RiTeK: A Dataset for Large Language Models Complex Reasoning over Textual Knowledge Graphs

Anonymous ACL submission

Abstract

Answering complex real-world questions often requires accurate retrieval from textual knowledge graphs (TKGs), as the relational path information from TKGs could enhance the inference ability of Large Language Models (LLMs). However, the bottlenecks include the scarcity of existing TKGs, the limited expressiveness of their topological structures, and the lack of comprehensive evaluations of current retrievers on TKGs. To tackle these challenges, we first develop a Dataset[1] for LLMs Complex **R**eason**i**ng over **Te**xtual **K**nowledge Graphs (RiTeK) with a broad topological structure coverage. We synthesize realistic user queries that integrate diverse topological structures, relational information, and complex textual descriptions. We conduct rigorous expert evaluation to validate the quality of our synthesized queries. RiTeK also serves as a comprehensive benchmark dataset designed to evaluate the capabilities of retrieval systems built on LLMs. By assessing 11 representative retrievers on this benchmark, we observe that existing methods struggle to perform well, revealing notable limitations in current LLM-driven retrieval approaches. These findings highlight the pressing need for more effective retrieval systems tailored for semi-structured data.

1 Introduction

Although large language models (LLMs) have made a significant stride in natural language processing (NLP), complex question answering remains a challenge. Medical professionals, for instance, often need to express complex information that combines flexible inputs with specific, structured constraints. Consider the query: "*Which organ or tissue function, that circulates mother and fetus blood, is affected by Fetal Distress?*" compared to the simpler version: "*What does Fetal Distress affect?*" Accurately addressing such complex queries is critical, as it directly impacts healthcare diagnoses and treatment plans.

To effectively answer these queries, organizing the underlying knowledge using textual knowledge graphs (TKGs) becomes essential. TKGs integrate unstructured data—such as textual descriptions of nodes (e.g., the definition of the medical term *Placental Circulation*)—with structured data, like the relationships between entities within the graph (e.g., the relationship between *Fetal Distress* and *Placental Circulation* is *affects*). This integration enables TKGs to represent comprehensive knowledge tailored to specific applications, rendering them invaluable, especially in the medical field, where accuracy and reliability are critically important.

However, existing datasets (Wu et al., 2024b,a) exhibit one or more of the following limitations: they are overly simplistic, involving only 1-2 hop reasoning paths; they lack diverse topological structure templates [2] and rich relation types; or they fail to incorporate complex constraints [3]. Consequently, these datasets fall short in addressing the complexity of retrieval tasks involving TKGs, especially within the medical domain, where queries require more complex multi-hop reasoning, diverse topological structure templates, and multiple interdependent constraints. Moreover, the absence of textual properties in existing TKGs limits their effectiveness in delivering comprehensive answers.

To address this gap, we present a large-scale complex reasoning dataset over textual knowledge graphs (RiTeK) within the medical domain. In

[1]The dataset is available here: https://anonymous.4open.science/r/Riteck_sumission_version-026B/readme.md

[2]A topological structure is a graph that is an abstract of the query graphs of the same pattern, as shown in Li and Ji (2022).

[3]Constraints are particularly important in KBQA as they help filter out irrelevant information from large knowledge bases, narrowing the search space and improving both efficiency and accuracy

this progress, one primary technical challenge we address is the accurate simulation of user queries with different reasoning types (e.g., six topological structures in Figure 1) within TKGs, ensuring that these queries are relevant and reflective of real-world medical scenarios involving patients, doctors, and medical scientists. This challenge arises from the interdependence of textual and relational information, the complexity of medical terminology and relationships, and the lack of textual descriptions of medical terms. We refer to the framework of Wu et al. (2024b) to simulate user queries and construct precise ground truth answers. However, our focus is primarily on the medical domain, incorporating richer topological structures that extend beyond the traditional 2- and 3-hop structures to better reflect real-world scenarios in the medical domain. Additionally, the textual descriptions of each node are more detailed, enhancing the overall context and understanding. With RiTeK, we delve deeper into retrieval tasks on TKGs, evaluate the capability of current retrieval systems, and provide insights for future advancement.

Key features of RiTeK include the following: (1) it incorporates rich ontological structures and detailed textual descriptions, with content quality rigorously validated by medical experts to ensure high reliability; (2) the benchmark queries are constructed to capture complex relational dependencies and nuanced linguistic characteristics; and (3) the queries demand context-sensitive reasoning, where effective retrieval hinges not only on the model's reasoning capabilities but also on its ability to semantically align with the entity constraints embedded within the question.

We also delve deeper into retrieval tasks on RiTeK, evaluate the capability of current retrieval systems, and provide insights for future advancement, and , highlighting challenges in handling textual and relational data with more compelxy ontology structure and latency on large-scale SKBs with millions of entities or relations

2 Related Work

Datasets of Question Answering over Document. This area of research centers on extracting answers from document sources (Rajpurkar, 2016; Dunn et al., 2017; Joshi et al., 2017; Trischler et al., 2016; Welbl et al., 2018; Yang et al., 2018; Jin et al., 2021, 2019; Hendrycks et al., 2020). For example, SQuAD (Rajpurkar, 2016) assesses a model's ability to interpret and retrieve answers from a single document, focusing on comprehension within a defined context. PubMedQA (Jin et al., 2019) focuses on understanding and reasoning within the context of complex biomedical scientific texts. MedQA-CS (Yao et al., 2024b) aims at simulating authentic clinical scenarios encountered in the clinical examination tasks of medical education. However, unstructured QA datasets often fall short in providing the depth needed for complex relational reasoning to effectively tackle complex user inquiries. In contrast, our research involves queries that demand more complex relational reasoning, challenging the model's ability to navigate and utilize structured information effectively.

Datasets of Question Answering over Knowledge Graph. The structure QA dataset challenges models to retrieve answers from the structured database, such as knowledge graph (Zhang et al., 2018; Yih et al., 2016; Gu et al., 2021; Bao et al., 2016; Trivedi et al., 2017). For example, MetaQA (Zhang et al., 2018) challenges models to generate the relational path with multi-hops. To test the models' abilities to decompose the constraint information in the queries, WebQuestionsSP (Yih et al., 2016) is proposed. GrailQA (Gu et al., 2021) aims to facilitate the answering of more complex questions, as it allows queries to involve up to four relations and optionally includes functions such as counting, superlatives, and comparatives. However, these datasets primarily focus on relational information, the lack of textual information limits its questions within predefined relationships and entities, which constrains the breadth of available information.

Datasets of Question Answering over Textual Knowledge Graph. To integrate textual information into knowledge graphs and queries, STARK (Prime, Amazon, Mag) (Wu et al., 2024b) is proposed. To the best of our knowledge, Stark is the only work that focuses on combining relational information with textual information in the question answering over TKGs. However, this dataset lacks sufficient topological structure coverage, hindering the ability to handle complex queries, particularly in the medical domain. The absence of detailed node descriptions further challenges the model's ability to understand the query information. RiTeK addresses these issues by incorporating richer topological structures and more extensive textual information into knowledge graphs and queries, resulting in more comprehensive and nuanced responses

with deeper insights drawn from abundant textual data.

3 Problem Statement

Textual Knowledge Graph A Textual Knowledge Graph (TKG) is defined as a graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{D})$, where $\mathcal{E}$ denotes a set of entities and $\mathcal{R}$ denotes the set of relations among these entities. In a TKG, the entities and relations are usually organized as *facts* and each fact is defined as a triplet (h, r, t) where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ denote the head entity, tail entity and the relation between the two entities, respectively. Each entity e ($e = h$ or $e = t$) in $\mathcal{G}$ has a textual document $d^e \in \mathcal{D}$ describing the entity information.

Complex Question Answering over Textual Knowledge Graph Given a textual knowledge graph $\mathcal{G}$ and input query q, the model is expected to generate the answers $a \in \mathcal{E}$, which satisfy the relational constraints defined by the structure of $\mathcal{G}$ as specified in q, and the associated document d^e need to satisfy the the knowledge required to solve q.

Textual Triple Graph Unlike traditional knowledge graphs, where each node represents an entity and each edge denotes the relationship between nodes, in the textual triple graph, each node corresponds to a triple (head entity, relation, tail entity) along with the textual description of each entity. In this context, the relation indicates whether the two triples are connected. To be specific, let $\mathcal{G}* = (V, E)$ denote a graph consisting of a set of node V and a set of edges $E \in V \times V$. We denote by n the number of nodes in $\mathcal{G}$ and by m its number of edges. Each node $v = (h, r, t, T(h), T(t)) \in V$, $T(*)$ is the textual description of entity.

4 Dataset for LLMs Complex Reasoning over TKGs (RiTeK)

4.1 Medical Textual Knowledge Graph Construction

We construct two medical TKGs based on PharmKG (Zheng et al., 2021) and ADint (Xiao et al., 2024), as the increased number of entity and relation types introduces significant challenges for path retrieval in the question answering over textual knowledge graph. To enhance the entity attributes, we incorporate textual details from various databases, including Ensembl, UMLS, and Mondo Disease Ontology. As shown in Table 1, our constructed TKGs provide greater node textual coverage, along with more entity types and relation types. For further details on these two medical TKGs, please refer to Appendix A.2.

TKG Dataset	# Entities	# Relation	# Triple	# Coverage
Stark-Amazon	4	4	9,443,802	–
Stark-Mag	4	4	39,802,116	–
Stark-Prime	10	18	8,100,498	15.29%
RiTeK-PharmKG	3	29	500,958	95.61%
RiTeK-ADint	102	15	1,017,284	36.73%

Table 1: Datasets Statistics of constructed medical textual knowledge graphs. # Coverage refers to textual description coverage of each node. # Entities is the number of entity types. # Relation is the number of relation types. As the textual information for the provided nodes is difficult to analyze statistically, we have not provided the statistical information for Stark Mag and Amazon.

4.2 Question Answering Dataset Construction

QA Dataset	# queries	# topological structure	# instance rate	train/val/test
Stark-Amazon	9,100	1	4	0.65/0.17/0.18
Stark-Mag	13,323	4	1.25	0.60/0.20/0.20
Stark-Prime	1,1204	3	9.3	0.55/0.20/0.25
RiTeK-PharmKG	1,0235	6	11.33	0.80/0.10/0.10
RiTeK-ADint	5322	6	9.67	0.80/0.10/0.10

Table 2: Statistical Overview of the Textual KBQA benchmark Datasets. Instance rate refers to the average number of relational templates per topological structure.

4.2.1 Overview

We developed two question answering datasets **RiTeK PharmKG** and **RiTeK ADint** based on two textual knowledge graphs for complex reasoning. These datasets notably feature queries that integrate relational and textual knowledge, incorporating relational templates with broader coverage and higher instance rates. Additionally, to enhance their applicability in practical scenarios, these queries mimic real-life query patterns, exhibiting a natural-sounding quality and flexible formats. Specifically, RiTeK-PharmKG consists of 10,235 synthesized queries. To maximize the coverage of the question topology, we generate the queries following the six types of topological structure (e.g., multi-hop and multi-hop with constraints). For the synthesized queries, we developed 68 relational templates, crafted by a medical expert and detailed in Appendix B.1, to encompass various relation types and ensure practical relevance. The instance rate of

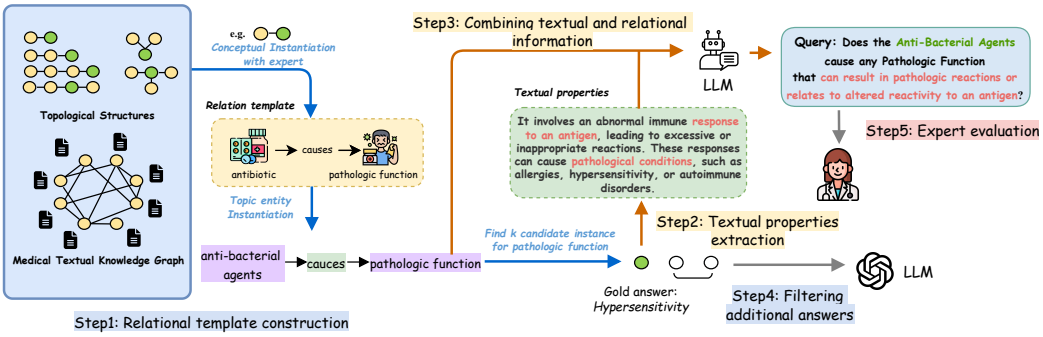

Figure 1: The process of constructing textual structured retrieval datasets involves five main steps, 1) Relational template construction: Create the relation template for TKG using the expert-designed topological structure. 2) Extract Textual Properties: Choose one node as the answer node that meets the relational requirement, and extract relevant textual properties. 3) Combine Information: Merge the relational information and textual properties to form a natural-sounding query. 4) Filtering additional answers: Check if the left nodes satisfy the textual properties to establish other ground truth nodes. 5) Expert Evaluation: The medical experts evaluate the naturalness, diversity, and practicality of the dataset.

11.33, which is higher than that of the current TKG dataset Start (Amazon, Mag, and Prime), highlights the higher diversity of this dataset. RiTeK-ADint consists of 5322 synthesized queries and convers 6 topological structures, with 58 relational templates. Further details are provided in Appendix B.2. To capture the diverse language styles used by different users, we follow Stark and simulate three distinct roles: medical scientist, doctor, and patient. We divide the synthesized queries on each dataset into training, validation, and testing subsets, with the ratios detailed in Table 2. Further details on the scale of our QA benchmarks can be found in Table 2.

4.2.2 Construction Pipline

We present the pipeline that generates the large-scale medical QA datasets on TKGs. The core idea is to entangle relational information and textual properties into the query, and accurately construct ground truth answers with more complex topological structures. The construction of the QA datasets (Figure 1) generally involves five steps, and the specific processes vary depending on the characteristics of each dataset. These steps are as follows:

Relational Template Construction. As shown in Figure 1 Step1, we first created templates based on the 6 designed topological structures (Li and Ji, 2022), which were evaluated by medical experts to ensure their practical relevance and value. Afterward, the topological structures are instantiated conceptually with experts. For instance, for the topological structure *Head entity–relation–tail entity*, the *"(antibiotic) causes <pathologic function>"* is a valid and common medical relation template, as antibiotics, particularly penicillin and cephalosporins, are well-known for triggering drug hypersensitivity reactions. This makes it a medically reasonable and frequently observed relationship. We then converted these relation templates into specific relationship queries, such as "*Anti-Bacterial Agents* causes *pathologic function*." Since each query could correspond to one or more candidate entities, we matched the queries with the textual KG to obtain k candidate entities.

Extracting Textual Properties. As shown in Figure 1 Step2, for the k candidate answers that meet the relationship criteria, we select one entity as the *gold answer* and use GPT-4 to extract textual properties from the entity's associated document. For instance, in the relationship "*Anti-Bacterial Agents* causes pathologic function," we selected "Hypersensitivity" as the gold answer and extracted its textual properties. These textual properties expand upon the concept of hypersensitivity, highlighting their key characteristics, which make it more likely to fulfill the inquirer's needs.

Combining Textual and Relational Information. As shown in Figure 1 Step3, after obtaining the relationship templates and textual properties, we combine these components to synthesize the queries. We chose GPT-4 as the LLM for query synthesis, as it excels at generating natural, human-like questions. Additionally, we optimized the prompt and incorporated instructions for different personas to make the queries more diverse and realistic. This approach enhances the quality of our dataset and increases the demands on our model's reasoning

capabilities.For details on using GPT-4 to generate this query, please refer to Appendix A.3.

Filtering Additional Answers. As shown in Figure 1 Step4, in addition to the gold answer from which the textual properties are extracted, we need to evaluate whether other candidates meet the requirements of the query in order to include them in the final answer set. We use multiple LLMs to assess whether each candidate's description satisfies the textual requirements of the query. Only candidates that pass validation by all LLMs will be added to the final answer set.

Human Evaluation. We invited four medical experts to evaluate 1000 synthetic queries sampled from two datasets. The evaluation was conducted using a 5-point Likert-like scale across three dimensions. Naturalness refers to how grammatically and logically human-like the queries sound. Diversity assesses whether the queries exhibit complex logical structures and encompass multiple entities, relations, and textual requirements. Practicality measures the real-world applicability of the generated queries and their likelihood of being used in everyday scenarios.

The scores were ultimately converted into percentages representing the rates of Positive and Acceptable responses. We found that the evaluation results provided by GPT-4 for our generated dataset were largely consistent with assessments from medical experts. For shorter queries, such as "What gene is inhibited by naloxone?", GPT-4 noted the limited relational and textual information contained within and consequently assigned a lower Diversity score. Both GPT-4 and medical experts agreed that certain rare relationship types, such as "an ancestor of", are infrequently encountered in everyday Q&A scenarios and are more common in medical education contexts. Only a very small number of queries exhibited issues with insufficient Practicality. The results of this evaluation are summarized in the Table 3. The data in the table represents the Positive/ Acceptable rates (%) from GPT-4.

	Naturalness	Diversity	Practicality
RiTeK-PharmKG	81.80/99.60	81.6/99.40	67.4/97.8
RiTeK-ADint	81.20/99.20	74.80/100	68.60/96.60

Table 3: Positive/Acceptable rates(%) from experts

4.2.3 Data Distribution Analysis

We chose Shannon Entropy and Type-Token Ratio (TTR) as the metrics to evaluate the query diversity generated in our two datasets. Shannon Entropy takes into account the frequency of each word, measuring the evenness of word distribution in the text, while Type-Token Ratio reflects the variety of words, with a higher value indicating greater diversity in the generated queries. As shown in Table 6, the TTR values for both RiTeK-PharmKG and RiTeK-ADint surpass those of STARK-Prime, demonstrating that the queries generated in our datasets exhibit high complexity and diversity. For Shannon Entropy, our results are comparable to STARK-Prime. Since our RiTeK-ADint dataset involves a wide range of non-pharmacological interventions (NPIs), lifestyle modifications, and environmental factors, it introduces a richer variety of specialized terminology and concepts into the synthesized queries. This expanded vocabulary diversity leads to significantly higher Shannon Entropy compared to the other medical domain datasets. However, since our two datasets are derived from the medical domain, the frequent repetition of specialized medical terminology, as well as the more concentrated vocabulary compared to general-domain texts, results in slightly lower Shannon Entropy for our datasets than for the other two general-domain datasets. For more analysis about the distribution of query lengths and answer length, please refer Appendix A.4.

5 Experiments

5.1 Retrieval Models and Evaluation Metrics

We evaluated the 9 representative retrieval models on our benchmark datasets under both zero-shot and few-shot settings. In addition to our benchmark dataset, we also evaluated the models on Stark-Prime (Wu et al., 2024b), a textual question answering dataset with minimal ontology in the query, including:

- GPT-4 (Achiam et al., 2023): We use GPT-4 with the instruction to generate the answers directly.
- Random Walk (Lovász, 1993): Starting from the topic entity, a random walk algorithm is applied to explore paths in the textual triple graph in the maximum depth d.
- MCTS (Chaslot, 2010): Starting from the topic entity, a monte carlo tree search algorithm is applied to explore paths in the textual triple graph in the maximum depth d. In this work, we set the $d = 3$.
- Chain-of-Thought (COT) (Wei et al., 2022): We designed the instruction to guide GPT-4 in gen-

		RiTeK-PharmKG						RiTeK-ADint						Stark-Prime					
		Exact Match			Rouge-1			Exact Match			Rouge-1			Exact Match			Rouge-1		
	Approach	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Zero-Shot	GPT-4	11.39	10.90	11.03	15.56	15.50	15.30	7.26	12.10	8.03	13.71	27.64	16.35	5.23	6.81	4.65	11.31	16.35	11.31
	+Random Walk (Lovász, 1993)	12.27	11.86	11.96	14.69	14.15	14.30	15.12	22.68	16.52	20.87	32.92	23.25	7.50	8.20	6.48	13.90	17.31	13.32
	+MCTS (Chaslot, 2010)	17.17	16.54	16.68	19.09	18.44	18.60	16.97	24.41	18.35	22.82	34.69	25.20	7.64	8.36	6.52	14.04	17.45	13.38
	+COT (Wei et al., 2022)	13.11	16.42	13.70	17.53	22.57	18.40	10.52	19.78	11.95	17.79	37.25	20.97	6.47	8.23	5.81	12.61	17.99	12.47
	+TOT (Yao et al., 2024a)	7.31	7.32	7.22	13.21	14.67	13.42	3.97	9.65	5.28	12.90	25.44	15.96	2.99	3.08	2.55	9.50	9.81	8.65
	+GOT (Besta et al., 2024)	3.56	4.20	3.75	10.86	11.84	11.06	2.61	3.32	2.81	15.09	17.63	15.84	1.99	2.20	1.78	9.89	9.34	8.72
	+ TOG (Sun et al., 2023)	29.85	**38.19**	**31.14**	31.38	**40.37**	**32.92**	23.08	**40.63**	25.81	27.81	**48.93**	31.54	12.14	**15.76**	**11.27**	**18.67**	**24.75**	**18.42**
	+G-retriver (He et al., 2024)	11.21	13.39	11.60	15.01	18.54	15.62	10.97	19.05	12.52	17.27	32.99	20.41	6.23	6.61	5.17	12.01	14.92	11.40
	+KAR (Xia et al., 2024)	**30.95**	23.99	25.18	**33.65**	26.11	27.50	**39.59**	24.00	**27.29**	**46.54**	28.87	**32.80**	**12.02**	14.49	11.12	18.04	22.20	17.61
Few-Shot	GPT-4	13.75	15.54	14.04	16.84	19.84	17.49	17.57	17.91	17.48	25.50	28.08	26.04	7.79	6.41	5.91	14.03	13.53	12.14
	+Random Walk (Lovász, 1993)	11.02	13.28	11.32	14.46	17.88	14.92	22.99	22.79	22.75	29.10	29.07	28.95	9.93	6.93	7.34	16.54	13.02	13.45
	+MCTS (Chaslot, 2010)	17.79	17.11	17.30	20.97	20.29	20.48	19.51	27.32	20.91	24.71	36.25	26.96	9.57	6.89	7.14	15.92	12.55	12.88
	+COT (Wei et al., 2022)	17.29	16.91	16.99	21.55	20.97	21.13	18.57	18.12	18.26	26.68	26.62	26.53	8.13	5.91	5.99	14.03	13.53	12.14
	+TOT (Yao et al., 2024a)	14.74	14.74	14.63	19.22	19.14	18.97	13.28	13.17	13.21	24.65	24.72	24.60	12.84	10.11	10.36	6.93	4.85	5.06
	+GOT (Besta et al., 2024)	12.10	12.22	12.06	17.38	17.31	17.19	15.84	15.32	15.42	26.20	25.89	25.91	5.37	3.73	3.78	12.69	9.98	10.17
	+TOG (Sun et al., 2023)	**29.14**	**42.33**	**32.36**	**30.40**	**44.00**	**33.88**	**26.50**	**47.13**	**33.83**	**29.46**	**49.69**	**36.43**	**14.41**	**20.39**	**16.40**	**19.75**	**26.61**	**20.14**
	+G-retriver (He et al., 2024)	12.51	12.14	12.22	15.94	15.44	15.57	17.47	17.50	17.32	24.87	24.92	24.71	7.72	5.75	5.86	14.63	11.92	12.10
	+KAR (Xia et al., 2024)	27.35	27.43	26.99	29.74	29.76	29.34	34.68	33.42	33.48	40.15	38.55	38.88	13.01	15.50	12.21	19.00	23.10	18.00
Supervised	G-retriver (He et al., 2024)	38.71	37.11	37.62	39.78	39.18	39.31	47.93	47.16	47.41	54.68	54.00	54.24	16.14	16.47	14.11	17.21	27.86	19.21
	GCR (Luo et al., 2024)	44.38	**57.28**	47.71	46.04	**58.83**	49.44	43.52	**60.78**	48.07	49.47	**65.57**	54.24	**19.03**	**26.89**	**18.94**	**28.01**	**37.18**	**28.75**
	GNN-RAG (Mavromatis and Karypis, 2024)	**50.78**	49.28	**49.72**	**51.66**	50.29	**50.73**	**51.04**	50.59	**50.55**	**56.49**	56.09	56.09	16.00	15.04	14.50	24.78	23.51	22.99

Table 4: Results of various approaches for question answering with complex reasoning on RiTeK-PharmKG, RiTeK-ADint and Stark-Prime. P refers to the Precision, R refers to the recall. In the experiments, the GPT-4 version is GPT4o-mini.

erating the answer step by step, with the output formatted as step-by-step reasoning: explanation, answer: medical terms.

- Tree-of-Thought (TOT) (Yao et al., 2024a):We structured the reasoning process as a tree search, where multiple intermediate reasoning paths are explored in parallel. GPT-4 evaluates and expands promising paths based on a voting or scoring mechanism.
- Graph-of-Thought (GOT) (Besta et al., 2024): We represented the reasoning process as a graph structure, where nodes capture different reasoning states and edges denote transitions. GPT-4 traverses the graph to aggregate information and synthesize the final answer.
- Think-on-Graph (TOG) (Sun et al., 2023): is a reasoning framework that enables large language models to interactively perform beam search over knowledge graphs, discovering and evaluating promising reasoning paths without additional training.
- G-retriever (He et al., 2024): A RAG-based approach that retrieves query-relevant subgraphs using the Prize-Collecting Steiner Tree (PCST) algorithm to enhance LLM understanding and reasoning over textual graphs.
- KAR (Xia et al., 2024): A knowledge-aware query expansion method that augments LLMs with structured document relations from a knowledge graph, using relation-aware filtering to improve retrieval for semi-structured queries.

We evaluated the 3 representative retrieval models on our benchmark datasets and Stark-Prime under supervised learning settings, including:

- G-retriever (He et al., 2024):A RAG-based approach that retrieves query-relevant subgraphs using the Prize-Collecting Steiner Tree (PCST) algorithm to enhance LLM understanding and reasoning over textual graphs.
- GCR (Luo et al., 2024): A knowledge-aware query expansion method that augments LLMs with document-based relational signals to improve retrieval for semi-structured queries.
- GNN-RAG (Mavromatis and Karypis, 2024). A method that uses a GNN to retrieve relevant answers and extract the shortest paths connecting the topic entity and answers, which are then verbalized and fed into the LLM to enhance retrieval-augmented generation (RAG) performance.

We evaluated the outputs of different methods using several metrics, including Exact Match (EM)(Rajpurkar, 2016; Li and Huang, 2023), which assesses whether the predicted sequence exactly matches the reference, awarding credit only for perfect matches. Additionally, we employed ROUGE-1(Cohan and Goharian, 2016) to measure unigram overlap between the predicted and reference sequences, providing partial credit for shared words even when the sequences are not identical. To ensure fairness in the comparison, the instructions and examples are the same for both the zero-shot and few-shot settings, respectively.

5.2 Results and Discussion

Table 4 shows the experiment results of various approaches based on Excat Match and Rougle-1. We have the following observations. Zero-shot and few-shot setting: (1) We observed that the baseline models struggle to generate the correct answers on RiTeK-PharmKG and RiTeK-ADint. For GPT-4 and GPT+COT, they are challenges in utilizing reasoning information from the graph. Although GPT+COT can utilize step-by-step reasoning, it still relies on the inherent knowledge of the LLM, which limits its ability to apply clear logical reasoning based on knowledge graphs. For the Random Walk, while it can provide reasoning paths, its random nature limits its ability to accurately identify the correct path information. However, it could get the better performance than GPT-4 in RiTek-Adint and Stark-Prime in the zero/few-shot setting. (2) Tree-of-Thought (ToT) and Graph-of-Thought (GoT) attempt to guide LLM reasoning through structured prompting, encouraging step-by-step or graph-based logical thinking. However, on complex textual KBQA datasets like RiTeK-PharmKG and RiTeK-ADint, both methods consistently underperform, with F1 scores far below those of retrieval-augmented approaches like KAR (e.g., ToT: 13.42 vs. KAR: 27.50 in zero-shot). This suggests that the internal knowledge and reasoning capabilities of LLMs alone are insufficient for tasks that require fine-grained relational understanding and the integration of attribute information from the query. Despite their logical scaffolding, ToT and GoT struggle to recover factual precision without access to external structured knowledge. (3) KAR achieves strong performance on medical datasets like RiTeK-PharmKG and RiTeK-ADint, outperforming baselines in both zero-shot and few-shot settings. Its main strength lies in combining textual semantics with structured KG relations to generate accurate and context-aware query expansions. However, KAR relies on retrieving the top-n relevant documents; however, determining an appropriate value for n and the optimal order in which to select documents is non-trivial. (4) G-Retriever shows moderate performance across medical datasets, but generally underperforms compared to methods like KAR or TOG in both zero-shot and few-shot settings. For example, on Stark-Prime, its ROUGE-1 F1 score (5.17 vs. 11.12 zero-shot) lags significantly behind KAR, This indicates a weaker ability to handle complex relational constraints, particularly when the answer's attributes are embedded in the query. Its main strength lies in interpretable subgraph selection using PCST, which enhances explainability and helps mitigate hallucinations. (5)TOG performs moderately in zero-shot settings but shows strong gains in few-shot scenarios, achieving top-tier ROUGE-1 F1 scores like 37.11 on Riteck-ADint and 36.43 on Stark-Prime. This highlights its ability to leverage demonstrations to guide accurate reasoning over knowledge graphs, especially in complex biomedical tasks.

In the setting of supervised fine tuning, GCR achieves the best overall performance across all three medical benchmarks in the supervised setting, with scores like 57.28 ROUGE-1 F1 on ADint and 49.72 on Stark-Prime, demonstrating its strength in generating faithful, KG-grounded answers. However, GCR relies on pre-constructed KG-Trie indices. We found that GNN-RAG achieves better performance on the RiTeK-PharmKG and RiTeK-ADint datasets, demonstrating its ability to retrieve relevant path information from the graph. However, since it primarily relies on shortest paths, it may overlook critical reasoning information embedded in more complex or indirect graph structures.

5.3 Analysis

5.3.1 Effect of Different LLMs on Retriever Effectiveness

In this part, we analyze the influence of different LLMs on the retrievers. Table 5 presents the performance of three retrieval settings, G-retriver, GNN-RAG, and without retriever, in three LLMs of the backbone: Llama 3.1 8b, Llama2-chat-7b, and Biomixtral 7b, on three datasets. Overall, G-retriver consistently outperforms other approaches across most metrics, particularly in Rouge-1 F1 scores. For instance, on RiTeK-ADint, G-retriver with Llama 3.1 8b achieves the highest F1 score of 56.87, while the GNN-RAG and no retriever baselines lag behind. Similarly, G-retriver reaches 55.02 F1 on Biomixtral for the same dataset, showcasing its robustness across model sizes. In contrast, GNN-RAG shows variable performance, sometimes underperforming even compared to the no-retriever baseline, such as on Stark-Prime using Biomixtral. The "w/o retriever" baseline, representing an LLM without retrieval augmentation, performs surprisingly well in some settings, indicating that strong LLMs alone can capture a sig-

		llama 3.1 8b						llama2-chat-7b						Biomixtral 7b					
		Exact Match			Rouge-1			Exact Match			Rouge-1			Exact Match			Rouge-1		
	Approach	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
RiTeK-PharmKG	G-retriver (He et al., 2024)	**36.97**	**46.07**	**38.31**	33.04	47.03	38.41	38.71	37.11	37.62	39.78	39.18	39.31	**43.01**	**41.59**	**42.01**	**43.95**	**42.69**	**43.10**
	GNN-RAG	33.21	43.01	37.31	21.00	44.89	26.00	**50.78**	**49.28**	**49.72**	**51.66**	50.29	**50.73**	39.93	39.12	39.26	41.69	40.89	41.08
	w/o retriever	32.45	43.40	34.23	**47.60**	**46.59**	**46.84**	38.91	37.63	38.02	40.57	39.31	39.72	41.49	39.43	40.05	41.25	41.06	40.99
RiTeK-ADint	G-retriver (He et al., 2024)	**50.83**	**50.07**	**50.31**	**57.34**	**56.61**	**56.87**	47.93	47.16	47.41	54.68	54.00	54.24	48.34	47.48	47.75	55.58	54.73	55.02
	GNN-RAG	40.88	40.90	40.43	44.43	45.01	45.49	**51.04**	**50.59**	**50.55**	**56.49**	**56.09**	**56.09**	**50.83**	**50.07**	**50.31**	**57.34**	**56.61**	**56.87**
	w/o retriever	49.59	48.48	48.82	55.23	54.29	54.61	46.58	45.82	46.06	51.66	49.91	46.47	49.79	48.93	49.20	56.43	53.80	54.15
Stark-Prime	G-retriver (He et al., 2024)	**16.14**	**16.47**	**14.11**	17.21	27.86	19.21	10.15	8.45	8.17	21.75	18.08	18.40	12.22	**11.54**	10.58	23.15	21.37	20.72
	GNN-RAG	7.81	16.67	9.35	**18.13**	**27.50**	**19.65**	**16.00**	**15.04**	**14.50**	**24.78**	**23.51**	**22.99**	11.20	10.31	10.65	17.98	18.09	18.32
	w/o retriever	12.96	14.99	11.91	16.80	25.75	18.41	11.77	10.38	9.68	20.65	21.59	17.66	**12.96**	11.12	**10.73**	**24.83**	**21.59**	**21.68**

Table 5: Performance of different retrieval models across backbone LLMs. "w/o retriever" denotes an LLM without retrieval augmentation.

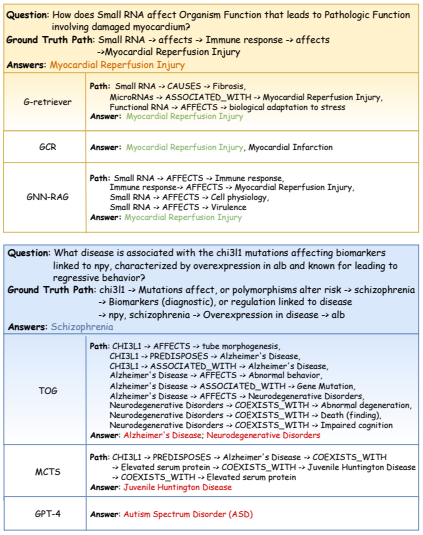

Figure 2: A case study on RiTeK

nificant amount of relevant knowledge. For example, on RiTeK-ADint with Biomixtral, it achieves a Rouge-1 F1 score of 54.15, close to the GNN-RAG. However, in most cases, retrieval-augmented methods still yield superior performance. Notably, Biomixtral 7b tends to outperform the other two LLMs when combined with retrieval, especially in recall and F1. These results suggest that both the choice of retrieval strategy and the backbone LLM significantly impact end-task performance.

5.3.2 Case Study of Path and Answer Quality

We conduct a qualitative analysis to compare the reasoning paths and predicted answers from different retrieval models on two biomedical question-answering examples. As shown in Figure 2, all models successfully predicted the correct answer *Myocardial Reperfusion Injury* in the first case, although their reasoning paths varied in granularity and relevance. G-RETRIEVER and GNN-RAG produced informative multi-hop paths that partially overlapped with the ground truth.

In contrast, for the second question involving *CHI3L1* and schizophrenia, only the ground truth path led to the correct answer. All baseline models failed: TOG and MCTS generated incorrect reasoning chains centered around *Alzheimer's Disease* and *Juvenile Huntington Disease*, while GPT-4 hallucinated *Autism Spectrum Disorder*. These errors reveal the challenge of modeling rare or indirect biomedical associations, especially when entity relations involve subtle phenotypic markers. This case highlights the importance of precise multi-hop reasoning and clinically aligned retrieval in semi-structured biomedical graphs.

6 Conclusion

We present RiTeK, the first dataset specifically designed to evaluate the capability of models in handling complex reasoning over textual knowledge graphs (TKGs). This dataset offers diverse topological structures, relational types, entity types, and queries that integrate relational and textual information, requiring sophisticated reasoning across TKGs. RiTeK also includes rich textual descriptions for each node. To ensure the authenticity and accuracy of the queries, medical experts performed stringent validation. RiTeK sets a new standard for evaluating real-world retrieval systems. We evaluated 11 retrieval models on our benchmark dataset. Our experiments on RiTeK reveal significant challenges faced by current models in effectively handling both textual and relational information, especially under complex topological structures involving intricate relations and entities. RiTeK paves the way for future research aimed at advancing retrieval systems by emphasizing the need to enhance reasoning capabilities, particularly in retrieving complex reasoning paths under answer attribute constraints.

7 Limitations

RiTeK is currently limited to queries that involve only a single topic entity and rely solely on the textual and structural information present in the graph. Future work should explore the inclusion of multiple topic entities and incorporate additional modalities, such as images, to enable a more comprehensive and robust information retrieval system.

Although we employed four medical experts for human evaluation, increasing the number of qualified domain experts would improve the statistical significance and robustness of our findings. Future work should consider expanding the pool of experts and addressing issues of fairness, and potential biases inherent in LLMs.

References

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

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers*, pages 2503–2514.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Guillaume Maurice Jean-Bernard Chaslot Chaslot. 2010. Monte-carlo tree search.

Arman Cohan and Nazli Goharian. 2016. Revisiting summarization evaluation for scientific articles. *arXiv preprint arXiv:1604.00400*.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Mingchen Li and Lifu Huang. 2023. Understand the dynamic world: An end-to-end knowledge informed framework for open domain entity state tracking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 842–851.

Mingchen Li and Shihao Ji. 2022. Semantic structure based query graph prediction for question answering over knowledge graph. *arXiv preprint arXiv:2204.10194*.

László Lovász. 1993. Random walks on graphs: A survey. In *Combinatorics, Paul Erdős is Eighty*, volume 2, pages 1–46. János Bolyai Mathematical Society.

Linhao Luo, Zicheng Zhao, Chen Gong, Gholamreza Haffari, and Shirui Pan. 2024. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*.

Costas Mavromatis and George Karypis. 2024. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*.

P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II 16*, pages 210–218. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis N Ioannidis, Karthik Subbian, Jure Leskovec, and James Zou. 2024a. Avatar: Optimizing llm agents for tool-assisted knowledge retrieval. *arXiv preprint arXiv:2406.11200*.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis N Ioannidis, Karthik Subbian, James Zou, and Jure Leskovec. 2024b. Stark: Benchmarking llm retrieval on textual and relational knowledge bases. *arXiv preprint arXiv:2404.13207*.

Yu Xia, Junda Wu, Sungchul Kim, Tong Yu, Ryan A Rossi, Haoliang Wang, and Julian McAuley. 2024. Knowledge-aware query expansion with large language models for textual and relational retrieval. *arXiv preprint arXiv:2410.13765*.

Yongkang Xiao, Yu Hou, Huixue Zhou, Gayo Diallo, Marcelo Fiszman, Julian Wolfson, Li Zhou, Halil Kilicoglu, You Chen, Chang Su, et al. 2024. Repurposing non-pharmacological interventions for alzheimer's disease through link prediction on biomedical literature. *Scientific reports*, 14(1):8693.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Zonghai Yao, Zihao Zhang, Chaolong Tang, Xingyu Bian, Youxia Zhao, Zhichao Yang, Junda Wang, Huixue Zhou, Won Seok Jang, Feiyun Ouyang, et al. 2024b. Medqa-cs: Benchmarking large language models clinical skills using an ai-sce framework. *arXiv preprint arXiv:2410.01553*.

Wen Tau Yih, Matthew Richardson, Chris Meek, Ming Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shuangjia Zheng, Jiahua Rao, Ying Song, Jixian Zhang, Xianglu Xiao, Evandro Fei Fang, Yuedong Yang, and Zhangming Niu. 2021. Pharmkg: a dedicated knowledge graph benchmark for bomedical data mining. *Briefings in bioinformatics*, 22(4):bbaa344.

A Appendix

A.1 TKG resources

Ensembl [4], [5], and Mondo Disease Ontology [6].

A.2 Medical textual knowledge graph construction

We construct two medical TKGs based on PharmKG (Zheng et al., 2021) and ADint (Xiao et al., 2024), as the increased number of entity and relation types introduces significant challenges for path retrieval in the question answering over textual knowledge graph. We present the statistics of the relational structure in Table 1 and introduce each TKG as follows:

PharmKG Textual Knowledge Graph: We leverage the existing medical knowledge graph PharmKG (Zheng et al., 2021) which is a multi-relational, attribute-rich biomedical knowledge graph (KG) constructed from six publicly available databases that provide high-quality structured information. These databases include OMIM, DrugBank, PharmGKB, Therapeutic Target Database (TTD), SIDER, and HumanNet. PharmKG consists of over 500,000 distinct interconnections between genes, drugs, and diseases, encompassing 29 types of relationships within a vocabulary of approximately 8,000 disambiguated entities. To enhance the entity attributes, we incorporate textual details from various databases, including Ensembl, UMLS,

[4] https://useast.ensembl.org/index.html
[5] https://www.nlm.nih.gov/research/umls/index.html
[6] https://mondo.monarchinitiative.org/

and Mondo Disease Ontology, as supplementary data sources.

ADInt Textual Knowledge Graph: ADInt(Xiao et al., 2024) is a comprehensive knowledge graph (KG) constructed from biomedical literature, focusing on non-pharmacological interventions (NPI) and their associations with Alzheimer's disease (AD). ADInt includes 162,212 entities spanning 113 UMLS semantic types, which, upon further classification, consist of 25,604 drugs, 16,474 diseases, 46,060 genes and proteins, 2,525 dietary supplements (DS), and 128 complementary and integrative health (CIH) interventions. Moreover, ADInt contains 1,017,284 triples, capturing 15 distinct relation types, offering a rich dataset for exploring the intricate relationships between NPIs and AD. Same as PharmKG, we also incorporate textual details from various databases, including Ensembl, UMLS, and Mondo Disease Ontology, as supplementary data sources.

A.3 Thr prompt of Combining Textual and Relational Information

You are a creative assistant tasked with generating natural, diverse, and realistic queries by combining textual properties and relational templates. Write the query from the perspective of a <persona>, ensuring it is concise, human-like, and paraphrased while retaining the original meaning.

Consider the following characteristics for the persona:

- Doctor: Formulate direct and practical questions aimed at diagnosing and treating. These questions should focus on side effects, symptoms, complications, and other clinically relevant aspects.
- Medical Scientist: Generate detailed and specific questions reflecting the complexity of scientific inquiry. These questions should explore etiology, pathophysiology, genetic factors, pathways, proteins, or molecular functions.
- Patient: Create straightforward questions that avoid professional medical terminology. These questions should focus on practical concerns, such as symptoms, effects, inheritance, or other relatable aspects, and may include more context from daily life.

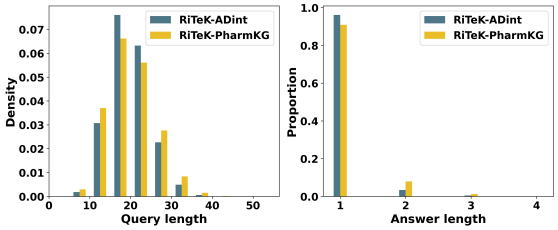

Figure 3: Distribution of query lengths and answer lengths on RiTeK-ADint and RiTeK-PharmKG datasets

Textual Properties: [<input_textual_properties>]

Relational Templates: [<input_relational_templates>]

Persona: <input_persona> (e.g., Doctor, Medical Scientist, Patient)

Ensure the query is realistic and diverse, leveraging flexibility in how the textual and relational elements are presented. Avoid directly copying the input phrases; instead, paraphrase them while retaining their original meaning. Please output only the generated query without any additional comments or explanations.

A.4 Data Analysis of query length and answer length

We analyzed the distribution of query lengths (i.e., the number of words in each query) to assess the complexity of the queries and the amount of information they contain. As shown in the Figure 3, the query lengths range from 5 to 40 words, with approximately 69% and 61% of queries in the two datasets having lengths between 15 and 25 words.

Then, we analyzed the proportion of ground truth answers associated with each query. Generally, the more ground truth answers there are, the less precise the textual requirements in the query tend to be. To increase the difficulty of the question-answering task, we filtered out queries with too many ground truth answers during the dataset creation process, retaining only those with a maximum of three ground truth answers. In both datasets, over 90% of queries have a single ground truth answer, indicating that our queries are enriched with detailed textual information from entity attributes. This introduces more challenges when developing new graph retrieval methods

B Relational Template

B.1 RiTeK-PharmKG

1. Gene -> [Production by cell population] -> Gene

2. Gene -> [Enhance response, or activate, stimulate] -> Gene
3. Gene -> [Relationships involving regulation and pathways] -> Gene
4. Gene -> [Binding, ligand] -> Gene
5. Gene -> [Affects expression/production] -> Gene
6. Gene -> [Gene-Gene] -> Gene
7. Chemical -> [Chemical-Chemical] -> Chemical
8. Disease -> [Ancestors of disease] -> Disease
9. Disease -> [Associations between diseases] -> Disease
10. Gene -> [Interactions] -> Chemical
11. Chemical -> [Interactions] -> Gene
12. Gene -> [Interactions] -> Gene
13. Gene -> [Interactions] -> Disease
14. Gene -> [Drug targets] -> Disease
15. Gene -> [Role in pathogenesis, or promotes progression] -> Disease
16. Gene -> [Mutations affect, or polymorphisms alter risk] -> Disease
17. Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene
18. Disease -> [Overexpression in disease] -> Gene
19. Chemical -> [Treatment or therapy] -> Disease
20. Chemical -> [Side effect or adverse event] -> Disease
21. Chemical -> [Inhibits cell growth] -> Disease
22. Chemical -> [Role in pathogenesis] -> Disease
23. Chemical -> [Prevents, suppresses, or alleviates, reduces] -> Disease
24. Disease -> [Biomarkers (progression)] -> Chemical
25. Chemical -> [Agonism, activation, or antagonism, blocking] -> Gene
26. Chemical -> [Binding, ligand] -> Gene
27. Chemical -> [Affects expression/production] -> Gene
28. Chemical -> [Inhibits] -> Gene
29. Gene -> [Transport, channels] -> Chemical
30. Gene -> [Metabolism, pharmacokinetics] -> Chemical
31. Gene -> [Enzyme activity] -> Chemical
32. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Drug targets] -> Disease
33. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Role in pathogenesis, or promotes progression] -> Disease
34. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Mutations affect, or polymorphisms alter risk] -> Disease
35. Gene -> [Relationships involving regulation and pathways] -> Gene -> [Binding, ligand] -> Gene
36. Gene -> [Binding, ligand] -> Gene -> [Affects expression/production] -> Gene
37. Gene -> [Interactions] -> Gene -> [Interactions] -> Chemical
38. Gene -> [Transport, channels] -> Chemical -> [Agonism, activation, or antagonism, blocking] -> Gene
39. Gene -> [Metabolism, pharmacokinetics] -> Chemical -> [Binding, ligand] -> Gene
40. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Enhance response, or activate, stimulate] -> Gene
41. Gene -> [Interactions] -> Chemical -> [Treatment or therapy] -> Disease
42. Gene -> [Interactions] -> Chemical -> [Side effect or adverse event] -> Disease
43. Gene -> [Interactions] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene

44. Chemical -> [Treatment or therapy] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene

45. Disease -> [Associations between diseases] -> Disease -> [Ancestors of disease] -> Disease

46. Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene

47. Gene -> [Interactions] -> Gene -> [Transport, channels] -> Chemical

48. Gene -> [Metabolism, pharmacokinetics] -> Chemical -> [Binding, ligand] -> Gene

49. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Drug targets] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene

50. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Mutations affect, or polymorphisms alter risk] -> Disease -> [Overexpression in disease] -> Gene

51. Gene -> [Transport, channels] -> Chemical -> [Agonism, activation, or antagonism, blocking] -> Gene -> [Binding, ligand] -> Chemical

52. Gene -> [Metabolism, pharmacokinetics] -> Chemical -> [Binding, ligand] -> Gene -> [Inhibits] -> Chemical

53. Gene -> [Interactions] -> Chemical -> [Treatment or therapy] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene

54. Gene -> [Interactions] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene -> [Transport, channels] -> Chemical

55. Gene -> [Role in pathogenesis, or promotes progression] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene -> [Metabolism, pharmacokinetics] -> Chemical

56. Chemical -> [Agonism, activation, or antagonism, blocking] -> Gene -> [Drug targets] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene

57. Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene -> [Role in pathogenesis, or promotes progression] -> Disease

58. Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene -> [Metabolism, pharmacokinetics] -> Chemical -> [Side effect or adverse event] -> Disease

59. Gene -> [Production by cell population] -> Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Relationships involving regulation and pathways] -> Gene

60. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Binding, ligand] -> Gene -> [Affects expression/production] -> Gene

61. Gene -> [Relationships involving regulation and pathways] -> Gene -> [Gene-Gene] -> Gene -> [Binding, ligand] -> Gene

62. Gene -> [Interactions] -> Gene -> [Interactions] -> Gene -> [Transport, channels] -> Chemical

63. Gene -> [Interactions] -> Gene -> [Interactions] -> Gene -> [Metabolism, pharmacokinetics] -> Chemical

64. Gene -> [Enhance response, or activate, stimulate] -> Gene -> [Mutations affect, or polymorphisms alter risk] -> Disease -> [Overexpression in disease] -> Gene

65. Gene -> [Enzyme activity] -> Chemical -> [Affects expression/production] -> Gene -> [Chemical-Chemical] -> Chemical

66. Gene -> [Interactions] -> Chemical -> [Role in pathogenesis] -> Disease -> [Overexpression in disease] -> Gene

67. Chemical -> [Side effect or adverse event] -> Disease -> [Biomarkers (diagnostic), or regulation linked to disease] -> Gene -> [Mutations affect, or polymorphisms alter risk] -> Disease

68. Chemical -> [Inhibits cell growth] -> Disease -> [Overexpression in disease] -> Gene -> [Role in pathogenesis, or promotes progression] -> Disease

B.2 RiTeK-ADint

1. Amino Acid, Peptide, or Protein -> [affects] -> Cell Function
2. Amino Acid, Peptide, or Protein -> [affects] -> Disease or Syndrome
3. Amino Acid, Peptide, or Protein -> [causes] -> Anatomical Abnormality
4. Amino Acid, Peptide, or Protein -> [interacts with] -> Pharmacologic Substance
5. Anatomical Abnormality -> [affects] -> Organ or Tissue Function
6. Anatomical Abnormality -> [complicates] -> Disease or Syndrome
7. Anatomical Abnormality -> [manifestation of] -> Genetic Function
8. Antibiotic -> [affects] -> Molecular Function
9. Antibiotic -> [causes] -> Pathologic Function
10. Antibiotic -> [disrupts] -> Cell Component
11. Antibiotic -> [treats] -> Disease or Syndrome
12. Bacterium -> [causes] -> Cell or Molecular Dysfunction
13. Bacterium -> [interacts with] -> Human
14. Biologically Active Substance -> [affects] -> Organism Function
15. Biologically Active Substance -> [causes] -> Injury or Poisoning
16. Biologically Active Substance -> [disrupts] -> Gene or Genome
17. Body Part, Organ, or Organ Component -> [produces] -> Immunologic Factor
18. Cell Component -> [affects] -> Molecular Function
19. Cell Component -> [produces] -> Nucleic Acid, Nucleoside, or Nucleotide
20. Cell Function -> [affects] -> Mental or Behavioral Dysfunction
21. Cell Function -> [produces] -> Biologically Active Substance
22. Cell or Molecular Dysfunction -> [affects] -> Neoplastic Process
23. Cell or Molecular Dysfunction -> [manifestation of] -> Pathologic Function
24. Cell -> [produces] -> Organic Chemical
25. Congenital Abnormality -> [affects] -> Virus
26. Congenital Abnormality -> [manifestation of] -> Organism Function
27. Diagnostic Procedure -> [affects] -> Genetic Function
28. Disease or Syndrome -> [affects] -> Organ or Tissue Function
29. Disease or Syndrome -> [associated with] -> Therapeutic or Preventive Procedure
30. Disease or Syndrome -> [manifestation of] -> Cell or Molecular Dysfunction
31. Finding -> [manifestation of] -> Pathologic Function
32. Gene or Genome -> [produces] -> Amino Acid, Peptide, or Protein
33. Genetic Function -> [affects] -> Human
34. Genetic Function -> [produces] -> Cell Component
35. Hazardous or Poisonous Substance -> [affects] -> Mental or Behavioral Dysfunction
36. Hazardous or Poisonous Substance -> [disrupts] -> Organ or Tissue Function
37. Health Care Activity -> [affects] -> Disease or Syndrome
38. Human -> [interacts with] -> Human
39. Immunologic Factor -> [affects] -> Pathologic Function
40. Indicator, Reagent, or Diagnostic Aid -> [interacts with] -> Hazardous or Poisonous Substance
41. Injury or Poisoning -> [disrupts] -> Genetic Function
42. Medical Device -> [treats] -> Mental or Behavioral Dysfunction

43. Mental or Behavioral Dysfunction -> [affects] -> Organism Function
44. Molecular Function -> [affects] -> Virus
45. Neoplastic Process -> [affects] -> Bacterium
46. Neoplastic Process -> [associated with] -> Neoplastic Process
47. Nucleic Acid, Nucleoside, or Nucleotide -> [interacts with] -> Immunologic Factor
48. Organ or Tissue Function -> [produces] -> Immunologic Factor
49. Organic Chemical -> [affects] -> Pathologic Function
50. Organic Chemical -> [interacts with] -> Pharmacologic Substance
51. Organism Function -> [affects] -> Disease or Syndrome
52. Pathologic Function -> [associated with] -> Therapeutic or Preventive Procedure
53. Pathologic Function -> [manifestation of] -> Organ or Tissue Function
54. Pharmacologic Substance -> [affects] -> Genetic Function
55. Pharmacologic Substance -> [treats] -> Sign or Symptom
56. Sign or Symptom -> [manifestation of] -> Genetic Function
57. Therapeutic or Preventive Procedure -> [affects] -> Neoplastic Process
58. Virus -> [interacts with] -> Human

C Relational MCTS

C.1 Motivation and Approach Overview

Monte Carlo Tree Search (MCTS) has made significant advancements in mathematics, as it dynamically explores and evaluates potential solutions, balancing exploration and exploitation to optimize decision-making in complex, high-dimensional spaces. However, the effectiveness of MCTS in TKGs has not yet been explored. To address this gap, this paper investigates the effectiveness of MCTS and proposes an improved version *Relational MCTS* that dynamically retrieves relational

	Shannon Entropy	Type-Token Ratio
	Medical domain	
RiTeK-ADint	10.04	0.187
RiTeK-PharmKG	9.61	0.157
STARK-PRIME	9.63	0.143
	General domain	
STARK-AMAZON	10.39	0.179
STARK-MAG	10.25	0.180

Table 6: Query diversity measurement

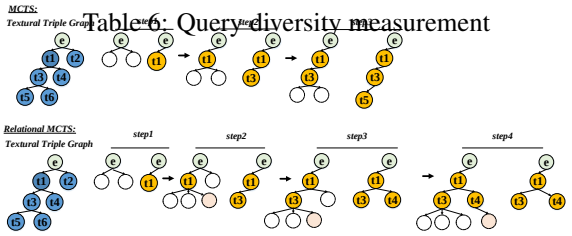

Figure 4: Expansion progress of MCTS and Relational MCTS. The pink shallow circle refers to the $Stop$ sign. Each node tn refers to a triple.

paths from the TKGs for the special query. Specifically, it involves four steps: 1) To improve the retrieval efficiency, we begin our path retrieval by creating the textual triple graph (TTG) for each textual KG. 2) Subsequently, the node embeddings in the TTG and the input query embedding are computed using a pre-trained language model, such as a SentenceBERT (Baek et al., 2023). Notably, each node embedding includes its associated textual information. These embeddings will be used to calculate the reward in the MCTS and Relational MCTS. 3) Proposed Relation MCTS is used to search the relational path information. 4) Finally, the retrieved relational path information will be verbalized and fed into the LLM, such as GPT-4, along with the input query to predict the final answers. Please refer to Appendix **??** for detailed information on steps 2 and 3.

C.2 MCTS vs Relational MCTS

In traditional MCTS, a trajectory typically extends from the root to a terminal node. However, in the context of QA tasks over textual knowledge graphs, many relational templates incorporate constraints, which pose challenges for standard MCTS. Furthermore, the depth of the reasoning path can vary depending on the specific question being posed; for example, some queries may require a reasoning depth of three hops, while others may only require two. Consequently, MCTS is hard to appropriately recognize or adapt to the required depth for each individual query. To address this issue, we propose an enhanced MCTS specifically designed to retrieve the relevant inference path information for the input query, which we refer to as Relational MCTS.

Same as traditional MCTS, starting from the

initial topic entity e, we perform multiple searches consisting of selection, expansion, simulations, and back-propagation. The key distinction between MCTS and Relational MCTS lies in the expansion phase, as illustrated in Figure 4. In the expansion phase of MCTS, the next move is typically selected from the set of children of the current node. For instance, in MCTS, the candidate moves from node $t1$ are $t3$ and $t4$. In contrast, in Relational MCTS, the candidate moves from node $t1$ include $t3$, $t4$, a $Stop$ action, and the sibling nodes of $t1$. At the same depth level, if two sibling nodes exist, such as in Step 4 of Relational MCTS, the next move is selected from the children of these sibling nodes. For example, when node $t4$ is selected, its candidate moves are $t5$, $t6$, and the $Stop$ action. The inclusion of the $Stop$ action allows Relational MCTS to automatically halt and generate a relevant relational path (1 or 2 hops) for the input query.

Selection Starting from the topic entity, the algorithm navigates through promising child nodes based on specific strategies (e.g., UCT), continuing until a leaf node is reached.

Expansion: At the leaf node, unless it represents a terminal state of the game, one or more feasible new child nodes are added to illustrate potential future moves.

Simulation or Evaluation: From the newly added node, the algorithm conducts random simulations— often termed "rollouts"—by selecting moves arbitrarily until a game's conclusion is reached, thereby evaluating the node's potential.

Backpropagation: Post-simulation, the outcome (win, loss, or draw) is propagated back to the root, updating the statistical data (e.g., wins, losses) of each traversed node to inform future decisions.

C.3 Answer Generation

In the progress of inference path generation, we encountered challenges in recognizing the internal entity within the inference path. This difficulty arose because some internal entities were not present in the query. Therefore, after obtaining the reasoning paths using Relational MCTS, we utilized only the relations within these paths that form the relational information. We then verbalized this information and provided it as input to a downstream LLM, such as GPT-4 or LLaMA.