ChatKBQA: A Generate-then-Retrieve Framework for Knowledge Base Question Answering with Fine-tuned Large Language Models

Anonymous ACL submission

Abstract

Knowledge Base Question Answering (KBQA) aims to answer natural language questions over large-scale knowledge bases (KBs), which can be summarized into two crucial steps: knowledge retrieval and semantic parsing. However, three core challenges remain: inefficient knowledge retrieval, mistakes of retrieval adversely impacting semantic parsing, and the complexity of previous KBQA methods. To tackle these challenges, we introduce ChatKBQA, a novel and simple generate-then-retrieve KBQA framework, which proposes first generating the logical form with fine-tuned LLMs, then retrieving and replacing entities and relations with an unsupervised retrieval method, to improve both generation and retrieval more directly. Experimental results show that ChatKBQA achieves new state-of-the-art performance on standard KBQA datasets, WebQSP, and CWQ. This work can also be regarded as a new paradigm for combining LLMs with knowledge graphs (KGs) for interpretable and knowledge-required question answering. Our code is publicly available[1].

1 Introduction

Knowledge Base Question Answering (KBQA) is a classical NLP task to answer natural language questions based on facts over a large-scale knowledge base (KB), such as Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), and DBpedia (Auer et al., 2007), which are composed of structured knowledge graphs (KGs) built from triples consisting of (head entity, relation, tail entity). Previous KBQA methods primarily addressed two core issues: knowledge retrieval (Yao et al., 2007) and semantic parsing (Berant et al., 2013). Knowledge retrieval mainly aims to locate the most relevant entities, relations, or triples according to the question from KB, to narrow the scope of consideration. Then, semantic parsing essentially converts the question from unstructured natural language into a structured logical form (such as S-expression (Gu et al., 2021)), which can then be converted into an executable graph database query (such as SPARQL (Pérez et al., 2006)) to obtain precise answers and interpretable paths, as shown in Figure 1.

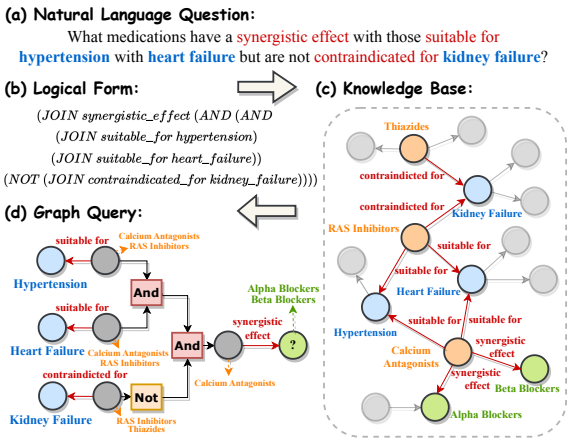


Figure 1: An example of KBQA task to answer a natural language question by converting the question to a graph query which can be executed over Knowledge Base.

Previous KBQA work (Miller et al., 2016; Sun et al., 2019; Zhang et al., 2022) proposed different knowledge retrieval methods with technologies of named entity recognition (NER) (Devlin et al., 2019), entity linking (Li et al., 2020) or subgraph retrieval (Zhang et al., 2022) to align natural language questions with structured KB. After retrieving factual triples, some studies (Yih et al., 2016; Lan and Jiang, 2020; Jiang et al., 2023b) utilized strategies of step-wise query graph generation and search answers with semantic parsing. On the other hand, other work (Ye et al., 2022; Hu et al., 2022b; Shu et al., 2022; Yu et al., 2023; Zhang et al., 2023) performed semantic parsing by using a seq2seq model like T5 (Raffel et al., 2020) to generate a

[1]Anonymous Github Code: https://anonymous.4open.science/r/ChatKBQA

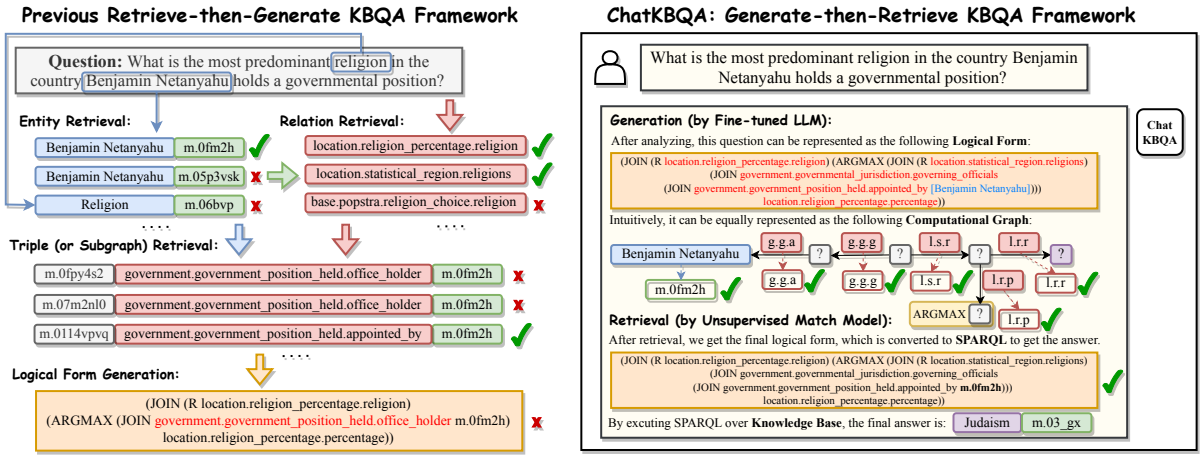


Figure 2: Comparison of the previous retrieve-then-generate KBQA framework (left) and our proposed generate-then-retrieve KBQA framework, ChatKBQA (right). Note that labels such as "g.g.a" etc. in the computational graph are acronyms for relation names such as "government.government_position_held.appointed_by".

logical form and then converted it to an SPARQL query to fetch answers when executed over KB.

Despite this, three main challenges remain, as shown on the left side of Figure 2: (i) **Low retrieval efficiency.** Traditional methods first identify the span of candidate entities and then do entity retrieval and relation retrieval. Since the structure of natural language questions differs from KB facts, most approaches require training dedicated models for extraction and linking inefficiently. (ii) **Incorrect retrieval results will mislead semantic parsing.** Previous methods have utilized retrieved triples also as input of reference to the seq2seq model along with the original question. However, since the retrieved triples are not always accurate, they adversely impact semantic parsing outcomes. Additionally, if there are numerous retrieved triples, the seq2seq model requires a much longer context length. (iii) **Multiple processing steps make KBQA a redundantly complex task.** Previous work decomposed the KBQA task into multiple sub-tasks (Hu et al., 2022b; Shu et al., 2022; Yu et al., 2023), forming a complex pipeline, which made reproduction and migration challenging. In the era when large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023; Zhao et al., 2023) are restructuring traditional NLP tasks (Chung et al., 2022; Pan et al., 2023), a more straightforward solution utilizing LLMs to reformulate the traditional KBQA paradigm is promising.

To overcome these challenges, we introduce **ChatKBQA**, a novel generate-then-retrieve KBQA framework based on open-source LLMs, such as Llama (Touvron et al., 2023), ChatGLM (Zeng et al., 2023) and Baichuan (Yang et al., 2023). As illustrated on the right side of Figure 2, ChatKBQA simplifies KBQA into two efficient phases: generating logical forms and then retrieving relevant entities and relations. **In the generation phase**, leveraging instruction tuning (Mangrulkar et al., 2022), fine-tuned LLMs exhibit high accuracy in semantic parsing of natural language questions without retrieval. The generated logical forms are not only mostly correct in skeleton (entities and relations masked) but also semantically consistent or close to the ground truth in terms of entities and relations. **In the retrieval phase**, ChatKBQA proposes an unsupervised retrieval method that employs phrase-level semantic retrieval within knowledge bases to improve generation accuracy and retrieval efficiency further. Additionally, ChatKBQA features a plug-and-play characteristic, ensuring compatibility with various LLMs and retrieval models, making it a flexible solution for KBQA tasks.

To valid the performance of our proposed framework, we conduct experiments on two standard KBQA datasets, WebQSP (Yih et al., 2016) and ComplexWebQuestions (CWQ) (Talmor and Berant, 2018), with both settings of using and not using golden entities. The experimental results demonstrate that ChatKBQA achieves a new state-of-the-art performance in the KBQA task. We also set up additional experiments to validate that our generate-then-retrieve approach improves both generation and retrieval results efficiency. Finally, we also discuss how insights from this framework lead us to envision future combinations of LLMs and KGs for knowledgable and interpretable Q&A.

2 Related Work

2.1 Knowledge Base Question Answering

Existing Knowledge Base Question Answering (KBQA) methods can be broadly categorized into Information Retrieval-based (IR-based) and Semantic Parsing-based (SP-based) methods. Recently, there have been some KBQA methods based on large language models (LLM-based) as well.

(a) IR-based KBQA methods (Miller et al., 2016; Sun et al., 2019; Saxena et al., 2020; He et al., 2021; Shi et al., 2021; Zhang et al., 2022) primarily retrieve relevant factual triples or text from KBs based on natural language questions, forming a subgraph to determine answers.

(b) SP-based KBQA methods focus on translating questions into logical forms executable against KBs, such as SPARQL, query graph, and S-expression. Some SP-based approaches (Yih et al., 2016; Chen et al., 2019; Lan et al., 2019; Bhutani et al., 2019; Lan and Jiang, 2020; Jiang et al., 2023b) utilize strategies of **step-wise** query graph generation and search for semantic parsing. Alternatively, other SP-based methods (Das et al., 2021; Ye et al., 2022; Cao et al., 2022; Shu et al., 2022; Hu et al., 2022b; Xie et al., 2022; Yu et al., 2023; Zhang et al., 2023) employ **seq2seq** models to generate S-expressions completely and offer various enhancements to the semantic parsing process.

(c) LLM-based KBQA methods (Jiang et al., 2023a; Gu et al., 2023; Sun et al., 2024) utilize the thinking capabilities of LLMs to find answers by retrieving from the graph in a step-wise manner.

In this paper, our proposed ChatKBQA is the first SP-based KBQA method using fine-tuned LLMs, which innovatively proposes a generate-then-retrieve approach to simplify KBQA method.

2.2 Large Language Models

With ChatGPT (OpenAI, 2023) displaying the prowess of decoder-only large language models (LLMs), many traditional NLP tasks are becoming simplified (Zhao et al., 2023). Subsequently, open-source LLMs like Llama (Touvron et al., 2023), ChatGLM (Zeng et al., 2023), and Baichuan (Yang et al., 2023) emerged and can be supervisedly fine-tuned (SFT) using Parameter-Efficient Fine-Tuning (PEFT) technologies (Mangrulkar et al., 2022) such as LoRA (Hu et al., 2022a), QLoRA (Dettmers et al., 2023), P-Tuning v2 (Liu et al., 2022a), and Freeze (Geva et al., 2021), enhancing the capabilities of LLMs for specific tasks.

2.3 Knowledge Retrieval for KBQA

General retrieval methods are typically divided into lexical methods, such as BM25 (Robertson and Zaragoza, 2009), and dense retrieval models, such as Dense Passage Retrieval (DPR) (Karpukhin et al., 2020), SimCSE (Gao et al., 2021), and Contriever (Izacard et al., 2022). In KBQA task, to better utilize knowledge related to the question from KB, ELQ (Li et al., 2020) and FACC1 (Evgeniy et al., 2013) are commonly used to entity retrieval.

In this paper, our ChatKBQA framework proposes a phrase-level retrieval method for entities and relations in an unsupervised manner after LLM's generation of logical form, improving both generation performance and retrieval efficiency.

3 Preliminaries

In this section, we define two basic concepts of our work: the knowledge base and the logical form, followed by the problem statement for KBQA task.

Definition 1: Knowledge Base (KB). A KB $\mathcal{K} = \{(s, r, o)|s \in \mathcal{E}, r \in \mathcal{R}, o \in \mathcal{E} \cup \mathcal{L}\}$ is an RDF graph consisting of triples (s, r, o) where s is an entity, r is a relation , and o can be an entity or a literal. Each entity $e \in \mathcal{E}$ in the entity set $\mathcal{E}$ is represented by a unique ID, e.g., e.id="m.0fm2h", which can be queried to get its label as e.label="Benjamin Netanyahu". Each relation $r \in \mathcal{R}$ in the relation set $\mathcal{R}$ has a multiple-level label, e.g. r="government.government_position_held. appointed_by".

Definition 2: Logical Form. A logical form is a structured representation of a natural language question. Taking the S-expression as an example, a logical form usually consists of projection and various operators. Projection operation represents a one-hop query of a triple (s, r, o) on s or o, where, $(?, r, o)$ is denoted as (JOIN r o), while $(s, r, ?)$ is denoted as (JOIN (R r) s). Other operators, e.g. "AND", "COUNT", and "ARGMAX", are introduced in Appendix A.

Problem Statement. For KBQA task, given a natural language question Q, and a knowledge base $\mathcal{K}$, we need to first convert Q into a logical form $F = \texttt{Sp}(Q)$, where $\texttt{Sp}(.)$ is a semantic parsing function. Then convert F to the equivalent SPARQL query $q = \texttt{Convert}(F)$, where $\texttt{Convert}(.)$ is the fixed conversion function. Finally, the final set of answers $A = \texttt{Execute}(q|\mathcal{K})$ is obtained by executing q against $\mathcal{K}$, where $\texttt{Execute}(.)$ is the query execution function.

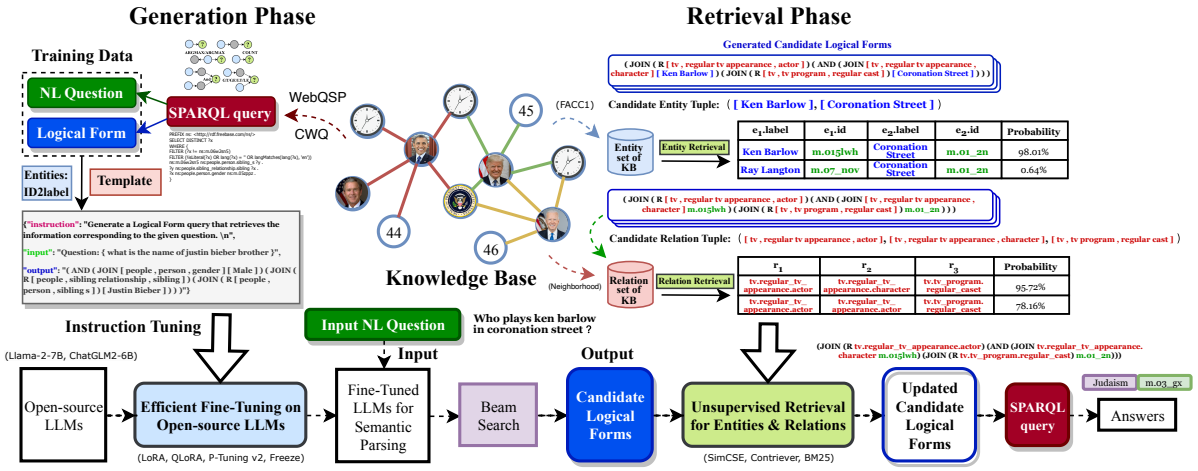


Figure 3: The overview of ChatKBQA framework for generate-then-retrieve KBQA method with fine-tuned LLMs and unsupervised retrieval for entities and relations in candidate logical forms.

4 Methodology

In this section, we first present an overview of the ChatKBQA framework as shown in Figure 3, and introduce efficient fine-tuning on large language models (LLMs), logical form generation by fine-tuned LLMs, unsupervised retrieval for entities and relations, and interpretable query execution.

4.1 Overview of ChatKBQA

ChatKBQA is a generate-then-retrieve KBQA framework with fine-tuned LLMs. First, the ChatKBQA framework needs to efficiently fine-tune an open-source LLM based on the (natural language question, logical form) pairs in the KBQA dataset by instruction tuning. The fine-tuned LLM is then used to convert the new natural language questions to according candidate logical forms by semantic parsing. Then, ChatKBQA retrieves the entities and relations in these logical forms at the phrase level, and searches for the logical forms that can be executed against KB after being converted to SPARQL. Finally, the converted SPARQL generates the final set of answers, resulting in interpretable and knowledge-required responses to natural language questions.

4.2 Efficient Fine-Tuning on LLMs

To construct the instruction fine-tuning training data, ChatKBQA first converts the SPARQL corresponding to the natural language questions of the train set in the KBQA dataset into equivalent logical forms and then replaces the entity IDs (e.g., "m.06w2sn5") in these logical forms with the corresponding entity tags (e.g., "[Justin Bieber]"), to let LLMs understand entity labels better than meaningless entity IDs. We then combine the natural language question (e.g. "What is the name of Justin Bieber's brother?") and the processed corresponding logical form (e.g. "(AND (JOIN [people, person, gender] [Male]) (JOIN (R [people, sibling relationship, sibling]) (JOIN (R [people, person, sibling s]) [Justin Bieber])))") as "input" and "output" respectively, and add "instruction" as "Generate a Logical Form query that retrieves the information corresponding to the given question." constitutes the instruction fine-tuning training data for LLMs.

ChatKBQA employs Parameter Efficient Fine-Tuning (PEFT) (Mangrulkar et al., 2022) techniques including various efficient fine-tuning methods, such as LoRA (Hu et al., 2022a), QLoRA (Dettmers et al., 2023), P-tuning v2 (Liu et al., 2022a), and Freeze (Geva et al., 2021), to minimize the cost of fine-tuning LLMs with a large number of parameters. ChatKBQA can switch between all the above fine-tuning methods as well as open-source LLMs, such as Llama-2-7B (Touvron et al., 2023), ChatGLM2-6B (Zeng et al., 2023) and Baichuan2-7B (Yang et al., 2023).

4.3 Logical Form Generation by LLMs

Through fine-tuning, the LLMs have acquired expertise in semantic parsing, enabling them to convert natural language questions into logical forms. We apply the fine-tuned LLMs to perform semantic parsing on the new questions in the test set and observe that approximately 63% of the samples match the ground truth logical forms exactly. When em-

ploying beam search, the set of candidate logical forms $\mathcal{C}$ generated by our LLMs includes approximately 74% of the instances with correct logical forms, indicating that fine-tuned LLMs possess effective learning and parsing abilities for semantic parsing tasks. In addition, by replacing the entities and relations in the candidate logical forms with "[]" (for example, "(AND (JOIN [] []) (JOIN (R []) (JOIN (R []) [])))"), more than 91% of the samples contain the candidate skeleton. Hence, the next step involves retrieving the entities and relations in the logical form with the corresponding ones from the KB to enhance performance further.

4.4 Unsupervised Retrieval for Ents and Rels

Due to the strong generative capabilities of fine-tuned LLMs for logical form skeletons, we employ an unsupervised retrieval approach during the retrieval phase. This method involves subjecting the entities and relations in the candidate logical forms to phrase-level semantic retrieval and replacement. The result is a final logical form that can be executed as a SPARQL query against the KB.

Algorithm 1 Unsupervised Retrieval

Input : Candidate logical form list generated from LLM $\mathcal{C}$, top-k threshold k_e, k_r, k_1, k_2, probability threshold t_e, t_r, t_1, t_2, the entity set of KB $\mathcal{E}$
Output : The equivalent SPARQL query q

$\mathcal{C}' \leftarrow \emptyset$ **foreach** $F \in \mathcal{C}$ **do**
  foreach $e \in F$ **do**
    $e_{list} \leftarrow \emptyset$ **foreach** $e' \in \mathcal{E}$ **do**
      $s_e \leftarrow \texttt{SimiEntities}(e, e')$
      $e_{list}.append((e', s_e))$
    $e_{list} \leftarrow \texttt{TopKwithThreshold}(e_{list}, k_e, t_e)$
    $F.attach(e_{list})$
  $F_{list} \leftarrow \texttt{PermuteByEntity}(F)$
  $\mathcal{C}'.append(\texttt{TopKwithThreshold}(F_{list}, k_1, t_1))$
$\mathcal{C}'' \leftarrow \emptyset$ **foreach** $F \in \mathcal{C}'$ **do**
  foreach $e \in F$ **do**
    $r_{list} \leftarrow \emptyset$ **foreach** $r \in \texttt{Neighborhood}(\mathcal{E}_F)$ **do**
      $s_r \leftarrow \texttt{SimiRelations}(r, r')$
      $r_{list}.append((r', s_r))$
    $r_{list} \leftarrow \texttt{TopKwithThreshold}(r_{list}, k_r, t_r)$
    $F.attach(r_{list})$
  $F_{list} \leftarrow \texttt{PermuteByRelation}(F)$
  $\mathcal{C}''.append(\texttt{TopKwithThreshold}(F_{list}, k_2, t_2))$
foreach $q \in \mathcal{C}''$ **do**
  $q = \texttt{Convert}(F)$ **if** *q is valid to execute* **then**
    return q
return $\emptyset$

Specifically, as shown in the Algorithm 1, the input is the generated candidate logical form list $\mathcal{C}$, and we traverse each of these logical forms F in order. First, we perform the entity retrieval. For each entity e in F, we compute the similarity $s_e \leftarrow \texttt{SimiEntities}(e, e')$ with the label of each entity e' in the knowledge base $\mathcal{K}$ entity set $\mathcal{E}$. We sort the retrieved entities based on the similarities, take the top k_e and greater than the threshold t_e to get the retrieval result for that entity $e_{list} \leftarrow \texttt{TopKwithThreshold}(e_{list}, k_e, t_e)$. Function `PermuteByEntity` performs permutation on the retrieved entities at each position, and we get the result F_{list} after entity retrieval. Based on probabilities in F_{list}, we take top k_1 and greater than threshold t_1 to get a new candidate logical form list $\mathcal{C}'.append(\texttt{TopKwithThreshold}(F_{list}, k_1, t_1))$.

Then, we perform the relation retrieval. Similar to entity retrieval, but different in that for each relation r in $F \in \mathcal{C}'$, we compute the similarity $s_r \leftarrow \texttt{SimiRelations}(r, r')$ with each candidate relation r' according to the neighborhood of entity set of the logical form $\mathcal{E}_F$. We also sort the retrieved relations according to the similarities, take the top k_r and greater than the threshold t_r to get the retrieval result $r_{list} \leftarrow \texttt{TopKwithThreshold}(r_{list}, k_r, t_r)$. By permuting the retrieval results of the relations at each position, we get the result F_{list} after relation retrieval and then take top k_2 and greater than the threshold t_2 to get a new list of candidate logical forms $\mathcal{C}''.append(\texttt{TopKwithThreshold}(F_{list}, k_2, t_2))$.

Given a query, unsupervised retrieval methods such as SimCSE (Gao et al., 2021), Contriever (Izacard et al., 2022), and BM25 (Robertson and Zaragoza, 2009), require no additional training to identify the top k most semantically similar candidates from the set of retrieved answers. ChatKBQA can switch between all the above unsupervised retrieval methods. We also discuss the retrieval complexity in Appendix B.

4.5 Interpretable Query Execution

After retrieval, we get a final candidate logical form list $\mathcal{C}''$, which we sequentially iterate through the logical form $F \in \mathcal{C}''$ and convert to the equivalent of the SPARQL query $q = \texttt{Convert}(F)$. When the first q that can be executed against KB $\mathcal{K}$ is found, we execute to get the final answer set $A = \texttt{Execute}(q|\mathcal{K})$. With this approach, we can also get a complete reasoning path for natural language questions based on SPARQL query with good interpretability. To summarize, ChatKBQA proposes a thought taking both the advantages of using LLMs to do natural language semantic parsing for graph query generation and calling external KBs to interpretably reason with queries.

5 Experiments

This section presents the experimental setup, results, and analysis. We answer the following research questions (RQs): **RQ1**: Does ChatKBQA outperform other KBQA methods? **RQ2**: Does the main components of ChatKBQA work? **RQ3**: Why use Generate-then-Retrieve method instead of Retrieve-then-Generate method? **RQ4**: Why use fine-tuned open-source LLMs instead of calling ChatGPT or training traditional T5 models? **RQ5**: Does Generate-then-Retrieve method improve retrieval efficiency? **RQ6**: Does ChatKBQA has plug-and-play characteristics?

5.1 Experimental Setup

Datasets. All experiments are conducted on two standard KBQA datasets: WebQuestionsSP (WebQSP) (Yih et al., 2016) containing 4,737 natural language questions with SPARQL queries and ComplexWebQuestions (CWQ) (Talmor and Berant, 2018) containing 34,689 natural language questions with SPARQL queries. Both datasets are based on Freebase (Bollacker et al., 2008) KB. More details of datasets are in Appendix C.

Baselines. We compare ChatKBQA with numerous KBQA baseline methods, including IR-based methods (Miller et al., 2016), SP-based methods (Yih et al., 2016; Das et al., 2021), and LLM-based methods (Jiang et al., 2023a) in Section 2. Details of more baselines are in Appendix D.

Evaluation Metrics. Following previous work (Shu et al., 2022; Yu et al., 2023), we use F_1 score, Hits@1, and Accuracy (Acc) to denote coverage of all the answers, single top-ranked answer, and strict exact-match accuracy, respectively.

Hyperparameters and Enviroment. We fine-tune LLMs 100 epochs on WebQSP and 10 epochs on CWQ with batch size 4 and learning rate 5e-5, detailed in Apendix E. All experiments were done on a single NVIDIA A40 GPU (48GB), with results averaged from five randomly seeded experiments.

5.2 Main Results (RQ1)

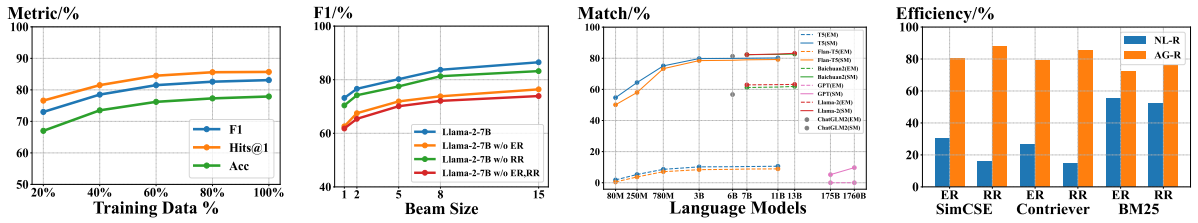
For the KBQA task, Table 1 lists the experimental results for our proposed generate-then-retrieve ChatKBQA framework, with the best setup of LoRA (Hu et al., 2022a) fine-tuning Llama-2-7B (Touvron et al., 2023) (beam size = 15) on WebQSP, Llama-2-13B (Touvron et al., 2023) (beam size = 8) on CWQ, with SimCSE (Gao et al., 2021) for unsupervised retrieval, and other baseline models. We can see that ChatKBQA has a significant improvement over all existing KBQA methods on both WebQSP and CWQ datasets. The F_1 score, Hits@1, and Acc have improved by about 4, 4, and 4 percentage points on WebQSP and about 4, 16, and 4 percentage points on CWQ, respectively, compared to the previous best results, which reflects ChatKBQA's superior KBQA capability to reach the new state-of-the-art performance.

Model	WebQSP			CWQ		
	F1	Hits@1	Acc	F1	Hits@1	Acc
IR-based KBQA Methods						
KV-Mem	34.5	46.7	-	15.7	21.1	-
PullNet	-	68.1	-	-	47.2	-
EmbedKGQA*	-	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 KBQA Methods (step-wise)						
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	-
SP-based KBQA Methods (seq2seq)						
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 KBQA Methods						
StructGPT*	72.6	-	-	-	-	-
PanGu	79.6	-	-	-	-	-
ToG*	-	82.6	-	-	69.5	-
ChatKBQA (ours)	79.8	83.2	73.8	77.8	82.7	73.3
ChatKBQA* (ours)	**83.5**	**86.4**	**77.8**	**81.3**	**86.0**	**76.8**

Table 1: KBQA comparison of ChatKBQA with other baselines on WebQSP and CWQ datasets. * denotes using oracle entity linking annotations. The results of the models are mainly taken from their original paper. For our proposed ChatKBQA framework, we display the results of the best setup on WebQSP and CWQ, respectively. The best results in each metric are in **bold**.

5.3 Ablation Study (RQ2)

To validate the effectiveness of the generation and retrieval phases of ChatKBQA, we ablate the two phases separately. For the generation phase, we use 20%, 40%, 60%, and 80% of the training data for fine-tuning versus full training set fine-tuning. For the retrieval phase, to validate entity retrieval (ER) and relation retrieval (RR) separately, we remove ER or RR from the framework and obtain three simplified variants for comparison.



(a) (b) (c) (d)

Figure 4: (a) Ablation study in ChatKBQA generation phase. (b) Ablation study in ChatKBQA retrieval phase. (c) Comparison with other language models in the generation phase. (d) Comparison of retrieval efficiency between retrieval from nature language questions (NL-R) and generated logical forms (AG-R) in the retrieval phase.

Effectiveness of LLM's Fine-tuning. As shown in Figure 4(a), the performance of KBQA gets better as the training volume increases, proving the effectiveness of fine-tuning. We also observe that the F1 score has exceeded 70% when only using 20% training data to fine-tune, which indicates that the fine-tuned LLMs are also effective at learning from a limited dataset. As shown in Figure 4(b), we also utilize beam search to improve the generation performance, detailed in Appendix F.

Effectiveness of Entity Retrieval (ER). As shown in Figure 4(b), ER improves about 15 percentage points on average over no oracle entity linking in the F1 score at different beam sizes. This is because, after LLM's fine-tuning, the generated logical forms contain entities unseen in the train set, which can be further aligned to KB after retrieving the entities from the KB entity set.

Effectiveness of Relation Retrieval (RR). As shown in Figure 4(b), RR enhances the F1 score by an average of 5% across various beam sizes in ablation experiments. Although relations are rarely directly present in natural language questions, the number of thousand-level relations in the KB is still small compared to the tens of millions of entities, and the LLM perceives relational information well during fine-tuning. Thus, RR does not improve performance as much as ER, but combined with ER, RR makes KBQA perform at its best.

5.4 Generate-then-Retrieve Or Retrieve-then-Generate (RQ3)

To verify that our proposed LLM-based Generate-then-Retrieve method is better than previous Retrieve-then-Generate methods, we add Top1, Top2, Top5, and Top10 retrieval knowledge fragments obtained in DECAF (Yu et al., 2023) to the instruction, respectively, compared with the fine-tuning of Llama-2-7B without retrieval.

Fine-tuning Settings	WebQSP			
	Max Token↓	EM↑ %	BM↑ %	SM↑ %
Llama-2-7B w/o R	**512**	**63.5**	**74.7**	**91.1**
Llama-2-7B w Top1 R	612	58.5	72.3	88.4
Llama-2-7B w Top2 R	712	59.7	73.6	89.0
Llama-2-7B w Top5 R	1012	55.6	68.3	85.3
Llama-2-7B w Top10 R	2012	53.1	67.9	84.8

Table 2: Comparison of whether or not utilizing retrieval results before fine-tuning Llama-2-7B for logical form generation in ChatKBQA.

As shown in Table 2, we find that without retrieval is better than with retrieval in the logical form generation in terms of extract match ratio (EM), match after beam search ratio (BM), and skeleton match ratio (SM), because the information obtained from retrieval will **have erroneous interfering information** and **increase Max Token of instruction**, which leads to catastrophic forgetting of the original problem for LLMs and increases the difficulty of training. At the same time, we observe that Llama-2-7B fine-tuning without retrieval achieves a BM of 74.7% and SM hits 91.1%, with good performance because of LLM's well-learned schema of entities and relations, which provides the basis for the retrieval after generation.

5.5 Comparison with ChatGPT and T5 in Generation Phase (RQ4)

To illustrate why ChatKBQA chooses to fine-tune open-source generative LLMs such as Llama-2-7B and ChatGLM2-6B, we replace the LLMs in the generation phase with ChatGPT and GPT-4 (OpenAI, 2023) with API call in a zero-shot setting, T5 (Raffel et al., 2020) and Flan-T5 (Chung et al., 2022) with seq2seq training, respectively, and observe their results in Extract Match (EM) and Skeleton Match (SM) results without beam search.

Comparison with zero-shot ChatGPT. As shown in Figure 4(c), ChatGPT and GPT-4, al-

though having large parametric quantities, cannot generate standard logical forms well because they aren't open-source to be fine-tuned. They can generate the SPARQL language, but it is challenging to build the correct query skeleton, entities, and relations because they cannot perceive the complex structure of the external KB well through designing prompts in limited context length.

Comparison with fine-tuned T5 & Flan-T5. While T5 and Flan-T5 can capture the skeletons well after fine-tuning, the EM is only about 10%, which is much worse than the 63% of Llama-2-7B, and therefore does not guarantee subsequent unsupervised entity and relation retrieval. Fine-tuned open-source LLMs such as Llama-2-7B (Touvron et al., 2023) and ChatGLM2-6B (Zeng et al., 2023) show stronger semantic parsing ability than models such as T5 and ChatGPT and can generate higher-quality logical forms in both EM & SM.

5.6 Analysis of Efficiency of Retrieval in Retrieval Phase (RQ5)

To embody the Generate-then-Retrieve method improving the efficiency of retrieval, we compare entity retrieval (ER) and retrieval (RR) after logical form generation (AG-R) with traditional retrieval from natural language questions (NL-R). We define the efficiency of retrieval as the average similarity ranging [0,1] between the text to be retrieved and the set of retrieved answers, which is scored by different retrieval models. Note that the BM25 score needs to be mapped to the similarity range of [0,1].

Efficiency gains in both ER & RR. As Figure 4(d) shows, all three retrieval methods SimCSE (Gao et al., 2021), Contriever (Izacard et al., 2022), and BM25 (Robertson and Zaragoza, 2009) consider AG-R to be more efficient than NL-R for both ER and RR. This is due to NL-R still needs to determine the boundaries of the entities or relations. However, this step has been completed in AG-R after LLM generates the logical forms.

RR has more efficiency gains than ER. Moreover, although the generated logical form has fewer kinds of relations than entities in general, the relations generally exist implicitly in natural language questions. Thus, relations are more difficult to determine the boundaries than entities in natural language questions, and the generation of logical forms with the help of fine-tuned LLMs can help us to better determine the boundaries of relations, resulting in a more significant improvement in the efficiency of RR over ER.

ChatKBQA Framework			WebQSP		
LLMs	**Tuning**	**Retrieval**	**F1**	**Hits@1**	**Acc**
Baichuan2-7B	LoRA	SimCSE	79.1	81.5	74.1
Baichuan2-13B	LoRA	SimCSE	79.4	82.1	74.4
ChatGLM2-6B	LoRA	SimCSE	79.8	82.7	74.5
Llama-2-7B	LoRA	SimCSE	80.0	82.4	75.2
Llama-2-13B	**LoRA**	**SimCSE**	**82.6**	**85.2**	**77.5**
Llama-2-13B	QLoRA	SimCSE	81.9	85.0	76.9
ChatGLM2-6B	P-Tuning v2	SimCSE	74.6	77.8	70.6
Llama-2-13B	Freeze	SimCSE	81.7	84.7	76.8
Llama-2-13B	LoRA	Contriever	81.5	83.6	76.8
Llama-2-13B	LoRA	BM25	79.8	80.5	72.7

Table 3: Plug-and-play performance comparison of ChatKBQA framework for replacing LLMs, tuning methods, and unsupervised retrieval methods, respectively, with the beam size all set as 8.

5.7 Plug-and-Play Characteristics (RQ6)

ChatKBQA is a KBQA framework based on LLMs with plug-and-play characteristics that can flexibly replace three parts: LLM, efficient tuning method, and unsupervised retrieval method. We choose Llama-2-13B (Touvron et al., 2023) for LLM, LoRA (Hu et al., 2022a) for the tuning method, and SimCSE (Gao et al., 2021) for the retrieval method as the basic variant, setting the beam size for all variants to 8 for comparison.

We replace Baichuan2-7B (Yang et al., 2023), Baichuan2-13B (Yang et al., 2023), ChatGLM2-6B (Zeng et al., 2023), Llama-2-7B (Touvron et al., 2023) **in the LLM part**, QLoRA (Dettmers et al., 2023), P-Tuning v2 (Liu et al., 2022a), Freeze (Geva et al., 2021) **in the tuning part**, and Contriever (Izacard et al., 2022), BM25 (Robertson and Zaragoza, 2009) **in the retrieval part**. Benefiting from the plug-and-play characteristics of the ChatKBQA framework, as the LLMs and the methods of tuning and retrieval are upgraded, the KBQA task will be solved better with good flexibility and extensibility. More details are in Appendix G.

6 Conclusion

In this work, we introduce ChatKBQA, a generate-then-retrieve KBQA framework that utilizes advanced fine-tuned LLMs, which overcomes traditional challenges like retrieval inefficiencies, semantic parsing errors, and complexity of KBQA methods. Experimental results on WebQSP and CWQ benchmarks show that ChatKBQA achieves a new state-of-the-art KBQA performance. Its simplicity, flexibility, and plug-and-play make it an effective approach for combining LLM and KG in interpretable knowledge-required KBQA tasks.

Limitations

In Appendix H, we provide an error analysis of ChatKBQA, revealing significant room for improvement. Furthermore, we also discuss more limitations of ChatKBQA for future directions, such as in the design of the training set, the decomposition of complex questions, support for various graph query languages, and applications in specific domains, which are detailed in Appendix I.

Ethics Statement

This paper investigates the problem of Knowledge Base Question Answering. We use large language models and retrieval methods to promote generation and retrieval performance. Therefore, we believe it does not violate any ethics.

References

Farah Atif, Ola El Khatib, and Djellel Difallah. 2023. Beamqa: Multi-hop knowledge graph question answering with sequence-to-sequence prediction and beam search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 781–790, New York, NY, USA. Association for Computing Machinery.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Nikita Bhutani, Xinyi Zheng, and H V Jagadish. 2019. Learning to answer complex questions over knowledge bases with query composition. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 739–748, New York, NY, USA. Association for Computing Machinery.

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.

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 2021. ReTraCk: A flexible and efficient framework for knowledge base question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 325–336, Online. Association for Computational Linguistics.

Yongrui Chen, Huiying Li, Guilin Qi, Tianxing Wu, and Tenggou Wang. 2023. Outlining and filling: Hierarchical query graph generation for answering complex questions over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8343–8357.

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.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. 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.

Gabrilovich Evgeniy, Ringgaard Michael, and Subramanya Amarnag. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0).

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.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, WWW '21, page 3477–3488, New York, NY, USA. Association for Computing Machinery.

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

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. 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.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Xixin Hu, Xuan Wu, Yiheng Shu, and Yuzhong Qu. 2022b. 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.

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

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

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*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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.

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, Online. Association for Computational Linguistics.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks.

Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. 2022b. Uniparser: Unified semantic parser for question answering on knowledge base and database. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8858–8869, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

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.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

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

Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. 2006. Semantics and complexity of sparql. In *The Semantic Web - ISWC 2006*, pages 30–43, Berlin, Heidelberg. Springer Berlin Heidelberg.

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. *J. Mach. Learn. Res.*, 21(1).

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.

Priyanka Sen, Armin Oliya, and Amir Saffari. 2021. Expanding end-to-end question answering on differentiable knowledge graphs with intersection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8805–8812, Online and Punta Cana, Dominican Republic. 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, Andrew Arnold, Tania Bedrax Weiss, Fernando Pereira, and William W Cohen. 2020. Faithful embeddings for knowledge base queries. In *Advances in Neural Information Processing Systems*, volume 33, pages 22505–22516. Curran Associates, Inc.

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.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. 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. 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, New Orleans, Louisiana. 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.

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.

Yuanmeng Yan, Rumei Li, Sirui Wang, Hongzhi Zhang, Zan Daoguang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Large-scale relation learning for question answering over knowledge bases with pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3653–3660, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models.

Yiyu Yao, Yi Zeng, Ning Zhong, and Xiangji Huang. 2007. Knowledge retrieval (kr). In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pages 729–735.

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, Matthew Richardson, Chris 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, 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*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130b: An open bilingual pre-trained model. 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.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Appendix

A Operators in Logical Form

Various operators include "AND" (AND E_1 E_2) to denote taking the intersection of E_1 and E_2, "COUNT" (COUNT E_1) to denote counting E_1, "ARGMAX" (ARGMAX E_1 r) to denote taking the max literal obtained after the projection of E_1 in the r relation, "ARGMIN" (ARGMIN E_1 r) to denote taking the min literal obtained after the projection of the r relation for E_1, "GT" (GT E_1 l) means to take the portion of E_1 that is greater than l, "GE" (GE E_1 l) to denote taking the part of E_1 greater than or equal to l, "LT" (LT E_1 l) to denote taking the part of E_1 less than l, "LE" (LE E_1 l) to denote taking the part of E_1 which is less than or equal to l, where E_1 or E_2 denote a sublayer logical form.

B Retrieval Complexity Analysis

During the retrieval phase, we measure the complexity of the algorithm using two indicators: the number of times vector similarity is calculated and the number of attempts to execute the logical form. Assuming the beam size in the generation phase is set to b, the size of the KB entity set is E, and the average logical form skeleton has n_e entities, the complexity of entity retrieval is O(bn_eE). For each entity's position, we select entities that rank in the top k_e in similarity and are greater than the threshold t_e for replacement. For the logical form as a whole, we select the top k_1 logical forms with a combined probability greater than the threshold t_1 as the result of entity retrieval.

In the relation retrieval phase, similarly, assuming the size of the KB relation set is R, and the average logical form skeleton has n_r entities, the complexity of entity retrieval is O(k_1n_rR). For each position's relation, we select relations that rank in the top k_r in similarity and are greater than the threshold t_r for replacement. For the logical form as a whole, based on the combination probability of the relation retrieval results, we select the top k_2 logical forms greater than the threshold t_2 as the result of relation retrieval.

Therefore, the complexity of the number of vector similarity calculations is O(bn_eE + k_1n_rR). For the number of attempts to execute the logical form, we initially attempt with the first b logical forms; if none can be executed, we proceed with entity retrieval and attempt up to k_1 times. If there is still no executable logical form, we move to relation retrieval and attempt up to k_2 times. Thus, the complexity of the number of logical form execution attempts is O(b + k_1 + k_2).

In this way, for KBQA tasks with large entity and relation sets, other parameters are much smaller than E and R, making the complexity of vector similarity calculations in the order of O(n) and the complexity of logical form execution attempts in the order of O(1), both of which are controllable.

C Dataset Statistics

As shown in Table 4, this is the statistical information of the two KBQA datasets, WebQSP and CWQ, made by the ChatKBQA experiment.

WebQSP dataset (Yih et al., 2016) is developed to evaluate the importance of gathering semantic parses compared to just answers for a set of questions. WebQSP consists of 4,737 KBQA questions, with 34 logical form skeletons and 2,461 entities involved. There are 628 relations specified within the dataset, which is divided into a training set of 3,098 questions and a test set of 1,639 questions. This dataset utilizes Freebase as its knowledge base and is tailored for developing systems that can process and answer natural language questions using structured data.

CWQ dataset (Talmor and Berant, 2018) is designed to answer complex questions requiring reasoning over multiple web snippets, which contains a large set of complex questions in natural language and is versatile in its applications. CWQ is considerably larger with 34,689 questions, underpinned by 174 logical form skeletons. It encompasses a more extensive set of entities amounting to 11,422 and includes 845 relations. The training set comprises 27,639 questions, supplemented by a validation set of 3,519 questions and a test set of 3,531 questions. CWQ also leverages Freebase as its knowledge base and is designed for complex question-answering tasks that require the interpretation and synthesis of information from various sources.

D More Baseline KBQA Methods

We compared ChatKBQA with more KBQA models as follows in order of publication.

KV-Mem (Miller et al., 2016) uses a key-value structured memory model to enhance document comprehension and question-answering by encoding facts and reasoning over them for accurate predictions.

Dataset	#Question	#Skeleton(LF)	#Entity	#Relation	#Train	#Valid	#Test	KB
STAGG	4,737	34	2,461	628	3,098	-	1,639	Freebase
CWQ	34,689	174	11,422	845	27,639	3,519	3,531	Freebase

Table 4: Dataset statistics, where the columns respectively indicate the number of all KBQA questions, logical form skeletons, participant entities, participant relations, and questions in train/valid/test sets, followed by KB's name.

Model	WebQSP			CWQ		
	F1	Hits@1	Acc	F1	Hits@1	Acc
KV-Mem	34.5	46.7	-	15.7	21.1	-
STAGG	71.7	-	63.9	-	-	-
GRAFT-Net	62.8	67.8	-	32.7	36.8	-
UHop	68.5	-	-	29.8	-	-
Topic Units	67.9	68.2	-	36.5	39.3	-
TextRay	60.3	72.2	-	33.9	40.8	-
PullNet	-	68.1	-	-	47.2	-
QGG	74.0	73.0	-	40.4	44.1	-
EmbedKGQA*	-	66.6	-	-	44.7	-
EmQL*	-	75.5	-	-	-	-
NSM+h*	67.4	74.3	-	44.0	48.8	-
GrailQA Ranking*	70.0	-	-	-	-	-
ReTraCk*	74.7	74.6	-	-	-	-
TransferNet	-	71.4	-	-	48.6	-
Relation Learning	64.5	72.9	-	-	-	-
Rigel*	-	73.3	-	-	48.7	-
CBR-KBQA	72.8	-	69.9	70.0	70.4	67.1
Subgraph Retrieval*	64.1	69.5	-	47.1	50.2	-
RnG-KBQA	75.6	-	71.1	-	-	-
Program Transfer*	76.5	74.6	-	58.7	58.1	-
TIARA*	78.9	75.2	-	-	-	-
UniK-QA	79.1	-	-	-	-	-
ArcaneQA	75.6	-	-	-	-	-
GMT-KBQA	76.6	-	73.1	77.0	-	72.2
Uni-Parser*	75.8	-	71.4	-	-	-
UnifiedSKG	73.9	-	-	68.8	-	-
UniKGQA*	72.2	77.2	-	49.4	51.2	-
DECAF	78.8	82.1	-	-	70.4	-
BeamQA*	-	73.4	-	-	-	-
HGNet*	76.6	76.9	70.7	68.5	68.9	57.8
SKP	-	79.6	-	-	-	-
StructGPT*	72.6	-	-	-	-	-
FC-KBQA	76.9	-	-	56.4	-	-
PanGu	79.6	-	-	-	-	-
ToG*	-	82.6	-	-	69.5	-
ChatKBQA (ours)	79.8	83.2	73.8	77.8	82.7	73.3
ChatKBQA* (ours)	**83.5**	**86.4**	**77.8**	**81.3**	**86.0**	**76.8**

Table 5: KBQA comparison of ChatKBQA with other baselines on WebQSP and CWQ datasets. * denotes using Oracle entity linking annotations. The results of the models are mainly taken from their original paper. For our proposed ChatKBQA framework, we display the results of the best setup on WebQSP and CWQ, respectively. The best results in each metric are in **bold**.

STAGG (Yih et al., 2016) presents a KBQA method using semantic parse labeling, showing improvements in query accuracy compared to relying solely on question-answer pairs.

GRAFT-Net (Sun et al., 2018) introduces a novel graph convolution-based neural network that enhances open-domain question answering by combining information from knowledge bases and text documents into a single model.

UHop (Chen et al., 2019) introduces a framework for unrestricted-hop relation extraction to handle queries requiring any number of relational hops in a knowledge graph, improving the capability to answer complex and indirect questions.

Topic Units (Lan et al., 2019) utilizes a wide range of knowledge base units for question answering, employing a generation-and-scoring approach and reinforcement learning to enhance the identification and ranking of relevant topic units.

TextRay (Bhutani et al., 2019) decomposes complex questions into simpler queries, processes them individually, and combines the results, using a semantic matching model.

PullNet (Sun et al., 2019) presents a method that iteratively constructs a question-specific subgraph from knowledge bases and text for effective multi-hop reasoning in open-domain question answering.

QGG (Lan and Jiang, 2020) introduces a method that enhances complex question answering by generating flexible query graphs for multi-hop questions and integrating constraints early.

EmbedKGQA (Saxena et al., 2020) introduces a method that uses knowledge graph embeddings to improve multi-hop question answering, addressing knowledge graph sparsity.

EMQL (Sun et al., 2020) presents a method that combines centroid-sketch entity set representations with neural retrieval over embedded knowledge base triples.

NSM$_{+h}$ (He et al., 2021) introduces a teacher-student framework for multi-hop KBQA, where the teacher network learns intermediate supervision signals through forward and backward reasoning to enhance the student network's reasoning capability.

GrailQA Ranking (Gu et al., 2021) presents a BERT-based KBQA model, demonstrating the critical role of pre-trained contextual embeddings, focusing on three levels of generalization - i.i.d., compositional, and zero-shot.

ReTraCk (Chen et al., 2021) introduces a neural semantic parsing framework, which combines

retriever, transducer, and checker components for efficient and effective KBQA.

TransferNet (Shi et al., 2021) introduces a model that combines a transparent, attention-based approach with the ability to handle both label and text relations in a unified framework.

Relation Learning (Yan et al., 2021) presents a method that integrates pre-trained language models with auxiliary tasks like relation extraction and reasoning.

Rigel (Sen et al., 2021) introduces a method for enhancing end-to-end question answering using differentiable knowledge graphs, and adds an intersection operation to handle multiple-entity questions more effectively.

CBR-KBQA (Das et al., 2021) employs a case-based reasoning framework that retrieves similar cases (questions and logical forms) from a nonparametric memory, then reuses and revises these cases to generate logical forms for new questions, demonstrating its capability to handle complex questions and unseen relations without retraining.

Subgraph Retrieval (Zhang et al., 2022) introduces a method devising a trainable subgraph retriever (SR) decoupled from the reasoning process, which efficiently retrieves relevant subgraphs for question answering, enhancing performance by focusing on more relevant and smaller subgraphs and combining with subgraph-oriented reasoners.

RnG-KBQA (Ye et al., 2022) introduces a framework that combines ranking and generation, using a rank-and-generate approach, where a ranker model identifies candidate logical forms and a generation model refines them.

Program Transfer (Cao et al., 2022) proposes a novel two-stage parsing framework with an efficient ontology-guided pruning strategy for complex KBQA, which involves a sketch parser that translates questions into high-level program sketches and an argument parser that fills in detailed arguments.

TIARA (Shu et al., 2022) introduces a novel method that enhances question answering over knowledge bases by using multi-grained retrieval, which improves the performance of pre-trained language models by focusing on the most relevant knowledge base contexts, including entities, logical forms, and schema items, and employs constrained decoding to control the output space, reducing generation errors and enhancing robustness in various generalization settings.

UniK-QA (Oguz et al., 2022) proposes a framework that integrates structured, unstructured, and semi-structured knowledge sources, such as text, tables, lists, and knowledge bases, which flattens all data into text and applies a unified retriever-reader model.

ArcaneQA (Gu and Su, 2022) introduces a generation-based KBQA model that addresses large search space and schema linking challenges in KBQA, which employs dynamic program induction for efficient search space navigation and dynamic contextualized encoding for improved schema linking.

GMT-KBQA (Hu et al., 2022b) proposes a multi-task learning framework with a shared T5 encoder to improve question answering over knowledge bases by simultaneously learning entity disambiguation, relation classification, and logical form generation.

Uni-Parser (Liu et al., 2022b) unifies semantic parsing for question answering on both knowledge bases and databases by using a three-module approach: primitive enumeration, ranking, and compositional generation.

UnifiedSKG (Xie et al., 2022) unifies 21 structured knowledge grounding tasks into a text-to-text format, leveraging T5 models and multi-task learning to improve performance across diverse tasks and facilitate zero-shot and few-shot learning investigations.

UniKGQA (Jiang et al., 2023b) integrates retrieval and reasoning for multi-hop question answering over knowledge graphs, employing a unified architecture that combines a semantic matching module and a matching information propagation module, enhanced by pre-training and fine-tuning strategies.

DECAF (Yu et al., 2023) combines the generation of logical forms and direct answers, leveraging a sequence-to-sequence framework with retrieval from linearized knowledge bases.

BeamQA (Atif et al., 2023) combines sequence-to-sequence prediction and beam search for multi-hop knowledge graph question answering, using a fine-tuned BART model for path generation and a novel beam search execution algorithm to traverse the knowledge graph and find answers.

HGNet (Chen et al., 2023) proposes a hierarchical query graph generation approach with an outlining stage for structural constraints and a filling stage for instance selection.

SKP (Dong et al., 2023) introduces structured knowledge-aware pre-training tasks, an efficient

linearization strategy, and an interval attention mechanism, leading to significant improvements in subgraph retrieval and encoding.

StructGPT (Jiang et al., 2023a) enhances LLMs' reasoning over structured data using an Iterative Reading-then-Reasoning (IRR) approach, which includes specialized interfaces for efficient data access, a novel invoking-linearization-generation procedure, and iterative reasoning to effectively utilize structured data in answering complex questions.

FC-KBQA (Zhang et al., 2023) introduces a Fine-to-Coarse composition framework for question answering over knowledge bases, utilizing fine-grained component detection, middle-grained component constraints, and coarse-grained component composition.

PanGu (Gu et al., 2023) proposes a grounded language understanding framework that combines a symbolic agent and a neural language model, which allows for the incremental construction of valid plans and utilizes the language model to evaluate the plausibility of these plans.

ToG (Sun et al., 2024) integrates LLMs with KGs for deep and responsible reasoning, using a beam search algorithm in KG/LLM reasoning, which allows the LLM to dynamically explore multiple reasoning paths in KG and make decisions accordingly, enhancing LLMs' deep reasoning capabilities for knowledge-intensive tasks.

E Hyperparameter Settings

We use the grid search method to select the optimal hyperparameter settings for the network. The F1 score of KBQA predicted without oracle entity linking is chosen as the evaluation metric. The hyperparameters that we can adjust and the possible values of the hyperparameters are first determined according to the structure of our model in Table 6.

Afterward, the different hyperparameter choices are combined to judge the merit of the hyperparameter combinations. The optimal hyperparameter combinations of the model are obtained by circular traversal of all combinations. The optimal hyperparameter combinations are shown in **bold**.

For example, WebQSP hyperparameter choices select the Llama-2-7B model, as shown by bolded values, for optimal model performance. LoRA is the fine-tuning type chosen, suggesting low-rank adjustments to model parameters. A train batch size of 4, learning rate of 5e-4, and 50 training epochs indicate a preference for moderate-sized data processing batches and a faster learning rate over many epochs. Test batch size of 4 and beam size of 5 indicate evaluation and prediction generation configuration. The retrieval algorithm was SimCSE because it compares sentence embeddings well. The top-k and threshold values for Entity Retrieval (ER) and Relation Retrieval (RR) were set to balance retrieving relevant information and computational efficiency.

F Effectiveness of Beam Search

Beam search is a heuristic algorithm usually used in sequence generation tasks, which expands the search space by generating multiple highly probable logical forms instead of only one. As shown in Figure 4(b), an increase in beam size enhances the likelihood of executing SPARQL queries based on candidate logical forms, improving the KBQA performance.

G Plug-and-Play Settings

ChatKBQA has a plug-and-play characteristic, as shown in 3 parts, including the Open-source LLMs, PEFT methods, and Unsupervised Retrieval methods, all of which have different candidates. The following is a description of these candidates.

G.1 Open-source Large Language Models

In the open-sourced macro modelling part, we choose Llama-2, ChatGLM2, and Baichuan2.

Llama-2-7B / Llama-2-13B (Touvron et al., 2023): Part of Meta AI's Llama series, these models are auto-regressive transformers with 7 and 13 billion parameters, trained on 2 trillion tokens. They are optimized for dialogue and general language tasks, leveraging supervised fine-tuning and reinforcement learning for better alignment with human preferences.

ChatGLM2-6B (Zeng et al., 2023): Developed by Tsinghua University, this 6.2 billion-parameter bilingual Chinese-English chat model improves upon its predecessor with enhanced performance, longer context support, and efficient inference. It's designed for fluent, coherent conversations in both languages.

Baichuan2-7B / Baichuan2-13B (Yang et al., 2023): From Baichuan Intelligent Technology, these multilingual models have 7 and 13 billion parameters and are trained on 2.6 trillion tokens. They support Chinese and English, offering competitive performance on various language process-

Hyperparameter	WebQSP	CWQ
LLM Selection	**Llama-2-7B**	**Llama-2-13B**
Fine-tuning Type	{**LoRA**, QLoRA, P-tuning v2, Freeze}	{**LoRA**, QLoRA, P-tuning v2, Freeze}
Train Batch Size	{1, 2, 3, **4**}	{1, 2, 3, **4**}
Learning Rate	{**5e-5**, 5e-4, 5e-3}	{**5e-5**, 5e-4, 5e-3}
Train Epoch	{10, 50, **100**}	{**10**, 50, 100}
Test Batch Size	{**1**, 2, 3, 4}	{**1**, 2, 3, 4}
Beam Size	{1, 2, 5, 8, **15**}	{1, 2, 5, **8**}
Retrieval Type	{**SimCSE**, Contriever, BM25}	{**SimCSE**, Contriever, BM25}
ER Top k_e	{5, 10, **50**, 100}	{5, 10, **50**, 100}
ER Threshold t_e	{0.0, 0.0001, **0.001**, 0.01, 0.1}	{0.0, 0.0001, **0.001**, 0.01, 0.1}
ER Top k_1	{10, 30, **50**, 100, 1000}	{10, **30**, 50, 100, 1000}
ER Threshold t_1	{**0.0**, 0.0001, 0.001, 0.01, 0.1}	{**0.0**, 0.0001, 0.001, 0.01, 0.1}
RR Top k_r	{3, 5, **15**, 30}	{3, 5, **15**, 30}
RR Threshold t_r	{0.0, 0.0001, 0.001, **0.01**, 0.1}	{0.0, 0.0001, 0.001, **0.01**, 0.1}
RR Top k_2	{30, **300**, 3000, 10000}	{40, 400, **4000**, 10000}
RR Threshold k_2	{**0.0**, 0.0001, 0.001, 0.01, 0.1}	{**0.0**, 0.0001, 0.001, 0.01, 0.1}

Table 6: Hyperparameter Search.

ing benchmarks and are available for open-source commercial use.

G.2 Parameter-Efficient Fine-Tuning Methods

In the PEFT part, we choose LoRA, QLoRA, P-tuning v2, and Freeze.

LoRA (Low-Rank Adaptation) (Hu et al., 2022a) is a PEFT method that introduces low-rank matrices to adapt large pre-trained models. Instead of fine-tuning all parameters, LoRA modifies only a small number of additional trainable parameters, effectively reducing the computational cost. It alters the weights of a pre-trained model in a low-rank decomposed space, allowing for efficient adaptation while maintaining the original model's structure and size.

QLoRA (Quantized Low-Rank Adaptation) (Dettmers et al., 2023) is an extension of LoRA, combining low-rank adaptation with quantization techniques. It aims to further reduce the computational and memory overhead associated with fine-tuning large models. By quantizing the additional low-rank matrices introduced in LoRA, QLoRA provides a more memory-efficient approach to adapting pre-trained models.

P-tuning v2 (Liu et al., 2022a) advances the concept of prompt tuning, where trainable prompts are added to a fixed pre-trained model to guide its predictions. P-tuning v2 introduces trainable continuous prompts at the embedding layer and employs a sophisticated bi-level optimization strategy. This approach enhances the model's ability to adapt to specific tasks with minimal parameter updates, making it more efficient than traditional fine-tuning methods.

Freeze (Geva et al., 2021) is a parameter-efficient approach where most of the layers of a pre-trained model are frozen, and only a small fraction of the parameters are fine-tuned. This technique significantly reduces the computational resources required for fine-tuning, making it ideal for scenarios with limited budgets. By selectively updating only certain layers or parts of a model, Freeze retains the general knowledge of the pre-trained model while adapting it to specific tasks.

G.3 Unsupervised Retrieval Methods

In the Unsupervised Retrieval part, we choose SimCSE, Contriever and BM25.

SimCSE (Gao et al., 2021) is an unsupervised method for generating sentence embeddings using contrastive learning. It enhances semantic understanding by using variations of the same sentence to train neural networks, improving performance in tasks like textual similarity and natural language inference.

Contriever (Izacard et al., 2022) is an unsupervised technique for creating dense passage embeddings, designed for effective retrieval in large document collections. It focuses on semantic content, offering an advanced alternative to traditional keyword-based retrieval methods.

BM25 (Robertson and Zaragoza, 2009) is a probabilistic ranking function used in search engines. It evaluates document relevance to a search query, improving upon models like TF-IDF by incorporating document length normalization and term frequency saturation.

H Error Analysis

We analyze the questions in the WebQSP test set that were not answered correctly by ChatKBQA without oracle entity linking, and errors can be summarized as follows.

Logical form skeleton error (40.10%). We discover that the majority of the errors are caused by ChatKBQA failing to provide the correct logical form skeleton for the question, e.g. predicting `"(JOIN (R []) (JOIN (R []) []))"` as `"(JOIN (R []) [])"`. This is due to the limited representation of certain complex skeletons in train set.

Entity retrieval error (27.17%). Then, a portion of the samples that predicted the correct logical form skeletons, but did not retrieve the correct entities, e.g. predicting `"(JOIN (R []) m.0d3k14)"` as `"(JOIN (R []) m.07618sw)"`.

Relation retrieval error (19.48%). In the case of successful skeleton prediction and entity retrieval, errors in relation retrieval can also lead to failed logical form generation that does not match the ground truth, e.g. predicting `"(JOIN (R finance.currency.countries_used) m.0kz1h)"` as `"(JOIN (R finance.currency.currency_code) m.0kz1h)"`.

SPARQL convertion error (13.26%). Finally, a small proportion of the remaining errors arise from the fact that, although the generated logical form is consistent with the ground truth, it fails to execute or the answers are inconsistent when converted to SPARQL, which may be caused by the loss of the conversion from logical form to SPARQL.

I Discussion of LLM combined with KG.

I.1 Insights from ChatKBQA.

(1) We propose a straightforward KBQA framework that uses fine-tuned open-source large models for the first time. (2) Innovatively, we adopt a generate-then-retrieve approach to enhance generation outcomes and retrieval efficiency separately, ultimately boosting KBQA performance. (3) Our framework has plug-and-play capabilities, allowing flexible replacement of LLMs and retrieval models to address the KBQA challenge. (4) Our approach introduces a new paradigm for LLMs to conduct interpretable knowledge-based Q&A, offering a fresh perspective on merging LLMs and KGs.

To summarize, ChatKBQA proposes a thought taking both the advantages of using LLMs to do natural language semantic parsing for graph query generation and calling external KBs to interpretably reason with queries, which we name Graph Query of Thoughts (GQoT), a promising LLM+KG combination paradigm to better utilize the external knowledge, improve Q&A's interpretability, and avoid LLM's hallucinations.

I.2 Future Directions.

ChatKBQA still has much room for improvement, such as in the design of the training set, the decomposition of complex questions, support for various graph query languages, and applications in specific domains, which are our future research directions:

Training set design: ChatKBQA is the first method to fine-tune open-source large models using unsupervised retrieval methods for the KBQA task, achieving state-of-the-art results. Therefore, the effectiveness of fine-tuning depends on the quality of the dataset used to map natural language to logical forms. In future work, we plan to enhance the training set by extracting computation graphs from the knowledge graph using graph sampling, then converting them into natural language, and exploring ways to achieve maximum training effectiveness with the least amount of training data.

Decomposition of complex questions: We have seen that for some simple tasks, such as one-hop and two-hop queries, ChatKBQA performs very well because the logical form skeletons involved are very similar and the fine-tuned LLM can generate them effectively. However, generating the corresponding long logical forms for more complex questions is a challenge. Therefore, in future work, we plan to use techniques such as CoT or Agent to decompose natural language questions into simpler logical forms for better performance.

Support for various graph query languages: Currently, ChatKBQA converts generated logical forms into SPARQL queries in two datasets, as the Freebase KB stores knowledge in RDF format. We will explore more KBs and datasets, such as those using the Cypher language like Neo4j, where the methodology of generating and then retrieving with ChatKBQA is also promising.

Open-domain and specific-domain applications: There is a demand for precision knowledge question answering in fields such as open-domain, medicine, finance, and telecommunications. We can first use UIE or LLM information extraction technology to build a knowledge graph, then fine-tune ChatKBQA to understand the structure of the knowledge graph, achieving interpretable knowledge Q&A in open and specific domains.