summary	0.4800
CoTKR	0.5167
CoTKR+PA	**0.5567**

Table 13: GPT-4-score on GraphQuestions. We use Llama-3 as knowledge rewriter, ChatGPT as question-answering model, and 2 hop as retrieval method.

mation, and generating more concise summaries. These improvements are exemplified by three representative cases, as shown in Table 14.

C.6 Detailed Experimental Results

This section presents all the experimental results of this study. As shown in Table 15, Table 16, and Table 17, CoTKR/CoTKR+PA achieves the best performance in most scenarios. This indicates that CoTKR is effective for both open-source LLMs after training and closed-source LLMs using ICL. Besides, the results also reveal the robustness of CoTKR, demonstrating its applicability across KGQA systems with various retrieval methods and QA models.

D AI Assistants in Research or Writing

In this research, we primarily utilize ChatGPT for the construction of training data and as the foundational model for knowledge rewriting and QA. For academic writing, ChatGPT is used to correct grammatical errors.

Effect	Question	k^+	k^{++}
Avoiding Redundant Reasoning Step	Katy börner is the curator for what exhibition?	Reason: I need to know which exhibition Katy Börner curated. Summarize: Katy Börner curated the exhibition "Places & Spaces". Reason: I need to know the subjects of the exhibition "Places & Spaces" curated by Katy Börner. Summarize: The subjects of the exhibition "Places & Spaces" curated by Katy Börner include Cartography, Information visualization, Information design, and Scientometrics.	Reason: I need to know which exhibition Katy Börner curated. Summarize: Katy Börner curated the exhibition "Places & Spaces".
Key Points Supplementation	Person of interest deals with how many tv subjects?	Reason: I need to know how many TV subjects are associated with "Person of Interest". Summarize: "Person of Interest" is associated with multiple TV subjects such as Espionage, Mass surveillance, Hacker, and Vigilante.	Reason: I need to know how many TV subjects are associated with "Person of Interest". Summarize: "Person of Interest" deals with a total of 5 TV subjects including Espionage, Mass surveillance, Hacker, Surveillance, and Vigilante.
More Concise Summarization	Who are the owners of gree?	Reason: I need to know who owns GREE, Inc. Summarize: GREE, Inc. is owned by GREE, Inc. itself, according to the triple (GREE, internet.website.owner, GREE, Inc.).	Reason: I need to know who owns GREE, Inc. Summarize: GREE, Inc. is owned by GREE, Inc. itself.

Table 14: Representative cases for improvements through data augmentation.

KR LLMs	**Methods**	**GrailQA**			**GraphQuestions**		
		Acc	**Recall**	**EM**	**Acc**	**Recall**	**EM**
		ChatGPT as QA model					
None	No Knowledge	28.91	22.81	20.14	35.87	25.76	22.09
	Triple	57.76	49.67	44.73	55.03	46.65	41.63
Llama-2	KG-to-Text	54.75	47.35	42.44	49.73	40.00	33.74
	Summary	58.14	51.38	46.38	52.94	44.70	38.41
	CoTKR	58.64	52.33	47.88	51.36	45.20	39.96
	CoTKR+PA	**59.25**	**53.52**	**49.64**	**56.78**	**47.99**	**42.46**
Llama-3	KG-to-Text	55.76	48.41	43.90	52.40	45.06	39.83
	Summary	57.55	51.06	46.80	54.95	46.86	40.75
	CoTKR	58.33	52.55	48.65	53.19	47.23	43.17
	CoTKR+PA	**61.51**	**56.08**	**52.67**	**56.37**	**49.31**	**45.26**
ChatGPT	KG-to-Text	56.32	49.05	44.73	53.53	45.59	41.17
	Summary	58.54	51.81	47.29	**55.62**	**48.93**	**44.97**
	CoTKR	**59.87**	**53.19**	**49.02**	54.28	48.18	44.68
		Mistral as QA model					
None	No Knowledge	29.44	23.13	20.30	38.20	26.92	22.13
	Triple	54.47	47.78	43.25	51.32	45.97	41.67
Llama-2	KG-to-Text	49.49	42.91	38.41	44.59	37.98	32.82
	Summary	54.10	47.79	43.15	49.85	42.33	36.45
	CoTKR	56.75	51.10	46.71	50.19	43.73	38.54
	CoTKR+PA	**58.15**	**52.98**	**49.13**	**55.07**	**47.02**	**41.71**
Llama-3	KG-to-Text	50.64	44.32	40.13	49.06	43.04	38.25
	Summary	53.84	47.71	43.49	52.03	44.30	38.50
	CoTKR	56.47	51.33	47.36	52.65	46.48	42.21
	CoTKR+PA	**59.31**	**54.13**	**50.24**	**54.82**	**47.76**	**43.09**
ChatGPT	KG-to-Text	51.04	44.87	40.97	49.14	43.04	38.83
	Summary	54.44	48.16	43.97	52.28	47.10	43.30
	CoTKR	**57.28**	**51.14**	**47.09**	**52.82**	**47.13**	**43.55**

Table 15: The overall results of CoTKR and the baselines on GrailQA and GraphQuestions using 2-Hop as retrieval method. For each combination of the Knowledge Rewriter (KR) LLM and the QA model, the best and second-best results are highlighted in bold and underlined, respectively.

KR LLMs	Methods	GrailQA			GraphQuestions		
		Acc	Recall	EM	Acc	Recall	EM
		ChatGPT as QA model					
None	No Knowledge	28.91	22.81	20.14	35.87	25.76	22.09
	Triple	58.42	49.06	43.87	48.43	40.44	36.83
Llama-2	KG-to-Text	52.80	43.93	39.01	41.67	32.09	27.68
	Summary	57.62	49.13	43.80	46.47	36.84	31.44
	CoTKR	**58.32**	50.32	45.57	45.34	37.21	32.73
	CoTKR+PA	**58.32**	**50.77**	**46.41**	**49.52**	**40.06**	**35.16**
Llama-3	KG-to-Text	55.14	46.64	41.96	44.68	34.51	30.19
	Summary	59.19	50.83	46.19	44.97	36.97	32.44
	CoTKR	58.89	51.13	46.93	46.18	38.59	35.03
	CoTKR+PA	**59.69**	**52.34**	**48.14**	**50.44**	**42.03**	**37.58**
ChatGPT	KG-to-Text	55.61	47.00	42.38	**46.14**	36.64	31.98
	Summary	58.76	50.47	45.66	**46.14**	**39.20**	**35.66**
	CoTKR	**59.66**	**51.06**	**46.40**	45.18	38.03	34.36
		Mistral as QA model					
None	No Knowledge	29.44	23.13	20.30	38.20	26.92	22.13
	Triple	55.58	47.10	42.07	43.34	36.98	33.86
Llama-2	KG-to-Text	47.33	39.90	35.64	35.24	28.35	25.05
	Summary	52.94	44.94	40.12	43.47	34.56	29.52
	CoTKR	55.69	48.37	43.77	45.39	37.07	32.73
	CoTKR+PA	**57.13**	**50.17**	**45.94**	**49.02**	**39.46**	**34.53**
Llama-3	KG-to-Text	49.83	42.40	38.07	39.21	31.86	28.48
	Summary	54.33	46.73	42.35	41.63	34.41	30.15
	CoTKR	56.76	49.75	45.57	44.38	37.14	33.57
	CoTKR+PA	**58.27**	**51.00**	**46.72**	**48.14**	**39.34**	**34.91**
ChatGPT	KG-to-Text	50.35	42.69	38.40	39.46	33.36	29.52
	Summary	54.98	47.15	42.63	41.42	35.60	31.86
	CoTKR	**56.84**	**48.87**	**44.17**	**43.42**	**36.23**	**32.57**

Table 16: The overall results of CoTKR and the baselines on GrailQA and GraphQuestions using BM25 as retrieval method. For each combination of the Knowledge Rewriter (KR) LLM and the QA model, the best and second-best results are highlighted in bold and underlined, respectively.

KR LLMs	Methods	GrailQA			GraphQuestions		
		Acc	Recall	EM	Acc	Recall	EM
		ChatGPT as QA model					
None	No Knowledge	28.91	22.81	20.14	35.87	25.76	22.09
	Triple	77.41	67.14	61.47	83.17	70.89	62.76
Llama-2	KG-to-Text	80.05	70.20	65.06	76.24	64.47	57.58
	Summary	85.14	75.90	70.03	80.88	69.31	60.00
	CoTKR	87.18	78.46	73.56	80.67	70.73	63.72
	CoTKR+PA	**87.79**	**79.86**	**75.50**	**83.67**	**75.02**	**68.85**
Llama-3	KG-to-Text	82.17	73.10	68.28	78.66	68.32	59.92
	Summary	85.44	77.01	72.07	83.55	72.47	64.47
	CoTKR	88.27	80.35	75.93	85.51	75.89	69.14
	CoTKR+PA	**91.48**	**84.02**	**79.93**	**87.85**	**79.66**	**73.95**
ChatGPT	KG-to-Text	80.39	71.19	66.46	83.05	73.60	67.22
	Summary	86.53	77.88	72.87	88.39	79.26	72.99
	CoTKR	**87.77**	**78.97**	**74.38**	**89.81**	**80.09**	**73.49**
		Mistral as QA model					
None	No Knowledge	29.44	23.13	20.30	38.20	26.92	22.13
	Triple	74.12	65.83	60.79	83.26	72.17	64.22
Llama-2	KG-to-Text	72.47	64.18	59.53	71.19	59.62	51.98
	Summary	78.87	70.43	64.71	78.71	67.45	58.29
	CoTKR	84.27	76.57	71.60	78.66	68.75	61.80
	CoTKR+PA	**85.81**	**79.26**	**74.52**	**81.38**	**72.88**	**67.06**
Llama-3	KG-to-Text	74.54	66.58	62.38	76.37	66.56	58.75
	Summary	79.64	71.65	66.89	80.84	70.07	62.21
	CoTKR	84.73	77.86	73.31	84.18	75.04	68.64
	CoTKR+PA	**87.80**	**81.24**	**76.65**	**84.63**	**77.38**	**73.24**
ChatGPT	KG-to-Text	72.53	65.06	60.92	81.71	72.35	65.43
	Summary	81.34	72.53	67.53	86.47	77.15	70.56
	CoTKR	**84.77**	**76.72**	**72.11**	**88.60**	**78.95**	**72.36**

Table 17: The overall results of CoTKR and the baselines on GrailQA and GraphQuestions using GS as retrieval method. For each combination of the Knowledge Rewriter (KR) LLM and the QA model, the best and second-best results are highlighted in bold and underlined, respectively.