LLM-Based Multi-Hop Question Answering with Knowledge Graph Integration in Evolving Environments

Anonymous ACL submission

Abstract

The rapid obsolescence of information in Large Language Models (LLMs) has driven the development of various techniques to incorporate new facts. However, existing methods for knowledge editing still face difficulties with multi-hop questions that require accurate fact identification and sequential logical reasoning, particularly among numerous fact updates. To tackle these challenges, this paper introduces Graph Memory-based Editing for Large Language Models (GMeLLo), a straightforward and effective method that merges the explicit knowledge representation of Knowledge Graphs (KGs) with the linguistic flexibility of LLMs. Beyond merely leveraging LLMs for question answering, GMeLLo employs these models to convert free-form language into structured queries and fact triples, facilitating seamless interaction with KGs for rapid updates and precise multi-hop reasoning. Our results show that GMeLLo significantly surpasses current state-of-the-art knowledge editing methods in the multi-hop question answering benchmark, MQuAKE, especially in scenarios with extensive knowledge edits.

1 Introduction

As the widespread deployment of LLMs continues, the imperative to keep their knowledge accurate and up-to-date, without incurring extensive retraining costs, becomes increasingly evident (Sinitsin et al., 2020). Several approaches have been proposed in prior works to address this challenge, with some focusing on the incremental injection of new facts into language models (Rawat et al., 2020; De Cao et al., 2021; Meng et al., 2022; Mitchell et al., 2022a). Interestingly, certain methodologies in the literature diverge from the conventional path of updating model weights, opting instead for an innovative strategy involving the use of external memory to store the edits (Mitchell et al., 2022b; Zhong et al., 2023).

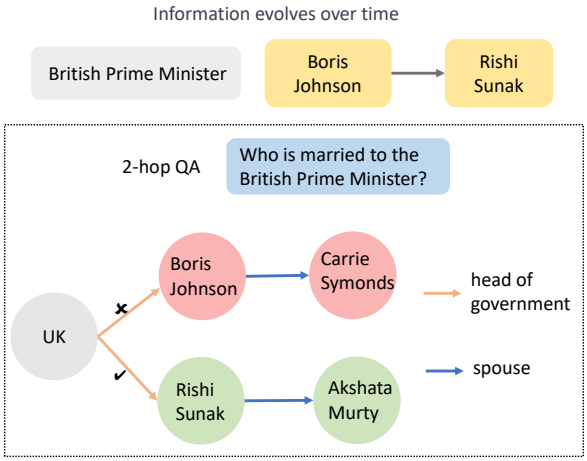

Figure 1: Multi-hop question answering in dynamic domains (Zhong et al., 2023). Dynamic nature of information: Changes over time may trigger subsequent modifications. For instance, a transition in the British Prime Minister, such as from Boris Johnson to Rishi Sunak, necessitates corresponding adjustments, like the change in the British Prime Minister's spouse.

As LLMs operate as black boxes, modifying one fact might inadvertently alter another, making it challenging to guarantee accurate revisions. In this paper, we introduce GMeLLo, an effective approach designed to synergize the strengths of LLMs and KGs in addressing the multi-hop question answering task after knowledge editing (Zhong et al., 2023). An illustrative example of our focus is presented in Figure 1. Following an update regarding the information of the British Prime Minister, it becomes evident that the corresponding spouse information should also be modified.

As depicted in Figure 2, our GMeLLo comprises the following key steps:

- We utilize LLMs to translate edited fact sentences into triples, employing these triples to update the KG and ensure its information remains up to date.
- Given a question, we utilize LLMs to extract

its relation chain, encompassing the primary entity and its connections with other unknown entities. After populating a template, we convert the relation chain into a formal query and use it to search the updated KG.

- In addition, we retrieve the most pertinent edited facts based on the question and prompt LLMs to generate an answer in accordance with these facts.

- In instances where the answer provided by the LLM conflicts with that from the KG, we prioritize the answer from the KG as the final response.

LLMs, trained on extensive sentence corpora (Brown et al., 2020; Rae et al., 2022; Chowdhery et al., 2023), are expected to encapsulate a wide range of commonly used sentence structures. As a result, they are invaluable tools for analyzing sentences and extracting entities and relations. Once the correct relation chain and edited triples are obtained, using a formal query to interrogate the KG in a Knowledge-based Question Answering (KBQA) (Cui et al., 2017) manner ensures precision in the searching process. In cases where KBQA fails, we still have LLMs for question answering (QA) to ensure comprehensive coverage. GMeLLo outperforms current state-of-the-art (SOTA) methods on two datasets from the MQuAKE benchmark, affirming its effectiveness in multi-hop question answering within an evolving environment.

2 Related Work

This work utilizes both KGs and LLMs to address the challenge of multi-hop question answering, with a particular focus on scenarios involving evolving factual knowledge. Therefore, we review existing literature on multi-hop question answering, knowledge editing, and the augmentation of LLMs with knowledge graphs[1].

2.1 Multi-Hop Question Answering

Multi-hop question answering is more challenging because it requires not only recalling facts but also appropriately aggregating and chaining them. Facts can be sourced from a knowledge graph (Lin et al., 2018; Cheng et al., 2023; Zhong et al., 2023), tables (Yin et al., 2016), free-form text (Yang et al., 2018; Welbl et al., 2018), or a heterogeneous combination of these sources (Chen et al., 2020; Mavi et al., 2022; Lei et al., 2023). With the development of LLMs, prompt-based methods combined with an optional retrieval module have become a popular approach for handling multi-hop question answering (Khattab et al., 2022; Press et al., 2023; Zhong et al., 2023). While most previous works focus on a static information base, our approach targets a dynamic domain, accommodating changes in facts.

2.2 Knowledge Editing

As highlighted in Yao et al. (2023), two paradigms exist for editing knowledge: modifying model parameters and preserving model parameters.

2.2.1 Modifying Model Parameters

In the case of modifying model parameters, this can be further categorized into meta-learning or locate-and-edit approaches. Meta-learning methods, as discussed in (De Cao et al., 2021; Mitchell et al., 2022a), utilize a hyper network to learn the necessary adjustments for editing LLMs. The locate-then-edit paradigm, as demonstrated in (Dai et al., 2022; Meng et al., 2022, 2023; Li et al., 2023a; Gupta et al., 2023; Zhang et al., 2024), involves initially identifying parameters corresponding to specific knowledge and subsequently modifying them through direct updates to the target parameters.

2.2.2 Preserving Model Parameters

In the case of preserving model parameters, the introduction of additional parameters or external memory becomes necessary. The paradigm of additional parameters, as presented in (Dong et al., 2022; Hartvigsen et al., 2022; Huang et al., 2022), incorporates extra trainable parameters into the language model. These parameters are trained on a modified knowledge dataset, while the original model parameters remain static. On the other hand, memory-based models (Mitchell et al., 2022b; Zhong et al., 2023) explicitly store all edited examples in memory and employ a retriever to extract the relevant edit facts for each new input, guiding the model in generating the edited output.

While previous evaluation paradigms have primarily focused on validating the recall of edited facts, Zhong et al. (2023) introduced MQuAKE, a benchmark that includes multi-hop questions involving counterfactual or temporal edits. The two

[1]Due to space constraints, some of the literature is located in Appendix B.

datasets within MQuAKE assess whether methods can accurately answer questions where the response should change due to edited facts.

3 GMeLLo: Graph Memory-based Editing for Large Language Models

In this section, we introduce our method GMeLLo for multi-hop question answering with knowledge editing (Figure 2).

3.1 Extracting Fact Triples from Edited Information Using LLMs

KGs play a pivotal role in enhancing the capabilities of LLMs by offering external knowledge for improved inference and interpretability, as demonstrated by recent studies (Pan et al., 2023; Rawte et al., 2023). Apart from merely storing updated information in an external memory, such as a list of separate sentence statements as seen in conventional approaches (Zhong et al., 2023), we utilize the KG to maintain inherent connections and ensure the integration of the latest information.

In our approach, we leverage Wikidata (Vrandečić and Krötzsch, 2014), a widely recognized KG, as the foundational knowledge base. When updated facts are received, we utilize LLMs to extract entities from the sentences and determine their relationships (selecting a relation from the predefined list). This process generates edited fact triples, which are then used to update the KG (see Figure 2). Updating the KG with an edited fact triple involves identifying the connections in the KG based on the subject entity and relation, breaking these connections, and establishing a new connection based on the triple.

We incorporate in-context learning (Dong et al., 2023) to ensure the LLMs have thorough understanding of the task. Furthermore, given the possibility that LLMs may generate relations not present in the predefined relation list (Chen et al., 2024), we use a retrieval model to identify the most similar relation (i.e., the closest relation in the embedding space) from the predefined relation list. The integration of retrieval model makes the triple extraction process more robust.

3.2 Extracting Relation Chain from Questions Using LLMs

As the world evolves rapidly, the training data for LLMs can quickly become outdated. However, since the evolution of linguistic patterns typically progresses at a slower pace, the extensive training data of LLMs should enable them to effectively comprehend most sentence patterns. In this paper, we employ LLMs to extract the relation chain from a sentence, encompassing the mentioned entity in the question and its relations with other unidentified entities. Similar to the fact triple exaction 3.1, we task LLMs with selecting a relation from a predefined list to mitigate varied representations of the same relation. Take a question sentence from the MQuAKE-CF (Zhong et al., 2023) dataset as an example,

Question
What is the capital of the country of citizenship of the child of the creator of Eeyore?
Relation Chain
```
Eeyore->creator->?x->child->?y
->country of citizenship
->?z->capital->?m
```

The presented question necessitates a 4-hop reasoning process. With "Eeyore" as the known entity in focus, the journey to the final answer involves identifying its creator '?x', moving on to the creator's child '?y', obtaining the child's country of citizenship '?z', and culminating with the retrieval of the country's capital '?m'. All the relations, such as 'creator,' 'child,' 'country of citizenship,' and 'capital,' are chosen from a predefined list of relations. The relation chain encapsulates all essential information for deriving the answer.

To enable LLMs to extract relation chains and generate outputs in a structured template, we provide several examples of relation chain extraction in the prompt and utilize in-context learning(Dong et al., 2023), as detailed in Appendix A.4.

3.3 Converting Relation Chain into a Formal Query

Once the relation chain is obtained, the next step involves integrating the known entity and the relations into a formal query template. Consider a KG represented in RDF[2] format and a corresponding SPARQL[3] query, the relation chain elucidated in Section 3.2 should be represented as follows,

```
PREFIX ent: <http://www.kg/entity/>
PREFIX rel: <http://www.kg/relation/>
SELECT DISTINCT ?id ?label WHERE {
   ent:E0 rel:R0 ?x.
```

[2] https://www.w3.org/RDF/
[3] https://www.w3.org/TR/sparql11-query/

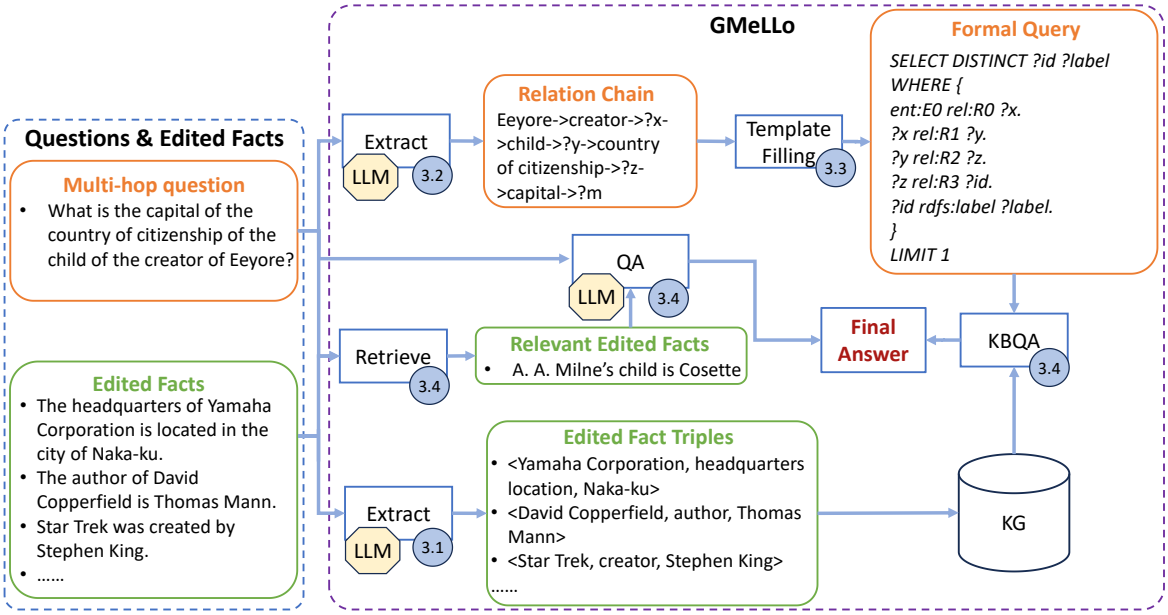

Figure 2: The illustration depicts our proposed method, GMeLLo. We begin by utilizing LLMs to extract entities and relations from edited facts, resulting in a list of edited fact triples. These triples are then used to update a KG. Similarly, we employ LLMs to extract relation chains from a given question. By populating this information into a template, we generate a formal query suitable for use in KBQA (Lan et al., 2022). Simultaneously, we utilize LLMs for question answering, providing an answer based on the relevant edited facts retrieved. In cases where the LLM's answer contradicts that of the KG, we defer to the KG's answer as the final response.

```
    ?x rel:R1 ?y.
    ?y rel:R2 ?z.
    ?z rel:R3 ?id.
    ?id rdfs:label ?label.
}
LIMIT 1
```

In this context, "ent" and "rel" serve as prefixes for entity and relation, respectively. The identifier "E0" uniquely represents "Eeyore" within the KG, while the identifiers for "creator," "child," "country of citizenship," and "capital" are denoted as "R0", "R1", "R2", and "R3" respectively. After identifying the entity "?id", we retrieve its string label "?label" as the final answer.

3.4 Integrating LLM-based QA and KBQA

This subsection outlines the integration of the proposed KBQA module with the LLM-based QA module within the GMeLLo framework.

LLM-based question answering. When a question arises, we retrieve the top-x relevant facts using the pre-trained Contriever (Izacard et al., 2022) model from a list of edited fact sentences. We then prompt the LLMs to generate answers based on the question and these pertinent facts. Compared to the "split-answer-check" pipeline in MeLLo (Zhong et al., 2023), this LLM-based QA method is expected to be simpler and yield more accurate results when the facts are provided accurately.

However, addressing multi-hop questions, especially those where the edited facts pertain to intermediary hops, presents a challenge in accurately retrieving the relevant information and performing correct multi-hop question answering. This challenge is particularly pronounced when dealing with a large volume of edited facts. For instance, accurately identifying the relevant fact given the question in Figure 2 and producing the correct final answer is difficult.

KBQA. To address the challenges of LLM-based question answering, we integrate responses from KBQA to refine the outputs from the LLMs, as detailed in Sections 3.1-3.3. When the relation chain and fact triples are accurately derived, the KBQA system provides the correct answer. However, if the relation chain is incorrectly extracted, the search path in the KG may become invalid, leading the KBQA system to yield no output. In such instances, we accept the response from the LLMs as the final answer.

4 Experiment

In the upcoming section, we will conduct experiments to demonstrate the effectiveness of employ-

ing our GMeLLo methodology.

4.1 Experiment Setup

4.1.1 Dataset

Our experiment focuses on the multi-hop question-answering benchmark, MQuAKE (Zhong et al., 2023), which comprises two datasets: MQuAKE-CF [4], designed for counterfactual edits, and MQuAKE-T, specifically tailored for updates in temporal knowledge.

The MQuAKE-CF dataset comprises 3,000 N-hop questions (N $\in \{2, 3, 4\}$), each linked to one or more edits. This dataset functions as a diagnostic tool for examining the effectiveness of knowledge editing methods in handling counterfactual edits. The MQuAKE-T dataset consists of 1,868 instances, each associated with a real-world fact change. Its purpose is to evaluate the efficacy of knowledge editing methods in updating obsolete information with contemporary, factual data. A table of statistics is available in Appendix A.1.

4.1.2 Evaluation Settings

To evaluate our models, we adhere to the testing settings outlined by Zhong et al. (2023). Specifically, instances are batched in groups of size k, with $k \in 1, 100, 1000, 3000$ for MQuAKE-CF, and $k \in 1, 100, 500, 1868$ for MQuAKE-T. For example, in the MQuAKE-CF dataset, when $k = 100$, the 3000 instances are split into 30 groups, and we report the average performance as the final result.

For each test instance, the dataset includes three multi-hop questions that convey the same meaning. In alignment with Zhong et al. (2023), if the model correctly answers any one of these questions, we consider the instance to be accurately resolved.

4.1.3 Baselines

To demonstrate the effectiveness of our approach, we conduct comparisons with the following SOTA knowledge editing methods.

- MEND (Mitchell et al., 2022a). It trains a hyper-network to generate weight updates by transforming raw fine-tuning gradients based on an edited fact.
- MEMIT (Meng et al., 2023). It updates feed-forward networks across various layers to incorporate all relevant facts.
- MeLLo (Zhong et al., 2023). It employs a memory-based approach for multi-hop question answering, storing all updated facts in an external memory.

Given the substantial costs associated with training, deploying, and maintaining larger LLMs (Li et al., 2023b), and the challenges of scaling up knowledge editing methods that require model parameter modifications, this paper primarily focuses on smaller LLMs, specifically GPT-J (6B) (Wang and Komatsuzaki, 2021) and Vicuna (7B) (Chiang et al., 2023). However, to showcase GMeLLo's effectiveness with larger LLMs in practical scenarios, we also report the performance of both MeLLo and GMeLLo on the MQuAKE-CF dataset when $k = 3000$.

4.1.4 Knowledge Graph Setting

Considering Wikidata's community-driven nature, guaranteeing a dynamic and comprehensive dataset across a spectrum of knowledge domains, we use Wikidata (Vrandečić and Krötzsch, 2014) as the foundational KG for this experiment. To align the relations in the question and fact sentences with those in WikiData (Vrandečić and Krötzsch, 2014), we follow the following steps:

- We select the first 500 item properties[5] from WikiData as the base relations. Items represent either concrete or abstract entities, such as a person (Piscopo and Simperl, 2019).
- Next, we employ GPT-3.5-Turbo to examine each multi-hop question in the test samples to determine if it contains any of the base relations.
- Afterward, we rank the frequencies of each relation and choose the top 50 relations as candidates for use in relation chain extraction and edited fact triple extraction.

To stay updated with the latest information on WikiData, we utilize the WikiData API service[6] and the WikiData Query Service[7]. The correctness of our KBQA result hinges on the accurate extraction of both edited fact triples and relation chains. If the relation chain is found to be incorrect, we

[4] Following Zhong et al. (2023), our experiments on MQuAKE-CF are carried out on a randomly sampled subset of the complete dataset, comprising 3000 instances in total(1000 instances for each of 2, 3, 4-hop questions).

[5] https://www.wikidata.org/w/index.php?title=Special:ListProperties/wikibase-item&limit=500&offset=0

[6] https://www.wikidata.org/w/api.php

[7] https://query.wikidata.org/sparql

Base Model	Method	MQuAKE-CF				MQuAKE-T			
		k=1	k=100	k=1000	k=3000	k=1	k=100	k=500	k=1868
GPT-J-6B	MEMIT	12.3	9.8	8.1	1.8	4.8	1.0	0.2	0.0
	MEND	11.5	9.1	4.3	3.5	38.2	17.4	12.7	4.6
	MeLLo	20.3	12.5	10.4	9.8	85.9	45.7	33.8	30.7
	GMeLLo	**76.3**	**53.4**	**49.5**	**49.0**	**86.9**	**82.1**	**81.5**	**81.5**
Vicuna-7B	MeLLo	20.3	11.9	11.0	10.2	84.4	56.3	52.6	51.3
	GMeLLo	**71.3**	**46.5**	**42.5**	**41.9**	**97.1**	**86.3**	**85.4**	**85.1**

Table 1: Performance comparison of GMeLLo and other approaches on the MQuAKE-CF and MQuAKE-T datasets using GPT-J-6B or Vicuna-7B as the base language models. Adhering to the methodology outlined by Zhong et al. (2023), instances are grouped into batches of size k. For the MQuAKE-CF dataset, k varies from 1 to 3000, and for the MQuAKE-T dataset, it ranges from 1 to 1868. For example, in the MQuAKE-CF dataset, when $k = 100$, the 3000 instances are organized into 30 groups, and the average performance reported as the final result. The metric used is accuracy.

conduct an online search on WikiData to determine if the relation chain leads to an entity that could potentially yield an incorrect answer for the specific question, which takes about 1 second.

4.1.5 Strategies for Managing Unforeseen Relationships

As previously noted, since LLMs may produce relations that are similar in meaning but not identical, we employ the pretrained Contriever model (Izacard et al., 2022) to retrieve the most similar relation (i.e., the closest relation in the embedding space) from the base list of relations. This replacement is performed when undefined relations are encountered during both edited fact triple extraction and relation chain extraction.

4.2 Main Results

As shown in Table 1, our GMeLLo significantly outperforms all existing methods on the both the MQuAKE-CF dataset and the MQuAKE-T dataset (Zhong et al., 2023), particularly when handling a large number of edits.

The performance degradation in MeLLo is primarily due to its challenges in identifying relevant facts as the number of edits increases. When k=1, the model utilizes only the facts directly related to the input question for context. However, as k increases, the model faces the challenge of discerning relevant facts from a broader memory. Our proposed GMeLLo model mitigates this by employing an explicit symbolic graph representation, which enhances the system's ability to update and retrieve relevant facts effectively. This feature significantly boosts the scalability of GMeLLo, making it well-suited for real-world question answering applications that require managing large volumes of rapidly changing information.

In addition, we evaluated MeLLo and GMeLLo using two larger models, GPT-3.5-Turbo-Instruct and GPT-3.5-Turbo[8], on the MQuAKE-CF dataset with k=3000[9]. The accuracy rates achieved by MeLLo and GMeLLo with GPT-3.5-Turbo-Instruct were 30.7% and 51.4%, respectively. While GMeLLo achieved an accuracy of 66.4% with GPT-3.5-Turbo, the same model consistently returned errors when tested with MeLLo, suggesting that the prompts may require modification for compatibility with chat completion models. These results indicate that GMeLLo performs well even when scaled to larger LLMs.

4.3 Ablation Study

To gain a comprehensive understanding of the performance of various components, i.e., LLM-based QA and KBQA, we conduct an experiment to illustrate the impact of LLM-based QA and KBQA as the number of edits increases.

As demonstrated in Table 2, the performance of KBQA remains consistent, as all edited facts are converted to triples and all relation chains are extracted from the test questions, regardless of the value of 'k'. However, as the parameter 'k' increases, more edited facts are stored in the external memory. Consequently, selecting the relevant edits to accurately answering the questions becomes increasingly challenging for LLM-based QA.

[8]https://platform.openai.com/docs/models/gpt-3-5-turbo

[9]The model `text-davinci-003` used in Zhong et al. (2023) was deprecated on January 4, 2024.

Base Model	Method	MQuAKE-CF				MQuAKE-T			
		k=1	100	1000	3000	k=1	100	500	1868
GPT-J-6B	QA	71.0	24.2	14.3	12.2	32.3	18.0	15.7	15.5
	KBQA	43.3	43.3	43.3	43.3	80.2	80.2	80.2	80.2
	GMeLLo	**76.3**	**53.4**	**49.5**	**49.0**	**86.9**	**82.1**	**81.5**	**81.5**
Vicuna-7B	QA	**72.6**	27.0	16.5	13.5	96.9	63.0	59.2	58.2
	KBQA	35.9	35.9	35.9	35.9	73.6	73.6	73.6	73.6
	GMeLLo	71.3	**46.5**	**42.5**	**41.9**	**97.1**	**86.3**	**85.4**	**85.1**

Table 2: Ablation study of GMeLLo. QA involves directly using LLM for answering the multi-hop questions. KBQA involves using LLM to transform edited fact sentences into triples, update WikiData, convert question sentences into relation chains, and generate formal KG queries for question answering. GMeLLo combines these methods by using KBQA to correct answers from LLM-based QA.

When k=1 and all relevant facts are provided to the LLMs for question answering, the LLM-based QA proves to be quite effective. However, a more realistic scenario involves multiple edits occurring simultaneously, where each question is asked separately (i.e., k>1). The performance showcased in Table 2 demonstrates the effectiveness of our GMeLLo, highlighting that KBQA serves as a valuable enhancement to LLM-based QA within evolving environments.

4.3.1 Further Analysis

To evaluate the impact of KBQA on LLM-based QA within the GMeLLo framework, we conducted an analysis comparing the responses from LLMs to those from the KG. We consider the KG's response as the final answer. Therefore, comparing to only using LLM-based QA, if the answer from LLMs is correct but the answer from the KG is incorrect, this leads to a decline in performance. Conversely, if the answer from LLMs is incorrect but the answer from the KG is correct, performance improves. If the KBQA provides no response, performance remains unchanged. As illustrated in Table 3, when there are discrepancies between KBQA and LLM-based QA responses, the likelihood of KBQA providing the correct answer increases as the parameter k increases.

4.4 Qualitative Analysis

Table 2 illustrates that Vicuna exhibits superior performance in directly handling the QA task, particularly when provided with the exact edited facts. Conversely, GPT-J excels in sentence analysis tasks, showcasing its high performance in the KBQA task.

4.4.1 Inferior Performance of GPT-J in QA

Table 2 shows that the performance of GPT-J and Vicuna in conducting QA tasks is comparable on the MQuAKE-CF dataset when k=1. However, GPT-J exhibits notably lower performance on the MQuAKE-T dataset. Further analysis revealed that GPT-J struggles in answering questions with only an edited fact pertaining to its intermediary information, such as:

Sample from MQuAKE-CF
Facts: *Midfielder is associated with the sport of Gaelic football*
Question: *What is the capital of the country where the sport associated with Kieron Dyer's specialty was first played?*
Predicted Answer: *Bondi Junction*
Answer: *Dublin*

Sample from MQuAKE-T
Facts: *The name of the current head of the Philippines government is Bongbong Marcos*
Question: *Who is the head of government of the country that Joey de Leon is a citizen of?*
Predicted Answer: *Benigno Aquino III*
Answer: *Bongbong Marcos*

However, it can achieve the correct answer in KBQA because it accurately extracts the fact triple and relation chain of the question. Given that all test samples in MQuAKE-T contain only one edited fact, while approximately 63.6% of test samples in MQuAKE-CF consist of more than two edited facts, GPT-J is able to connect most of the information together. Therefore, it achieves better performance in the MQuAKE-CF dataset.

Base Model	Scenario			MQuAKE-CF				MQuAKE-T			
	LLM	KG	Performance	k=1	100	1000	3000	k=1	100	500	1868
GPT-J-6B	✖	✔	↑	8.1	22.9	24.9	25.0	44.0	47.2	47.9	48.0
	✔	✖	↓	12.5	2.4	1.2	0.7	0.7	0.4	0.3	0.3
	✔	◯	-	34.2	7.0	4.0	3.7	7.1	2.8	2.4	2.3
Vicuna-7B	✖	✔	↑	7.7	17.8	19.6	20.0	4.2	19.7	21.4	21.7
	✔	✖	↓	21.8	3.9	2.0	1.2	7.2	4.2	4.0	3.9
	✔	◯	-	32.7	7.4	4.0	3.4	35.7	19.8	18.1	17.7

Table 3: Further analysis for scenarios where the answers from LLM and KG contradict each other. The values are expressed as percentages. It is important to note that the total number of test questions is three times the number of test instances. For instance, in MQuAKE-CF, each test instance comprises three distinct questions with the same meaning, totaling 9,000 test questions. Symbols used: ↑ indicates improved performance, ↓ indicates reduced performance, and ◯ denotes no response from KBQA, resulting in no impact on the final output (-).

4.4.2 Inferior Performance of Vicuna in KBQA

Compared to GPT-J, Vicuna performs less effectively in the KBQA task. Aside from misunderstandings, the main reasons are as follows:

- It often makes errors in the sequence. For example, given the fact "The author of Misery is Richard Dawkins", its output fact triple is "Richard Dawkins->author->Misery". However, the correct sequence is "Misery->author->Richard Dawkins".

- It frequently makes errors in selecting a relation from the list. For example, it often outputs a relation chain as "Mike->citizenship->country->head of state", instead of "Mike->country of citizenship->head of state".

It is important to note that even if the relation chain is incorrect, the KBQA system may still provide the correct answer because of some loops in WikiData, such as the country of the USA is the USA.

Although Vicuna is not as effective overall, we still find that in some cases it can correctly extract relations, but cannot provide the correct answer directly. An example is given as follows:

Sample from MQuAKE-CF
Facts: *Point guard is associated with the sport of cricket*
Question: *What is the capital of the country from which Erik Spoelstra's sport comes?*
Predicted Answer: *Miami*
Answer: *London*

4.5 Further Discussion

KG offers a clearer representation of multi-hop information and its updates. In GMeLLo, we harness the strengths of both KBQA and LLM-based QA, benefiting from KBQA's high precision and LLM-based QA's extensive coverage. Our experiments reveal that GPT-J excels in extracting relation chains and fact triples, whereas Vicuna demonstrates superior performance in LLM-based QA. Given that KBQA and LLM-based QA operate as separate modules in GMeLLo, we can optimize their use by employing different LLMs in each module, maximizing their effectiveness in practical applications.

5 Conclusion

In this paper, we present GMeLLo, a method designed for multi-hop question answering in dynamic environments. Except leveraging LLMs for question answering, we also leverage the capabilities of LLMs to extract the triples from edited fact sentence to update KG, and use the capabilities of LLMs to analyze question sentences and generate a relation chain, and finally get the formal query by filling in a formal query template. Finally, we combine KBQA and LLM-based QA to bolster the multi-hop question answering capability within a dynamic environment. This approach capitalizes on the strengths of both LLMs and KGs. By utilizing LLMs for analyzing question sentences and QA to ensure the coverage, and KBQA to provide accurate results, we achieve a synergy between these two methodologies.

Limitations

Despite the promising results, it is important to acknowledge that this investigation is still in its early stages. Although our performance significantly surpasses baseline approaches in multi-hop questions in dynamic domains, particularly for large knowledge bases and complex questions, there is still room for further improvement. Our future research includes

- Leveraging more sophisticated prompting techniques, such as Chain of Thought (CoT) (Wei et al., 2022), to enable more accurate multi-hop reasoning.
- Refining the predefined relation list to enhance its accuracy.
- Enhancing the KG to support more complex question answering, such as inquiries involving historical information.

We believe these improvements can further enhance the performance and scalability of the system, enabling it to handle more complex and diverse real-world applications.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pages 70–98, Toronto, ON, Canada. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024. Is a large language model a good annotator for event extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17772–17780.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zhen Cheng, Jianwei Niu, Shasha Mo, and Jia Chen. 2023. Genboost: Generative modeling and boosted learning for multi-hop question answering over incomplete knowledge graphs. In *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1131–1138.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. Kbqa: learning question answering over qa corpora and knowledge bases. *Proc. VLDB Endow.*, 10(5):565–576.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket

Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8214–8232.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *NeurIPS 2022 Workshop on Robustness in Sequence Modeling*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2022. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Fangyu Lei, Xiang Li, Yifan Wei, Shizhu He, Yiming Huang, Jun Zhao, and Kang Liu. 2023. S3HQA: A three-stage approach for multi-hop text-table hybrid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1731–1740, Toronto, Canada. Association for Computational Linguistics.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023a. Pmet: Precise model editing in a transformer.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253, Brussels, Belgium. Association for Computational Linguistics.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass editing memory in a transformer. *The Eleventh International Conference on Learning Representations (ICLR)*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning*.

Zhijie Nie, Richong Zhang, Zhongyuan Wang, and Xudong Liu. 2024. Code-style in-context learning for knowledge-based question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18833–18841.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*.

Alessandro Piscopo and Elena Simperl. 2019. What we talk about when we talk about wikidata quality: a literature survey. In *Proceedings of the 15th International Symposium on Open Collaboration*, OpenSym '19, New York, NY, USA. Association for Computing Machinery.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong,

Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis & insights from training gopher.

Ankit Singh Rawat, Chen Zhu, Daliang Li, Felix Yu, Manzil Zaheer, Sanjiv Kumar, and Srinadh Bhojanapalli. 2020. Modifying memories in transformer models. In *International Conference on Machine Learning (ICML) 2021*.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

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

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. In *International Conference on Learning Representations*.

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

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.

Pengcheng Yin, Zhengdong Lu, Hang Li, and Kao Ben. 2016. Neural enquirer: Learning to query tables in natural language. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 29–35, San Diego, California. Association for Computational Linguistics.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024. Knowledge graph enhanced large language model editing. *CoRR*, abs/2402.13593.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

A Implementation

A.1 Dataset Statistics

Table 4 provides a summary of the statistics for the MQuAKE-CF and MQuAKE-T datasets.

	#Edits	2-hop	3-hop	4-hop	Total
MQuaKE-CF	1	513	356	224	1,093
	2	487	334	246	1,067
	3	-	310	262	572
	4	-	-	268	268
	All	1,000	1,000	1,000	3,000
MQuaKE-T	1 (All)	1,421	445	2	1,868

Table 4: Statistics of MQuAKE dataset (Zhong et al., 2023).

A.2 Hyperparmers Settings

To ensure reproducibility, we set the temperature to zero in all experiments. Table 5 shows that retrieving the top-6 edited facts from external memory provides the best average performance on the MQuAKE-CF dataset for $k > 1$. Consequently, we include top-6 edited facts in the prompt for subsequent experiments on this dataset when $k > 1$. Similarly, for the MQuAKE-T dataset when $k > 1$, we opted to incorporate the top-1 edited fact in the prompt.

A.3 Predefined Relations Utilized in the Prompts for Relation Chain and Fact Triple Extraction

After filtering by GPT-3.5-Turbo, the first 50 relations utilized in MQuAKE-CF dataset are: ['country of origin','sport', 'country of citizenship', 'capital', 'continent', 'official language', 'head of state', 'head of government', 'creator', 'country', 'author', 'headquarters location', 'place of birth','spouse', 'director / manager','religion or worldview', 'genre', 'work location', 'performer','manufacturer', 'developer', 'place of death', 'employer', 'educated at','member of sports team', 'head coach', 'languages spoken, written or signed', 'notable work', 'child', 'founded by', 'location', 'chief executive officer', 'original broadcaster', 'chairperson', 'occupation', 'position played on team / speciality','member of', 'language of work or name', 'director', 'league', 'home venue', 'native language', 'composer', 'place of origin (Switzerland)', 'officeholder','religious order', 'publisher', 'original language of film or TV show', 'ethnic group','military branch'].

After GPT-3.5-Turbo filtering, the MQuAKE-T dataset includes a total of 35 relations. The relation list is ['head of government', 'country of citizenship', 'head of state', 'country of origin', 'country', 'headquarters location', 'location', 'sport', 'performer', 'genre', 'developer', 'employer', 'manufacturer', 'place of death', 'place of birth', 'author', 'member of', 'capital', 'member of sports team', 'chief executive officer', 'notable work', 'director / manager', 'original broadcaster', 'creator', 'work location', 'educated at', 'located in the administrative territorial entity', 'head coach', 'place of publication', 'location of formation', 'director', 'producer', 'transport network', 'continent', 'child']

	k=100	k=1000	k=3000	Average
Top-4	15.6	**9.1**	7.2	10.63
Top-5	**16.8**	8.3	6.9	10.67
Top-6	16.6	8.5	7.4	**10.83**
Top-10	15.3	9.0	**8.0**	10.77
Top-100	8.2	4.7	3.7	5.53

Table 5: Hyperparameter search for top-x in Vicuna-based QA systems on the MQuAKE-CF dataset.

A.4 Prompt Setup and Post-Processing

The prompts used for edited fact triple extraction, relation chain extraction, and LLM-based QA are depicted in Figures 3, 4, and 5. The edited triple can be regarded as a specialized relation chain, with only one relation between entities and all entities known. All samples in the prompt are selected from the complete MQuAKE-CF dataset, ensuring they are distinct from the test samples. To improve the performance of LLMs in extracting relation chains and ensure that outputs conform to a specified format, we employ a 4-shot learning ap-

Prompt for Transforming the Edited Sentences to Triples

Sentence: The headquarters of University of Cambridge is located in the city of Washington, D.C.
Relation Chain: University of Cambridge->headquarters location->Washington, D.C.
.....
Given the above samples, please help me analyze the relation chain of the following sentence. All the relations should be selected from ['country of origin','sport', ...].
Sentence: The chief executive officer of Boeing is Marc Benioff
Relation Chain:

Figure 3: The prompt used for transforming edited fact sentences to triples.

Prompt for Transforming the Question Sentences to Relation Chains

Question: What is the birthplace of the author of "The Little Match Girl"?
Relation Chain: The Little Match Girl->author->?x->place of birth->?y
.....
Given the above samples, please help me analyze the relation chain of the following sentence. All the relations should be selected from ['country of origin','sport', ...].
Question: What is the continent where the CEO responsible for developing Windows 8.1 was born?
Relation Chain:

Figure 4: The prompt used for transforming question sentences to relation chains.

Prompt for LLM-based QA

Facts: Hans Christian Andersen was born in the city of Brittany
Question: What is the birthplace of the author of "The Little Match Girl"?
Answer: Brittany
.....
Facts: Windows 8.1 was developed by Boeing; The chief executive officer of Boeing is Marc Benioff; California is located in the continent of Europe; Marc Benioff was born in the city of California
Question: What is the continent where the CEO responsible for developing Windows 8.1 was born?
Answer:

Figure 5: The prompt used in LLM-based QA.

proach for the MQuAKE-CF dataset and a 3-shot learning approach for the MQuAKE-T dataset. For MQuAKE-CF, the approach involves presenting the model with samples of one 2-hop question, one 3-hop question, and two 4-hop questions. In contrast, for MQuAKE-T, the model is presented with one 2-hop question, one 3-hop question, and one 4-hop question.

To address the limitations of GPT-J and Vicuna in conforming to the desired output format, we establish a heuristic rule for extracting essential information from their outputs. For instance, in the context of relation chain extraction, this heuristic is outlined as follows:

- Narrow the attention to the output sentence containing the "->" indicator.
- Divide the sentence based on the "->" delimiter.
- Regard the initial segment as the predicted entity. Subsequently, process the following segments sequentially as relations, provided they do not begin with "?".

A.5 Strategies for Managing Sequence Errors in Extracting Fact Triples

While LLMs consistently identifies relations accurately—such as 'head of state,' 'chief of department,' and 'head of government'—it often makes errors in their sequencing. To address this, we employ Spacy[10] to detect instances where the object of an edited triple is not a person. If it is not, we adjust the sequence of the object and subject in the triple accordingly.

B The Distinctions Between Our GMeLLo and Other Methods

While both GMeLLo and MeLLo (Zhong et al., 2023) are memory-based models targeting multi-hop question answering in an evolving environment, they differ in the following aspects:

- MeLLo employs in-context learning to direct LLMs in splitting the question into sub-questions, answering each, and verifying against relevant edited facts for contradictions. In contrast, GMeLLo retrieves pertinent edited facts for the multi-hop question and presents them alongside the question to LLMs for answering.
- Except storing edited facts as isolated sentences in an external memory, we leverage LLMs to translate these sentences into triples and update the KG. In addition to obtaining an answer from LLMs, we utilize KBQA to enhance the precision of multi-hop question answering within an evolving environment.

Recently, the advent of LLMs has spurred the development of LLM-based KBQA systems (Baek et al., 2023; Sen et al., 2023; Nie et al., 2024). However, our GMeLLo are different from these works in the following aspects:

- Firstly, we consider question answering in a dynamic environment, where changes in the knowledge graph need to accounted for, whereas they do not.
- Secondly, we focus on multi-hop questions, whereas they deal with standard KBQA tasks, including intersection and difference questions etc.
- Thirdly, the KBQA and LLM-based QA are handled separately, using the KBQA answer as the final answer. In contrast, they retrieve triples from the knowledge graph and incorporate them into the prompt to guide LLM-based QA.

[10] https://spacy.io/

Model	Method	Number of Hops			
		2	3	4	Avg
GPT-J-6B	MEND	13.9	11.3	9.5	11.5
	MEMIT	22.5	6.0	8.4	12.3
	MeLLo	-	-	-	20.3
	GMeLLo	89.5	73.7	65.6	76.3

Table 6: The breakdown performance on the MQuAKE-CF dataset with respect to the number of hops when $k = 1$.

C Multi-Hop Performance Analysis

We study the breakdown of performance on the MQuAKE-CF dataset with respect to the number of hops when $k = 1$. Table 6 provides the hop-specific performance of different methods. Although MQuAKE did not provide the hop performance for MeLLo, it can be inferred that the average hop performance should not exceed 65.6%, given that the overall performance is 20.3%.