Semi-Structured Chain-of-Thought: Integrating Multiple Sources of Knowledge for Improved Language Model Reasoning

Anonymous ACL submission

Abstract

An important open question in the use of large language models for knowledge-intensive tasks is how to effectively integrate knowledge from three sources: the model's parametric memory, external structured knowledge, and external unstructured knowledge. Most existing prompting methods either rely on one or two of these sources, or require repeatedly invoking large language models to generate similar or identical content. In this work, we overcome these limitations by introducing a novel semi-structured prompting approach that seamlessly integrates the model's parametric memory with unstructured knowledge from text documents and structured knowledge from knowledge graphs. Experimental results on open-domain multi-hop question answering datasets demonstrate that our prompting method significantly surpasses existing techniques, even exceeding those that require fine-tuning.

1 Introduction

Large language models (LLMs) have demonstrated an impressive breadth of capabilities in the field of natural language processing (NLP). LLMs can be adapted to achieve strong performance on a wide variety of tasks without additional training by *few-shot prompting*: conditioning generation on instructions and several exemplars (Brown et al., 2020b). However, few-shot prompting may produce hallucinations due to an under-representation of knowledge in training datasets (OpenAI, 2023), which compromises their suitability for tasks in which a high degree of factual accuracy is necessary.

Recent works have explored alternative prompting strategies to mitigate this issue in knowledge-intensive tasks. These include guiding the model to generate intermediate steps before producing the final answer (Wei et al., 2022; Wang et al., 2023b) and using external tools, such as information retrievers, to utilize external knowledge (Yao et al., 2023b; Trivedi et al., 2023). These strategies have somewhat addressed LLM hallucination issues and have enhanced the model's capability to tackle knowledge-intensive tasks. However, there are still limitations to these prompting techniques.

First, current methods do not fully utilize all available knowledge sources. Knowledge sources can be categorized into parametric memory inside the model, and structured or unstructured external knowledge. Parametric memory is learned from large datasets during pre-training and stored in the model's parameters. Unstructured external knowledge typically refers to text-based knowledge, such as paragraphs from Wikipedia documents, while structured external knowledge is usually in the form of tables or knowledge graphs such as Wikidata (Vrandečić and Krötzsch, 2014). Most existing prompting strategies utilize only one or two types of knowledge sources. For example, Chain-of-Thought (CoT; Wei et al., 2022) and several follow-up works (Zhang et al., 2023; Yao et al., 2023a; Long, 2023) mainly focus on invoking knowledge stored internally in the model's parametric memory, which cannot be updated without further training. Other methods like Shao et al. (2023) and Trivedi et al. (2023) attempt to integrate parametric memory within the model with external unstructured text, by making multiple calls to LLMs and using external information retrievers to search for documents relevant to the task. These approaches omit the large amount of information stored in knowledge graphs.

Second, there is a lack of seamless synergy between LLMs and external tools. This leads to costly repeated calls to LLMs and post-processing of external tool results during the reasoning process. Consequently, most recent methods have only been tested on a limited number of examples in small-scale experiments, such as in Trivedi et al. (2023), Yoran et al. (2023) and Jiang et al. (2023b). For instance, Gou et al. (2023), Yao et al. (2023b),

and Wang et al. (2023a) use external tools to provide feedback on content generated by LLMs, and then repeatedly invoke LLMs to regenerate content based on this feedback. This often results in the model regenerating a large amount of the same content, which is inefficient and difficult to scale. Moreover, new hallucinations may be introduced in the regeneration process.

Our work addresses the research question of how to efficiently integrate the three primary knowledge sources during inference time: parametric memory of LLMs, external structured knowledge, and external unstructured knowledge. We propose a semi-structured chain of thought method, focusing on multi-hop reasoning question answering. First, we use an LLM with few-shot prompting to parse the question into a masked semi-structured reasoning chain, followed by syntactic-based filtering to retain only those reasoning chains that are syntactically correct. Then we use external tools such as document retrievers and entity linkers to sequentially fill in the masks within the reasoning chain. Finally, we call upon LLMs as needed to fill in any remaining masks to arrive at the final answer.

To demonstrate the effectiveness of our approach, we conduct extensive experiments using open-source LLMs across various model sizes on several multi-hop question answering datasets. Our contributions are as follows:

1. We propose a simple, intuitive, and efficient inference-time method to integrate various sources of knowledge for reasoning.
2. We compare our method with existing approaches on multi-hop question answering datasets. Our method achieves state-of-the-art performance, surpassing even those alternatives that require supervised fine-tuning.
3. We conduct detailed analyses to investigate the significance of each element of our method and make our code publicly available.[1]

2 Related work

LLM reasoning with CoT. Since Wei et al. (2022) first proposed CoT prompting, a variety of CoT-style approaches have been proposed to further improve the reasoning capabilities of LLMs. Whereas the original CoT utilized greedy decoding, Wang et al. (2022) showed that prediction accuracy can be improved by sampling a diverse set of CoT reasoning paths and selecting the final answer from the plurality of predictions among samples. Kojima et al. (2022) demonstrated that even zero-shot CoT significantly improves the reasoning abilities of LLMs across a variety of tasks. Madaan and Yazdanbakhsh (2022) proposed a concise CoT which prunes intermediate steps to only contain key text and patterns necessary for reasoning.

Least-to-most prompting (Zhou et al., 2022) extends the ability of CoT to generalize to harder problems than seen in examples by breaking down complex reasoning tasks into easier sub-problems. Yoran et al. (2023) propose multi-chain reasoning, which reasons over multiple CoTs rather than only aggregating their answers. Tree of Thoughts (Yao et al., 2023a) also considers multiple reasoning paths while enabling look aheads and backtracking through self-evaluation. In contrast with our Semi-CoT methodology, the aforementioned approaches rely solely on the use of the LLMs parametric memory for reasoning and do not integrate CoT reasoning with external knowledge sources.

Retrieval-augmented approaches for multi-hop reasoning. To address the lack of knowledge necessary for complex reasoning tasks in the parametric memory of LLMs, several methods which retrieve information from external sources have been proposed. Press et al. (2023) introduced Self-Ask, which prompts the LLM to ask follow-up questions which can be answered by external search engines. Demonstrate-Search-Predict (DSP) (Khattab et al., 2022) breaks down multi-hop questions into sub-problems, leveraging retrieval models to help generate intermediate answers using external documents. ReAct (Yao et al., 2022) and IRCoT (Trivedi et al., 2023) interleave CoT with retrieval of documents from external sources to reduce hallucinations and improve accuracy on multi-hop QA. Iter-RetGen (Shao et al., 2023) uses complete model outputs to retrieve relevant documents over multiple iterations to refine the answer.

Verify-and-edit (Zhao et al., 2023) identifies cases where sampled CoTs have lower-than-average consistency and then retrieves external knowledge from search engines and document repositories to edit the rationales generated by the LLM. Jiang et al. (2023b) also propose an adaptive approach for retrieving external knowledge by identifying the presence of low-probability tokens in intermediate steps. LLM-Augmenter (Peng et al., 2023) uses multiple sub-modules to iteratively re-

[1]We will release our code upon publication under the MIT license.

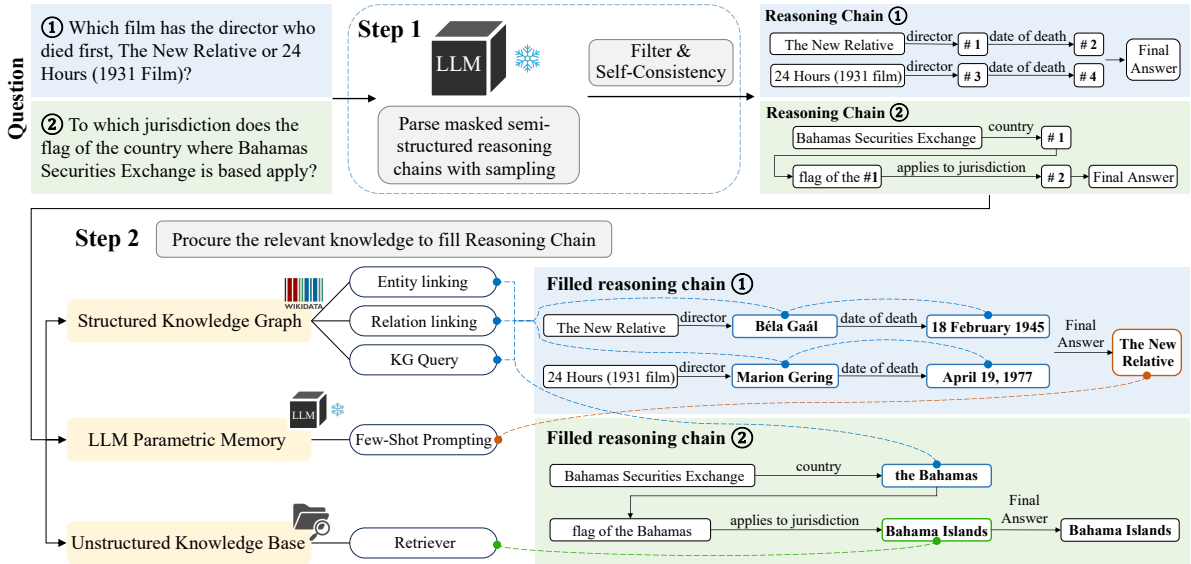

Figure 1: Overview of our approach using different sources of knowledge.

fine generations through fact checking and retrieval of supporting information from external knowledge sources. These approaches differ from ours primarily in their exclusive focus on integrating external knowledge from text sources, whereas our use of a semi-structured CoT format enables the seamless integration of both unstructured text and structured knowledge sources.

Structured knowledge integration with LLMs. Relatively little prior work has explored strategies for integrating structured knowledge with frozen LLMs at inference time. ERNIE 3.0 (Sun et al., 2021) and SKILL (Moiseev et al., 2022) both train LLMs using triples from structured knowledge graphs to infuse knowledge into the LLM's parametric memory. QA-GNN (Yasunaga et al., 2021) and GreaseLM (Zhang et al., 2022) integrate knowledge graph embeddings with language model encodings through the use of GNNs. Recently, Jiang et al. (2023a) proposed an iterative approach for retrieving and reasoning over information from structured knowledge sources during LLM inference by linearizing retrieved knowledge into text. In contrast, our approach facilitates the integration of both unstructured text and structured knowledge without requiring specialized linearization interfaces.

3 Methodology

Overview We focus on multi-hop question answering tasks. To synthesize the answers to a given multi-hop question, we synergistically integrate an LLM, structured knowledge from an external Knowledge Graph (KG), and unstructured textual knowledge from an external text knowledge base (TKB). Our method unfolds in two steps:

1. We prompt an LLM to parse the input question into a semi-structured reasoning chain with masks as placeholders.
2. We procure knowledge from three sources to fill in the masks in the reasoning chain, thereby deriving the final answer. We leverage knowledge from the LLM via few-shot prompting, use entity and relation linking models to ground the reasoning chain to the KG, and employ a dense retriever to fetch relevant documents from the TKB.

No components of our method require any additional training. A major strength of our method is its simplicity, intuitiveness, and ease of use. We present an overview of our method in Figure 1.

3.1 Semi-Structured Reasoning Chain Parsing

Translating a question into a semi-structured reasoning chain can be seen as a semantic parsing task, where the multi-hop question serves as the utterance, and the reasoning chain serves as the logical form. Unlike standard semantic parsing tasks such as text2SQL, the logical forms we parse cannot be directly used to retrieve answers from a database. Instead, the subsequent steps involve populating the masked placeholders in the reasoning chains with accurate knowledge to derive the final answer.

Drawing inspiration from Brassard et al. (2022) and the annotation of Trivedi et al. (2022), we posit that a good reasoning chain should use the most rel-

evant set of facts to coherently connect the question to the answer, creating a minimal bridge between the two. To establish this bridge, our reasoning chain uses a series of triplet structures, each expressed as `(head, relation, tail)`. The `head`, `relation`, and `tail` can appear as specific entity mentions or relations in a knowledge graph, as free-form text, or as masks symbolized by a # followed by an ascending number (e.g., `#1`, `#2`) indicating parts that need to be populated. These masks can be either of the two aforementioned forms. The reasoning chain ends with a masked final answer, denoted as `final answer: #answer` or `final answer: #<number>` based on specific reasoning logic. This blended structure allows us to merge structured and unstructured knowledge, creating a unified reasoning pathway.

We use an LLM to parse masked reasoning chains from each input question with few-shot prompting. For example, for the question:

> "To which jurisdiction does the flag of the country where the Bahamas Securities Exchange is based apply?",

the corresponding reasoning chain is:

> "(Bahamas Securities Exchange, country, `#1`); (flag of the `#1`, applies to jurisdiction, `#2`); final answer: `#2`"

A natural question is: why not adhere strictly to either structured or unstructured formats? The answer lies in the varying complexity of grounding different knowledge pieces to their sources, and the extent to which these sources cover the knowledge.

For example, understanding the positions held by George Washington through his Wikipedia page requires a deep dive into a long document, semantic understanding, and temporal reasoning. While challenging to associate with unstructured sources, this knowledge is easily depicted in the Wikidata knowledge graph with a series of triples, such as (George Washington, position held, Commander-in-Chief) (as shown in the example in Appendix A Figure 2). Nevertheless, the scope of knowledge graphs is finite; we cannot encapsulate every fact in a structured form. Therefore, semi-structured triplets with free-form text are also essential to ground knowledge with other unstructured sources.

As in other semantic parsing tasks, LLMs can produce both syntactic and semantic errors when parsing reasoning chains. We define those reasoning chains that do not follow the pre-defined format as reasoning chains with "syntactic errors". We define reasoning chains that are syntactically correct but logically flawed, meaning the reasoning chains cannot coherently connect the question to the answer, as "semantic errors". To address syntactic errors, we employ the LLM to parse multiple reasoning chains for each input question through sampling. We then filter out any chains that have syntactic errors (see Appendix B for the types of errors we filter). Finally, we apply the Self-Consistency strategy (Wang et al., 2022) to derive the final masked semi-structured reasoning chain. We leave addressing semantic errors to future work.

3.2 Masked Reasoning Chain Filling

We procure factual knowledge from three sources to fill the masks within parsed reasoning chains: the LLM's parametric memory, the structured triples of the KG, and the unstructured documents from the TKB. We generally prioritize parametric memory last due to its lower reliability in knowledge retrieval. The choice between structured and unstructured knowledge often depends on the predominant knowledge type required for the specific task. In our experiments, we have explored all possible sequences of their application. We iteratively fill the masks in each reasoning chain triplet from left to right, leveraging these knowledge sources. If we encounter instances where the triplets in the chains cannot be grounded to the available structured or unstructured knowledge, we leave those masks unfilled. In each iteration, we extract valid triplets (where both the head and the relation are unmasked, and only the tail is masked) from the generated masked reasoning chains for filling.

Use of Structured Knowledge Algorithm 1 presents our algorithm for filling the reasoning chain using a structured knowledge source. For the KG grounding of a specific triplet (line 8), we take a triplet from Figure 1 as an example: (24 Hours (1931 film), director, `#3`). We begin by conducting entity linking for "24 Hours (1931 film)" and relation linking for "director." We then query the KG to identify the entity corresponding to "`#3`", which is Marion Gering. This allows us to fill in the reasoning chain's "`#3`" mask.

Use of Unstructured Knowledge Algorithm 2 presents our algorithm for filling reasoning chains using unstructured textual knowledge. Here, our approach for filling each valid triplet adheres to the classic retrieve-and-read strategy. For instance,

Algorithm 1: Fill the Masked Reasoning Chains with Structured Knowledge

Input:
RC: the generated masked reasoning chains
KG: the external knowledge graph
EL: the entity linking model
RL: the relation linking model

$T_q \leftarrow \emptyset$
while ***true*** **do**
  $T_v \leftarrow$ extract valid triplets from RC
  $T_v \leftarrow T_v \setminus \{t \mid t \in T_q\}$
  if $T_v = \emptyset$ **then**
    break;
  $T_q \leftarrow T_q \cup T_v$
  $T_g \leftarrow$ ground each triplet in T_v onto KG using EL and RL
  if $T_g \neq \emptyset$ **then**
    $T_a \leftarrow$ Query KG using T_g
    if $T_a \neq \emptyset$ **then**
      $RC \leftarrow$ fill RC with information from T_a
return RC

Algorithm 2: Fill the Masked Reasoning Chains with Unstructured Knowledge

Input:
LLM: the pre-trained language model
RC: the generated masked reasoning chains
TKB: the external unstructured text knowledge base
$Retriever$: the dense retriever
k: the number of top documents to retrieve

$T_q \leftarrow \emptyset$
while ***true*** **do**
  $T_v \leftarrow$ Extract valid triplets from RC
  $T_v \leftarrow T_v \setminus \{t \mid t \in T_q\}$
  if $T_v = \emptyset$ **then**
    break;
  $T_q \leftarrow T_q \cup T_v$
  $Q_{gen} \leftarrow$ few-shot prompt LLM to generate single-hop questions for each triplet in T_v
  $D_{ret} \leftarrow$ use $Retriever$ to retrieve the top k documents for each question in Q_{gen} from TKB
  $T_a \leftarrow$ few-shot prompt LLM to answer each question in Q_{gen} based on corresponding documents in D_{ret}
  if $T_a \neq \emptyset$ **then**
    $RC \leftarrow$ fill RC with information from T_a
return RC

consider the triplet from Figure 1: (Bahamas Securities Exchange, country, #1). We first prompt the LLM to convert it into a straightforward single-hop question: In which country is the Bahamas Securities Exchange located? Next, we employ the retriever to fetch relevant documents from the TKB. Finally, we combine the retrieved documents with the question and prompt the LLM again to obtain the answer: the Bahamas.

Use of Parametric Memory We utilize the parametric memory of the LLM in the final step of our method. Specifically, we employ few-shot prompting to have the LLM fill in any remaining masks. For instance, in the case of Figure 1, question 1, even after all the masks in the reasoning chain have been filled, we may still need to rely on the LLM for a final comparison of time to arrive at the final answer.

4 Experiments

4.1 Datasets

We evaluate our proposed method on three knowledge-intensive multi-hop reasoning datasets: 2WikiMultihopQA (2Wiki; Ho et al., 2020), MuSiQue-Ans (MuSiQue; Trivedi et al., 2022) and Bamboogle (Press et al., 2023). Following previous works (Press et al., 2023; Shao et al., 2023; Chen et al., 2023), due to the unavailability of labels for the test sets of 2Wiki and MuSiQue, we utilize their training sets to develop our prompts and the development sets as our test sets. Bamboogle only provides a test set, and we test our method on its entire test set. Additional dataset details are in Appendix C.

Evaluation Metrics For the 2Wiki and MuSiQue datasets, we employ the official evaluation methods from their respective code bases to compute the Exact Match (EM) and F1 scores for the answers. The Bamboogle code base does not offer an official evaluation method. Instead, we use the evaluation code from MuSiQue to calculate the metrics.

4.2 Implementation Details

Base Models In all of our experiments, we use few-shot in-context learning and do not perform any model training. We utilize three different sizes of language models from the LLAMA 2 family (7b, 13b, and 70b) (Touvron et al., 2023) as our base models because of their strong empirical results and open-source availability. Additionally, we also use the LLAMA-65b model in some of our experiments for a fair comparison with previous methods. We employ these base models to parse semi-structured reasoning chains, convert masked triplets into single-hop questions for answering, and fill in reasoning chains.

Prompts All of our in-context learning exemplars are randomly sampled from the training set of the corresponding dataset, and their semi-

structured reasoning chains are manually annotated by the authors. Since Bamboogle does not have a training set, we use examples from the MuSiQue training set for this purpose. More details about the prompts can be found in Appendix D.

Knowledge Sources and Usage 2Wiki and MuSiQue were initially created for reading comprehension, with each question accompanied by multiple context paragraphs, including supporting and distracting paragraphs. Following Chen et al. (2023) and Trivedi et al. (2023), we adapt them for an open-domain setting by collecting all context paragraphs from the training, development, and test sets' questions to serve as an unstructured TKB. Bamboogle was originally an open-domain setup. We follow the original setup and use Google Search API [2] to access the entire web as the unstructured TKB. We host a local Wikidata (Vrandečić and Krötzsch, 2014) endpoint as our structured knowledge source for all the datasets. We employed the pre-trained Contriever (`Contriever-msmarco`) (Izacard et al., 2022) for document retrieval, the BLINK model (Wu et al., 2020) for entity linking, and the all-MiniLM-L6-v2 model from Sentence-Transformers (Reimers and Gurevych, 2019) for relation linking. See Appendix E for more details.

4.3 Baselines

On 2Wiki and MuSiQuem datasets, we compare our proposed method with the following state-of-the-art prompting methods: standard few-shot prompting (standard) (Brown et al., 2020a), CoT prompting (Wei et al., 2022), Self-Ask (Press et al., 2023), ITER-RETGEN (Shao et al., 2023), ReAct (Yao et al., 2023b), and DSP (Khattab et al., 2022). We also contrast our method with Data Synthesis (Chen et al., 2023), which requires fine-tuning. Data Synthesis generates millions of synthetic question-answer pairs and fine-tunes the base LLM on the generated data, utilizing a prompting method similar to ReAct. More detailed baseline descriptions are provided in Appendix F.

The Bamboogle dataset does not provide an official evaluation method[3], which makes it challenging to directly compare our results with previously reported model performance. Considering the cost of Google Search API call, we replicate the best-performing baselines, Self-Ask (see analysis in Section 4.4), based on the LLAMA2-70b model on Bamboogle for a fair comparison[4].

4.4 Main Results

We present the performance comparison between the baselines and our method in Tables 1 and 2.

Which baseline methods perform best? We compare the baselines on larger datasets, 2Wiki and MuSiQue, as shown in the first two sections of Table 1. For those methods not involving fine-tuning, Self-Ask is particularly effective. When using the text-davinci-003 as the base model, Self-Ask excels in all metrics across all datasets, indicating that simply generating clear multi-round sub-questions and retrieving relevant documents is more effective than more complex methods such as ITER-RETGEN. This holds true when Self-Ask is applied to the LLAMA-65b model, where it also performs on par with DSP. On the other hand, the Data Synthesis method, fine-tuned on more than one million synthesized question-answer pairs, surpasses other inference-only methods in all metrics on both datasets. The weakest performance is observed in the standard few-shot prompting method, which does not generate any intermediate steps.

Does our method outperform the baseline methods? Our method significantly improves over the standard few-shot prompting across all models, datasets, and metrics. For example, compared to the standard few-shot prompting, our KG+Text+Model method improves the exact match score by 148%, 179%, and 135% on the 2Wiki, MuSiQue, and Bamboogle datasets, respectively.

When compared with the existing state-of-the-art inference-only prompting method, Self-Ask, our method still exhibits substantial superiority across all models, datasets, and metrics. When using the same base model, our KG+Text+Model method surpasses the exact match score of Self-Ask by 120% and 60% on the 2Wiki and MuSiQue datasets, respectively (based on LLAMA-65B), and also exceeds it by 10% on the Bamboogle dataset (based on LLAMA-70B).

Compared to the Data Synthesis+SC method that involves fine-tuning, our KG+Text+Model approach surpasses it by 53% in exact match score on 2Wiki, using the same base LLM, LLAMA-65b. Additionally, our method shows very similar

[2] https://serpapi.com/

[3] The original work relies on human evaluation for model performance.

[4] Our replication of Self-Ask uses the officially released code: https://github.com/ofirpress/self-ask.

Method	Inference Only	Base Model	2Wiki EM	2Wiki F1	MuSiQue EM	MuSiQue F1
Self-Ask *	✓	text-davinci-003	0.37	0.49	0.28	0.42
CoT *	✓	text-davinci-003	0.30	0.40	0.19	0.31
ITER-RETGEN *	✓	text-davinci-003	0.36	0.47	0.26	0.42
ReAct *	✓	text-davinci-003	0.28	0.39	0.23	0.37
Data Synthesis + SC ♢	✗	llama-65b	0.51	0.60	0.31	0.42
DSP ♢	✓	llama-65b	0.36	0.44	0.21	0.29
Self-Ask ♢	✓	llama-65b	0.35	0.42	0.20	0.28
Ours						
Standard	✓	llama-65b	0.29	0.34	0.11	0.21
Text + KG + Model	✓	llama-65b	0.58	0.63	0.32	0.41
KG + Text + Model	✓	llama-65b	0.78	0.82	0.32	0.40
Standard	✓	llama2-7b	0.27	0.33	0.07	0.15
Text + KG + Model	✓	llama2-7b	0.44	0.49	0.24	0.33
KG + Text + Model	✓	llama2-7b	0.65	0.70	0.25	0.34
Standard	✓	llama2-13b	0.28	0.34	0.09	0.18
Text + KG + Model	✓	llama2-13b	0.50	0.56	0.32	0.40
KG + Text + Model	✓	llama2-13b	0.68	0.73	0.31	0.40
Standard	✓	llama2-70b	0.33	0.39	0.14	0.24
Text + KG + Model	✓	llama2-70b	0.73	0.77	**0.40**	**0.50**
KG + Text + Model	✓	llama2-70b	**0.82**	**0.87**	0.39	0.48

Table 1: Evaluation results on 2Wiki and MuSiQue datasets. The highest performance is **bolded**. SC is self-consistency (Wang et al., 2023b). Text/KG/Model refers to the use of retrieved paragraphs/the knowledge graph/model's parametric knowledge to fill the reasoning chains. ∗ indicates results from Shao et al. (2023), who used a smaller-scale evaluation. We use their results for comparison to identify the strongest baselines. ♢ indicates results from Chen et al. (2023), who employed the same evaluation settings as ours.

Method	EM	F1
Self-Ask ♡	0.49	0.62
Standard	0.23	0.34
Text + KG + Model	**0.54**	0.67
KG + Text + Model	**0.54**	**0.69**

Table 2: Evaluation results on Bamboogle dataset using LLAMA2-70b as the base model. The highest performance is **bolded**. Text/KG/Model refers to the use of retrieved paragraphs/the knowledge graph/model's parametric knowledge to fill the reasoning chains. ♡ indicates our replication.

performance on MuSiQue, with a slightly higher exact match and a slightly lower F1 score. When employing LLAMA2-70b as the base model, our approach again significantly outperforms the Data Synthesis+SC method on both datasets.

Does the order of using knowledge sources matter? The impact of using knowledge sources in different sequences varies across datasets. For the 2Wiki dataset, prioritizing structured knowledge from knowledge graphs to fill reasoning chains proves to be significantly more effective than starting with retrieved textual knowledge, regardless of the model size. This can likely be attributed to the dataset's integration of a substantial amount of structured knowledge at the time of its creation, making its questions more amenable to decomposition into structured triplets. In this case, entity linking models can more easily identify and match structured knowledge. Hence, for 2Wiki, knowledge graphs may serve as a more reliable source of knowledge compared to retrieved text knowledge. However, the MuSiQue and Bamboogle datasets, which mainly consist of a composition of single-

Model	2Wiki		Mu		Ba	
	EM	F1	EM	F1	EM	F1
Model	0.46	0.52	0.24	0.33	0.42	0.55
Text + Model	0.62	0.67	0.39	0.49	**0.57**	**0.70**
KG + Model	0.81	0.85	0.28	0.38	0.40	0.55
KG + Text + Model	**0.82**	**0.87**	0.39	0.48	0.54	0.69
Text + KG + Model	0.73	0.77	**0.40**	**0.50**	0.54	0.67

Table 3: Performance of using different knowledge sources on 2Wiki, MuSiQue (Mu), and Bamboogle (Ba) with LLAMA2-70b as the base model.

hop questions, seem less sensitive to the preference for knowledge sources, rendering the sequence in which knowledge is used less impactful.

Does the model size matter? Our method benefits from increased model size, resulting in continuous performance improvements, regardless of the order in which sources of knowledge are utilized. This trend aligns with the majority of current LLM prompting methods. Notably, our method also significantly boosts the performance of the smaller LLAMA2-7b model, allowing it to surpass the LLAMA2-70b model with standard few-shot prompting, which is ten times larger, across all datasets and metrics.

5 Analysis

What is the effect of different knowledge sources? We conduct ablation experiments on all the datasets, with the results presented in Table 3. As discussed in Section 4.4, the characteristics of the dataset significantly influence the contributions of the knowledge sources to the final performance of the method. On the 2Wiki dataset, using the KG and LLM's parametric memory yields comparable results to using all three knowledge sources. However, for MuSiQue and Bamboogle datasets, combining external text knowledge with model memory alone was enough to match or exceed the performance of using all knowledge sources.

What is the impact of using an oracle KG? The 2Wiki dataset provides relation triples as gold evidence for each question. We utilize these triples to construct an Oracle KG, significantly smaller than Wikidata and employ string matching to link the triplets in our reasoning chain to the Oracle KG. This approach aims to simulate our method's performance with enhanced entity linking and relation linking. The results are presented in Appendix G Table 7. Using the Oracle KG indeed improves performance, but the extent of this improvement is considerably smaller compared to the enhancement our method achieves over standard few-shot prompting techniques and other state-of-the-art prompting methods. This outcome also suggests that our approach can effectively leverage large-scale KGs without the need for task-specific fine-tuning of the entity linking model.

What are the errors in our method? To better understand the errors of our proposed method, we run our LLAMA2-70b-based KG+Text+Model method on a sample of 200 training instances from the 2Wiki and MuSiQue datasets. We manually annotate 37 and 50 errors sampled from these instances (totaling 87 errors). (In 2Wiki, there are only 37 errors.) Table 8 in Appendix H provides the error types and corresponding examples. In the 2Wiki dataset, the majority of the errors, approximately 78%, are false negatives, where the model prediction are alternative phrasings of the gold answers (e.g., the model predicts "1", but the reference answer is "one"). The remaining errors include 19% knowledge retrieval errors, where the model fails to correctly utilize relevant knowledge to fill the reasoning chain (e.g., failing to retrieve the correct dates), and 3% reasoning chain errors, where the model's initial reasoning chain is incorrect (e.g., confusing who in the question a requested position refers to). However, in the MuSiQue dataset, 52% of errors are due to knowledge retrieval, 26% are reasoning chain errors, and 22% are false negatives. These analyses suggest that our method might be underestimated for the 2Wiki dataset, which relies more on structured knowledge. For datasets like MuSiQue, which depend more on unstructured knowledge, performance could be significantly improved by fine-tuning knowledge retrieval methods, such as document retrievers, and filtering out reasoning chains with semantic errors.

6 Conclusion

In this work, we explored integrating knowledge from three different sources through semi-structured reasoning chains. We found that our approach significantly improved the performance of LLMs on multi-hop reasoning tasks, surpassing other state-of-the-art prompting methods. In future work, we plan to improve our method through semantic error filtering in reasoning chain parsing and knowledge retriever fine-tuning.

Limitations

In this work, we exclusively investigate utilizing large language models from the open-source LLAMA family as our base models. We do not examine other potentially more robust open-source or proprietary models. We also only focus on the inference capabilities of these models without engaging in any training or fine-tuning processes. It is likely that training could improve performance, though it would require significant additional computational resources.

Ethics Statement

Wikipedia contains certain biases (Falenska and Çetinoğlu, 2021), and we use data from Wikipedia in the text knowledge bases and knowledge graphs for retrieval, so we are potentially introducing similar biases into our method.

References

Ana Brassard, Benjamin Heinzerling, Pride Kavumba, and Kentaro Inui. 2022. COPA-SSE: Semi-structured explanations for commonsense reasoning. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3994–4000, Marseille, France. European Language Resources Association.

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. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mingda Chen, Xilun Chen, and Wen-tau Yih. 2023. Efficient open domain multi-hop question answering with few-shot data synthesis. *arXiv preprint arXiv:2305.13691*.

Agnieszka Falenska and Özlem Çetinoğlu. 2021. Assessing gender bias in Wikipedia: Inequalities in article titles. In *Proceedings of the 3rd Workshop on Gender Bias in Natural Language Processing*, pages 75–85, Online. Association for Computational Linguistics.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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*.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

O. Khattab, Keshav Santhanam, Xiang Lisa Li, David Leo Wright Hall, Percy Liang, Christopher Potts, and Matei A. Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *ArXiv*, abs/2212.14024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Jieyi Long. 2023. Large language model guided tree-of-thought.

Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 1581–1588, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. ♫ MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

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

Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. 2023a. Boosting language models reasoning with chain-of-knowledge prompting.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

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

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

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.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. *arXiv preprint arXiv:2304.13007*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Appendix

A Comparison of Structured and Unstructured Knowledge Sources

In Figure 2, we show that sometimes structured knowledge is easier to use compared to unstructured knowledge.

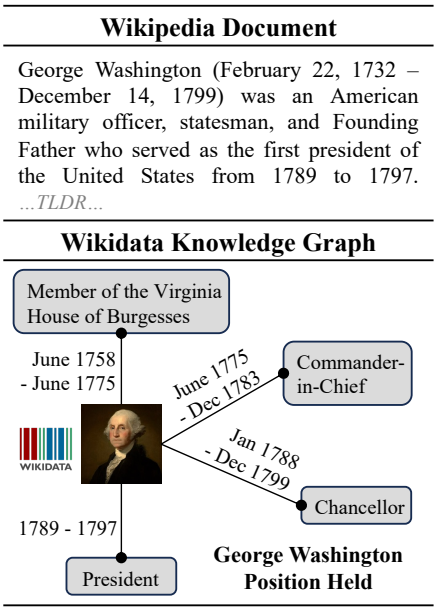

Figure 2: Comparison of structured and unstructured knowledge.

B Pre-defined Syntactic Errors

When filtering sampled masked reasoning chains, we filter out chains that contain the following syntactic errors.

- ***MaskUnderflowError/MaskOverflowError***: The two errors refer to the number of unique masks in the reasoning chain being either below the minimum threshold or above the maximum threshold. By pre-setting the quantity of masks, we can limit the number of masks included in the reasoning chain generated by LLM. The number of hops in a multi-hop question is typically finite. For different tasks, we can set the possible number of masks based on prior knowledge. For instance, in multi-hop question-answering tasks, the reasoning chain should have at least two unique masks and rarely involve more than four hops. Thus, we can set the minimum number of masks to two and the maximum to four, filtering out reasoning chains that don't meet these criteria.
- ***DiscontinuousMaskError***: This error indicates that the sequence numbers of masks in the reasoning chain are not sequential, such as jumping from #1 to #3.
- ***MissingMaskNumberError***: This error refers to an issue where, apart from the final answer mask "#answer", the generated mask lacks a number following the "#".
- ***MissingTripleMaskError***: The error refers to the case where neither the head nor the tail in the generated triple is masked.
- ***IncorrectTripleRelationMaskError***: We define that only the head and tail can be masked. This error points to cases where the relation in a triple is masked.
- ***IncorrectTripleFormatError***: This error indicates that the generated triple doesn't follow the "head >> relation >> tail" format. For example, the LLM might split the relation into two parts separated by ">>", turning the triple into "head >> relation words >> relation words >> tail."
- ***FinalAnswerFormatError***: This error is flagged when the generated reasoning chain doesn't conclude with the format "final answer: #answer."

These error definitions can be expanded upon based on specific tasks or results from error analysis. After filtering, we employ the self-consistency method to obtain the final reasoning chain from the filtered chains.

C Datasets

2WikiMultihopQA 2WikiMultihopQA is a multi-hop question answering dataset that contains various reasoning types and is developed based on HotpotQA (Yang et al., 2018). However, it excludes the single-hop and context-dependent multi-hop question types. The dataset contains 192,606 questions in total, split into 167,454 for training, 12,576 for development, and 12,576 for testing. We evaluate our method on the entire development set.

MuSiQue MuSiQue-Ans (MuSiQue; Trivedi et al., 2022) is comprised of answerable multi-hop questions that are constructed from a large number of single-hop questions in a bottom-up method. These questions are designed to force the model to connect all supporting facts for connected reasoning instead of seeking shortcuts to find the answers. MuSiQue consists of a total of 24,814 questions,

split into 19,938 training samples, 2,417 development samples, and 2,459 test samples. Following previous works (Press et al., 2023; Shao et al., 2023; Chen et al., 2023), we focus on 2-hop questions (1, 252 in the development set). This is because, as also discussed in Press et al. (2023), many of the automatically composed 3-hop and 4-hop questions in MuSiQue are unnatural and even challenging for humans to comprehend.

Bamboogle Bamboogle (Press et al., 2023) is designed to evaluate the compositionality skills of models. It consists of 125 two-hop questions from various categories, all annotated by humans. These questions are based on Wikipedia but cannot be directly answered through a Google search and have not previously appeared on the web.

D In-Context Learning Exemplars

In our experiments on the 2Wiki and MuSiQue datasets, we employ the following settings:

- For parsing semi-structured reasoning chains and using the model's parametric memory to fill in masks, we use a 25-shot setting.
- For converting masked triplets into single-hop questions and answering these questions, we use a 15-shot setting.
- For standard few-shot prompting, we use a 25-shot setting.

The experiment in the Bamboogle dataset differs from those mentioned above in that, for parsing semi-structured reasoning chains, using the model's parametric memory to fill in masks, and in standard few-shot prompting, we use a 4-shot setting.

We present examples of the in-context learning exemplars used throughout our method in Table 4 and exemplars used in the standard few-shot prompting experiments in Table 5.

E Implementation Details

Knowledge Sources and Usage Details The TKB for 2Wiki consists of 398,354 paragraphs, and for MuSiQue, it contains 139,409 paragraphs. For querying unstructured text knowledge sources, we use the pre-trained Contriever (`Contriever-msmarco`) (Izacard et al., 2022) to retrieve the top 10 most relevant paragraphs for each single-hop question converted from a structured triplet. We employ the off-the-shelf BLINK entity linking model (Wu et al., 2020) to link the head of the structured triplet to Wikipedia, and use WikiMapper [5] to obtain the corresponding Wikidata entry. For relation linking, we use the `all-MiniLM-L6-v2` model from SentenceTransformers (Reimers and Gurevych, 2019) to encode the relation string in the structured triplet and all the property labels from Wikidata and calculate the cosine similarity between them to find the most similar Wikidata property. We utilize the results of entity and relation linking to construct simple SPARQL queries and execute them on the Wikidata endpoint to obtain the entities corresponding to the masks.

Model Inference We use the vLLM library v0.1.3 (Kwon et al., 2023) for model inference. All inference is conducted on eight Nvidia A6000 GPUs. The total GPU hours are around 780 hours.

F Baselines Details

We compare our approach to the following state-of-the-art methods:

Standard Few-Shot Prompting Standard few-shot prompting (Standard) (Brown et al., 2020a) simply involves prompting the model to generate answers using the few-shot in-context learning exemplars. We utilize a 25-shot setting.

CoT Prompting CoT prompting (Wei et al., 2022) incorporates intermediate reasoning steps into the few-shot exemplars, guiding the model to generate step-by-step thought processes before producing the final answer.

Self-Ask Unlike the CoT approach, where the model generates the entire thinking process and final answer in one pass, Self-Ask (Press et al., 2023) adopts a multi-round self-questioning method. Given a multi-hop question, the Self-Ask method prompts the LLM to generate explicit intermediate questions and uses a search engine to find answers to these questions. This process continues until the LLM produces the final answer.

ITER-RETGEN ITER-RETGEN (Shao et al., 2023), similar to Self-Ask, answers multi-hop questions through iterative prompting of LLMs and retrieval. However, unlike Self-Ask, ITER-RETGEN does not prompt LLMs to generate explicit sub-questions for each iteration. Instead, ITER-RETGEN utilizes content generated by the model in the previous iteration along with the given

[5] https://github.com/jcklie/wikimapper

Parse semi-structured reasoning chains on the MuSiQue Dataset
Question: What are the spirits associated with Shintoism called in the language of Lala DX? Reasoning Chain: LaLa DX » language » #1; spirits associated with Shintoism in #1 » name » #2; final answer: #2 Question: When did the person with the famous quote we came we saw we conquered die? Reasoning Chain: famous quote "we came we saw we conquered" » author » #1; #1 » date of death » #2; final answer: #2 ... Question: Which flag was made first between Cuba and the country with the immigration? Reasoning Chain: country with the immigration » name of the country » #1; flag made first between Cuba and #1 » name » #2; final answer: #2
Parse semi-structured reasoning chains on the 2WikiMultiHopQA Dataset
Question: Were Wessel Dammers and Robert Handcock (Rugby Union) from the same country? Reasoning Chain: Wessel Dammers » country of citizenship » #1; Robert Handcock (rugby union) » country of citizenship » #2; final answer: #answer Question: Which award the director of film The Blue Umbrella (2005 Film) won? Reasoning Chain: The Blue Umbrella » director » #1; #1 » award received » #2; final answer: #answer ... Question: Where was the place of death of Strut-Harald's father? Reasoning Chain: Strut-Harald » father » #1; #1 » place of death » #2; final answer: #answer
Convert triplets to single-hop questions
Triplet: ("basit ali", "place of birth", ?) Question: where was basit ali born? Triplet: ("instincts", "performer", ?) Question: who performed instincts? ... Triplet: ("Notre Dame", "last time won national championship in football", ?) Question: When was the last time Notre Dame won a national championship in football?
Answer triplet converted single-hop questions
Context: The Blue Umbrella (2005 film) ...It was directed by Vishal Bhardwaj and starred ldots. Question: Who is the director of The Blue Umbrella? Answer: Vishal Bhardwaj Context: Marie Louise Coidavid(1778 - 2013 March 11, 1851), was the Queen of the Kingdom of Haiti 1811 ̆2013 20 as the spouse of Henri I of Haiti. Question: Who is the director of The Blue Umbrella? Answer: unknown ... Context: Charles Haughey died of prostate cancer in 2006, at the age of eighty. Question: What was the cause of death for Charles Haughey? Answer: prostate cancer
Use parametric memory to fill the masks
Question: Were Wessel Dammers and Robert Handcock (Rugby Union) from the same country? Reasoning Chain: Wessel Dammers » country of citizenship » #1; Robert Handcock (rugby union) » country of citizenship » #2; final answer: #answer Filled reasoning chain: Wessel Dammers » country of citizenship » Dutch; Robert Handcock (rugby union) » country of citizenship » New Zealand; final answer: no Question: Which award the director of film The Blue Umbrella (2005 Film) won? Reasoning Chain: The Blue Umbrella » director » #1; #1 » award received » #2; final answer: #answer Filled reasoning chain: The Blue Umbrella » director » Vishal Bhardwaj; Vishal Bhardwaj » award received » National Film Award for Best Music Direction; final answer: National Film Award for Best Music Direction ... Question: Where was the place of death of Strut-Harald's father? Reasoning Chain: Strut-Harald » father » #1; #1 » place of death » #2; final answer: #answer Filled reasoning chain: Strut-Harald » father » Gorm the Old; Gorm the Old » place of death » Jelling; final answer: Jelling

Table 4: Examples of in-context learning exemplars used in our method.

Question: Were Wessel Dammers and Robert Handcock (Rugby Union) from the same country?
Answer: no

Question: Which award the director of film The Blue Umbrella (2005 Film) won?
Answer: National Film Award for Best Music Direction
...
Question: Where was the place of death of Strut-Harald's father?
Answer: Jelling

Table 5: Examples of in-context learning exemplars used in the standard few-shot prompting.

Model	MuSiQue Recall@20	2Wiki Recall@20
Single-round	0.64	0.67
Multi-round	0.77	0.70

Table 6: Performance comparison of iterative and one-time retrieval for MuSiQue and 2Wiki datasets.

Model	EM	F1
Oracle KG + Text + Model	0.89	0.91
Wikidata KG + Text + Model	0.82	0.87

Table 7: Comparison of using Wikidata and Oracle KG on 2Wiki with LLAMA2-70b as the base model.

question to perform retrieval. It then prompts LLMs with the question and the retrieved contents using a CoT prompting approach to generate answers.

ReAct ReAct prompting (Yao et al., 2023b) employs a few-shot prompting strategy to prompt the model to engage in multiple rounds of reasoning and action-taking. "Reasoning" refers to the model generating a thought process based on the input question or observations from the previous actions. An "action" refers to task-specific, which may also include utilizing APIs, such as Wikipedia search, to acquire external knowledge.

DSP DSP (Khattab et al., 2022) employs Python programs to outline the process required to answer multi-hop questions. These programs integrate the use of language models and retrievers to retrieve relevant documents and make predictions for the final answers. DSP also uses existing labels from the training set and applies semi-supervised learning techniques to bootstrap label training examples' programs, which are then used as exemplars for in-context learning.

Data Synthesis Data Synthesis (Chen et al., 2023) focuses on multi-hop question-answering tasks by establishing a framework that utilizes few-shot in-context learning to prompt LLMs to generate more than one million question-answer pairs. The synthesized data is then used to fine-tune the LLMs. During the inference stage, Data Synthesis employs a reasoning and action alternation method similar to ReAct, leveraging prompts to guide the fine-tuned LLMs in answering multi-hop questions.

G Comparison of Using Wikidata and Oracle KG

We present the performance comparison on 2Wiki between using Wikidata and Oracle KG in Table 7.

H Error Analysis

We show the types of errors annotated and corresponding examples in Table 8.

I Comparison of Single-round Retrieval and Multi-round Retrieval

When using unstructured text to fill in reasoning chains, we need to retrieve documents from the knowledge sources. There are two options for retrieval: one is to directly use multi-hop questions as the query for the retriever, fetching all relevant content in one shot and then using them to fill the reasoning chain. The other option is to iteratively use single-hop questions, converted from triplets, as the retrieval query to fetch relevant documents. We adopt the latter approach. In Table 6, we report the performance of these two retrieval methods in terms of recall@20. The results show that multi-round retrieval surpasses single-round retrieval, especially in the MuSiQue dataset. This is because the questions in the MuSiQue dataset often require bridging reasoning, where the next hop of reasoning depends on the knowledge from the previous hop, and the questions usually lack the intermediary information that acts as a bridge.

#	Type	Frequency		Question	Reasoning Chain	Prediction	Answer
		2Wiki	MuSiQue				
1	False Negative	29	11	How many natioanl female outdoor track and field championships have been won by the university where Bob Allison was educated?	Bob Allison » university where he was educated » James Madison University.; James Madison University. » number of national female outdoor track and field championships won » 1.	1	one
2	Knowledge Retrieval	7	26	Which film has the director who died earlier, It'S Never Too Late (1956 Film) or The Slaughter Rule?	It's Never Too Late (1956 film) » director » Michael McCarthy; Michael McCarthy » date of death » May 7th 1959; The Slaughter Rule » director » Alex Smith; Alex Smith » date of death » 2006	It's Never Too Late (1956 film)	The Slaughter Rule
3	Reasoning chain	1	13	What position was it that the person who said man cannot live by bread alone was the Son of God?	person who said man cannot live by bread alone » Son of God » Jesus.; Jesus. » position » king and judge of his kingdom	king and judge of his kingdom	the Arian position

Table 8: Error categories and examples on 2Wiki and MuSiQue.