Chain-of-Question: A Progressive Question Decomposition Approach for Complex Knowledge Base Question Answering

Yixing Peng[1], **Quan Wang**[2], **Licheng Zhang**[1], **Yi Liu**[3] **and Zhendong Mao**[1]*
[1]University of Science and Technology of China, Hefei, China
[2]Beijing University of Posts and Telecommunications, Beijing, China
[3]State Key Laboratory of Communication Content Cognition,
People's Daily Online, Beijing, China
xk98@mail.ustc.edu.cn

Abstract

Complex KBQA leverages the knowledge base (KB) to answer complex natural questions involving complicated semantics like multi-hop reasoning. Existing methods involve a question decomposition process, *i.e.*, breaking a complex question into several simpler sub-questions, to assist obtaining logical forms for querying the KB. However, existing question decomposition process derives all sub-questions directly according to the original question, resulting in limitations when one sub-question relies on the answer from a previous one. In this work, we propose Chain-of-Question, a progressive question decomposition approach to address complex KBQA challenges. First, inspired by chain-of-thought, we design a prompt to guide LLM to sequentially decompose multiple semantically clear sub-questions and provide corresponding reference answers, where each step of the decomposition relies on the previous results. Next, we utilize the decomposition result to select relevant patterns (relation-entity pairs) as accurate and faithful auxiliary information for the following logical form generation. Finally, we jointly perform logical form generation and answer prediction, utilizing the predicted answer to supplement non-executable logical forms. Experimental results demonstrate that our method achieves state-of-the-art performance on multiple datasets.

1 Introduction

Knowledge Base Question Answering (KBQA) which empowers machines to answer natural questions using the knowledge base (KB) is a critical task in developing high-confidence AI systems. Semantic parsing, as a prevailing approach, transforms natural questions into logical forms with robust interpretability and executes them on the KB to generate answers, achieving notable performance (Das et al., 2021; Ye et al., 2022; Hu et al., 2022). However, addressing complex KBQA, which involves answering complex questions containing complicated semantics such as multi-hop reasoning, remains a challenge. For machines, comprehending complex questions directly and obtaining accurate and complete logical forms can often be difficult, consequently leading to the non-executable problem.

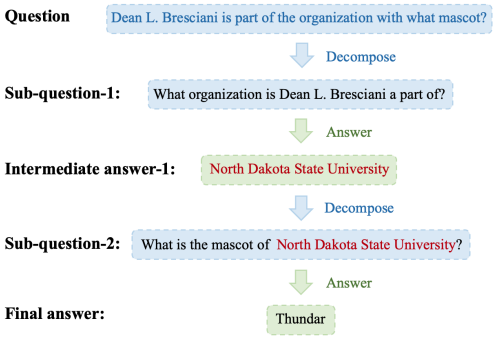


Figure 1: An example of progressive question decomposition. Red denotes the intermediate answer and its occurrence in the subsequent sub-question.

A natural approach to comprehend complex question is question decomposition, the core idea of which is to break down complicated semantics into simpler semantic components. Therefore, many studies utilize question decomposition to assist semantic parsing, which involves question decomposition process to understand the question and then integrate the decomposition result as auxiliary information for obtaining the logical form. In some prior methods (Talmor and Berant, 2018; Min et al., 2019), a question is decomposed into two sub-questions, where the relationship between these two sub-questions could be coordinate (connected using conjunctive descriptions) or inclusion (one is included in the other). However, they cannot handle complex questions that involve more semantic components and types of relationships. (Huang et al., 2023) proposed question decomposition tree to better model the semantic structure of questions, They linearize the question decomposition tree into

*Corresponding author: Zhendong Mao.

Findings of the Association for Computational Linguistics ACL 2024, pages 4763–4776

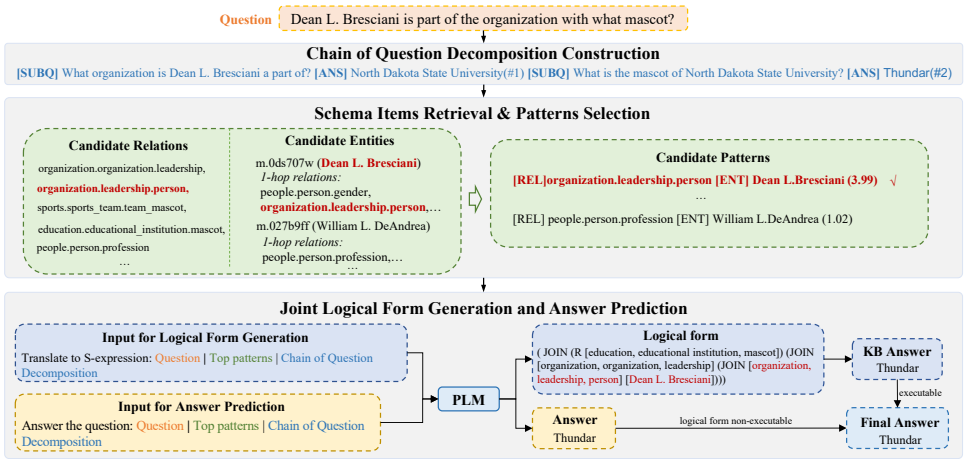


Figure 2: Overview of CoQ. Red highlights the top patterns and the corresponding schema items. The relations under candidate entities are the 1-hop relations in the KB.

a question with inserted special tokens and directly generate the linearized question decomposition tree as the decomposition result, enabling the representation of questions with multiple semantic components that have different relationships.

However, in the aforementioned methods, the question decomposition process derives all sub-questions according to the original question directly. It is not suitable in cases where the next sub-question needs to be generated based on the answer to the preceding sub-question, like multi-hop reasoning which is quite common in complex KBQA. Actually, the progressive question decomposition, i.e., providing intermediate results during the decomposition and utilizing the previous step's results to guide the subsequent decomposition, makes the modeling of each question semantic component more accurate and interpretable. Moreover, it aligns more closely with human intuition in problem-solving. For example, considering the question "Dean L. Bresciani is part of the organization with what mascot?" human reasoning naturally tends to decompose it into a simpler sub-question like "What organization is Dean L. Bresciani part of?" Following the answer to this sub-question, the subsequent sub-question "What is the mascot of North Dakota State University" can be derived based on the intermediate answer, eventually leading to the final answer. In this process, the sub-questions clearly represent the corresponding semantic components, and the intermediate answers provide supplementary information to enhance the comprehension of the complex question. While some efforts (Wolfson et al., 2020; Zhu et al., 2023) explore this decomposition for text-based QA without relying on a KB, they struggle with the absence of intermediate answers and resort to complex training procedures involving methods like Hard-EM. More importantly, lacking of grounding to KBs makes it challenging for them to achieve satisfactory performance on complex KBQA requiring factual information.

To better comprehend complex questions and assist semantic parsing, we propose **C**hain-**o**f-**Q**uestion (CoQ), a progressive question decomposition approach that sequentially models semantic components of the question to address complex KBQA problems. The overall process is depicted in Figure 2. Firstly, inspired by chain-of-thought (CoT) (Wei et al., 2022) which highlights the significance of intermediate reasoning steps on enhancing the complex reasoning capabilities of large language models (LLMs) (Brown et al., 2020; Ouyang et al., 2022), we prompt the LLM with a limited number of samples to mimic the progressive question decomposition process and construct a chain of question decomposition. During this process, the model decomposes multiple semantically clear sub-questions and provide corresponding reference answers step by step, where each step of the decomposition relies on the previous result. Secondly, to improve the accuracy and faithfulness of the logical form, we retrieve relevant candidate schema items (relations and entities) according to their similarity to the question, and combine these candidate schema items into candidate patterns (relation-entity pairs) based on their connectivity in the KB. We then select several patterns that best match the semantic of the question. Finally, we feed the chain of question decomposition and these patterns as auxiliary information into a pre-trained language model for

multi-task learning of logical form generation and answer prediction. This process enhances the accuracy of logical form generation and enables the use of the predicted answer to solve the non-executable problem.

Our work incorporates LLM to address KBQA problem, which represents a significant and promising research direction. Previous studies (Omar et al., 2023; Tan et al., 2023) have demonstrated that directly employing LLMs to answer KBQA questions may yield less factual results. Thus, integrating semantic parsing methods becomes essential. To our knowledge, we are the first to utilize LLM for question decomposition to assist semantic parsing in the KBQA context. We harness the robust reasoning capabilities of LLM along with external knowledge to enhance question comprehension. This is complemented by semantic parsing to ensure accurate and factual answers, reducing reliance solely on LLMs.

We conduct experiments on three datasets, and compare our method against a series of strong baselines. Remarkably, on the challenging CWQ dataset, known for its complex logic, our method outperform all previous approaches with an improvement of 1.8 F1 and 8.6 Hit@1. On two other popular datasets, WebQSP and GraphQ, we also achieve outstanding performance. These experimental results demonstrate the effectiveness of our method in addressing complex KBQA problems.

2 Related Work

2.1 Complex KBQA

Distinguished from text-domain question answering (Khashabi et al., 2020; Peng et al., 2023), KBQA task involves mining information from the KB. Generally, there are two main paradigms employed to address complex KBQA: information retrieval (IR)-based methods (Saxena et al., 2020; Thai et al., 2022; Dai et al., 2023; Chen et al., 2022) and semantic parsing (SP)-based methods. The former builds a question-specific graph to capture relevant information and ranks entities based on their question relevance . The latter transforms a question to a symbolic logical form and query the KB to obtain the answers. Due to the interpretability and successful transition to generative paradigm recently (Das et al., 2021; Ye et al., 2022), SP-based methods often achieve better performance. To effectively utilize schema items as auxiliary information to enhance the accuracy of logical forms, (Hu et al., 2022) proposes a multi-task approach which jointly learns entity disambiguation, relation classification and logical form generation. Our work employs a question decomposition approach to assist semantic parsing, and have achieved state-of-the-art performance on multiple datasets.

2.2 Question Decomposition for KBQA

Many studies employ question decomposition to simplify the process of comprehending complex questions. (Talmor and Berant, 2018; Min et al., 2019) as split-based methods, classify the question into a single compositional types and use pointer network to split it into two parts. To better model the complex questions, (Huang et al., 2023) inserts special tokens in the question to decompose its different semantic components as a linearized question decomposition tree and employs a generative approach to obtain this linear question decomposition tree. (Jia et al., 2018) also devises rules to decompose questions with temporal constraints. Compare to their approach of deriving all the sub-questions directly according to the original question, our proposed progressive question decomposition method utilizes the results from the previous decomposition at each step, resulting in a more accurate and interpretable process of question decomposition. Recently, there has also been some works (Gu and Su, 2022; Gu et al., 2022) dynamically decomposing logical forms. However, the decomposition at each step is based on the origin question, and lacking of explicit modeling of semantic structure may lead to less accurate logical forms. For text-based QA, (Wolfson et al., 2020) constructs a question decomposition dataset through manual annotation, yet lacking sub-answers. (Zhu et al., 2023) optimizes sub-answers training through Hard-EM. However, the absence of grounding to KBs makes it challenging for them to achieve satisfactory performance on complex KBQA that require factual information.

2.3 Reasoning with LLMs

In recent years, numerous methods have emerged that enable LLMs to exhibit powerful reasoning capabilities. Chain-of-thought (CoT) (Wei et al., 2022) emphasizes the importance of intermediate reasoning steps in augmenting the complex reasoning abilities of LLMs. (Katz et al., 2022) implicitly decomposes relationships in the question using LLM, lacking explicit decomposition. (Omar et al., 2023; Tan et al., 2023) have demonstrated that di-

rectly utilizing LLMs to answer knowledge-based complex questions often results in answers with weak factual accuracy. Our approach utilizes LLMs to progressively decompose questions, model question semantic structure, and facilitate semantic parsing to address complex KBQA. It combines the strengths of strong reasoning abilities of LLMs and the factual accuracy of semantic parsing.

3 Method

In this section, we elaborate on our method CoQ for addressing complex KBQA problems. Figure 2 provides an overview of our approach, which consists of three main components: chain of question decomposition construction, relevant schema items retrieval and patterns selection, and performing joint logical form generation and answer prediction via multitask learning. We will provide the details in the following subsections.

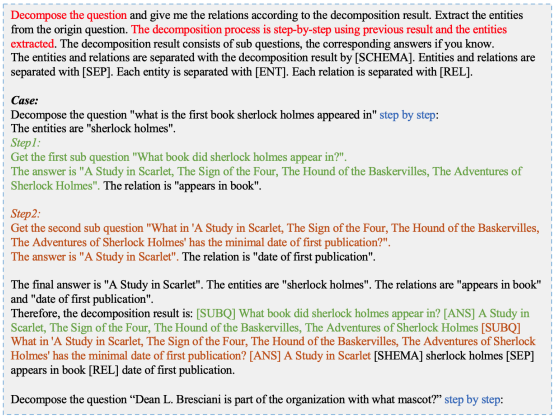
3.1 Preliminaries

KBQA can be defined as: Given a natural language question q, refer to the KB to obtain the corresponding answer a that exists in the KB. KB often represents knowledge in the form of subject-relation-object triples, denoted as (s, r, o), where s is an entity, r is a relation, and o can be either an entity or a literal (e.g., text descriptions, numeric values, etc). For complex KBQA, q contains complicated semantics and always involves multiple triples.

The SP-based method achieves KBQA by parsing q into a logical form y such as SPARQL or s-expression (Gu et al., 2021) to execute on the KB and obtain the answers. We use s-expression as the logical form following (Gu et al., 2021) as its trade-off on compositionality and readability. We use generative semantic parsing to obtain logical forms. The generation target is a normalized s-expression, with entities represented by their corresponding mentions and tokenized relations. After generation, we denormalize the s-expression, mapping mentions to entity IDs and restoring relations, and then convert the s-expression into SPARQL to execute on the KB. We unitedly name the relations, entities in an s-expression as the schema items.

3.2 Chain of Question Decomposition Construction

A progressive decomposition process can be formulated as: $(q_1, a_1) \rightarrow ... \rightarrow (q_m, a_m)$. The decomposition of each sub-question q_i takes into account the decomposition results from previous steps, and each intermediate answer a_i is derived based on its corresponding sub-question q_i. We define the chain of question decomposition $D = [\text{SUBQ}]\ q_1\ [\text{ANS}]\ a_1 ... [\text{SUBQ}]\ q_m\ [\text{ANS}]\ a_m$, which is a sequence of sub-questions and corresponding intermediate answers, separated by some special tokens.


Decompose the question and give me the relations according to the decomposition result. Extract the entities from the origin question. The decomposition process is step-by-step using previous result and the entities extracted. The decomposition result consists of sub questions, the corresponding answers if you know.
The entities and relations are separated with the decomposition result by [SCHEMA]. Entities and relations are separated with [SEP]. Each entity is separated with [ENT]. Each relation is separated with [REL].

Case:
Decompose the question "what is the first book sherlock holmes appeared in" step by step:
The entities are "sherlock holmes".
Step1:
Get the first sub question "What book did sherlock holmes appear in?".
The answer is "A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes". The relation is "appears in book".

Step2:
Get the second sub question "What in 'A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes' has the minimal date of first publication?".
The answer is "A Study in Scarlet". The relation is "date of first publication".

The final answer is "A Study in Scarlet". The entities are "sherlock holmes". The relations are "appears in book" and "date of first publication".
Therefore, the decomposition result is: [SUBQ] What book did sherlock holmes appear in? [ANS] A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes [SUBQ] What in 'A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes' has the minimal date of first publication? [ANS] A Study in Scarlet [SHEMA] sherlock holmes [SEP] appears in book [REL] date of first publication.

Decompose the question "Dean L. Bresciani is part of the organization with what mascot?" step by step:

Figure 3: Prompt for progressive question decomposition.

Inspired by CoT, we design a prompt to enable LLM to mimic the progressive decomposition process and construct the chain of question decomposition. As illustrated in Figure 3, the prompt starts with "Decompose the question" and the requirement "The decomposition process is step-by-step using previous result", which enables the model to understand the need for step-by-step decomposition of the question. As shown in the example, given a question q, we employ the prompt "Decompose the question q step by step." Subsequently, LLM will simulate the step-by-step question decomposition process. At each step, the LLM first get the sub-question, and accordingly get the answer of the sub-question. Each step of the decomposition relies on previous results. To make the decomposition result encompass schema items and facilitate retrieval in the subsequent subsection, we prompt the LLM to identify relevant relations and entities during the decomposition. Finally, the LLM organizes the sub-questions and intermediate answers to form the decomposition result, from which we extract the chain of question decomposition.

3.3 Schema Items Retrieval and Patterns Selection

We begin by initially retrieve several candidate relations and entities from KB, Subsequently, joint relation classification and entity disambiguation is

conducted for the further refinement and scoring of schema items (relations and entities). Relevant patterns selection is then based on these scores.

Retrieval of Candidate Schema Items Initially, our focus is on maximize the retrieval of schema items. The refinement process is deferred to later stages. To identify candidate entities for a given question, we follow (Hu et al., 2022) to employ the end-to-end entity linking model, ELQ (Li et al., 2020), and entity mention mappings from the FACC1 project (Evgeniy Gabrilovich and Subramanya, 2013) to obtain top-l candidate entities. Following the common practice in the literature, we adopt dense retrieval approach to obtain candidate relations. For each question, we select the top-l relations with the highest relevance scores. Please refer to Appendix A for specific retrieval details.

Relation Classification and Entity Disambiguation Ambiguity exists among the retrieved candidate entities, where multiple entities are retrieved for one mention. As a result, 1-1 mapping between mentions and entity IDs cannot be ensured during the denormalization for the generated logical forms. Therefore, entity disambiguation becomes necessary. We jointly perform two tasks: relation classification and entity disambiguation. These two tasks mutually reinforce each other to select correct schema items. As illustrated in Figure 4, both tasks share an encoder, utilizing distinct prefixes to denote different tasks which aligns with the characteristics of the T5 (Raffel et al., 2020). Therefore, we employ a shared T5 encoder to obtain representations for each input, which are then fed into separate networks to accomplish the two tasks. Relation classification aims to select the correct relations from the previously obtained candidate relations based on the question q and its corresponding chain of question decomposition D. We concatenate q, D and the relation r, apply the "Relation Classification" prefix, and input them to the T5 encoder to obtain a scalar score:

$$\begin{aligned} \mathbf{h}_{q,D,r} &= \text{Avgpool}(\text{T5Encoder}([q; D; r])); \\ s(q, D, r) &= \text{Sigmoid}(\text{Linear}(\mathbf{h}_{q,D,r})) \end{aligned} \tag{1}$$

and calculate the binary cross-entropy loss:

$$\begin{aligned} L^{\text{REL}} = -\frac{1}{l}\sum_{i=1}^{l}[u_i \cdot \log(s(q, D, r_i)) + \\ (1 - u_i) \cdot \log(1 - s(q, D, r_i))] \end{aligned} \tag{2}$$

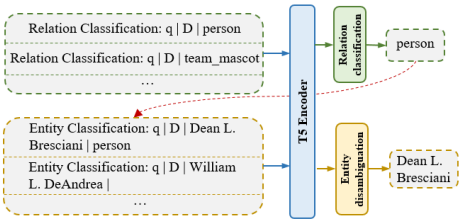


Figure 4: Joint relation classification and entity disambiguation. q is the question, D is the chain of question decomposition. Both tasks share one encoder. Red dashed line indicates subsequent task using result of previous task.

where u_i is the classification label of relation r_i, l denotes the number of candidate relations. We then set a score threshold to select refined relations.

Similar to relation classification task, we formalize the entity disambiguation task as a sequence classification task. We concatenate the label of entity e with its 1-hop neighboring relations to form a semantic enhanced context γ_e, represented as "$\text{label}_e \mid r_1 \mid r_2 \mid ...$". To maintain consistency with inference, we only concatenate the refined relations classified from the previous relation classification task. We concatenate q, D, γ_e and using "Entity Classification" prefix as input to the T5 encoder, obtain a scalar score:

$$\begin{aligned} \mathbf{h}_{q,D,e} &= \text{Avgpool}(\text{T5Encoder}([q; D; \gamma_e])); \\ s(q, D, e) &= \text{Sigmoid}(\text{Linear}(\mathbf{h}_{q,D,e})) \end{aligned} \tag{3}$$

and similar to (2), we calculate the binary cross entropy loss L^{ENT}.

The combined loss for jointly training is:

$$L = L^{\text{REL}} + L^{\text{ENT}} \tag{4}$$

Patterns Selection We define connected relation-entity pairs in the KB as patterns, which are denoted as (r, e). Patterns are built upon the foundation of schema items and can provide more accurate and faithful auxiliary information for the subsequent generation of logical forms. Firstly, we generate candidate patterns based on the previously obtained candidate relations and candidate entities. In order to represent the relevance of these patterns to different semantic components of the question in a more fine-grained manner, we decompose the chain of question decomposition D into multiple sub-chains, i.e., $D_1, D_2, ..., D_m$. Each sub-chain consists of a sub-question and its corresponding answer, that is, $D_i = [\text{SUBQ}]\, q_i\, [\text{ANS}]\, a_i$. For each pattern P, we obtain its relevance score with all

sub-chains based on the following formula:

$$s(P, D_i) = s(q, D_i, r) + s(q, D_i, e), i = 1, ..., m \tag{5}$$

where r and e are relation and entity, respectively, that form the pattern. $s(q, D_i, r)$ and $s(q, D_i, e)$ are obtained according to (1) and (3). We aggregate the relevance scores of each pattern with all sub-chains to obtain a final score, representing its alignment with the overall semantic of the question.

$$s(P) = \sum_{i=1}^{m} s(P, D_i) \tag{6}$$

Finally, we rank the candidate patterns based on their final scores and select the top-k patterns as auxiliary information for the subsequent generation of logical forms.

3.4 Joint Logical Form Generation and Answer Prediction

To enhance logical form generation and address non-executability, we propose integrating logical form generation with answer prediction. We utilize a transformer-based shared seq-to-seq model (Vaswani et al., 2017), instantiated as T5, and employ distinct prefixes to guide the model in performing separate tasks. This multi-task generation approach based on different prefixes has been proven effective in various studies (Raffel et al., 2020; Xie et al., 2022). Specifically, as shown in Figure 2, for the logical form generation task, we add a prefix to the concatenation of question q, chain of question decomposition D, and the selected top-k patterns to construct the input: "Translate to S-expression: $q \mid P_1 \mid ... \mid P_k \mid D$", which is called as γ_{LF}. We fed it to T5 and calculate the cross entropy loss using teacher forcing as follows:

$$\begin{aligned} &[\mathbf{h}_1, ..., \mathbf{h}_n] = \text{Encoder}(\gamma_{LF}); \\ &\mathbf{p}_j = \text{Decoder}(g_1, ..., g_{j-1}, \mathbf{h}_1, ...\mathbf{h}_n); \\ &L_{\text{LF}} = \frac{1}{m} \sum_{j=1}^{m} \log \mathbf{p}_{j,g_j} \end{aligned} \tag{7}$$

where n is number of tokens in γ_{LF}, $\mathbf{h}_i$ is the representation of the i-th token of the input. $g_1, ..., g_m$ are the tokens in the target logical form. $\mathbf{p}_j$ is the probability distribution over the decoding vocabulary at the j-th step.

For the answer prediction task, similar to the logical form generation, we add another prefix to construct the input: "Answer the question: $q \mid P_1 \mid ... \mid P_k \mid D$", which is called as γ_{ANS}. Similar to (7), we calculate the cross entropy loss of the answer prediction task L_{ANS} using teacher forcing.

Therefore, the final loss of joint logical form generation and answer prediction is:

$$L = L_{\text{LF}} + \alpha L_{\text{ANS}} \tag{8}$$

where α is a trade-off between the two loss terms.

During inference, we employ beam search to generate multiple logical forms, and execute them to obtain KB answers. We select the KB answer of the first executable logical form as the final answer. If none of the logical forms can be executed, we use the answer generated by the answer prediction task as the final answer. Through this approach, we can partially alleviate the non-executable issue.

4 Experiment

4.1 Setup

Datasets We conducted experiments on the ComplexWebQuestions (CWQ) (Talmor and Berant, 2018), WebQuestionsSP (WebQSP) (Yih et al., 2016) and GraphQ (Su et al., 2016) datasets, all of which are based on the Freebase. CWQ contains 34689 complex questions. These questions require up to 4-hop reasoning, making CWQ highly challenging. WebQSP is a popular dataset containing 4937 questions which require up to 2-hop reasoning. We adopt the same split for the training and test sets as done by (Ye et al., 2022) as there is no official split. GraphQ is also challenging due to the small size of training set and non-i.i.d setting.

Implementation Details We use Huggingface implementation of T5 base and BERT-base-uncased models. All experiments are conducted on a single V100 GPU. For question decomposition, we utilize gpt-3.5-turbo from OpenAI API and Llama-2 (7B) (Touvron et al., 2023). We utilize the entity mappings obtained during the entity disambiguation stage to transform the normalized s-expression back into its original form. Finally, we convert the generated s-expression into SPARQL and execute it on KB. Please refer to Appendix G for more implementation details and specific metrics.

4.2 Main Results

We present the experimental results on CWQ dataset in Table 1. The results of other comparative methods are taken from corresponding papers. Our method achieves 78.8 F1, surpassing

Methods	CWQ	
	F1	Hit@1
GPT-4	45.9	51.2
FC-KBQA (Zhang et al., 2023b)	56.4	-
Program Transfer (Cao et al., 2022)	-	58.1
CBR-KBQA (Thai et al., 2022)	70.0	70.4
QDTQA (Huang et al., 2023)	72.8	-
GMT-KBQA (Hu et al., 2022)	77.0	-
DecAF (T5-large) (Yu et al., 2022)	-	68.1
CoQ	**78.8**	**79.0**
CoQ (w Llama-2)	77.7	78.1

Table 1: QA performance on CWQ dataset.

Methods	WebQSP	
	F1	Hit@1
GPT-4	39.3	47.8
IR-based methods		
TransferNet (Shi et al., 2021)	-	71.4
NSM* (Saxena et al., 2020)	67.4	74.3
CBR-iKB (Thai et al., 2022)	-	78.3
SP-based methods		
CBR-KBQA (Das et al., 2021)	72.8	-
Rng-KBQA (Ye et al., 2022)	75.6	-
GMT-KBQA (Hu et al., 2022)	76.6	-
TIARA (Shu et al., 2022)	76.7	-
FC-KBQA (Zhang et al., 2023b)	76.9	-
Pangu (T5-base) (Gu et al., 2022)	77.3	-
DecAF (T5-large) (Yu et al., 2022)	77.1	**80.7**
CoQ	**78.1**	79.3
CoQ (w Llama2)	77.5	78.6

Table 2: QA performance on WebQSP dataset.

the prior SOTA (GMT-KBQA) (Hu et al., 2022). Our approach also achieves the best Hit@1 of 79.0, surpassing CBR-KBQA (Das et al., 2021) by a large margin of 8.6. Compared to these approaches that directly perform semantic parsing based on the original question, our method, which relies on question decomposition, offers a more effective means of modeling the structure of the question. Our approach also outperforms the previous question decomposition method QDTQA (Huang et al., 2023), demonstrating that our progressive question decomposition approach can effectively utilize previous step results to obtain more accurate decomposition results during the decomposition process. We also employ state-of-the-art LLM, GPT-4, to decompose questions and extract answers based on the prompts within our methodology, without querying KB. The performance is not satisfactory, which

Methods	GraphQ
UNDEPLAMBDA (Reddy et al., 2017)	17.7
PARA4QA (Dong et al., 2017)	20.4
SPARQA (Sun et al., 2020)	21.5
BERT+Ranking (Gu et al., 2021)	27.0
ArcaneQA (Gu and Su, 2022)	34.3
Pangu (T5-base) (Gu et al., 2022)	53.3
CoQ	**53.8**

Table 3: QA performance (F1) on GraphQ dataset.

Methods	CWQ		WebQSP	
	F1	Δ	F1	Δ
CoQ	78.8	-	78.1	-
w/o Decomp	73.0	-5.8	74.5	-3.6
w/o Interm. Answers	77.1	-1.7	76.9	-1.3
w/o Pattern	77.1	-1.7	77.3	-0.8
w/o Answer Prediction	73.9	-4.9	75.7	-2.4
w/o LF Generation	44.8	-34.0	25.5	-52.6

Table 4: Ablation study.

demonstrates the necessity of integrating LLM and other components within our approach. Additionally, we demonstrate the performance of the use of Llama2 (7B) for decomposition in our experiment remains highly competitive, further demonstrating the robustness and reproducibility of our approach.

On both WebQSP and GraphQ datasets, we achieve outstanding performance, as shown in Table 2 and Table 3. On WebQSP, our method achieves higher F1 score than all previous approaches. including the method with larger base model (e.g., DecAF using T5-large). Our Hit@1 score is also competitive compared to DecAF. This strongly indicates the effectiveness of our approach. Our method also performs better than the previous step-wise semantic parsing method Pangu (Gu et al., 2022), which confirms the importance of explicitly modeling the semantic structure of questions through decomposition. On GraphQ, CoQ achieves best performance across all the compared methods, affirming the adaptability of our approach across different datasets.

4.3 Analyses

Ablation Study To analyze the impact of individual modules on the full method, we compared the performance of the full method with various incomplete ablations, as shown in Table 4. First, we remove the chain of question decomposition and retrain the final T5 model, leading to F1 scores dropping of 5.8 and 3.66 on CWQ and WebQSP,

Methods	F1	Δ	Hit@1	Δ
CoQ	78.8	-	79.0	-
QDT	75.8	-3.0	75.9	-3.1

Table 5: Performance of different question decomposition approaches on CWQ dataset.

Metric	Relation		Entity	
	Sep	Joint	Sep	Joint
P	82.8	84.3	60.4	78.2
R	79.6	80.6	59.4	69.5
F	79.4	80.8	57.2	71.9

Table 6: Results of retrieving schema items seperately vs jointly.

respectively. This underscores the significance of our chain of question decomposition for addressing complex KBQA. The removal of intermediate answers during decomposition also results in performance degradation, demonstrating that intermediate answers contribute to a more comprehensive decomposition. We also attempted to use the placeholders described in (Zhang et al., 2023a) to replace intermediate answers during question decomposition. As shown in Table 9, the performance of using placeholder on WebQSP is inferior to the method of generating intermediate answers, further demonstrating the necessity of generating intermediate answers.

The performance drop observed upon removing patterns from the input of the final multi-task model demonstrates that selected patterns enhance the comprehension of complex questions and further improve the accuracy and faithfulness of logical form generation. We also separately removed the answer prediction task and the logical form generation task, resulting in varying degrees of performance decline. The decrease in performance upon removing the former indicates the benefit of the answer prediction task in effectively utilizing additional reference knowledge within the question decomposition chain. This complements the non-executable generated logical forms. The removal of the latter task leads to a significant performance decrease, emphasizing the impact of the logical form generation. As shown in Table 11, we also present the F1 performance of separately training two models and ensembling them and observe that the performance is lower compared to the jointly trained model. This further validates the complementary nature of both tasks for achieving optimal performance.

Methods	F1	Δ	Hit@1	Δ
CoQ	78.8	-	79.0	-
Random	76.9	-1.9	77.7	-1.3
Oracle	84.3	+5.5	85.4	+6.4

Table 7: Performance of different patterns selection approaches on CWQ dataset.

Decomposition Approach Study To further demonstrate the effectiveness of our progressive decomposition approach, we designed a variant that utilizes the linear question decomposition tree generated by QDTQA to replace our chain of question decomposition. The rest of methodology remains consistent with that of CoQ for answering the questions. The performance of this variant on CWQ dataset is presented in Table 5. It can be observed that its F1 score and Hit@1 score decrease by 3.0 and 3.1, respectively, compared to CoQ. This clearly demonstrates that the progressive decomposition process leads to improved accuracy and interpretability in questions comprehension, ultimately enhancing the accuracy of semantic parsing.

Schema Items Retrieval Study To demonstrate that joint relation classification and entity disambiguation can improve the retrieval results of schema items, in Table 6, we compare separate and joint approaches for relation classification and entity disambiguation on the CWQ dataset, It can be observed that joint training leads to a significant improvement in the final retrieval results of schema items, providing ample evidence for the mutual enhancement of the two tasks.

Table 7 illustrates the influence of different pattern selection methods on final predictions. Randomly selecting candidate schema items to compose patterns [1] is found to be detrimental to prediction performance, suggesting that incorporating noisy schema items into patterns may confuse the model. We also evaluate the model's QA performance using patterns constructed by oracle schema items, showing a significant improvement in final predictions. This underscores the importance of precise schema item retrieval for the model to accurately locate entities and relations, leading to more accurate logical form predictions and enhanced QA performance.

Efficiency Analysis We leveraged 100 randomly chosen samples to compare the average inference

[1]The random patterns here are randomly selected from the 1-hop neighbor relations of the entities.

times of our method with two semantic parsing methods, GMT-KBQA(Hu et al., 2022) and RNG-KBQA(Ye et al., 2022). GMT-KBQA (5.17s) demonstrated slightly higher inference efficiency than ours (7s), attributed to the consolidation of schema items retrieval and LF generation. When compared to the classical RNG-KBQA method (9.08s), our average inference time is shorter. This demonstrates the rationality of our approach in terms of efficiency.

5 Error Analysis

We conduct an analysis on 200 randomly select questions from the CWQ test set for which CoQ does not answer correctly (F1$<$1.0). The errors can be summarized as follows:

Schema Items Error (34.5%) The largest proportion of incorrect answers is attributed to errors in generating schema items, i.e., relations and entities, in the logical forms. Despite providing relevant patterns as auxiliary information to improve accuracy and faithfulness, the retrieval of schema items remains a significant challenge for complex KBQA.

Logical Form Structure Error (25.5%) Despite CoQ explicitly modeling the question structure through question decomposition and enhancing the accuracy of logical form structure, there are still some complex questions for which it cannot generate accurate logical form structures. The reason could be that CoQ does not enforce the decomposition results to fully match particularly these complex structures.

Answer Prediction Error (28.3%) The accuracy of the answer prediction task still has limitations, which results in some questions remaining not being correctly answered.

S-expression Conversion or Denormalization Error (11.7%) Some particularly complex SPARQL queries cannot be well converted into matching S-expressions, and errors may also occur during the denormalization phase of the generated logical forms.

6 Conclusion

This paper introduces Chain-of-Question (CoQ), a progressive question decomposition method for addressing complex KBQA. We first design a prompt that guides a LLM to progressively construct a chain of question decomposition, containing sub-questions and corresponding reference answers. Relevant schema items are then retrieved, and patterns are selected to improve the accuracy and faithfulness of logical forms. The final step involves jointly generating logical forms and predicting answers with the assistance of the chain of question decomposition and selected patterns, using predicted answers to mitigate non-executable issues. Experimental results demonstrate the method's state-of-the-art performance on multiple datasets. The impact of schema item retrieval on final predictions is also explored, revealing significant potential. Future research will focus on refining schema item accuracy and effectively modeling the semantic structure of questions.

7 Limitation

Similar to other pipeline-based KBQA methods, the primary limitation of this article lies in the lack of an end-to-end framework, which may lead to the error accumulation. Our current approach mainly involves three steps: question decomposition, schema items retrieval with pattern selection, and joint logical form generation and answer prediction. In Table 7, we experimented with using gold labels from the second step in the third step (using gold patterns for the third step), providing a quantification of the error accumulation. Despite the inherent error accumulation in pipeline methods, our work explores strategies to address compounding errors (Our method, when compared to the random patterns selection method, demonstrated an increase of 1.9 in F1 score and 1.3 in Hit@1 score in Table 7). Our method represents a preliminary exploration of utilizing LLM for progressive question decomposition to assist semantic parsing in addressing complex KBQA problems. In the future, there may be further attempts to explore more effective solutions for minimizing compounding errors.

8 Acknowledgements

This research is supported by National Science Fund for Excellent Young Scholars under Grant 62222212 and the General Program of National Natural Science Foundation of China under Grant 62376033.

References

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

Shulin Cao, Jiaxin Shi, Zijun Yao, Xin Lv, Jifan Yu, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jinghui Xiao. 2022. Program transfer for answering complex questions over knowledge bases. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8128–8140.

Ziyang Chen, Xiang Zhao, Jinzhi Liao, Xinyi Li, and Evangelos Kanoulas. 2022. Temporal knowledge graph question answering via subgraph reasoning. *Knowledge-Based Systems*, 251:109134.

Rong Dai, Xun Yang, Yan Sun, Li Shen, Xinmei Tian, Meng Wang, and Yongdong Zhang. 2023. Fedgamma: Federated learning with global sharpness-aware minimization. *IEEE Transactions on Neural Networks and Learning Systems*.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. *arXiv preprint arXiv:1708.06022*.

Michael Ringgaard Evgeniy Gabrilovich and Amarnag Subramanya. 2013. FACC1: Freebase annotation of ClueWeb corpora, Version 1 (Release date 2013-06-26, Format version 1, Correction level 0).

Yu Gu, Xiang Deng, and Yu Su. 2022. Don't generate, discriminate: A proposal for grounding language models to real-world environments. *arXiv preprint arXiv:2212.09736*.

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

Yu Gu and Yu Su. 2022. Arcaneqa: Dynamic program induction and contextualized encoding for knowledge base question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1718–1731.

Xixin Hu, Xuan Wu, Yiheng Shu, and Yuzhong Qu. 2022. Logical form generation via multi-task learning for complex question answering over knowledge bases. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1687–1696.

Xiang Huang, Sitao Cheng, Yiheng Shu, Yuheng Bao, and Yuzhong Qu. 2023. Question decomposition tree for answering complex questions over knowledge bases. *arXiv preprint arXiv:2306.07597*.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. Tequila: Temporal question answering over knowledge bases. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 1807–1810.

Uri Katz, Mor Geva, and Jonathan Berant. 2022. Inferring implicit relations in complex questions with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2548–2566.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907.

Belinda Z Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6433–6441.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109.

Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. 2023. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv preprint arXiv:2302.06466*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yixing Peng, Quan Wang, Zhendong Mao, and Yongdong Zhang. 2023. Sade: A self-adaptive expert for multi-dataset question answering. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits

of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. *arXiv preprint arXiv:1702.03196*.

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.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. Transfernet: An effective and transparent framework for multi-hop question answering over relation graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4149–4158.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. 2022. Tiara: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8108–8121.

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for qa evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 562–572.

Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. 2020. Sparqa: skeleton-based semantic parsing for complex questions over knowledge bases. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8952–8959.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions. *arXiv preprint arXiv:2303.07992*.

Dung Thai, Srinivas Ravishankar, Ibrahim Abdelaziz, Mudit Chaudhary, Nandana Mihindukulasooriya, Tahira Naseem, Rajarshi Das, Pavan Kapanipathi, Achille Fokoue, and Andrew McCallum. 2022. Cbr-ikb: A case-based reasoning approach for question answering over incomplete knowledge bases. *arXiv preprint arXiv:2204.08554*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2022. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The Eleventh International Conference on Learning Representations*.

Jiajie Zhang, Shulin Cao, Tingjia Zhang, Xin Lv, Jiaxin Shi, Qi Tian, Juanzi Li, and Lei Hou. 2023a. Reasoning over hierarchical question decomposition tree for explainable question answering. *arXiv preprint arXiv:2305.15056*.

Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. 2023b. Fc-kbqa: A fine-to-coarse composition framework for knowledge base question answering. *arXiv preprint arXiv:2306.14722*.

Wang Zhu, Jesse Thomason, and Robin Jia. 2023. Chain-of-questions training with latent answers for robust multistep question answering. *arXiv preprint arXiv:2305.14901*.

A Retrieval of Entities and Relations

To identify candidate entities for a given question, we follow (Hu et al., 2022) to employ the end-to-end entity linking model, ELQ (Li et al., 2020), and integrate entity mention mappings from the FACC1 project (Evgeniy Gabrilovich and Subramanya, 2013) to separately obtain the top-$l/2$ relevant candidate entities. These entities are then combined to form the final set of top-l candidate entities, leveraging both retrieval methods to enhance overall candidate entity coverage.

Following the common practice in the literature, we adopt a dense retrieval approach to obtain candidate relations. As shown in Figure 5, we employ a cross encoder to learn the interaction representation of the question and relations.

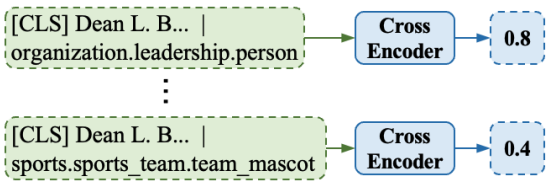


Figure 5: Relations retrieval with cross-encoder

Each pair (q, r) of question q and relation r is concatenated and used as input to BERT, and then we can obtain the relevance score of q and r as follows:

$$s(q, r) = \text{Linear}(\text{BERTCLS}[q; r]) \quad (9)$$

where Linear is a linear layer and BERTCLS is the representation of [CLS] in BERT. We then select top-l relations with the highest relevant scores for each question. We train cross-encoder using cross-entropy loss.

B Prompt for Question Decomposition

For each dataset, we randomly selected two samples from the training set to construct prompts, enabling LLM to mimic the progressive decomposition process and construct the chain of question decomposition. The complete prompts used for the CWQ and WebQSP datasets are illustrated in Figure 6 and Figure 7, respectively. Following the decomposition process of these two cases, given a question q highlighted in red, the LLM decomposes it into a sub-question at each step, and presents the corresponding intermediate answer (providing the actual answer if known, otherwise offering a numeric label prefixed with "#"). Each step of the decomposition relies on previous results.

Method	CWQ			WebQSP		
	3	5	7	2	3	4
CoQ (F1)	78.4	78.8	78.5	77.6	78.1	77.6

Table 8: Effects of different numbers of patterns on CWQ and WebQSP datasets.

Method	F1
CoQ (w Interm. Answer)	78.1
CoQ (w Placeholder)	77.1

Table 9: The F1 performance of CoQ with intermediate answers and with the use of placeholders on WebQSP.

C Effect of the Number of Patterns

We further investigated the impact of the number of top-k patterns selected for auxiliary information on the model's performance. For CWQ dataset, we tried k=3, 5, 7, while for WebQSP dataset, we tried k=2, 3, 4. The results indicated that setting k=5 for CWQ dataset and k=3 for WebQSP dataset achieved the best performance. We analyzed that if k is too small, it may result in insufficient auxiliary information retrieval, while if k is too large, it may introduce some irrelevant noise, affecting the accuracy of the results. The reason for requiring a larger value of k for achieving the best performance on CWQ dataset compared to WebQSP dataset is attributed to the fact that CWQ dataset involves more complex questions with multiple schema items, as compared to WebQSP dataset. Consequently, a larger number of relevant patterns are needed to assist in understanding these complex questions effectively.

D Effect of Hyper-parameter

To demonstrate the rationale behind our choice of the hyper-parameter α (i.e., weight for the an-

Dataset	0.1	0.2	0.3	0.4	0.6	0.8
CWQ	75.6	76.5	77.6	77.9	78.8	78.1
WebQSP	77.1	78.1	77.9	78.0	77.6	77.5

Table 10: The F1 performance with different α on CWQ and WebQSP datasets.

Decompose the question and give me the relations according to the decomposition result. Extract the entities from the origin question. The decomposition process is step-by-step using previous result and the entities extracted. The decomposition result consists of sub questions, the corresponding answers if you know. The entities and relations are separated with the decomposition result by [SCHEMA]. Entities and relations are separated with [SEP]. Each entity is separated with [ENT]. Each relation is separated with [REL].

Case1:
Decompose the question "Who was the 1996 coach of the team owned by Jerry Jones? " step by step:
The entities are "Jerry Jones", " 1996 ".
Step1:
Get the first sub question "What sports team's owners are Jerry Jones?".
The answer is "the Dallas Cowboys". The relation is "sports team's owners".
Step2:
Get the second sub question "Who was the 1996 coach of the Dallas Cowboys?".
The answer is "Barry Switzer". The relation is "coach".
The final answer is "Barry Switzer". The entities are "Jerry Joines" and "1996". The relations are "sports team's owners" and "coach".
Therefore, the decomposition result is: [SUBQ] What sports team's owners are Jerry Jones? [ANS] the Dallas Cowboys [SUBQ] Who was the 1996 coach of the Dallas Cowboys? [ANS] Barry Switzer [SHEMA] Jerry Jones [ENT] 1996 [SEP] sports team's owners [REL] coach.

Case2:
Decompose the question "Who was the governor of Arizona in 2009 that held his governmental position before 1998?" step by step:
The entities are "Arizona", "2009", "1998".
Step1:
Get the first sub question "What are the governors of Arizona in 2009?".
The answer is "#1" as you do not know. The relation is "governor".
Step2:
Get the second sub question "What in #1 held his governmental position before 1998?".
The answer is "#2" as you do not know. The relation is "government position held".
The final answer is "#2". The entities are "Arizona", "2009" and "1998". The relations are "governor" and "government position held".
Therefore, the decomposition result is: [SUBQ] What are the governors of Arizona in 2009? [ANS] #1 [SUBQ] What in #1 held his governmental position before 1998? [ANS] #2 [SCHEMA] Arizona [ENT] 2009 [ENT] 1998 [SEP] governor [REL] government position held.

Decompose the question "{q}" step by step:

Figure 6: Prompt for progressive question decomposition on CWQ dataset. For each case, green and blue denote different decomposition steps. Red represents the given original question.

Method	F1
CoQ (Jointly)	78.1
CoQ (Separately)	77.4

Table 11: The F1 performance on WebQSP of jointly or separately performing answer prediction and logical form generation.

swer prediction task), we further provide the performance of our model with different α in Table 10. The optimal alpha on the CWQ dataset (0.6) is higher than that on the WebQSP dataset (0.2). This may be because questions in the CWQ dataset are more complex, making it potentially more difficult to obtain accurate logical forms. Therefore, there is a greater need for assistance from the answer prediction task.

E Patterns Search Space

Our patterns search occurs within the sub-space of initially retrieved schema items. Assuming initial retrieval of N entities and M relations per question, the search space is N*M. In contrast, directly searching the original space involves scanning all KB schema items, significantly expanding the search space.

F Expandability Analysis

This work is a generic approach designed for complex KBQA. In theory, the proposed method does not impose restrictions on the syntax of logical forms or the representation of entities and relations. Therefore, it is applicable to new KBs and is not constrained by different datasets or languages. As a result, there are no limitations on its extension to new tasks.

G Implement Details and Metrics

The candidate number l in schema items retrieval is set as 10. For the joint relation classification and entity disambiguation model, the epoch and batch size is set to 10 and 6 on both two datasets. We select 5 top patterns for CWQ and 3 top patterns for WebQSP and GraphQ. For jointly logical form generation and answer prediction, on CWQ, the epoch is set to 15, the batch size is 16, and the trade-off is 0.6. On WebQSP and GraphQ, we set epoch as 20, batch size as 8 and the trade-off as 0.2. During the inference process, we set the beam size to 50 for both datasets. The random seed is set to 42 for all experiments.

We use F1 and Hit@1 as evaluation metrics. For obtaining Hit@1, we randomly select one answer from several obtained answers, repeating this process 5 times and taking the average as the final Hit@1 score.

Decompose the question and give me the relations according to the decomposition result. Extract the entities from the origin question. The decomposition process is step-by-step using previous result and the entities extracted. The decomposition result consists of sub questions, the corresponding answers if you know.
The entities and relations are separated with the decomposition result by [SCHEMA]. Entities and relations are separated with [SEP]. Each entity is separated with [ENT]. Each relation is separated with [REL].

Case1:
Decompose the question "what is the first book sherlock holmes appeared in" step by step:
The entities are "sherlock holmes".
Step1:
Get the first sub question "What book did sherlock holmes appear in?".
The answer is "A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes". The relation is "appears in book".
Step2:
Get the second sub question "What in 'A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes' has the minimal date of first publication?".
The answer is "A Study in Scarlet". The relation is "date of first publication".
The final answer is "A Study in Scarlet". The entities are "sherlock holmes". The relations are "appears in book" and "date of first publication".
Therefore, the decomposition result is: [SUBQ] What book did sherlock holmes appear in? [ANS] A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes [SUBQ] What in 'A Study in Scarlet, The Sign of the Four, The Hound of the Baskervilles, The Adventures of Sherlock Holmes' has the minimal date of first publication? [ANS] A Study in Scarlet [SHEMA] sherlock holmes [SEP] appears in book [REL] date of first publication.

Case2:
Decompose the question "when last did real madrid win the champions league?" step by step:
The entities are "Real Madrid C.F.", "UEFA Champions League Final".
Step1:
Get the first sub question "When are Real Madrid C.F.'s win the championship of UEFA Champions League Final?". The answer is "#1" as you do not know. The relation is "championship".
Step2:
Get the second sub question "What in #1 has max end date". The answer is "#2" as you do not know. The relation is "end date".
The final answer is "#2". The entities are "Real Madrid C.F." and "UEFA Champions League Final". The relations are "championship" and "end date".
Therefore, the decomposition result is: [SUBQ] When are Real Madrid C.F.'s win the championship of UEFA Champions League Final? [ANS] #1 [SUBQ] What in #1 has max end date [ANS] #2 [SCHEMA] Real Madrid C.F. [ENT] UEFA Champions League Final [SEP] championship [REL] end date.

Decompose the question "{q}" step by step:

Figure 7: Prompt for progressive question decomposition on WebQSP dataset. For each case, green and blue denote different decomposition steps. Red represents the given original question.

H Comparisons with Other Question Decomposition

Compared to other question decomposition methods, the novelty of our approach is manifested in the training complexity, decomposition process, and KB access methodology. For training complexity, unlike methods that require defining templates or complex training processes, we only require a small amount of demonstration, reducing manual involvement and training efforts, thereby facilitating generalization. In terms of decomposition process, unlike implicit non-progressive decomposition, our explicit progressive decomposition with reference to intermediate answers ensures more accurate and comprehensive decomposition. In KB access methodology, existing CoT-like question decomposition methods cannot access the entire KB and cannot utilize semantic parsing for accurate querying. Our approach not only leverages the reasoning capability of CoT but also assists semantic parsing in accurate querying across the entire KB. Additionally, to further demonstrate the significance of our design, Table 1 and 2 showcase that directly applying LLM for question decomposition fails to achieve satisfactory results in solving complex KBQA.

I Adaptation and Implementation for Different Use Cases

The adaptation and implementation our method for different use cases are not challenging. First, for adaptation to different datasets, the training part of our method is essentially similar to other mainstream KBQA methods (Hu et al., 2022; Shu et al., 2022). The models only require training BERT and T5, and the LLM part does not need training. The BERT model is used solely for the initial selection of schema items and can even reuse the trained model from other methods, allowing for training only the T5 model. If there is a desire to implement with other LLMs, it would only require retraining the T5 model.