Counter-intuitive: Large Language Models Can Better Understand Knowledge Graphs Than We Thought

Anonymous ACL submission

Abstract

Although the method of enhancing large language models' (LLMs') reasoning ability and reducing their hallucinations through the use of knowledge graphs (KGs) has received widespread attention, the exploration of how to enable LLMs to integrate the structured knowledge in KGs on-the-fly remains inadequate. Researchers often co-train KG embeddings and LLM parameters to equip LLMs with the ability of comprehending KG knowledge. However, this resource-hungry training paradigm significantly increases the model learning cost and is also unsuitable for non-open-source, black-box LLMs. In this paper, we employ complex question answering (CQA) as a task to assess the LLM's ability of comprehending KG knowledge. We conducted a comprehensive comparison of KG knowledge injection methods (from triples to natural language text), aiming to explore the optimal prompting method for supplying KG knowledge to LLMs, thereby enhancing their comprehension of KG. Contrary to our initial expectations, our analysis revealed that LLMs effectively handle messy, noisy, and linearized KG knowledge, outperforming methods that employ well-designed natural language (NL) textual prompts. This counter-intuitive finding provides substantial insights for future research on LLMs' comprehension of structured knowledge.

1 Introduction

By pretraining on vast amounts of data from various information sources, Large Language Models (LLMs), such as GPT-3 (Brown et al., 2020), ChatGPT, GPT-4 and LLaMA (Touvron et al., 2023), provide both regular users and researchers with a comprehensive and extensive foundational tool, which store a significant amount of information and can perform a wide range of tasks. However, when dealing with domain-specific knowledge, LLM often struggles to answer questions related to specialized knowledge or even generates statements that are factually incorrect – a phenomenon known as hallucination (Ji et al., 2023). In the current research community, the factual knowledge within a multitude of domains is stored in the form of knowledge graphs (KGs) respectively (Rizun et al., 2019). Consequently, researchers have conducted extensive research on the combination of KGs and LLMs.

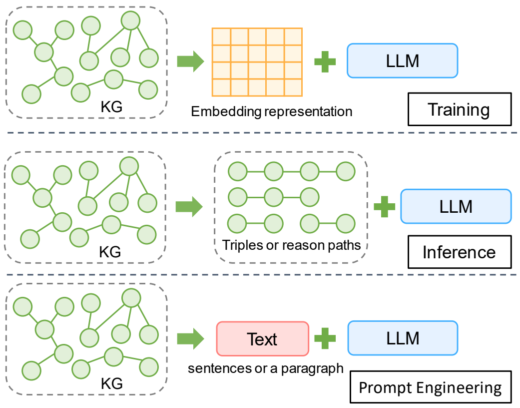


Figure 1: The method of KG-enhanced LLM.

As shown in Figure 1, KG-enhanced LLM is categorized into two primary stages: the training stage and the inference stage (Pan et al., 2024). Encoding the KG and injecting these distributed representations as a trainable parameter into LLM enables it to understand the semantic meaning hidden in structured knowledge (Zhang et al., 2019; Rosset et al., 2020). However, the increasing size of current models causes a significant resource-hungry problem. Moreover, training LLMs with billions of parameters may encounter limitations, especially when the model structure, training data, or training methods are not publicly accessible (Ufuk, 2023). In the enhancement stage, the integration of graph neural networks with path reasoning methods is essential for identifying potential reasoning paths. The performance of LLMs depends on the KG inference module (maybe need training). However, this approach lacks the consideration for leveraging LLMs in the process of KG understanding and reasoning.

Consequently, researchers have recently shown a keen interest in exploring how to supply high-quality relevant knowledge to pre-trained LLMs via constructing optimal prompts, thereby facilitating the model's comprehension of KG (Sorensen et al., 2022; White et al., 2023; Li et al., 2023; Wen et al., 2023; Hu et al., 2024). This more lightweighted approach involves converting the KG into linearly represented triples, reasoning paths, or natural language (NL) textual representations (sentences or a paragraph) and concatenating them to the input prompt to query the LLMs. To compensate for the loss of structured information, researchers strive to generate NL texts by effectively organizing structured knowledge (Sun et al., 2020; Brate et al., 2022). However, the generation of KG-to-text itself poses a significant challenge when dealing with sub-graphs containing numerous triples (tens or even hundreds).

In this work, we use complex question answering (CQA) based on KG to assess the LLM's understanding of externally injected KG knowledge. When LLM answers questions, it always needs to acquire the latest knowledge from external sources to assist in precise answering. Understanding the externally injected information, integrating it with the information inherent to LLM, are essential abilities for solving QA tasks. Therefore, choosing QA tasks to examine the model's knowledge understanding capability is the mainstream choice (Lan et al., 2021; Pan et al., 2024). The answers to such questions often extend beyond simply listing entity aliases; they involve tasks such as counting, sorting, comparing, judging authenticity, and other situations that require LLM's reasoning abilities (Tan et al., 2023).

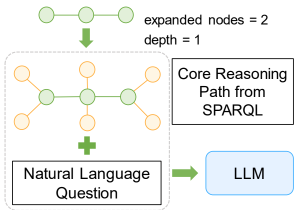
Following the intuitive cognition, compared with the structured data, we hypothesize that the NL textual data should be easier for LLMs to understand since it better aligns with the corpus (mostly written in NL) used for LLM pretraining (Zhou et al., 2023). To validate this hypothesis, we propose the following research questions: 1) How does sub-graph injection at varying scales impact LLM's reasoning ability in KGQA? 2) What is LLM's reasoning performance in a completed KG? 3) Is fluently organized knowledge of natural language texts superior to disorganized structured knowledge? 4) How robust are LLMs when dealing with sub-graphs from noisy or incomplete KG? 5) What factors should be considered when designing the prompt framework to leverage KG as external knowledge?

Based on the aforementioned questions, we conducted an evaluation of LLM's comprehension of KG across various dimensions using a unified experimental metric. Our experiments produce intriguing findings when the LLM with billions of parameters:

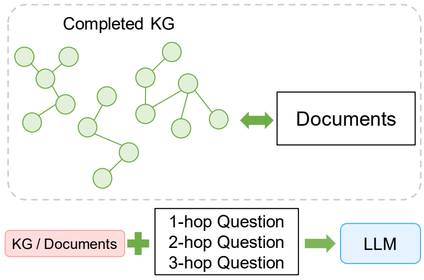
- LLMs consistently outperform well-crafted and fluent text prompts when presented with disorganized, noisy, and abstract knowledge prompts. This indicates their proficiency in organizing and understanding structured knowledge, which is beyond our expectations.
- Superfluous or irrelevant information does not necessarily degrade the reasoning capabilities of LLMs. They can enhance accuracy beyond expectations by discarding irrelevant information or harnessing relevant details.
- Even the marginally relevant knowledge can bolster the reasoning performance of LLMs. Experimental results show that the applicability of prompts with different knowledge-injection patterns varies for different LLMs. Prompts that perform well on certain models may not necessarily be effective on other models. Researchers need to conduct refined experiments to identify the knowledge-injection prompts that are more universally applicable.

2 Related Works

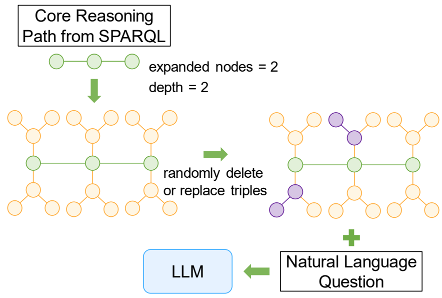
Extensive research conducts on injecting KG knowledge into LLM during training, enabling LLM to grasp the semantic meaning of the KG embeddings by co-training (Sun et al., 2019; Liu et al., 2020; Xiong et al., 2019; Su et al., 2021; He et al., 2021; Arora et al., 2022; Chen et al., 2022). ERNIE (Sun et al., 2019) is a notable approach that improves LLM by using knowledge masking techniques, including entity-level and phrase-level masking. K-BERT (Liu et al., 2020) recognizes the problem of knowledge noise caused by an overwhelming amount of triple input and suggests a soft position mechanism and a visible matrix to reduce this negative impact. Colake (Sun et al., 2020) suggests that relevant contextual knowledge can also boost the performance of LLM. Although these



(a) Expand the sub-graph by extracting the core reasoning path and utilize it as knowledge to support the LLM in QA. The expansion in the figure is the expansion of 2 nodes for each node of the core reasoning path, and the number of hop is 1.



(b) Employing a completed KG for generating multi-hop questions, we evaluate the performance of LLM in answering questions of varying complexity within knowledge.



(c) The core reasoning path is represented by green nodes, while yellow nodes represent extended nodes. Any replaced or deleted nodes are indicated in purple.

Figure 2: The impact of different KG on LLM.

methods have shown progress with relatively small-scale LLMs, their applicability to larger LLMs (such as GPT-3) poses challenges that require careful consideration of data processing, model structure, training methods, and other aspects. Furthermore, this line of approaches are not applicable to the untrainable LLMs (for instance ChatGPT and GPT-4[1]).

Triple-to-text generation has also been extensively studied as an alternative to KG comprehension. This line of work normally translates triple into NL text and then sends the NL text as external information to LLMs. To address the semantic gap between triples and natural language text, KBGen (Banik et al., 2012) employs a two-step approach containing content selection and surface realization to generate text. Similarly, (Zhu et al., 2019; Moryossef et al., 2019; Tang et al., 2022) treat such the triple-to-text conversion as a Seq2Seq task, utilizing the language model for the conversion. Graph neural networks (GNNs) are also employed for this generation task to extract structural information from KG sub-graphs (Zhao et al., 2020; Guo et al., 2020).

However, training the model to convert KG triples into NL text requires the text to fully cover the semantics of the triples, and the text needs to be fluent, natural, and easy for the model to understand. This introduces additional computational costs to the model and even worse, a sub-optimal translation model may exert a negative influence on the downstream tasks, i.e., error propagation.

[1] https://openai.com/gpt-4

3 Methods

This section presents our technical approach for evaluating the LLM's comprehension of KGs through the KG-based CQA task. In the KGQA task, whether it involves deducing answers from a multi-hop sub-graph related to the central entity or generating structured queries using semantic parsing, these operations necessitate the utilization of KG knowledge (in the form of triples). In cases where training LLM becomes impractical (for instance ChatGPT and GPT-4), it becomes necessary to investigate the efficacy of LLM in addressing problems involving KG at the textual level.

3.1 LLM's Understanding of Sub-graphs

We utilize SPARQL queries in the KGQA dataset to assist us in generating all reasoning paths. Specifically, we deployed a local start-point to retrieve all the variables in the constraints (namely in the "where" section) of the SPARQL. It is worth noting that originating from the subject entity of a question, multiple core reasoning paths exist in the KG, encompassing varying entities or relations, all leading to the identical answer node. We utilize all these multi-hop paths to conduct sub-graph expansion.

We anticipate that the LLM will be capable of providing answers by traversing such multi-hop sub-graphs along core reasoning paths that are associated with the answers. However, considering the constrained input length of the LLM, it becomes impractical to provide a fully populated KG as input. In order to quantitatively evaluate the impact of different scales of KG sub-graphs on QA, we propose a sub-graph generation method.

Figure 2a illustrates that the number of neighboring nodes and the number of hops expanded from each node on the core reasoning path are controlled by the parameters *expanded nodes* and *depth*. This controlled generation of sub-graphs allows us to precisely analyze the impact of KG scale on LLM. We have defined six levels of expansion methods as follows: (1) The sub-graph remains unexpanded, preserving only the core reasoning path. (2) Each node on the core path expands to a neighboring node with a 50% probability of retention. (3) Each node on the core path expands to a neighboring node. (4) Each node on the core path expands to two neighboring nodes, each at a distance of one hop. (5) Each node on the core path expands to a neighboring node, which is two hops away. (6) Each node on the core path expands to two neighboring nodes, both at a distance of two hops. According to this rule, levels (4) and (5) have the same number of nodes in the sub-graphs but exhibit different structures. This facilitates the observation of LLM's inference performance on multi-hop sub-graphs.

In order to inject knowledge into LLM, we refer to existing literature and devise five levels of KG injection methods: (1) Omitting KG sub-graphs (Without Triples). (2) Providing only unordered linearized triples (Triples). (3) Combining triples that share the same head and tail entities to form meta paths (Gao et al., 2020) (Meta Paths). (4) Utilizing heuristic rules to convert meta paths into natural language texts (Wang et al., 2022) (Natural Language Generated by Rules). (5) Employing a text generation model to convert meta paths into NL texts (Tang et al., 2022) (Fluent Natural Language). Different combinations of extension and injection modes can form a total of 25 possible combinations (Omitting KG sub-graphs is treated as one method, irrespective of the KG size). In the experimental section, we will thoroughly analyze the outcomes of these combinations.

3.2 Multi-hop Reasoning Capabilities of LLM

As discussed in Section 2, the NL text generated by the underperforming triple-to-text generation model will affect LLM's ability to understand KG and further perform CQA. To eliminate the impact of the Triple-to-text model on LLM, we leverage a Relation Extraction (RE) benchmark to generate QA pairs, in which the document and its relevant ground-truth triples are provided. Since the triples in each document can represent the logical structure of the entities, we consider them as a comprehensive knowledge graph mapping the document. Specifically, to generate QA pairs based on each document, we adopted the dataset construction method employed in LC-QuAD 2.0 (Dubey et al., 2019). We first filled the triples into many different templates and constructed questions, and then used ChatGPT to paraphrase the questions, ensuring the diversity and complexity of questions. Following this approach, as illustrated in Figure 2b, we generated 1-hop, 2-hop, and 3-hop questions for each document. This serves as the evaluation dataset to assess the LLM's ability to answer complex questions based on the ground-truth KGs.

Figure 2b illustrates the composition of the dataset. Additionally, apart from injecting the KG corresponding to the document, we firmly believe that the document itself represents fluent natural language text corresponding to this KG. By utilizing this dataset, we can eliminate any doubts regarding the model performance of KG-to-text generation, which may impact the sub-graph understanding evaluation task and hinder the effectiveness of the injection.

3.3 Robustness of LLM to Noisy Sub-graphs

In order to assess the capacity of LLM to comprehend noisy sub-graphs, we systematically sabotaged the external knowledge within the injecting sub-graphs. Specifically, we employ three approaches to alter the sub-graphs: (1) Nodes are proportionally deleted randomly. (2) Nodes are proportionally replaced with random irrelevant KG nodes. (3) All core reasoning paths in the sub-graph are removed. To ensure a smooth transition when replacing and deleting elements based on percentages, we employ the max expanded sub-graph(*expanded nodes*=2, *depth*=2, details can be found in Section 4.2). The operation of deletion and replacement is shown in Figure 2c, and its ratio ranges from 10% to 90%. The way to delete the core reasoning path is to delete all the green nodes in the graph. Deleting the core reasoning path can be viewed as a complete destruction of the path from the start-point node to the target node, preventing the utilization of other nodes as intermediate steps to reach the target.

3.4 KG Prompt Engineering

When it proves challenging to incorporate structured knowledge into the LLM through training, we turn to prevalent retrieval-augmented generation (RAG) techniques (Asai et al., 2023; Tang and Yang, 2024; Cuconasu et al., 2024; Yu et al., 2022). We explore a prompt method, treating triples as external data, to find the injection approach most suitable for LLMs.

We construct the training data using all triples and questions present in the dataset, employing a BERT-based cross-encoder to assess the relevance of the triples to the questions. The correlation score of a triple to a question is computed as follows:

$$Score = Cross_{BERT}(Triple \text{ [TRI] } Question), \quad (1)$$

where [TRI] is the special token that separates triples and questions. We can view the $Score$ as the probability of a certain triple appearing in the question's reason path.

During the inference, according to the equation 1, we use the cross-encoder model to compute $Score$, and select the top 100 triples that have the highest $Score$ from the KG as candidates. We perform the following operations on these triples to compose the `relevance` labels based on the $Score$: 1) Grouping: the triples are segregated into three groups based on two thresholds, namely the most relevant, moderately correlated, and low-correlation triples. The `relevance` label is the group a triple belongs to. 2) Ranking: we don't have `relevance` label for Ranking, instead, we rank the candidate triples in descending order based on the $Score$ and concatenate them in reverse order. 3) Scoring: the `relevance` label is $Score$. We append the `relevance` labels after the triples (illustrated in Table 6, Appendix A.2), explicitly informing the model about the confidence score of this input triple, and feed the triples together with the question into LLM to generate answers. We discuss model training and hyperparameter settings in Appendix A.2.

4 Experiment

Given the vast size of the training data, the external knowledge we provide to the LLM does not necessarily imply that the model has not previously learned this information. We do not want to create zero-shot knowledge that the model has never seen before. Instead, we want to use some false facts to evaluate the model's robustness to noisy knowledge. A prevalent scenario is that despite the model's understanding of certain knowledge, it still struggles with fact-intensive multi-hop reasoning tasks (Zhang et al., 2023). To address these challenges, techniques such as prompt engineering or RAG retain substantial significance for the LLMs. Our experiment aims to explore the five questions raised in the Introduction.

4.1 Experimental Setup

4.1.1 Evaluation metrics

Referring to ChatGPT's evaluation research for the KGQA task (Tan et al., 2023), we compare the string similarity between the LLM's predicted answers and the gold answers to the question, using an empirically set threshold of 0.7. The final score is determined by the proportion of questions answered correctly. We discuss the detailed information on input format in Appendix A.1.

4.1.2 Datasets and LLMs

Wikidata (Vrandečić and Krötzsch, 2014) is a large-scale, high-quality knowledge graph that is updated frequently. We carefully chose three KGQA datasets, i.e., QALD-7 (Usbeck et al., 2017), LC-QuAD 2.0 (Dubey et al., 2019), and KQAPro (Cao et al., 2020), that resort to SPARQL queries to retrieve knowledge from the underlying KG, Wikidata. The QALD-7 dataset has 215 train questions and 50 test questions written in NL. The LC-QuAD 2.0 dataset has 24k train questions and 6046 test questions. The KQAPro dataset consists of 94K train questions and 10k test questions.

From the training and test datasets of QALD-7, we filtered out questions whose reasoning paths did not meet our expansion rules, contained wrong answers or involved with single-hop SPARQL queries. After filtering, there are 64 questions left in QALD-7. For LC-QuAD 2.0 and KQAPro, we randomly sampled 2000 questions from each dataset, considering the high time cost when calling the interface of ChatGPT.

For evaluating LLM's ability to understand a completed KG, we chose DocRED (Yao et al., 2019). DocRED is a document-level RE dataset, where entities and relations in this dataset can be linked to Wikidata. The dataset consists of 5053 Wikipedia documents, each associated with a set of human-annotated KG triples. This dataset emphasizes cross-sentence reasoning, thus each document's mapping triples can form a complete small KG where entity nodes connect with each other through multi-hop relations. We consider this small KG as a complete structured representation of all entities and relations involved in the document. Following this approach, we select 800 documents

	Expanded Node Ratio	Expanded Nodes	Hop	Knowledge Injection Method														
				Without Triples			Unordered Triples			Meta Paths			Natural Language Generated by Rules			Natural Language		
				ChatGPT	Vicuna 7b	Vicuna 13b	ChatGPT	Vicuna 7b	Vicuna 13b	ChatGPT	Vicuna 7b	Vicuna 13b	ChatGPT	Vicuna 7b	Vicuna 13b	ChatGPT	Vicuna 7b	Vicuna 13b
QALD-7	0%	0	0	71.45	14.35	60.82	**84.01**	**74.59**	**78.02**	**84.01**	70.00	73.29	75.07	60.71	71.44	73.62	60.33	65.65
	33.33%	0.5	1	71.45	14.35	60.82	**84.00**	**66.44**	**79.42**	**84.00**	66.38	76.18	76.04	53.61	74.59	69.28	46.81	68.79
	50%	1	1	71.45	14.35	60.82	**84.01**	**72.41**	**79.90**	**84.01**	72.19	73.14	71.21	39.76	61.54	57.68	38.60	52.85
	66.67%	2	1	71.45	14.35	60.82	**84.01**	**67.34**	**81.11**	**84.01**	60.05	77.00	68.55	47.67	64.69	59.86	47.00	55.27
	66.67%	1	2	71.45	14.35	60.82	**84.01**	**58.65**	**78.21**	**84.01**	58.55	73.14	68.31	48.26	57.68	54.54	38.84	52.76
	85.72%	2	2	71.45	14.35	60.82	**82.56**	**65.70**	**77.00**	81.35	60.77	74.30	67.34	50.92	50.91	63.96	37.54	60.10
LC-QuAD 2.0	0%	0	0	16.42	2.60	13.47	**50.80**	**36.76**	**46.62**	50.29	33.47	39.53	23.28	11.32	17.34	21.12	10.04	16.88
	33.33%	0.5	1	16.42	2.60	13.47	**51.61**	**36.37**	**45.75**	50.16	31.88	39.12	26.98	14.13	21.10	23.32	10.54	18.12
	50%	1	1	16.42	2.60	13.47	**51.71**	**35.95**	**42.43**	48.55	30.73	38.22	24.47	12.12	20.47	20.49	10.79	17.70
	66.67%	2	1	16.42	2.60	13.47	**51.24**	**35.18**	**43.73**	47.57	29.38	36.29	23.13	12.59	18.91	19.34	9.74	17.04
	66.67%	1	2	16.42	2.60	13.47	**51.35**	**33.97**	**44.96**	48.82	30.11	37.66	23.85	11.52	19.69	21.00	9.44	17.27
	85.72%	2	2	16.42	2.60	13.47	**50.60**	**32.84**	**42.98**	47.23	28.53	36.20	21.80	10.80	18.03	17.94	8.92	15.14
KQAPro	0%	0	0	15.77	3.55	10.29	**49.33**	**23.10**	**36.69**	48.10	22.09	27.74	34.93	12.19	18.88	27.15	10.88	15.77
	33.33%	0.5	1	15.77	3.55	10.29	**51.45**	**25.39**	**33.33**	49.97	23.35	28.61	29.89	12.08	18.79	23.57	10.68	15.54
	50%	1	1	15.77	3.55	10.29	**52.54**	**25.64**	**31.60**	50.78	24.72	28.22	28.80	12.39	19.07	23.60	10.51	15.30
	66.67%	2	1	15.77	3.55	10.29	**52.46**	**28.47**	**31.04**	50.00	25.42	27.24	28.19	12.36	18.93	22.54	9.70	14.77
	66.67%	1	2	15.77	3.55	10.29	**54.19**	**26.85**	**32.72**	51.45	25.39	27.77	28.61	11.27	16.89	23.33	8.47	12.67
	85.72%	2	2	15.77	3.55	10.29	**54.03**	**27.63**	**32.75**	51.51	25.70	28.58	22.60	10.23	15.30	17.20	7.49	11.49

Table 1: The understanding of sub-graphs by LLMs.

(each map to a 3-hop sub-graph) and generate 1-hop, 2-hop, and 3-hop questions for each sub-graph (document). We use ChatGPT, Vicuna 7B, and 13B (Zheng et al., 2023) to evaluate all data, and all model parameters are fixed.

4.2 Results and Findings

4.2.1 How does sub-graph injection at varying scales impact LLM's reasoning ability in KGQA?

We initiate the expansion of the sub-graph by expanding the neighboring nodes of the core reasoning path in the KG. As illustrated in Figure 2a, we are using *depth* to indicate how many hops of the expansion will be, and *expanded nodes* to denote the number of neighbor nodes that are selected as the expanded nodes each hop, which is an iterative process. By regulating the scale of the sub-graph, we can ascertain the ratio of core reasoning triples amidst all the triples of the KG sub-graph. Table 1 presents the results of LLM's performance in CQA, considering various scale sub-graphs and different methods of knowledge injection.

In Table 1, the **bold** result denotes the maximum value for each model, observed row-wise, with unordered triples consistently outperforming other methods, including the NL text. The limited size of the QALD-7 dataset and its simplicity initially obscure the advantages of our approach. However, upon increasing the complexity of the questions and expanding the data scale (In LC-QuAD 2.0 and KQAPro), unordered triples demonstrated superior performance in knowledge injection. An underline signifies the maximum value of column-wise sub-graph expansion in each dataset. We find that a sub-graph with a high proportion of core inference paths does not necessarily yield the strongest model inference performance. Contrarily, the inclusion of irrelevant yet correct triples can actually enhance the model's performance. This observation aligns with findings from recent research (Cuconasu et al., 2024), which shows that irrelevant noise documents can improve the performance of RAG. Based on the results, we can draw the following conclusions: 1) Models with larger parameters always yield superior performance in that the 13B model generally outperforms the 7B model while ChatGPT exhibits the best performance. 2)LLM infused with any form of relevant knowledge consistently outperforms LLM without infusion by a significant margin. This demonstrates the importance of knowledge provision and highlights knowledge infusion as a cost-effective method. 3) Table 1 shows that when the size of the sub-graph is enlarged within the range of the LLM input length, the performance degradation is not obvious, and sometimes even increases. Furthermore, comparing the setting of '*expanded nodes*=2' with '*depth*=1' and the setting '*expanded nodes*=1' with '*depth*=2', the structure of the expanded nodes are different from each other. However, LLM is not sensitive to such the structural changes, but focuses on the knowledge of the triples.

4.2.2 What is LLM's reasoning performance in a completed KG?

In this experiment, we investigate the performance differences of the model when facing complex questions of varying difficulty levels, ranging from 1-Hop to 3-Hop. As described in Section 4.1.2, the triples mapping to each DocRED document are manually annotated, ensuring the completeness of sub-graphs. We evaluated 800 documents that meet the criteria for three-hop QA. We employed two methods of information injection: direct injection

DOCRED 800 Questions	ChatGPT	Vicuna 7b	Vicuna 13b
1-Hop Text	25.25	14.88	27.62
1-Hop Triple	**73.38**	**50.13**	**60.13**
2-Hop Text	14.25	9.50	15.37
2-Hop Triple	**19.88**	**11.00**	**16.38**
3-Hop Text	14.00	8.63	13.87
3-Hop Triple	**18.25**	**10.50**	**14.75**

Table 2: Multi-hop QA for completed KG.

of all triples in completed KG and direct injection of the document. The injected triples are also randomly shuffled. In the case of LLM, utilizing the given triples to deduce the structural information of the KG is essential to address questions. Similarly, the document's text exhibits a rigorous logical structure, yet it also encompasses an abundance of intricate details, thereby posing a non-trivial challenge for LLM in extracting question-relevant knowledge from its contents.

The experimental results are shown in Table 2. Notably, LLM exhibits exceptional proficiency in addressing 1-hop questions, thereby showcasing its capacity to directly locate answers within the set of triples. However, when confronted with extensive textual content encompassing diverse details, LLM encounters difficulty in effectively retrieving question-relevant information. This observation reveals LLM's inherent limitation in fact extraction from natural language text. Furthermore, the performance of LLMs drops significantly when answering questions with more than 2 hops, suggesting that improving the reasoning capabilities of models is still an important research area.

4.2.3 Is fluently organized knowledge of natural language texts superior to disorganized structured knowledge?

In the DocRED dataset, the NL document and its corresponding triples are manually annotated. Therefore, we consider the document and their mapping triples as ground-truth KG-to-text pairs, serving as a complement to the potential issues in Table 1. In Table 1, we hypothesized that the model's performance degradation under NL text prompts could be attributed to the lack of an optimized KG-to-text model. However, the triples in this experiment were derived from manually annotated natural language documents, which refutes our hypothesis. In Table 2, we consider the document as the most reasonable and fluent natural language text generated from the triples, which assists us in eliminating the concern presented in Table 1 regarding the potential decline in LLM performance caused by the subpar quality of the text generated by the MVP model (Tang et al., 2022).

When compared to methods that inject knowledge using NL text, LLM with structured knowledge consistently performs better. Fluent NL texts may introduce noise, such as function words, which can hinder LLM's ability to reorganize the core structured knowledge that it should pay attention to. This indicates that LLM has a strong understanding capacity for the structured input knowledge and excels in reasoning on the structured knowledge, surpassing our initial expectations.

4.2.4 How robust are LLMs when dealing with sub-graphs from noisy or incomplete KG?

Replacement and deletion operations require a larger number of triples. In order to make the number of triples as large as possible, we choose to expand each node in Table 1 to 2 adjacent nodes and to 2 hops. Deletion operation simulates incomplete KG scenario. In the replacement operation, the sub-graph is attacked, and some nodes are replaced to generate false fact information and simulate a noisy sub-graph. We randomly delete and replace KG sub-graphs according to specified percentages. Based on Table 3, we have the following two findings: 1) The random replacement of nodes in KG has a more significant impact on the inference performance of LLM compared to the random deletion. Incorrect facts are more likely to result in erroneous model outputs. In some instances, the model can provide correct answers without reliance on any external knowledge; however, the introduction of noisy knowledge can lead to erroneous model outputs. We show this case in A.3. 2) When triples are injected as external knowledge, the model's robustness to noisy knowledge decreases as its size increases. Despite larger models demonstrating superior answering performance, they exhibit a greater performance loss when subjected to random replacement and deletion of KG knowledge. There exists an inverse proportionality between a model's robustness and its size.

4.2.5 What factors should be considered when designing the prompt framework to leverage KG as external knowledge?

The results of the knowledge prompt injection methods, designed in Section 3.4, are shown in Table 4. We observed that the performance of various prompt methods is inconsistent across different

	ChatGPT						Vicuna 7b						Vicuna 13b					
	QALD-7		LC-QuAD 2.0		KQAPro		QALD-7		LC-QuAD 2.0		KQAPro		QALD-7		LC-QuAD 2.0		KQAPro	
Ratio	Delete	Replace	Delete	Replace	Delete	Replace	Delete	Replace	Delete	Replace	Delete	Replace	Delete	Replace	Delete	Replace	Delete	Replace
0%	82.56	82.56	50.60	50.60	54.03	54.03	65.70	65.70	32.84	32.84	27.63	27.63	77.00	77.00	42.98	42.98	32.75	32.75
10%	82.08	80.62	47.84	47.74	51.57	52.04	64.44	58.89	31.07	32.20	26.12	26.23	72.85	78.21	41.66	40.55	31.32	31.24
20%	82.08	80.62	46.23	46.02	49.66	49.91	70.00	58.65	31.83	29.59	26.09	26.17	75.51	81.16	39.43	39.43	30.96	31.07
30%	80.39	79.90	44.18	43.60	46.95	46.87	62.95	59.08	29.51	28.10	23.52	24.58	74.59	78.70	38.57	38.85	28.94	29.81
40%	80.39	79.90	42.49	42.41	43.82	44.10	55.27	57.68	27.83	27.47	23.60	23.94	77.00	73.84	36.12	36.83	27.93	28.19
50%	80.39	80.39	40.78	40.28	41.02	41.33	66.86	59.66	27.78	26.03	21.95	21.87	76.52	74.35	34.57	35.09	26.26	27.88
60%	80.39	80.39	36.25	35.99	37.28	37.08	63.24	61.30	24.50	22.56	21.37	19.16	75.31	78.21	33.50	32.94	25.53	25.36
70%	75.56	75.56	31.87	31.72	32.33	32.02	63.00	48.74	21.82	19.44	18.06	16.69	73.62	68.07	30.54	27.36	23.01	23.38
80%	65.85	63.91	26.28	25.82	27.21	27.46	50.00	47.25	18.54	16.50	14.57	13.49	65.17	66.14	25.56	23.79	20.41	20.25
90%	62.42	54.30	19.16	17.64	19.91	19.77	46.64	45.02	13.21	11.98	10.49	9.82	57.51	56.23	18.48	17.51	14.15	14.90
Degradation Ratio	**20.14**	**28.26**	**31.44**	**32.96**	**34.12**	**34.26**	19.06	20.68	19.63	20.86	17.14	17.81	19.49	20.77	24.50	25.47	18.60	17.85

Table 3: Randomly delete and replace nodes in the sub-graph. The Degradation Ratio quantifies the discrepancy between the model's peak performance and its poorest performance. Values in bold denote the maximum, while underlined values signify the second highest. Larger models exhibit the most pronounced performance degradation when faced with attacks.

Data Set	ChatGPT			Vicuna 7b			Vicuna 13b		
	Grouping	Ranking	Scoring	Grouping	Ranking	Scoring	Grouping	Ranking	Scoring
QALD-7	84.11	84.11	**84.11**	63.84	**64.54**	53.64	75.52	**77.81**	72.4
LC-QuAD 2.0	48.71	50.01	**52.48**	33.49	**35.72**	26.14	45.1	**45.13**	42.57
KQAPro	50.25	52.29	**54.03**	27.74	**31.32**	24.92	36.05	**37.64**	35.12

Table 4: KG prompt engineering. Distinct models exhibit unique preferences towards various prompting methods.

Dataset	ChatGPT	Vicuna 7b	Vicuna 13b
QALD-7	**79.42** (71.45)	**53.73** (14.35)	**72.13** (60.82)
LC-QuAD 2.0	**27.51** (16.42)	**18.37** (2.60)	**23.17** (13.47)
KQAPro	**30.09** (15.77)	**18.82** (3.55)	**22.18** (10.29)

Table 5: KG without reason path. In brackets are the performance of the model without any knowledge (form Table 1).

models. ChatGPT favors the knowledge injection method that incorporates confidence scores, while the Vicuna series exhibits a preference for ranking methods. This discrepancy may be attributed to variations in the training data and inherent tendencies of the respective models. This finding suggests that when designing a prompt method, the applicability of a given lightweight method across multiple models should be considered.

We discovered that when we delete triples up to 90%, the ability of LLM to answer questions consistently improved compared to having no information at all. In Table 5, We specifically removed all core reasoning paths within a sub-graph and observed that as long as some triples were present (even if they weren't particularly relevant), the QA performance of LLM remained superior to providing no information. This situation is illustrated in the Appendix A.3. We hypothesize that these triples stimulate LLM thinking with question-related information, thus driving the model's ability to generate accurate answers. This underscores the importance of designing rigorous experiments when proposing a knowledge injection method, in order to validate the robustness of the method. For instance, the results from the ablation studies demonstrate that adding low-correlation triples or noisy knowledge does not significantly enhance the performance of the LLM.

5 Conclusion

In this work, we investigate the proficiency of LLM in comprehending KG knowledge through challenging KGQA tasks. Experiments show that the presence of redundant or irrelevant knowledge does not necessarily diminish the reasoning capability of LLM. In fact, it can enhance accuracy by filtering out irrelevant information and leveraging relevant details, surpassing our initial expectations. LLMs consistently outperform well-crafted and fluent text prompts when confronted with disorganized, noisy, and abstract knowledge inputs. This demonstrates their proficiency in organizing and comprehending structured knowledge beyond what we anticipated. Moreover, when incomplete KG or noisy sub-graphs are introduced, LLM consistently outperforms scenarios without any structured knowledge. This discovery emphasizes the necessity for researchers to conduct rigorous experiments to validate the effectiveness of their proposed frameworks. Additionally, the preference for KGs among different models warrants the community's attention, necessitating ample experimental validation when proposing frameworks for LLMs.

6 Limitations

Despite our extensive research on LLMs' understanding of KGs, this paper presents certain limitations.

We utilized the QALD-7 (Usbeck et al., 2017) dataset for our study. However, the simplicity and limited number of questions within this dataset present a challenge. To conduct quantitative subgraph expansion experiments, we imposed strict restrictions on the inference path of the dataset. For instance, since the data from QALD-7 contains too many simple questions that do not meet the restriction of a 2-hop limitation, and some nodes in the core reasoning path also do not meet the augmentation method proposed in this paper, questions that do not meet the requirements are removed. After applying these stringent filters, the number of questions meeting the criteria in each dataset was further reduced. This issue is particularly pronounced with the QALD-7 dataset, potentially biasing the evaluation results and complicating our analysis.

Furthermore, we restricted our study to datasets based on Wikidata, which inherently limits the dataset selection. However, as the largest knowledge base that continues to be updated and developed, Wikidata remains the best choice. Other KGs, including DBpedia (Lehmann et al., 2015) and YAGO (Pellissier Tanon et al., 2020), have ceased updates. In the future, we will consider evaluating our model on different KG-based datasets.

Owing to the constraints of paper length, an analysis of model interpretability in the experimental results is not included. Due to the limitations of black-box models and training difficulty, our work has not explored the principles of understanding structured knowledge by models at the vector level. This will serve as the focus of our subsequent research. In the future, we will continue research into understanding why the counter-intuitive phenomena discussed in this paper occur in certain open-source LLMs.

References

Simran Arora, Sen Wu, Enci Liu, and Christopher Ré. 2022. Metadata shaping: A simple approach for knowledge-enhanced language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1733–1745.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Eva Banik, Claire Gardent, Donia Scott, Nikhil Dinesh, and Fennie Linag. 2012. Kbgen-text generation for knowledge bases as a new shared task. In *The seventh International Natural Language Generation Conference. Starved Rock, Illinois, USA.*, pages 141–146.

Ryan Brate, Minh-Hoang Dang, Fabian Hoppe, Yuan He, Albert Meroño-Peñuela, and Vijay Sadashivaiah. 2022. Improving language model predictions via prompts enriched with knowledge graphs. In *DL4KG@ ISWC2022*.

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.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2020. Kqa pro: A dataset with explicit compositional programs for complex question answering over knowledge base. *arXiv preprint arXiv:2007.03875*.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang. 2022. Dictbert: dictionary description knowledge enhanced language model pre-training via contrastive learning. *arXiv preprint arXiv:2208.00635*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. *arXiv preprint arXiv:2401.14887*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pages 69–78. Springer.

Hanning Gao, Lingfei Wu, Po Hu, and Fangli Xu. 2020. Rdf-to-text generation with graph-augmented structural neural encoders. In *IJCAI*, pages 3030–3036.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. Cyclegt: Unsupervised graph-to-text and text-to-graph generation via cycle training. *arXiv preprint arXiv:2006.04702*.

Lei He, Suncong Zheng, Tao Yang, and Feng Zhang. 2021. Klmo: Knowledge graph enhanced pretrained language model with fine-grained relationships. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4536–4542.

Nan Hu, Jiaoyan Chen, Yike Wu, Guilin Qi, Sheng Bi, Tongtong Wu, and Jeff Z. Pan. 2024. Benchmarking large language models in complex question answering attribution using knowledge graphs. *CoRR*, abs/2401.14640.

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

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv preprint arXiv:2105.11644*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Shiyang Li, Yifan Gao, Haoming Jiang, Qingyu Yin, Zheng Li, Xifeng Yan, Chao Zhang, and Bing Yin. 2023. Graph reasoning for question answering with triplet retrieval. *arXiv preprint arXiv:2305.18742*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. *arXiv preprint arXiv:1904.03396*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*, pages 583–596. Springer.

Mariia Rizun et al. 2019. Knowledge graph application in education: A literature review. *Acta Universitatis Lodziensis. Folia Oeconomica*, 3(342):7–19.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. *arXiv preprint arXiv:2007.00655*.

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*.

Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. *AI Open*, 2:127–134.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. *arXiv preprint arXiv:2010.00309*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

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. *arXiv preprint arXiv:2303.07992*.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Mvp: Multi-task supervised pre-training for natural language generation. *arXiv preprint arXiv:2206.12131*.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Furkan Ufuk. 2023. The role and limitations of large language models such as chatgpt in clinical settings and medical journalism. *Radiology*, 307(3):e230276.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th open challenge on question answering over linked data (qald-7). In *Semantic Web Challenges: 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28-June 1, 2017, Revised Selected Papers*, pages 59–69. Springer.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Jianing Wang, Wenkang Huang, Qiuhui Shi, Hongbin Wang, Minghui Qiu, Xiang Li, and Ming Gao. 2022. Knowledge prompting in pre-trained language model for natural language understanding. *arXiv preprint arXiv:2210.08536*.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2481–2491.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. *CoRR*, abs/2305.11206.

Yaoming Zhu, Juncheng Wan, Zhiming Zhou, Liheng Chen, Lin Qiu, Weinan Zhang, Xin Jiang, and Yong Yu. 2019. Triple-to-text: Converting rdf triples into high-quality natural languages via optimizing an inverse kl divergence. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 455–464.

A Appendix

A.1 LLM Input

As illustrated in Figure 3, the input to the LLM is primarily divided into four components: task instruction, in-context learning examples, external knowledge, and questions. Within the instruction section, we impose constraints on the model's output mode, mandating that different questions must be answered according to the prescribed format. For example, if the answer pertains to an entity, we stipulate that the answer should be presented as a list of entities devoid of any explanations. For counting questions, only numerical values are permissible. Unanswerable questions are indicated by returning *None*. This specification ensures that the format of the model's responses consistently aligns with our unified evaluation process.

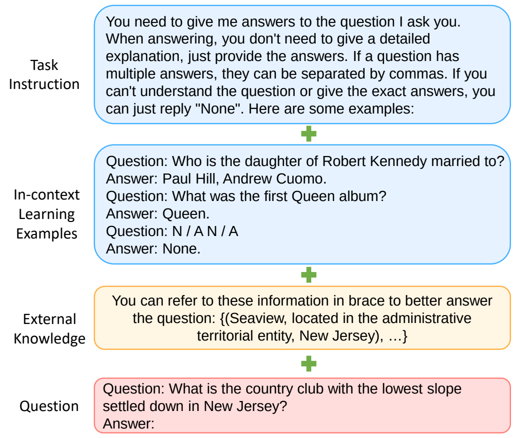


Figure 3: The structure of the input. The knowledge component is provided in triple format, and alternative formats (such as meta path or sentence) can be used to represent the knowledge.

During the experiment, we observed certain cases where the LLM's replies still failed to meet our criteria, occasionally returning sentences or blocks of text. To address this issue, we refer to the concept of in-context learning, wherein we provided multiple examples resembling the question to guide the model towards producing responses in the desired format. Regarding the knowledge injection method, we augmented the prompt with text derived from the KG, encompassing various formats, and supplied it to the LLM as additional knowledge. Finally, we appended the NL question that requires answers at the end of the model input.

A.2 Cross Encoder Training and Prompt Engineering

We utilize the BERT-Base (Devlin et al., 2018) as the foundation for our model. The training data comprises triples and corresponding questions. We partition the dataset into training and test sets in an 8:2 ratio. Triples in the reasoning path linked to the question are labeled as positive examples; otherwise, they are designated as negative examples. For the cross-encoder, the batch size is set at 50, we experimented with initial learning rates of {5e-4, 2e-5, 5e-5, 2e-5}, and the learning rate decays every 3 epochs. We set the multiplicative factor, gamma, for updating the learning rate to 0.2.

Upon training the model, it exhibits an accuracy of 98.89% in determining whether triples are pertinent to the question, i.e., whether they are part of the crucial reasoning path. This cross-encoder is employed to assign scores to the questions and their associated triples. By setting thresholds of 0.3 and 0.8 at either end to segment the triples, we can categorize them into high, medium, and low correlation groups relative to a question.

After the assignment of scores to triples by the cross-encoder, Table 6 illustrates the organizational format of the structured knowledge to the question "What trade structure did Straight to the point Gehry design?". This component is incorporated into the LLM as knowledge. For details on the incorporation format, refer to Appendix A.1.

A.3 Qualitative Example Study

We employ ChatGPT as our baseline model to elucidate the influence of external knowledge on the reasoning capability of the model. As shown in the first and second rows of Table 7, the model tends to commit errors when directly responding to the questions. Nevertheless, upon integrating a sub-graph devoid of inference paths, the model succeeds in providing correct responses. This is attributable to the model's capacity to draw analogies from similar knowledge, even though the external knowledge does not proffer direct answers. As illustrated in rows three and four of Table 7, 90% of the triples in the knowledge we gave have been replaced by wrong entities, which contain a lot of noisy knowledge. The model can provide correct answers when responding directly, however, the introduction of erroneous external knowledge leads to incorrect responses from the model. This indicates that the model lacks robustness against noisy information and is significantly influenced by the introduction of external inaccuracies.

Injection Method	Knowledge Prompt
Grouping	**Here are some triples that are highly relevant to the question:** (DZ Bank building, architect, Frank Gehry), (Gehry Tower, instance of, office building), ... **Here are some triples that are likely relevant to the question:** (IAC Building, architect, Frank Gehry), (Gehry Tower, architect, Frank Gehry) ... **Here are some triples that are less relevant to the question:** (Toledo Museum of Art, architect, Frank Gehry), (Vlado Miluni, notable work, Dancing House), ...
Ranking	**The triples are sorted from high to low according to their relevance score to the question for your reference:** (DZ Bank building, architect, Frank Gehry), (Dancing House, instance of, office building),(Gehry Tower, architect, Frank Gehry), (Dancing House, architect, Frank Gehry), (IAC Building, instance of, office building), ...
Scoring	**You can refer to these information to better answer the question. Each triple is followed by a confidence score of its relevance to the question, which helps in solving the question:** {(DZ Bank building, architect, Frank Gehry) ∣ 0.9981}, {(Toledo Museum of Art, architect, Frank Gehry) ∣ 0.0019}, {(Gehry Tower, instance of, office building) ∣ 0.998}, {(Vlado Miluni, notable work, Dancing House) ∣ 0.0023}...

Table 6: Prompt Orgnization.

Question	Knowledge	Without Knowledge Answer	With Knowledge Answer	Correct Answer
Which is the calling for the field of occupation of manga?	**without reason path:** (Jack Bauer, occupation, field agent), (field agent, instance of, profession), (Comic Gum, intended public, seinen), (manga, subclass of, manga), (seinen, subclass of, manga), (comedian, occupation, comedian), (Son Gokb, creator, Akira Toriyama), (field agent, occupation, field agent), (Nozomu Tamaki, occupation, mangaka), (Dragon Ball GT, after a work by, Akira Toriyama), (Akira Toriyama, occupation, mangaka), (seinen, intended public, seinen), (Douglas Adams, occupation, comedian), (comedian, instance of, profession)	Illustration, storytelling, comic art.	mangaka	mangaka
Is it true that David Koresh's given name was David or Wayne?	**without reason path:** (Waco siege, instance of, religious persecution), (Wayne Shorter, given name, Wayne), (David Bowie, given name, David), (Dave Arneson, given name, David), (religious persecution, statement supported by, David Koresh)	False.	True.	True.
Which is the island country for the nation of pound sterling?	**90% replace:** (Germany, diplomatic relation, Bahrain), (South Holland, contains the administrative territorial entity, Nieuw-Lekkerland), (Antwerp, twinned administrative body, Rotterdam), (Nieuw-Lekkerland, contains the: administrative territorial entity, Nieuw-Lekkerland), (Nieuw-Lekkerland, contains the administrative territorial entity, Nieuw-Lekkerland), (Nieuw-Lekkerland, instance of, village), (Antwerp, twinned administrative body, Rotterdam), (South Holland, contains the administrative territorial entity, Spijkenisse), (Nieuw-Lekkerland, instance of, village), (South Holland, contains the administrative territorial entity, Rijnsburg), (South Holland, contains the administrative territorial entity, Nieuw-Lekkerland), (South Holland, contains the administrative territorial entity, Rijnsburg), (Rijnsburg, contains the administrative territorial entity, Rijnsburg), (Nieuw-Lekkerland, instance of, village), (Rijnsburg, instance of, village), (European Netherlands, has part(s), South Holland)	United Kingdom.	None.	United Kingdom.
What is the inverse class for fiction?	**90% replace:** (The Night Watch, genre, historical fiction), (Gerry Adams, position held, Mary Lou McDonald), (Sinn Fin, chairperson, Mary Lou McDonald), (Sinn Fin, chairperson, Mary Lou McDonald), (Gerry Adams, position held, Mary Lou McDonald), (Martin McGuinness, member of political party, Sinn Fin), (Gerry Adams, position held, Mary Lou McDonald), (Sinn Fin, chairperson, Mary Lou McDonald), (Lynn Boylan, member of political party, Sinn Fin), (2001 United Kingdom general election, followed by, 2005 United Kingdom general election), (Martin McGuinness, candidacy in election, 2005 United Kingdom general election), (Martin McGuinness, member of political party, Sinn Fin), (2005 United Kingdom general election, candidate, Sinn Fin), (Mary Lou McDonald, replaced by, Mary Lou McDonald), (Martin McGuinness, member of political party, Sinn Fin), (Lynn Boylan, member of political party, Sinn Fin), (Martin McGuinness, member of political party, Sinn Fin), (Martin McGuinness, member of political party, Sinn Fin), (Gerry Adams, position held, Mary Lou McDonald), (Martin McGuinness, member of political party, Sinn Fin)	nonfiction.	historical fiction.	Nonfiction.

Table 7: The impact of external knowledge on LLM inference performance.