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 spurred the development of various techniques for incorporating new facts. To address the ripple effects of altering information, we introduce GMeLLo (Graph Memory-based Editing for Large Language Models), a straightforward yet highly effective method that harnesses the strengths of both LLMs and Knowledge Graphs (KGs). Instead of merely storing edited facts in isolated sentences within an external repository, we utilize established KGs as our foundation and dynamically update them as required. When faced with a query, we employ LLMs to derive an answer based on the relevant edited facts. Additionally, we translate each question into a formal query, tapping into the extensive data within the KG to obtain a more nuanced answer directly from it. In cases of conflicting answers, we prioritize the response derived from the KG as our final result. Our experiments demonstrate a substantial enhancement of GMeLLo over state-of-the-art (SOTA) methods on the MQuAKE benchmark—a dataset specifically designed for multi-hop question answering.

1 Introduction

As the widespread deployment of LLMs continues, the imperative to maintain their knowledge accuracy 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). As

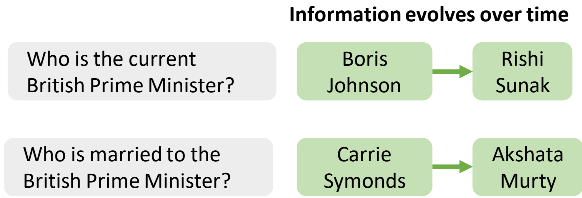

Figure 1: Dynamic nature of information: Changes over time may trigger subsequent modifications. For instance, a transition in the British Prime Minister, such as from Boris Johnson to Rishi Sunak, necessitates corresponding adjustments, like the change in the British Prime Minister's spouse.

LLMs operate as black boxes, modifying one fact might inadvertently alter another, making it challenging to guarantee accurate revisions. In light of this challenge, opting for an external memory system, rather than directly editing the LLMs, emerges as a prudent choice.

This paper introduces 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 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.

- Leveraging LLMs again, we analyze a query 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.

- Based on the query statement, we retrieve the most pertinent edited facts 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 chain of relations 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 retrieval process. In cases where KBQA fails, we still have LLMs for question answering (QA) to ensure comprehensive coverage. GMeLLo outperforms current SOTA models on the MQuAKE benchmark, affirming its effectiveness in multi-hop question answering within an evolving environment.

2 Related Work

The primary focus of this paper lies in exploring enhancing the multi-hop question answering within dynamic scenarios. Therefore, we delve into the related topic of knowledge editing. As highlighted in Yao et al. (2023), two paradigms exist for editing knowledge: modifying model parameters and preserving model parameters.

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), involves initially identifying parameters corresponding to specific knowledge and subsequently modifying them through direct updates to the target parameters.

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) proposed MQuAKE, a benchmark dataset comprising multi-hop questions with either counterfactual edits or temporal edits. This dataset assesses whether methods correctly answer questions where the response should change as a consequence of edited facts. 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 uses in-context learning to guide LLMs through splitting the question into sub-questions, answering each, and checking for contradictions with relevant edit facts. GMeLLo, on the other hand, retrieves a few relevant edit facts for the multi-hop question and presents them along with the question to LLMs for answering

- Rather than simply storing edited facts as isolated sentences in an external memory, we utilize LLMs to translate these sentences into triples and update the KG. Additionally, answers are obtained using KBQA to enhance the precision of multi-hop QA within an evolving environment.

Given that KG is a multi-relational graph consisting of entities as nodes and relations among them as typed edges (Saxena et al., 2020), it provides a more straightforward method for representing multi-hop information. Moreover, GMeLLo offers a means to seamlessly integrate the high precision of KBQA (Cui et al., 2017) with the extensive coverage of LLMs-based QA, enabling effective multi-hop question answering in dynamic environments.

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

In this section, we explore the details of our method, GMeLLo. Figure 2 provides a visual representation of the GMeLLo framework.

3.1 Utilizing KGs for Storing the Updated Correlated Facts to Enhance Multi-hop Reasoning

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 utilize an off-the-shelf KG, such as Wikidata (Vrandečić and Krötzsch, 2014), as the foundational source. Upon receiving updated facts, we employ LLMs to extract entities and their relationships, forming edited fact triples (Figure 2) that are then used to update the KG.

We incorporate in-context learning (Dong et al., 2023) to ensure the LLMs have a thorough understanding of the task. Furthermore, given the possibility that LLMs may generate relations not present in the KG's predefined list (Chen et al., 2024), we employ a retriever model to identify the most similar relation from the KG's list, which is detailed in Section 4.1.6. This relation retrieve procedure is also crucial during relation chain extraction.

3.2 Extracting the Relation Chain of a Question Sentence Using LLMs

With the world changing at a rapid pace, the training data for LLMs can quickly become outdated. Nevertheless, the evolution of patterns tends to occur at a relatively slower pace when compared to the intricate details. In this paper, we employ LLMs to extract the relation chain from a sentence, encompassing the mentioned entity and relations with other unidentified entities. To mitigate varied representations of the same relation, we task LLMs with selecting a relation from a predefined list. Take a question sentence from the MQuAKE 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->?id

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, moving on to the creator's child, obtaining the child's country of citizenship, and culminating with the retrieval of the country's capital. The relation chain encapsulates all essential information for arriving at the conclusive answer.

To ensure that LLMs comprehend the task of extracting the relation chain and generate output in a structured template, we employ in-context learning (Dong et al., 2023).

3.3 Converting the Relation Chain into a Formal Query for Retrieving Updated Information from KGs

Once the relation chain is obtained, the next step involves integrating the known entity and the relations into a formal query template. For instance, consider a KG represented in RDF[1] format and a corresponding SPARQL[2] query. The relation chain elucidated in Section 3.2 should be represented as follows, underscoring the seamless integration of the obtained information into a structured query framework.

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

[1] https://www.w3.org/RDF/

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

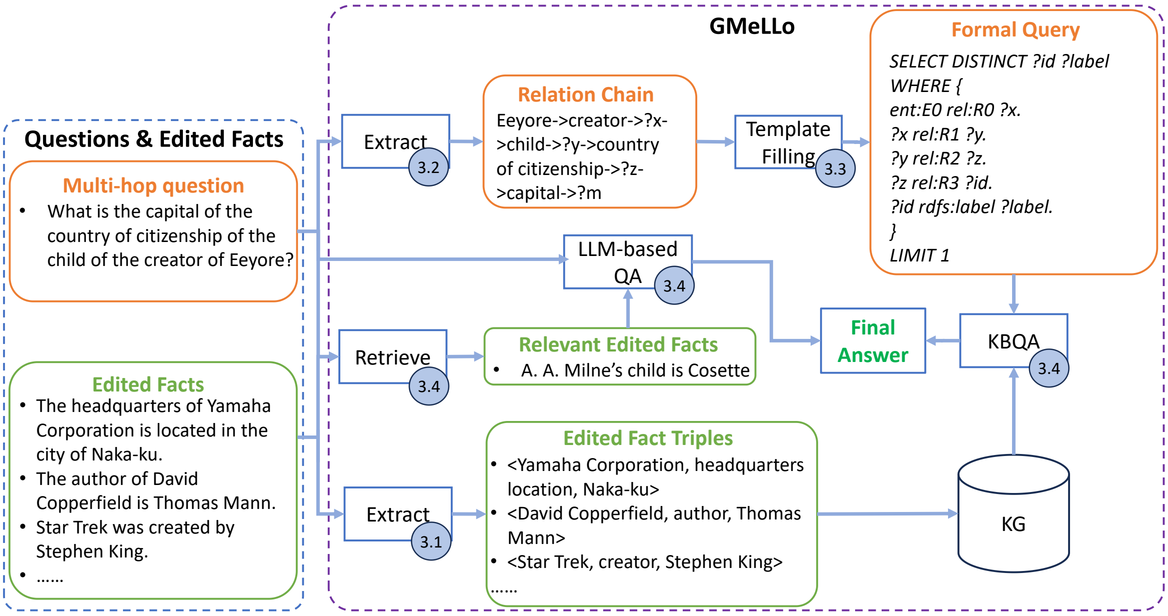

Figure 2: The illustration depicts our proposed method, GMeLLo. We begin by utilizing LLMs to extract entities and relations from edited facts, resulting in a list of edited fact triples. These triples are then used to update a KG. Similarly, we employ LLMs to extract relation chains from a given question. 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. In cases where the LLM's answer contradicts that of the KG, we defer to the KG's answer as the final response.

"R1", "R2", and "R3" respectively. After identifying the entity "?id", we retrieve its string label "?label" as the final answer.

3.4 Enhancing Multi-Hop Question Answering Using Knowledge Graph Integration

When a question arises, we retrieve the "top-x" [3] relevant facts using the pretrained Contriever (Izacard et al., 2022) model from a curated 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. To address this issue, we utilize answers from the KG to rectify responses from the LLMs. Once the relation chain and updated triples are derived accurately, the system will yield the correct answer. If the answer is not found within the KG, the system will output nothing, which does not affect the performance of the GMeLLo.

In conclusion, beyond tasking LLMs with question answering, we harness their powerful capabilities for analyzing both edited fact statements and questions. Post-analysis, we convert the edited fact sentences into edited fact triples, subsequently updating the KG. Likewise, we transform the question into a relation chain, culminating in a formal query generated by filling a template, obtaining an answer in a KBQA manner. Our approach leverages KBQA to substitute LLM answers in cases of inconsistency between the two responses. By amalgamating the high precision of KBQA with the expansive coverage of LLMs, our method excels in the multi-hop question answering domain following knowledge editing.

4 Experiment

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

[3]The "top-x" can be adjusted based on various scenarios. In the majority of cases, it should not exceed 4.

BaseModel	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
GPT-J-6B	MEND	11.5	9.1	4.3	3.5	38.2	17.4	12.7	4.6
GPT-J-6B	MeLLo	20.3	12.5	10.4	9.8	**85.9**	45.7	33.8	30.7
GPT-J-6B	**GMeLLo**	**50.9**	**29.2**	**27.7**	**27.1**	69.9	**65.1**	**64.9**	**64.8**
Vicuna-7B	MeLLo	20.3	11.9	11.0	10.2	**84.4**	56.3	52.6	51.3
Vicuna-7B	**GMeLLo**	**43.1**	**20.4**	**18.1**	**17.5**	75.0	**59.0**	**57.2**	**57.0**

Table 1: Performance results of GMeLLo (ours) on MQuaKE-CF and MQuaKE-T using either GPT-J-6B or Vicuna-7B as the base language model. Following the methodology of Zhong et al. (2023), instances are grouped into batches of size k, where k ranges from 1, 100, 1000, 3000 for MQuaKE-CF, and 1, 100, 500, 1868 for MQuaKE-T. For instance, with the MQuAKE-CF dataset, when k=100, the 3000 samples are divided into 30 groups, with the average performance reported as the final result. The metric used is multi-hop accuracy.

ing our GMeLLo methodology.

4.1 Experiment Setup

4.1.1 Dataset

Our experiment centers on the multi-hop question-answering dataset, MQuAKE (Zhong et al., 2023). This dataset comprises MQuAKE-CF[4], designed for counterfactual edits, and MQuAKE-T, tailored for temporal knowledge updates. These datasets enable the evaluation of methods under scenarios involving counterfactual changes and real-world temporal updates.

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.

4.1.2 Baseline Models

To demonstrate the effectiveness of our approach, we conduct comparisons with the following state-of-the-art knowledge editing methodologies.

- MEND (Mitchell et al., 2022a). It trains a hypernetwork 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.

Considering the significant costs associated with training, deploying, and maintaining larger LLMs (Li et al., 2023b), this paper primarily concentrates on smaller LLMs, specifically GPT-J (6B) (Wang and Komatsuzaki, 2021) and Vicuna (7B) (Chiang et al., 2023).

4.1.3 Evaluation Metric

In line with our paper's central emphasis on multi-hop question answering, we utilize accuracy as the primary metric, to evaluate the methods' performance in addressing multi-hop inquiries within dynamic environments.

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 opt for Wikidata (Vrandečić and Krötzsch, 2014) as the foundational KG for this experiment. Using LLMs along with 10 <edited fact, edited triple> pairs as samples in the prompt, we extract modified triples from the revised facts with the intention of using them to update the KG. To align the relationships in the questions of test samples with those in WikiData (Vrandečić and Krötzsch, 2014), we follow the following steps:

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

- 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[6] 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[7] and the WikiData Query Service[8]. Since WikiData may contain items with identical labels[9], we map the entity string in the edited fact triples and the relation chain to WikiData and select the first match as the candidate. We then verify if this entity corresponds to the intended one in the dataset. The correctness of our KBQA result hinges on two crucial criteria:

- The accurate extraction of both edited fact triples and relation chains.
- A precise match between the entity id retrieved from the WikiData API service for each entity string in the edited facts and relation chains and the intended entity id in the dataset.

If the relation chain is found to be incorrect, we 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 Prompt Setup and Post-Processing

Compared to MeLLo (Zhong et al., 2023), we adopt a strict evaluation approach, assessing only the first multi-hop question in the MQuAKE datasets for our GMeLLo, instead of considering all three and accepting any one correct. To enhance the comprehension of the relation chain extraction task by LLMs and ensure outputs adhere to a specified format, we utilize a 3-shot learning approach. This approach entails presenting the model with one 2-hop question sample, one 3-hop question sample, and one 4-hop question sample.

We also implement the in-context learning (Dong et al., 2023) for LLM-based QA. We provide 4 samples in the prompt for MQuAKE-CF: one 1-edit sample, one 2-edit sample, one 3-edit sample, and one 4-edit sample. When k>=100, we retrieve 4 relevant edit facts for each test sample. When k=1, the prompt consists of all the relevant facts for a specific test sample, given that the edit facts in the memory is less than 5.

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 "?".

4.1.6 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 demonstrates significantly superior performance compared to state-of-the-art models on the MQuAKE datasets, including the MQuAKE-CF dataset and MQuAKE-T dataset. Particularly noteworthy is its performance when handling multiple edits simultaneously. When k=3000 and using GPT-J as the base

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

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

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

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

[9] https://www.wikidata.org/wiki/Help:Label/general_principles

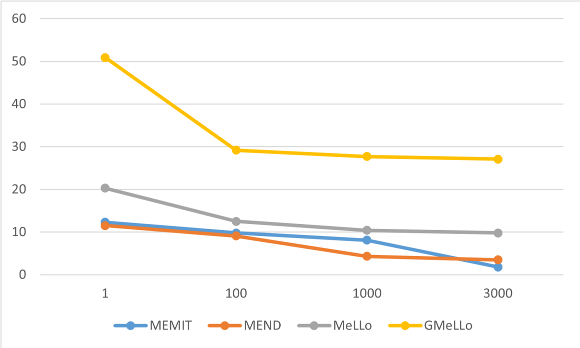

Figure 3: The performance comparison of different methods on MQuAKE-CF dataset when using GPT-J as the base model. The evaluation is conducted with varying numbers of edited instances (k) selected for editing, where k ranges from 1 to 3000.

model, GMeLLo shows an improvement of roughly 18% over MeLLo in MQuAKE-CF, and approximately 30% in MQuAKE-T.

As with many other approaches, we witness a significant decline from k=1 to k=100. This is understandable, as at k=1, all edited facts related to a question are fed into the prompt for LLMs to answer without requiring retrieval. However, the performance stabilizes thereafter. The graph in Figure 3 demonstrates that integrating KBQA enables GMeLLo to maintain higher performance levels, even with an increasing number of edits.

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 and accurately answering the questions becomes increasingly challenging for LLM-based QA.

As depicted in Table 2, when k=1 and all relevant facts are provided to the LLMs for question answering, the process proves to be more 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 this table demonstrates the effectiveness of our GMeLLo, highlighting that KBQA serves as a valuable enhancement to LLM-based QA within evolving environments.

4.4 The Impact of Entity Ambiguity on WikiData

In Section 4.1.4, we emphasize the importance of not only string matching but also the accurate mapping of entity strings to WikiData, ensuring precision in editing and searching. Table 3 reveals that out of 6015 edited facts in the MQuAKE-CF dataset, 1441 fail to map correctly to the intended entities in WikiData. Within these 1441 inaccurately transformed edit facts, 355 are correct in terms of string matching alone but are erroneously linked to unintended entities. Additionally, out of 3000 questions, the subject entity in 466 questions does not correctly match the intended entities in WikiData. Nearly half of these instances are correctly extracted by LLMs but are mismatched due to entity ambiguity.

We acknowledge that while some entities genuinely share the same labels but represent distinct entities, such as multiple individuals bearing identical names, others are indeed identical entities. This suggests that performance could further improve by addressing this issue and working with an enhanced KG, a direction we leave for future work.

4.5 Error 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.

4.5.1 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 under the same conditions. Further analysis revealed that GPT-J struggles in answering questions with only an edited fact pertaining to its intermediary information, such as:

- 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?

BaseModel	Method	MQuAKE-CF				MQuAKE-T			
		k=1	k=100	k=1000	k=3000	k=1	k=100	k=500	k=1868
GPT-J-6B	QA	**66.6**	11.1	7.2	6.4	20.9	10.7	9.4	9.1
GPT-J-6B	KBQA	24.2	24.2	24.2	24.2	63.5	63.5	63.5	63.5
GPT-J-6B	GMeLLo	50.9	**29.2**	**27.7**	**27.1**	**69.9**	**65.1**	**64.9**	**64.8**
Vicuna-7B	QA	**69.8**	15.6	9.1	7.2	**94.4**	56.9	52.9	52.0
Vicuna-7B	KBQA	14.0	14.0	14.0	14.0	37.3	37.3	37.3	37.3
Vicuna-7B	GMeLLo	43.1	**20.4**	**18.1**	**17.5**	75.0	**59.0**	**57.2**	**57.0**

Table 2: Performance comparison as edited facts increase among various methods. 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 questions for answering in a KBQA manner. GMeLLo combines these methods: opting for QA when KBQA yields no response and choosing KBQA when QA and KBQA answers differ.

Model	Edited Fact	Relation Chain
GPT-J-6B	355/1441	205/466
Vicuna-7B	345/2033	206/317

Table 3: The error rate of entity mapping from entity strings to entities in WikiData. Due to entity ambiguity in WikiData, a single string may correspond to multiple entities. In the context of GPT-J and MQuAKE-CF, '355/1441' in the edited fact indicates that out of 1441 errors in correctly extracting the fact triple, 355 errors stem from entity mapping.

- Predicted Answer: Benigno Aquino III
- Label: Bongbong Marcos

However, it is worth noting that all test samples in MQuAKE-T contain only one edited fact. In contrast, approximately 63.6% of test samples in MQuAKE-CF consist of more than 2 edited facts, which allows GPT-J to connect all the information together, resulting in improved performance.

4.5.2 Inferior Performance of Vicuna in KBQA

After analysis, we discovered that out of the 1868 test samples in the MQuAKE-T dataset, 130 samples did not capture the fact triples correctly due to not adhering to the output format. In addition, only 362 relation chains were accurately returned, whereas GPT-J returned 1382 correct relation chains.

It is important to note that even if the relation chain is incorrect, the KBQA system may still provide the correct answer. For instance, in the case of Vicuna, it consistently returns "citizen->country->head of government". Although this is mapped to the predefined relation list as "country of citizenship->country->head of government", whereas the golden chain is "country of citizenship->head of government", the predicted path still leads to the correct answer.

In addition, 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.

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—leveraging the high coverage of LLMs and the precision of using KGs. By utilizing LLMs for analyzing most question sentences and QA, and KBQA to provide accurate results, we achieve a synergy between the two methodologies.

[10] https://spacy.io/

Limitations

Nevertheless, it's important to note that this investigation is still in its early stages. Although our performance surpasses that of baseline approaches in the multi-hop question answering when editing multiple facts simultaneously, we recognize the potential for further improvement. Looking ahead, our future plans involve enhancing GMeLLo in the following key areas:

- Experiment with more sophisticated prompts, such as Chain of Thought (CoT) (Wei et al., 2022), to elevate performance.
- Mitigate the entity ambiguity in KGs to further improve the performance.
- Pioneering the integration of the strengths inherent in both LLMs and KGs, we aim to extend their application to diverse research endeavors.

References

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.

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*.

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*.

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.

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*.

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.

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*.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. 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.

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.

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 Appendix

The appendix will contain further elaboration on the details we used.

A.1 Prompts

The prompts used for edited fact triple extraction, relation chain extraction, and LLM-based QA are depicted in Figures 4, 5, and 6. 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.

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 4: 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 5: 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 6: The prompt used in LLM-based QA.

A.2 Relations

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’]

A.3 Further Details

All experiments are conducted on NVIDIA RTX A5000 GPUs, with the temperature of LLMs set to 0 across all tasks.