GRV-KBQA: A Three-Stage Framework for Knowledge Base Question Answering with Decoupled Logical Structure, Semantic Grounding and Structure-Aware Validation

Yuhang Tian[1]*, **Pan Yang**[1]*, **Dandan Song**[1]† , **Zhijing Wu**[1], **Hao Wang**[1]
[1]School of Computer Science and Technology, Beijing Institute of Technology, China
{tianyuhang,sdd}@bit.edu.cn, zaixiatongxin@gmail.com

Abstract

Knowledge Base Question Answering (KBQA) is a fundamental task that enables natural language interaction with structured knowledge bases (KBs). Given a natural language question, KBQA aims to retrieve the answers from the KB. However, existing approaches, including retrieval-based, semantic parsing-based methods and large-language model-based methods often suffer from generating non-executable queries and inefficiencies in query execution. To address these challenges, we propose GRV-KBQA, a three-stage framework that decouples logical structure generation from semantic grounding and incorporates structure-aware validation to enhance accuracy. Unlike previous methods, GRV-KBQA explicitly enforces KB constraints to improve alignment between generated logical forms and KB structures. Experimental results on WebQSP and CWQ show that GRV-KBQA significantly improves performance over existing approaches. The ablation study conducted confirms the effectiveness of the decoupled logical form generation and validation mechanism of our framework.

1 Introduction

Knowledge bases (KBs) such as Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), and DBpedia (Auer et al., 2007), which contain vast amounts of structured data in the form of triples, are widely used due to their structured nature and the accuracy of the information they represent. Knowledge Base Questioning Answering (KBQA) is a popular application of KB, aiming to retrieve accurate answers from the KB for a given question.

Previous methods for Knowledge Base Question Answering (KBQA) can be broadly categorized into two main classes: Information Retrieval-based (IR-based) (Miller et al., 2016; Sun et al., 2019; He et al., 2021a; Zhang et al., 2022) methods and Semantic Parsing-based (SP-based) methods. IR-based methods primarily focus on retrieving relevant subgraphs from the KB and performing reasoning over these subgraphs to derive the answer. Besides, SP-based methods (Das et al., 2021; Lan and Jiang, 2020; Ye et al., 2022; Shu et al., 2022) aim to transform natural language questions into executable logical forms, such as SPARQL queries, which are then executed directly on the KB to obtain the answer. Currently, as the large language models (LLMs) have exceptional generative capabilities and powerful learning abilities,

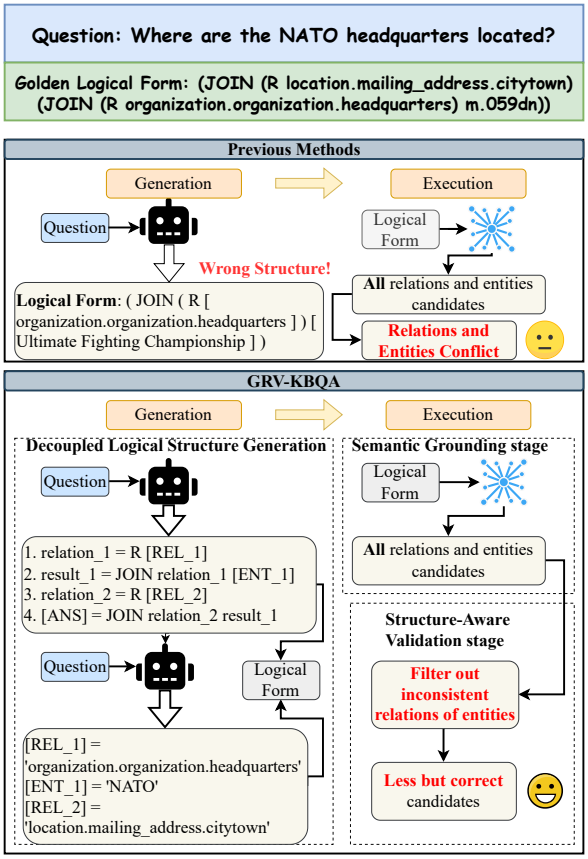

Figure 1: Comparison of GRV-KBQA with previous methods.

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2618–2632
November 4-9, 2025

several research efforts have proposed sequence-to-sequence (seq-to-seq) approaches based on LLMs, which directly generate logical forms, align them to the KB, and then execute them on the KB to retrieve answers. KB-Coder (Nie et al., 2024) and ARG-KBQA (Tian et al., 2024) have leveraged LLMs for few-shot in-context learning, where a small number of examples guides the model to generate logical forms. To fully leverage the capabilities of LLMs while avoiding the use of black-box models, ChatKBQA (Luo et al., 2024) fine-tunes open-source models to enable them to generate logical forms based on the given questions.

However, existing methods still face two key issues: **1. Challenges in Generating Correct Logical Forms:** They suffer from generating non-executable logical forms in many cases. For example, the basic logical structure of generated logical form is wrong, as shown in Figure 1. This limitation primarily arises from the one-step generation paradigm, which requires the model to simultaneously learn both the structural composition of the query and the semantic grounding of its components. This joint learning process increases the model's cognitive load, making it more prone to errors in both aspects, which can lead to errors in logical forms, including incorrect logical form structures (e.g., incorrect nesting of operations or missing/extra filtering conditions) and semantic mismatches (e.g., generating non-existent entities or relations). These errors often make the generated logical forms non-executable, preventing the framework from retrieving the correct answers. **2. Inefficiency in Aligning with the Knowledge Base:** Existing methods, when executing the generated logical form, require aligning the generated entities and relations to KB. However, previous methods generate all possible combinations of candidates without considering available pruning optimizations, as shown in Figure 1, which would result in an enormous search space, causing significant time to be wasted on invalid queries and greatly slowing down the query process.

To address these issues, we propose **GRV-KBQA**: A Three-Stage Framework for Knowledge Base Question Answering with Decoupled Logical Structure Generation, Semantic **Gr**ounding, and Structure-Aware **V**alidation. For Challenges in Generating Correct Logical Forms, we introduce **Decoupled Logical Structure Generation**, which simplifies the generation of nested logical structures by employing a two-phase compositional approach. This method leverages a subquery chain template with placeholders for entities, relations, and intermediate results, which are then specified based on the question semantics and instantiated with corresponding KB elements to produce the complete logical form.

For Inefficiency in Aligning with the Knowledge Base, we propose **Structure-Aware Validation**, which employ structure-aware validation techniques to verify the consistency and correctness of the relations and entities in the logical form, ensuring that the structure of the logic form complies with the constraints of the KB.

We conduct experiments on two widely used datasets, WebQSP (Yih et al., 2016a) and CWQ (Talmor and Berant, 2018a), and the results demonstrate that GRV-KBQA outperforms other baselines on these datasets, highlighting the effectiveness of Decoupled Logical Structure Generation and Structure-Aware Validation.

In summary, our contributions are as follows:

- We propose **Decoupled Logical Structure Generation**, a two-phase compositional approach that reduces the complexity of generating nested logical structures, leading to more accurate and executable logical forms.
- We present **Structure-Aware Validation**, which verifies the consistency and correctness of generated logical forms, ensuring they comply with the structural constraints of the KB.
- We validate the effectiveness of GRV-KBQA, through experiments on the WebQSP and CWQ datasets, demonstrating improved performance over existing methods.

2 Related Work

Traditional Knowledge Base Question Answering (KBQA) methods can generally be divided into two main categories: information retrieval (IR)-based methods and semantic parsing (SP)-based methods. In the era of LLMs, several methods have emerged that leverage LLMs to tackle the KBQA task.

IR-based methods (He et al., 2021b; Shi et al., 2021; Dong et al., 2023; Zhang et al., 2022; Jiang et al., 2023b) focus on extracting a subgraph from the KB that is specific to the given question, and then apply ranking algorithms to identify the top entities or employ text decoders to directly generate the answers.

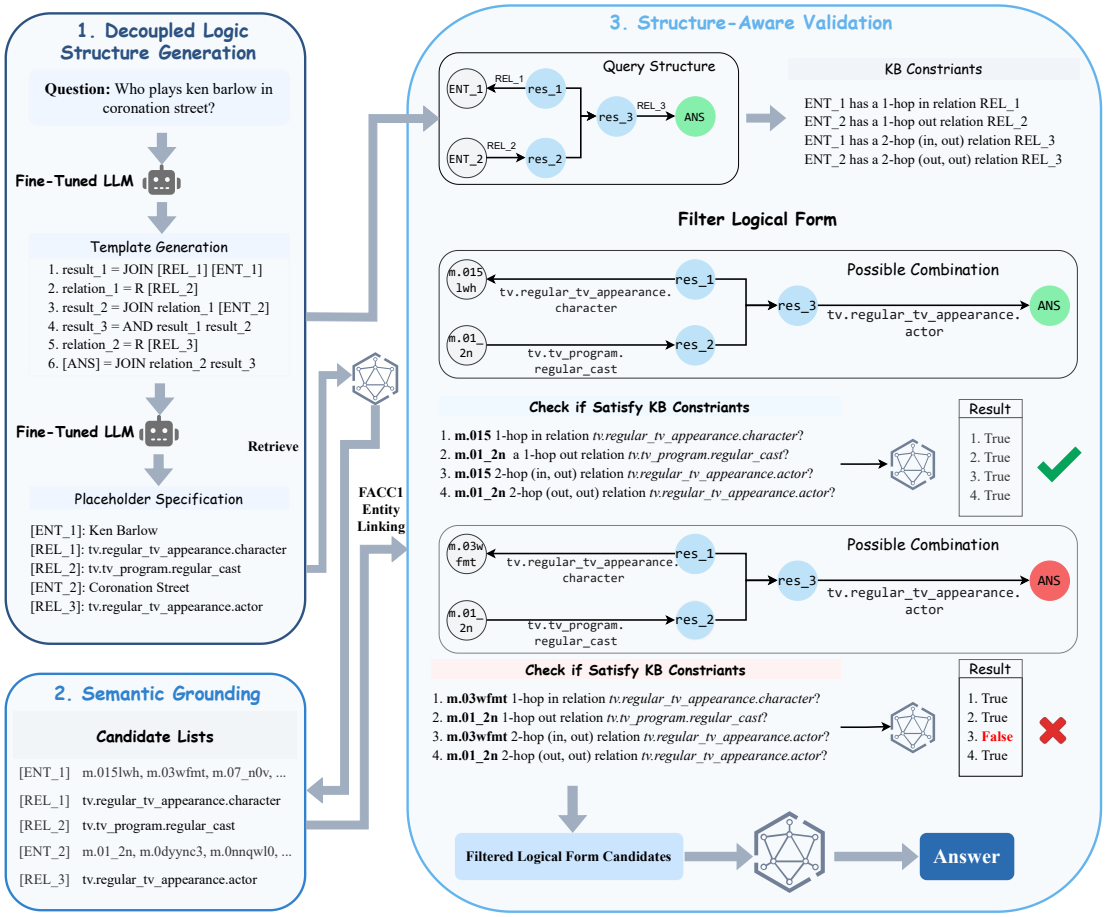

Figure 2: Overview of GRV-KBQA

SP-based methods transform natural language questions to logical forms and execute on the KB to generate answers. Yih et al. (2015) proposes a novel semantic parsing framework, where logical form generation is framed as a staged search problem, leveraging the KB to prune the search space. TIARA (Shu et al., 2022), DECAF (Yu et al., 2023), and RnG-KBQA (Ye et al., 2022) all employ sequence-to-sequence models to fully construct S-expressions and offer numerous enhancements to the semantic parsing process. FC-KBQA (Zhang et al., 2023) extracts pertinent fine-grained knowledge components from the KB and reconfigures them into intermediate-level knowledge pairs, which are subsequently used to generate the final logical forms.

LLM-based methods generate logical forms either by prompting or fine-tuning LLMs. StructGPT (Jiang et al., 2023a) proposes a framework that enhances the reasoning ability of LLMs by using specialized interfaces to collect relevant evidence. Li et al. (2024) enhances LLMs for KGQA by introducing question decomposition and atomic retrieval, using a decomposition tree to guide atomic-level KG subgraph retrieval for answering complex questions.. A series of training-free frameworks for few-shot in-context learning have been proposed by Li et al. (2023), Nie et al. (2024) and Tian et al. (2024) which employ LLMs to generate logical forms and ground them on the KB. Additionally, some methods (Sun et al., 2024; Huang et al., 2024) utilize strategies of stepwise query construction and search answers from the KB. In the realm of sequence-to-sequence approaches, ChatKBQA (Luo et al., 2024) proposes a generate-then-retrieve framework that fine-tunes open-source LLMs to generate logical forms.

In this paper, we present the GRV-KBQA framework, which separates the processes of logical form generation and semantic grounding while integrating structure-aware validation. Unlike previous approaches, our framework leverages the inherent structure of the KB to enhance the alignment of logical forms.

It is worth noting that KB-Binder also incorporates a KB-aware optimization module for execution. However, while KB-Binder (Li et al., 2023) filters candidates by retrieving a fixed two-hop subgraph, our approach considers the exact number of hops and the directionality specified by the query structure.

3 Preliminaries

3.1 Definitions

Knowledge Base (KB) A Knowledge Base (KB) is a RDF graph that stores various types of knowledge about the real world. It can be represented as a collection of triples, where each triple encodes a fact. Formally, a KB can be represented as $\mathcal{K} = \{(s_i, r_i, t_i)|i = 1, 2, \dots\}$ where s_i is an entity, t_i is either an entity or a literal and r_i is the relation between s_i and t_i. Each entity in the KB is associated with a unique entity ID, which allows precise identification of the corresponding entity node within the KB. Additionally, each entity is labeled with a natural language description, making it easier for humans to read and analyze. For example, the label of entity ID "`m.06by7`" is "`Rock Music`". Similarly, each relation is hierarchically labeled to accurately represent the complex relationships between entities, e.g., "`music.artist.genre`".

Logical Form A logical form is a highly structured intermediate representation derived from knowledge base queries. It hierarchically illustrates the process of retrieving answers from the KB. Compared to SPARQL queries, logical forms are more readable and easier to understand. In this work, the logical forms we employ primarily involve operators such as `JOIN`, `AND`, `TC`, `ARGMAX`, `ARGMIN`, and others. The `JOIN` operator is used for projection and selection. For instance, `(JOIN (R <rel>) s)` retrieves all tail entities that have the relation "`<rel>`" with the given head entity s. `(JOIN <rel> t)` retrieves all head entities that have the relation "`<rel>`" with the given tail entity t. The `AND` operator computes the intersection of two entity sets. `ARGMAX`, `ARGMIN`, `TC`, `LT`, and `GT` are conditional selection operators that filter results from a given entity set based on specified conditions.

3.2 Problem Statement

Knowledge Base Question Answering (KBQA) aims to retrieve answers from a structured KB given a natural language question. Given a natural language question q, the goal of KBQA is to generate a structured query Q that retrieves the correct answer set $\mathcal{A}$ from $\mathcal{K}$: $Q = Convert(q)$, where $Convert(\cdot)$ is the convert function. In this task, the structured query Q is often formulated in SPARQL or logical form. If Q is formulated as a logical form, it must be converted into a SPARQL query using a predefined transformation function. Finally, the SPARQL query is executed on the knowledge base $\mathcal{K}$ to obtain the final answer set $\mathcal{A}$.

4 Methodology

4.1 Overview

As shown in Figure 2, GRV-KBQA is a three-stage KBQA framework which leverages the generation capability of LLMs and consists of three parts: Decoupled Logical Structure Generation, Semantic Grounding, and Structure-Aware Validation.

Given a natural language question, GRV-KBQA first generates a subquery chain template. Based on the template, GRV-KBQA constructs a query structure graph which is a directed graph that defines the structural constraints that candidate entity IDs and relations must satisfy. Then, GRV-KBQA determines the semantic information for each placeholder based on the generated template and the given natural language question. It retrieves candidate entity IDs and relations from the knowledge base. GRV-KBQA leverages the generated relations to filter out entity IDs that do not maintain structural consistency with the query structure. It searches for the first valid candidate combination that generates an executable logical form f. The execution result of the generated logical form f would be the final answer set $\mathcal{A}$.

4.2 Decoupled Logical Structure Generation

One of the key challenges in KBQA is ensuring that logical forms are both structurally well-formed and semantically aligned with the KB. To address this, we introduce a decoupled approach that separates logical structure generation from semantic grounding. Specifically, we employ an LLM to perform two tasks. The first task is generating a subquery chain template based on the input question q. The second task aims to determine the textual content for each placeholder in the template based on the given question. Through these two tasks, we decouple the generation of the final logical form in terms of both structure and content. We fine-tune the LLM on these two tasks to enhance the performance.

Algorithm 1 Structure-Aware Validation

Input: Knowledge Base $\mathcal{K}$, Subquery Chain Template p, Threshold t_e, t_r, Top-k Num n_e, n_r
Output: The Final Answer Set $\mathcal{A}$

$\mathcal{T} \leftarrow$ *GetQueryStructure*(p)
$\mathcal{P}_e \leftarrow$ *GetEntityPlaceholders*(p)
$\mathcal{P}_r \leftarrow$ *GetRelationPlaceholders*(p)
$\mathcal{L}_e \leftarrow$ *GetCandidateListsWithThreshold*($\mathcal{P}_e, n_e, t_e$)
$\mathcal{L}_r \leftarrow$ *GetCandidateLists*($\mathcal{P}_r$)
for p_e, $L_{e_i} \in$ **zip**($\mathcal{P}_e$,$\mathcal{L}_e$) **do**
  $L'_{e_i} \leftarrow \emptyset$
  for $e \in L_{e_i}$ **do**
    $\mathcal{P}_a \leftarrow$ *GetAssociatedRelationPlaceholders*(p_e)
    if *CheckStructuralConsistency*($e, \mathcal{P}_a, \mathcal{T}, \mathcal{K}$) **then**
      L'_{e_i}.*Append*(e)
    end if
  end for
  L'_e.*Append*(L'_{e_i})
end for
$\mathcal{R} \leftarrow$ *FindNonEmptyResults*($\mathcal{L}'_e, \mathcal{L}_r, \mathcal{T}, \mathcal{K}$)
if $\mathcal{R} \neq \emptyset$ **then**
  return $\mathcal{R}$
end if
for $c \in$ *GetCombinations*($\mathcal{L}_e$) **do**
  $\mathcal{L}'_r \leftarrow \emptyset$
  for $e, p_e \in$ **zip**($c, \mathcal{P}_e$) **do**
    $\mathcal{P}_a \leftarrow$ *GetAssociatedRelationPlaceholders*(p_e)
    for $p_r \in \mathcal{P}_q$ **do**
      $r \leftarrow$*GetContent*(p_r)
      n_{hop}, direction $\leftarrow$ *GetConnection*($p_e, p_r, \mathcal{T}$)
      $\mathcal{R}_e \leftarrow$*GetRelations*(e, n_{hop}, direction, $\mathcal{K}$)
      $\mathcal{R}'_e \leftarrow$*TopkSimilarRelations*($r, \mathcal{R}'_e, t_r, n_r, \mathcal{K}$)
      $L'_{r_i} \leftarrow \mathcal{L}'_r$. *GetCandidates*($p_r$)
      if $L'_{r_i} = \emptyset$ **then**
        $\mathcal{L}'_r$.*SetCandidates*($p_r, \mathcal{R}'_e$)
      else
        $\mathcal{L}'_r$.*SetCandidates*($p_r, \mathcal{R}'_e \cap L'_{r_i}$)
      end if
    end for
  end for
  $\mathcal{R} \leftarrow$ *FindNonEmptyResults*($\mathcal{L}'_e, \mathcal{L}'_r, \mathcal{T}, \mathcal{K}$)
  if $\mathcal{R} \neq \emptyset$ **then**
    return $\mathcal{R}$
  end if
end for
return $\emptyset$

Task 1: Subquery Chain Template Generation The subquery chain template generation task aims to construct the structural template p of the final query based on the input question. The template p is organized as a sequence of subqueries and does not contain any specific semantic information. Instead, all entities and relations are replaced with placeholders such as `[ENT_i]` and `[REL_j]`.

We design this task to guide the model in learning query structural patterns that are independent of specific entities and relations. Meanwhile, the fine-grained subquery chain helps determine the overall graph structure of the query, which in turn benefits the subsequent validation process. Furthermore, the subquery chain intuitively illustrates the steps taken by the LLM to generate the final logical form, which can improve interpretability of the generation process of the final logical form.

Task 2: Placeholder Specification This task requires the model to determine the specific textual content for each placeholder in the template generated from Task 1, based on the natural language query, e.g. `"[ENT_1] = 'Ken Barlow'"`, `"[ENT_2] = 'Coronation Street'"`. The textual content is then used to retrieve entity IDs or relations from the knowledge base.

We adopt parameter-efficient fine-tuning (PEFT) methods, which includes various methods such as P-Tuning v2 (Liu et al., 2022), LoRA (Hu et al., 2021), and Freeze (Geva et al., 2021), to fine-tune the backbone LLM in our framework. The input and output formats of both tasks can be found in Appendix A. For both tasks, our optimization objective is next token prediction (NSP). The final loss is defined as:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2, \tag{1}$$

where $\mathcal{L}_1$ represents the loss for the first task, and $\mathcal{L}_2$ represents the loss for the second task.

4.3 Retrieval and Structure-Aware Validation

The LLM fine-tuned on Task 1 and Task 2 can exhibit strong semantic parsing capabilities and logical form generation ability.

Placeholder Candidate List Construction As shown in Algorithm 1, from the output of the fine-tuned LLM, we first extract the query structure $\mathcal{T}$ and the semantic information for each placeholder in the generated template. We utilize FACC1 entity linking method to retrieve entity IDs that match the textual content of entity placeholders.

Entity Filtering We discard entity IDs with occurrence probabilities below the threshold t_e and obtain a refined candidate entity ID list L_{e_i} for the entity placeholder e_i. For relation placeholders, we first assume that the model-generated relations are sufficiently accurate. Therefore, the candidate list L_{r_i} for the relation placeholder r_i contains only a single element, which corresponds to the generated textual content of the placeholder. For each candidate entity ID in the candidate list of entity placeholder e_i, we validate whether its connectivity in the knowledge base maintains structural consistency with the query structure $\mathcal{T}$. Only entity IDs that satisfy the structure constrain are retained in

the candidate list. Next, we rank the candidate entity ID list based on their occurrence probabilities in the knowledge base and select the top n_e entities as the final candidate entity set.

Logical Form Generation and Execution We enumerate all possible assignments of candidate entities and relations from their respective placeholder lists and try to construct valid logical forms. Each logical form is then converted into a SPARQL query using a fixed convert function and executed on the knowledge base. The first non-empty result is returned as the final answer set $\mathcal{A}$.

If none of the possible combinations yield a non-empty result set, we attempt to change the candidate lists for the relation placeholders. For each entity ID combination c_e, we take each entity ID in c_e as the central entity node and identify the relation placeholders associated with it and try to change the candidate lists of these placeholders. For each relation placeholder associated with the central entity, we retrieve relations that satisfy the directionality and distance constraints imposed by the query structure $\mathcal{T}$, from the knowledge base $\mathcal{K}$. We employ unsupervised retrieval methods like SimCSE (Gao et al., 2021), Conrtiver (Izacard et al., 2021) and BM25 (Robertson and Zaragoza, 2009), to compute the similarity between the retrieved relations and the model-generated relation. The retrieved relations are ranked based on their similarity to the model-generated relation. Relations with similarity scores below a predefined threshold t_r are discarded. From the remaining candidates, we select the top n_r relations as the updated candidate list for the relation placeholders. Then we try to find the first non-empty results set as our predicted answer with the updated candidate lists.

5 Experiments

In this section, we introduce the basic setup of the experiments, including the datasets, baselines, and implementation details. We also provide an analysis of the experimental results.

5.1 Setup

Datasets We evaluate the effectiveness of our approach on two widely used KBQA datasets: **WebQSP** (Yih et al., 2016a) and **ComplexWebQuestions (CWQ)** (Talmor and Berant, 2018a). Both datasets are constructed based on the Freebase (Talmor and Berant, 2018b) knowledge base and consist of a collection of knowledge-driven natural language questions. Each question is paired with a SPARQL query that retrieves the corresponding answer set from Freebase. Table 1 presents the statistics of WebQSP and CWQ.

Dataset	# Train	# Validation	# Test
WebQSP	3,098	-	1,639
CWQ	27,639	3,519	3,531

Table 1: Statistics (Number of QA Pairs) of the Datasets

Baselines We compare our method with several previous approaches. The baseline methods can be categorized into three main groups: **1. Information Retrieval-based (IR-based) Methods**, which are approaches that primarily rely on retrieval techniques to fetch relevant knowledge from the knowledge base; **2. Semantic Parsing-based (SP-based) Methods:** which are approaches that map natural language questions to structured queries, such as SPARQL, for precise answer retrieval; **3. Large Language Model-based (LLM-based) Methods:**, which are methods that leverage LLMs for semantic parsing, reasoning or knowledge retrieval. More details about the baselines can be found in Appendix B.

Metrics We adopt the same evaluation metrics as previous works (Shu et al., 2022; Yu et al., 2023; Luo et al., 2024) to evaluate the overlap between the retrieved answer set and the ground truth answer set. Specifically, we use **Accuracy (Acc)**, F_1 **score (F1)**, and **Hits@1** as our evaluation metrics.

Implementation Details While our method is generally applicable to relations of arbitrary hop distances and directions, in this paper, we restrict our consideration to relations within two hops. For WebQSP and CWQ, we use LLama2-7B and LLama2-13B (Touvron et al., 2023), respectively, as the backbone LLMs in our framework. We conduct our experiments on two NVIDIA A6000 GPUs with 48GB VRAM each. To reduce VRAM consumption during training, we apply LoRA (Hu et al., 2021). Further implementation details can be found in Appendix C.

5.2 Results and Analysis

Table 2 presents our experimental results, averaged over 3 runs. As can be observed, GRV-KBQA outperforms the existing baseline methods

on most evaluation metrics and demonstrates a significant improvement over information retrieval-based and semantic parsing-based methods across all the evaluation metrics. Compared to large language model-based approaches, GRV-KBQA also achieves an improvement of approximately 1% in F_1 and 1.5% in Accuracy. Furthermore, it is noteworthy that the improvement in Accuracy is particularly pronounced which indicates that our method is more effective at precisely identifying the correct answer.

Method	WebQSP			CWQ		
	F1	Hits@1	Acc	F1	Hits@1	Acc
IR-based Methods						
KV-Mem	34.5	46.7	-	15.7	21.1	-
Pull-Net	-	68.7	-	47.2	-	
Embed-KGQA	-	66.6	-	-	44.7	-
NSM+h	67.4	74.3	-	44.0	48.8	-
TransferNet	-	71.4	-	-	48.6	-
Subgraph Retrieval	64.1	69.5	-	47.1	50.2	-
SP-based Methods						
STAGG	71.7	-	63.9	-	-	-
UHop	68.5	-	-	29.8	-	-
Topic Units	67.9	68.2	-	36.5	39.3	-
QGG	74.0	73.0	-	40.4	44.1	-
UniKGQA	72.2	77.2	-	49.4	51.2	-
CBR-KBQA	72.8	-	69.9	70.0	70.4	67.1
RnG-KBQA	75.6	-	71.1	-	-	-
Program Transfer	76.5	74.6	-	58.7	58.1	-
TIARA	78.9	75.2	-	-	-	-
GMT-KBQA	76.6	-	73.1	77.0	-	72.2
UnifiedSKG	73.9	-	-	68.8	-	-
DecAF	78.8	82.1	-	-	70.4	-
FC-KBQA	76.9	-	-	56.4	-	-
LLM-based Methods						
StructGPT	72.6	-	-	-	-	-
Pangu	79.6	-	-	-	-	-
ToG	-	82.6	-	-	69.5	-
ChatKBQA	79.8	**83.2**	73.8	77.8	**82.7**	73.3
GRV-KBQA(ours)	**80.5**	**83.2**	**75.1**	**78.8**	82.5	**74.9**

Table 2: Comparison of Our Method with the Baseline Methods. Most of the results are directly taken from the corresponding original papers. The best result for each evaluation metric is highlighted in bold.

6 Ablation Study

In order to assess the contribution of each component in GRV-KBQA, we conduct an ablation study by systematically removing or modifying key steps and observing the impact on performance. Specifically, we design a series of experiments to evaluate the effectiveness of each component. Specifically, we conduct our ablation experiments on the WebQSP dataset.

6.1 Effectiveness of Task 1

The Subquery Chain Template Generation Task (Task 1) structures the KBQA pipeline by providing an intermediate logical form before entity and relation specification. To assess its impact, we bypass this step and allow the LLM to directly generate a complete logical form template without subqueries, as in *w/o T1* in Table 3.

Settings	F1	Hits@1	Acc
GRV-KBQA	80.5	83.2	75.1
w/o T1	80.2	83.0	74.7
w/o T2	78.4	82.6	70.5
w/o Validation	79.5	82.4	73.7

Table 3: Ablation Study Results on the WebQSP Dataset

Removing Task 1 leads to a drop in accuracy (75.1% $\rightarrow$ 74.7%) and F_1 score (0.805 $\rightarrow$ 0.802). The relatively minor impact suggests that the LLM can still infer the logical structure from the input question to some extent. However, Task 1 provides an explicit and modular breakdown of the reasoning process, which benefits interpretability and allows for easier analysis of error patterns in our framework. By decomposing complex queries into a structured form, Task 1 facilitates future refinements and debugging, making the overall KBQA pipeline more maintainable.

6.2 Effectiveness of Task 2

The Placeholder Specification task (Task 2) enables the LLM to determine the correct entities and relations for the placeholders in the structured template generated by Task 1. Instead of following this approach, *w/o T2* modifies Task 1 to generate a subquery chain that already contains specific entity and relation information, which means that the LLM must directly infer KB elements during subquery chain template generation.

As can be observed in Table 3, removing Task 2 results in a significant drop in accuracy (75.1% $\rightarrow$ 70.5%) and F_1 score (0.805 $\rightarrow$ 0.784). Without explicit placeholder resolution, the LLM must simultaneously generate the subquery chain and extract the relevant semantic information from the question, rather than resolving entity and relation placeholders in a separate step. This may lead to higher entity and relation misalignment rates. The substantial accuracy drop (4.6%) demonstrates that decoupling structure generation from semantic

grounding significantly improves retrieval correctness, thus leading to more precise answers.

6.3 Effectiveness of Structure-Aware Validation

The Structure-Aware Validation mechanism ensures that retrieved entities and relations comply with structural constraints derived from the generated template. In *w/o Validation*, this step is omitted, meaning all possible combinations are executed without structural checks.

As shown in Table 3, removing Structure-Aware Validation leads to a drop in accuracy from 75.1% to 73.7% and a 1% decrease in F_1 score. This suggests that precision declines due to an increased number of incorrect logical forms. Without structural validation, the framework is more likely to attempt to execute structurally inconsistent logical forms, which in turn results in more execution failures during query processing. This highlights the crucial role of validation in filtering structurally incompatible candidates, thus enhancing the reliability of the final logical forms.

To further illustrate the impact of Structure-Aware Validation, Table 4 presents a comparative analysis of two key factors: the average number of failed execution attempts and the exact match rate (EMR) between the executed logical forms and the ground-truth logical forms. The results show that removing validation significantly increases execution failures due to structurally inconsistent queries.

Settings	#Failed Execution (Average)	EMR
GRV-KBQA	3390119.26	66.99%
w/o Validation	1802539361.46	62.66%

Table 4: Impact of Structure-Aware Validation on Execution Failures and Logical Form Accuracy.

To verify the effectiveness of the Structure-Aware Validation step, we analyze the filtering rate of the candidate list for each placeholder during question processing and report the average across all questions. The filtering rates on the WebQSP and CWQ datasets are 90.0% and 83.8%, respectively. Additionally, the logical form match rate drops, which indicates that a structure-aware validation step can ensure greater alignment with the expected form. These findings confirm that enforcing structural constraints not only prevents invalid logical forms from being executed but also improves the overall accuracy of query generation.

Settings	F1	Hits@1	Acc
GRV-KBQA	80.5	83.2	75.1
ChatGPT(1-shot)	13.8	16.5	11.4

Table 5: Comparison Between GRV-KBQA and ChatGPT API

6.4 Do We Have to Fine-Tune an LLM?

LLMs have demonstrated impressive zero-shot and few-shot generation abilities, which also raises the question of whether fine-tuning is truly necessary for our framework.

To investigate this, we compare GRV-KBQA with a variant that replaces the fine-tuned LLM with an off-the-shelf LLM API (`gpt-4o-2024-11-20`). Instead of fine-tuning the backbone LLM on Task 1 and Task 2, this variant performs one-shot generation, where ChatGPT is prompted to directly produce the subquery chain template and the semantic information for each placeholder. By following the same validation step as our framework, this setup allows us to assess whether a non-fine-tuned LLM, guided only by one-shot prompting, can achieve comparable performance.

As the results shown in Table 5, the performance gap between GRV-KBQA (fine-tuned LLM) and ChatGPT(1-shot) is substantial, which indicates the necessity of fine-tuning for our framework. The drastic performance drop suggests that one-shot generation is highly unreliable for our framework. Without explicit fine-tuning, ChatGPT struggles to generate well-formed subquery chains, which leads to a high failure rate in execution. These results demonstrate that fine-tuning plays a crucial role in our framework. While LLMs exhibit impressive generation abilities, direct prompting alone is insufficient for generating accurate, executable logical forms.

6.5 Exact Match and Efficiency Comparison with Baseline

To further demonstrate the advantages of our approach over ChatKBQA, we additionally compare the two methods in terms of exact match rate.

Method	WebQSP	CWQ
ChatKBQA	59.40%	46.6%
GRV-KBQA	66.99%	60.0%

Table 6: Comparison with ChatKBQA on EMR

As shown in Table 6, our GRV-KBQA framework significantly outperforms ChatKBQA in terms of exact match rate (EMR) on both WebQSP and CWQ. Specifically, GRV-KBQA achieves an improvement of 7.59% points on WebQSP and 13.4% points on CWQ.

These results further demonstrate that GRV-KBQA can effectively increase the likelihood of generating executable logical forms that match the gold-standard answers. By separating query structure modeling from semantic grounding and verifying structural consistency before execution, GRV-KBQA reduces errors in both generation and alignment, leading to more reliable query execution.

We further compare the efficiency of GRV-KBQA and ChatKBQA on the WebQSP dataset in terms of average processing time per question. Our method achieves an average speed of 0.9 minutes per question, compared to approximately 1.06 minutes for ChatKBQA. This further highlights the efficiency advantage of our approach.

7 Conclusion

In this paper, we propose GRV-KBQA, a three-stage framework for knowledge base question answering (KBQA) that decouples logical structure generation from semantic grounding while incorporating structure-aware validation. Existing KBQA approaches, struggle with errors in logical forms, incorrect entity and relation alignment, or inefficiencies in candidate selection, leading to execution failures. Our framework addresses these challenges by explicitly modeling the logical form through subquery chain generation and placeholder specification, and ensuring logical form validity through structural constraints.

Experimental results on WebQSP and CWQ show that GRV-KBQA significantly improves performance over existing approaches, which demonstrates its effectiveness in generating more accurate and executable logical forms. The ablation study conducted further validates that decoupling logical structure generation from semantic grounding improves entity alignment, while structure-aware validation enhances execution reliability.

Limitations

While GRV-KBQA improves KBQA performance by decoupling logical structure generation from semantic grounding and enforcing structure-aware validation, it still has several limitations.

First, our approach relies on fine-tuning an LLM for subquery chain generation and placeholder specification. While fine-tuning enhances accuracy, it requires substantial computational resources and may limit generalization to unseen queries.

Second, the structure-aware validation stage introduces computational overhead, as verifying the structural consistency of candidate entities requires multiple KB queries. This can be inefficient on large-scale KBs, potentially limiting real-world deployment.

Finally, as illustrated in Appendix D, our method propagates errors across multiple stages, meaning that mistakes in subquery chain generation or placeholder resolution can negatively affect subsequent validation and ultimately lead to query failures.

Ethics Statement

In this work, we use publicly available datasets and do not collect any personally identifiable information. All datasets and models are utilized in full compliance with their intended purposes and respective licenses. The primary goal of this work is to address the two key issues observed in recent large language model-based KBQA approaches. We condemn any potential misuse.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFC3302100) and the National Natural Science Foundation of China (Grant No. 62476025).

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. *semantic web*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

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, Dublin, Ireland. Association for Computational Linguistics.

Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. UHop: An unrestricted-hop relation extraction framework for knowledge-based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 345–356, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guanting Dong, Rumei Li, Sirui Wang, Yupeng Zhang, Yunsen Xian, and Weiran Xu. 2023. Bridging the kb-text gap: Leveraging structured knowledge-aware pre-training for kbqa. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 3854–3859, New York, NY, USA. Association for Computing Machinery.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yu Gu, Xiang Deng, and Yu Su. 2023. Don't generate, discriminate: A proposal for grounding language models to real-world environments. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4928–4949, Toronto, Canada. Association for Computational Linguistics.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021a. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 553–561, New York, NY, USA. Association for Computing Machinery.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021b. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 553–561.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiang Huang, Sitao Cheng, Shanshan Huang, Jiayu Shen, Yong Xu, Chaoyun Zhang, and Yuzhong Qu. 2024. Queryagent: A reliable and efficient reasoning framework with environmental feedback based self-correction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5014–5035. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *CoRR*, abs/2112.09118.

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. *Preprint*, arXiv:2305.09645.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 969–974, Online. Association for Computational Linguistics.

Yunshi Lan, Shuohang Wang, and Jing Jiang. 2019. Knowledge base question answering with topic units. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5046–5052. International Joint Conferences on Artificial Intelligence Organization.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning for knowledge base question answering. *Preprint*, arXiv:2305.01750.

Yading Li, Dandan Song, Changzhi Zhou, Yuhang Tian, Hao Wang, Ziyi Yang, and Shuhao Zhang. 2024. A framework of knowledge graph-enhanced large

language model based on question decomposition and atomic retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11472–11485, Miami, Florida, USA. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Haoran Luo, Haihong E, Zichen Tang, Shiyao Peng, Yikai Guo, Wentai Zhang, Chenghao Ma, Guanting Dong, Meina Song, Wei Lin, Yifan Zhu, and Anh Tuan Luu. 2024. ChatKBQA: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2039–2056, Bangkok, Thailand. Association for Computational Linguistics.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

Zhijie Nie, Richong Zhang, Zhongyuan Wang, and Xudong Liu. 2024. Code-style in-context learning for knowledge-based question answering. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18833–18841. AAAI Press.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

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

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. 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 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Alon Talmor and Jonathan Berant. 2018a. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018b. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Yuhang Tian, Dandan Song, Zhijing Wu, Changzhi Zhou, Hao Wang, Jun Yang, Jing Xu, Ruanmin Cao, and HaoYu Wang. 2024. Augmenting reasoning capabilities of LLMs with graph structures in knowledge base question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11967–11977, Miami, Florida, USA. Association for Computational Linguistics.

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. *Preprint*, arXiv:2307.09288.

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

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1321–1331. The Association for Computer Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016a. 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, Berlin, Germany. Association for Computational Linguistics.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016b. 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, Berlin, Germany. Association for Computational Linguistics.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2023. DecAF: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The Eleventh International Conference on Learning Representations*.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784, Dublin, Ireland. Association for Computational Linguistics.

Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. 2023. FC-KBQA: A fine-to-coarse composition framework for knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1002–1017, Toronto, Canada. Association for Computational Linguistics.

A Input and Output Format

Table 8 and Table 9 present the input and output formats used for fine-tuning on the two tasks.

B Baselines

In this section, we introduce the baselines used in our experiments.

B.1 IR-Based Methods

KV-Mem (Miller et al., 2016) introduces a key-value memory network that facilitates fact encoding and reasoning for knowledge-based question answering. It enhances the model's ability to integrate relevant information and infer correct answers.

PullNet (Sun et al., 2019) integrates retrieval and reasoning in a unified framework for multi-hop KBQA, iteratively constructing a question-specific subgraph from both text corpus and knowledge bases. By leveraging graph convolutional networks (GCNs) for node expansion and answer extraction, it enables effective reasoning over large, incomplete KBs.

EmbedKGQA (Saxena et al., 2020) leverages knowledge graph embeddings to address multi-hop

	WebQSP	CWQ
Hyper-parameters for LLM Fine-Tuning		
Base Model	**LLama2-7B**, LLama3-8B, QWen2.5-7B	**LLama2-13B**, LLama3-11B, QWen2.5-14B
Batch Size	16	32
Learning Rate	1e-4, 2.5e-4, **5e-4**	2.5e-4, 5e-4, **1e-3**
Epoch	80, 100, **120**	8, 10, **12**
PEFT Method	**LoRA**, P-Tuningv2, Freeze	**LoRA**, P-Tuningv2, Freeze
LoRA Rank	8	8
LoRA Alpha	8	8
LoRA Dropout	0	0
Hyper-parameters for Inference		
Beam Size	15	8
Inference Batch Size	1	1
Hyper-parameters for Answer Retrieval		
n_e	50	50
n_r	5	5
t_e	1e-6	1e-5
t_r	0	0
Relation Retrieval	SimCSE	SimCSE

Table 7: Implementation Details of GRV-KBQA

Input	Question: Who is keyshia cole dad ?
Output	1. result_1 = JOIN [REL_1] [ENT_1] 2. relation_1 = R [REL_2] 3. result_2 = JOIN relation_1 [ENT_2] 4. [ANS] = AND result_1 result_2

Table 8: An Example of the Input and Output for Task 1

Input	Question: Who is keyshia cole dad ? 1. result_1 = JOIN [REL_1] [ENT_1] 2. relation_1 = R [REL_2] 3. result_2 = JOIN relation_1 [ENT_2] 4. [ANS] = AND result_1 result_2
Output	[ENT_1] = 'Male' [REL_1] = 'people.person.gender' [REL_2] = 'people.person.parents' [ENT_2] = 'Keyshia Cole'

Table 9: An Example of the Input and Output for Task 2

reasoning in KBQA, effectively handling knowledge graph sparsity without relying on external textual information. It also removes the constraint of selecting answers from a fixed local neighborhood, improving answer retrieval in incomplete KGs.

NSM$_{+h}$ (He et al., 2021a) employs a teacher-student framework for multi-hop KBQA, where a teacher network provides intermediate reasoning supervision to stabilize learning. By leveraging bidirectional reasoning, it enhances the reliability of intermediate entity distributions, effectively mitigating spurious reasoning issues.

TransferNet (Shi et al., 2021) introduces a unified framework for multi-hop KBQA that supports both structured (knowledge graph) and unstructured (text corpus) relations. By dynamically attending to different parts of the question and transferring entity scores through activated relations, it achieves interpretable multi-step reasoning with improved optimization stability.

Subgraph Retrieval (Zhang et al., 2022) introduces a trainable subgraph retriever (SR) that is decoupled from the reasoning process, mitigating reasoning bias caused by partial subgraphs. This approach enhances any subgraph-based KBQA model in a plug-and-play manner and improves retrieval effectiveness through weakly supervised pre-training and end-to-end fine-tuning.

B.2 SP-Based Methods

STAGG (Yih et al., 2016b) applies supervised semantic parsing to KBQA, demonstrating that learning from structured representations yields higher accuracy than relying solely on answer annotations.

UHop (Chen et al., 2019) introduces an unrestricted-hop framework for KBQA that eliminates fixed hop constraints by employing a transition-based search mechanism instead of exhaustive relation-chain search. This approach enables dynamic halting, improves efficiency, and significantly enhances performance on long relation paths.

Topic Units (Lan et al., 2019) introduces topic unit linking, expanding beyond traditional entity linking to improve KBQA answer retrieval. It employs a generation-and-scoring approach to refine topic unit selection and uses reinforcement learn-

ing for joint optimization of topic unit linking and answer ranking.

CBR-KBQA (Das et al., 2021) introduces a neuro-symbolic case-based reasoning framework for KBQA, combining a nonparametric memory to store question-logical form pairs with a retrieval-based parametric model that generates logical forms for new queries. It enables few-shot adaptability, allowing the model to handle unseen KB entities and relations without further training.

QGG (Lan and Jiang, 2020) enhances query graph generation by incorporating both constraints and multi-hop reasoning in a unified framework. By integrating constraints early in the process, it effectively prunes the search space, improving efficiency in complex KBQA tasks.

RnG-KBQA (Ye et al., 2022) introduces a Rank-and-Generate framework that combines contrastive ranking and generation to improve KBQA generalization. It first ranks candidate logical forms retrieved from the knowledge graph, then refines them with a tailored generation model, effectively addressing the coverage issue while maintaining strong generalization capability.

Program Transfer (Cao et al., 2022) enhances program induction for KBQA by leveraging annotated programs from resource-rich KBs to supervise learning on low-resource KBs. It introduces a two-stage parsing framework, combining sketch parsing for high-level program structure generation and argument parsing for retrieving KB arguments, with ontology-guided pruning to improve search efficiency.

TIARA (Shu et al., 2022) enhances PLM-based KBQA by incorporating multi-grained retrieval to provide relevant KB context and constrained decoding to improve logical form generation. This approach improves both coverage and generalization, effectively addressing semantic understanding and execution correctness challenges.

GMT-KBQA (Hu et al., 2022) enhances generation-based KBQA by integrating multi-task learning to jointly optimize entity disambiguation, relation classification, and logical form generation. It first retrieves candidate entities and relations via dense retrieval, then leverages auxiliary tasks to improve generalization and reduce errors from noisy auxiliary information.

UnifiedSKG (Xie et al., 2022) unifies structured knowledge grounding (SKG) tasks into a text-to-text format, enabling systematic research across diverse tasks, including KBQA. It benchmarks T5 with multi-task prefix-tuning, demonstrating SOTA performance on most SKG tasks and facilitating investigations into zero-shot and few-shot learning.

UniKGQA (Jiang et al., 2023b) unifies retrieval and reasoning for multi-hop KGQA by integrating them at both the model architecture and parameter learning levels. It employs a semantic matching module for question-relation alignment and a matching information propagation module to guide reasoning along KG edges, ensuring a more cohesive retrieval-reasoning process.

DecAF (Yu et al., 2023) combines logical form generation and direct answer prediction to enhance KBQA accuracy while mitigating execution failures. Unlike most prior methods, it relies on free-text retrieval instead of entity linking tools, improving adaptability across datasets.

FC-KBQA (Zhang et al., 2023) introduces a Fine-to-Coarse Composition framework that enhances both generalization and executability in KBQA. By extracting fine-grained knowledge components from KBs and reformulating them into middle-grained knowledge pairs, it enables more robust logical form generation.

B.3 LLM-Based Methods

StructGPT (Jiang et al., 2023a) improves LLMs' capability in structured data reasoning by employing an Iterative Reading-then-Reasoning (IRR) approach. It integrates specialized interfaces for structured data interaction, adopts a retrieval-augmented reasoning mechanism, and utilizes an iterative process to enhance structured data comprehension and complex question answering.

Pangu (Gu et al., 2023) introduces a grounded language understanding framework that integrates a symbolic agent with a neural language model. This approach enables incremental construction of valid plans while leveraging the language model to assess the plausibility of generated plans, improving structured reasoning and decision-making.

ToG (Think-on-Graph) (Sun et al., 2024) integrates large language models (LLMs) with knowledge graphs (KGs), treating the LLM as an agent that interactively explores entities and relations to enhance reasoning. It employs an iterative beam search mechanism to discover optimal reasoning paths, enabling deep reasoning, knowledge traceability, and training-free adaptability across various LLMs and KGs.

ChatKBQA (Luo et al., 2024) introduces a generate-then-retrieve KBQA framework, first gen-

Question	Who will play Mr. Gray in the film?
Template Generation	1. relation_1 = R [REL_1] 2. result_1 = JOIN relation_1 [ENT_1] 3. relation_2 = R [REL_2] 4. [ANS] = JOIN relation_2 result_1
Placeholder Specification	[REL_1] = 'film.film_character.portrayed_in_films' [ENT_1] = 'Mr. Gray' [REL_2] = 'film.performance.actor'
Golden Logical Form	(JOIN (R film.performance.actor) (JOIN (R film.film_character.portrayed_in_films) m.0nfjvrm))
Executed Logical Form	(JOIN (R film.performance.actor) (JOIN (R film.film_character.portrayed_in_films) m.0v4dfpr))

Table 10: A Representative Case for Error Analysis

erating logical forms using fine-tuned LLMs, then refining them via unsupervised entity and relation retrieval. This approach enhances retrieval efficiency, mitigates error propagation, and simplifies KBQA pipelines, achieving state-of-the-art performance on WebQSP and CWQ.

C Implementation Details

Table 7 presents the implementation details of our experiments.

D Error Analysis

Although our method achieves superior performance compared to existing baselines, we observe that it still struggles on certain types of questions. In this section, we present a representative example to analyze the reasons behind the failure and discuss potential directions for future improvements.

Table 10 presents this representative case. In this example, the subquery chain generated by our method is structurally correct. Although the structural reasoning is logically sound, the query ultimately returns an incorrect answer due to semantic grounding failure. We find that the issue lies in the generation of the entity placeholder `[ENT_1]`. The model generates it as "Mr. Gray", whereas the correct entity in the knowledge base is "Christian Grey" (entity ID: `m.0nfjvrm`).

This issue may stem from discrepancies between the parametric knowledge encoded in the large language model and the factual knowledge stored in the knowledge base. Our method relies on the model's parametric knowledge to implicitly complete contextual information from the input question and to guide the construction of logical forms. Given the vast pretraining corpora of large language models, they are often able to correctly infer background context and relevant entities, enabling accurate semantic interpretation in most cases.

However, in some instances, the model's internal knowledge may interfere with the semantic grounding process, which leads to inconsistencies between the generated logical form and the actual structure of the knowledge base. Although our proposed structure-aware validation mechanism can filter out some invalid entity–relation combinations from a structural perspective, it cannot fully address cases where the input query is inherently ambiguous and the placeholder generation process lacks direct alignment with the KB. In such cases, the textual content of placeholders may remain semantically vague, and the validation mechanism is limited in its ability to correct these types of errors.