HTML: Hierarchical Topology Multi-task Learning for Semantic Parsing in Knowledge Base Question Answering

Aziguli Wulamu[1], **Lyu Zhengyu**[1], **Kaiyuan Gong**[1], **Yu Han**[1], **Zewen Wang**, **Zhihong Zhu**[2] and **Bowen Xing**[1*]
[1]Beijing Key Laboratory of SMART Traditional Chinese Medicine for Chronic Disease Prevention and Treatment
Beijing Key Laboratory of Knowledge Engineering for Materials Science
School of Computer and Communication Engineering,
University of Science and Technology Beijing
[2]Jarvis Lab, Tencent

Abstract

Knowledge base question answering (KBQA) aims to answer natural language questions by reasoning over structured knowledge bases. Existing approaches often struggle with the complexity of mapping questions to precise logical forms, particularly when dealing with diverse entities and relations. In this paper, we propose Hierarchical Topology Multi-task Learning (HTML), a novel framework that leverages a hierarchical multi-task learning paradigm to enhance the performance of logical form generation. Our framework consists of a main task: generating logical forms from questions, and three auxiliary tasks: entity prediction from the input question, relation prediction for the given entities, and logical form generation based on the given entities and relations. Through joint instruction-tuning, HTML allows the mutual guidance and knowledge transfer among the hierarchical tasks, capturing the subtle dependencies between entities, relations, and logical forms. Extensive experiments on public benchmarks show that HTML markedly outperforms both supervised fine-tuning methods and training-free methods based on powerful large language models (e.g., GPT-4), demonstrating its superiority in question understanding and structural reasoning.

1 Introduction

Knowledge base question answering (KBQA) is a critical task that bridges unstructured natural language questions with structured knowledge bases (KBs) to deliver precise answers (Lan et al., 2021, 2023). Generally, KBQA involves knowledge retrieval (Yao et al., 2007) and semantic parsing (Berant et al., 2013; Eyal et al., 2023; Banerjee et al., 2023) to obtain the final answer. Regarding the input question, knowledge retrieval may include approaches like named entity recognition (Mayhew et al., 2020) and entity linking (Huang et al., 2020), which provide the meta knowledge for semantic parsing. As the core component in KBQA, semantics parsing aims to translate unstructured natural language questions into the corresponding structured logical forms (e.g. S-expression (Gu et al., 2021)), which is then converted into an executable graph database query (e,g, SPARQL (Harris and Shadbolt, 2005; Pérez et al., 2009)) for retrieving answers from knowledge bases, such as Freebase (Bollacker et al., 2008), Wikidata (Vrandecic and Krötzsch, 2014) and DBpedia (Auer et al., 2007).

With the advances of pre-trained large language models (LLMs), state-of-the-art (SOTA) works widely adopt LLMs to enhance semantic parsing to generate possible logical forms (Wang et al., 2023; Sun et al., 2024; Luo et al., 2024b; Peng et al., 2024; Xu et al., 2024; Dehghan et al., 2024; Luo et al., 2024a; Xiong et al., 2024). ChatKBQA(Luo et al., 2024a) fine-tuning large language models (LLMs) to unify the phrases of knowledge retrieval and semantic parsing, directly generating the logical form only based on the question. In the post-processing retrieval phrase, the logical form is refined to leverage the KBs and converted into the executable query. CoQ(Peng et al., 2024) first decomposes the question into smaller ones and extracts the reference knowledge, based on which the logical form is then generated.

Different from the supervised fine-tuning methods, a number of works (Peng et al., 2024; Sun et al., 2024; Xiong et al., 2024) leverage commercial LLMs' (e.g. GPT4 (OpenAI, 2023)) strong reasoning abilities via in-context learning (Xie et al., 2022; Min et al., 2022) in a training-free manner. This line of methods queries the APIs to generate the executable query (e.g., SPARQL) or logical form to interact with KBs, always consisting of multiple steps and API calls. ToG (Sun et al., 2024) method implements an LLM-KG integrating paradigm via treating the LLM as an agent, which iteratively performs beam search on the KB,

*Corresponding author, bwxing714@gmail.com

Findings of the Association for Computational Linguistics: ACL 2025, pages 9307–9321

identifies the most promising reasoning paths, and ultimately returns the most likely reasoning results.

Despite the recently achieved progress, we discover that there exist three issues in semantic parsing, resulting in inaccurately mapping complex questions to logical forms or excusable queries, especially when handling diverse entities and relations that require multi-hop reasoning.

(1) **Meta knowledge deficiency**. This issue lies in the methods that perform semantic parsing without meta knowledge extraction process. Without the candidate entities and relations, it is hard for LLMs to directly generate totally correct logical forms only based on the questions.

(2) **Error propagation**. Some existing works first extract the reference knowledge and then leverage the results to guide the logical form generation. However, there may exist noisy knowledge in the extraction results, thus the error prorogation issue would harm the semantic parsing phrase.

(3) **Insufficient knowledge alignment while high cost**. Even equipped with powerful commercial LLMs, in-context-learning cannot sufficiently align the key information in the question with the structured knowledge in the KBs, resulting in worse performances than SFT methods (as shown in Table 1). Besides, training-free methods generally require a large number of API calls, which is usually an unforgettable cost.

To tackle the above issues, in this paper, we propose a novel hierarchical task topology paradigm in semantic parsing, as shown in Fig. 1. Regarding semantic parsing as the main task, we discompose it into three subtasks: entity recognition and alignment (EntRA), entity-aware relation extraction (EARE) and logical form skeleton generation (SkelGen). EntRA aims to extract the entities in the question and predict the aligned ones in the KBs. EARE aims to predict the relations existing among the given entities. SkelGen aims to generate the correct logical form with the given entities and the relations among them. The three subtasks are intertwined with each other with intrinsic hierarchical topologies, and they collaboratively support semantic parsing.

To further implement our paradigm, we propose Hierarchical Topology Multi-task Learning (HTML) to effectively model the interdependencies among semantic parsing, EntRA, EARE and SkelGen within a unified instruction-tuning framework, as shown in Fig. 2. After constructing the instructions for each task, we mix them in specific proportions to form an instruction-following training set for fine-tuning the open-resource LLMs. By leveraging mutual guidance and knowledge transfer across these hierarchically organized tasks, HTML effectively models the compositional nature of KBQA, where entity-relation dependencies and structural reasoning are tightly intertwined. In the inference stage, only the semantic parsing instruction is adopted while the captured hierarchical topology multi-task knowledge support accurate generation of logical forms. Experimental results on standard benchmarks demonstrate that HTML significantly outperforms both SFT and training-free SOTA approaches, demonstrating stronger capability in question understanding and structural reasoning. This work advances the KBQA field by illustrating how hierarchical task decomposition and multi-task instruction-tuning can be integrated to address the challenge of semantic parsing.

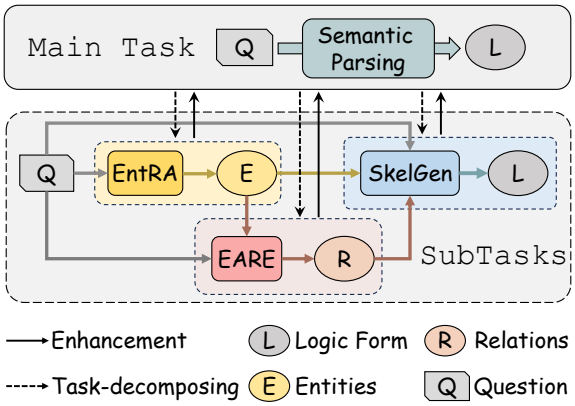

Figure 1: Illustration of the hierarchical task topology paradigm of HTML.

2 HTML

Accurately mapping natural language questions to structured logical forms remains a significant challenge in KBQA. To address this fundamental problem, we propose an innovative Hierarchical Topological Multi-Task Learning (HTML) framework (shown in Fig. 2), which implements a hierarchical task topology paradigm (shown in Fig. 1). In this section, we depict the details of HTML.

2.1 Hierarchical Task Topology

We decompose the semantic parsing (SP) phase in KBQA into several interdependent subtasks: Entity Recognition and Alignment (EntRA), Entity-Aware Relation Extraction (EARE), and Logical Form Skeleton Generation (SkelGen). As shown in Fig. 1, EntRA takes the question as input and

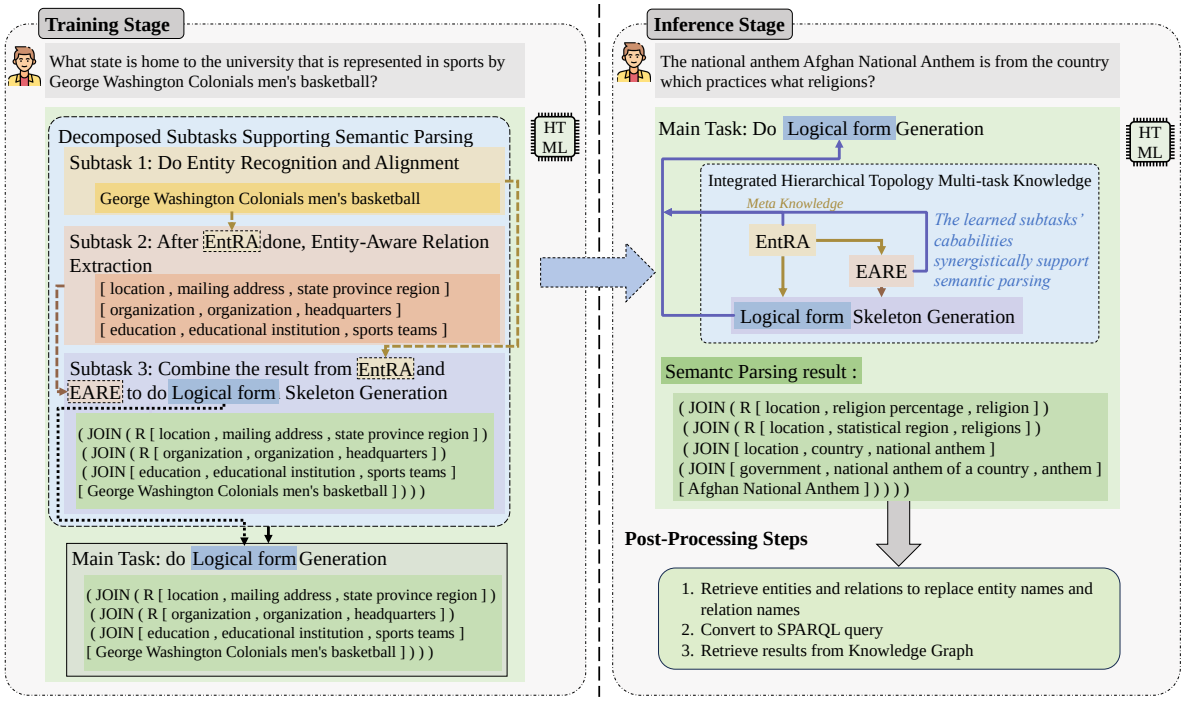

Figure 2: Illustration of the overall framework of HTML.

generates the entity candidates aligned in the KBs. EARE accepts the question and aligned entities as the input and outputs the relations existing among the entities. As for SkelGen, it serves for logical form generation based on the given entity and relation candidates. This paradigm systematically decomposes the challenge semantic parsing task into hierarchical subtasks, which are deeply coupled and intertwined. They work together to provide the meta knowledge for semantic parsing, facilitating the enhancement of the model's capacity for question understanding and structural reasoning.

2.2 Task-specific Instructions

Semantic Parsing (Main Task) The primary instruction $\mathcal{I}_{SP}$ directs LLMs to convert the natural language question text into the corresponding logical form. Specifically, our designed $\mathcal{I}_{SP}$ is as follows:

Instruction:
Please generate the corresponding **logical form** of this **question**: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*

Response:
(AND (JOIN [location , statistical region , child labor percent] (JOIN [measurement unit , dated percentage , rate] ["1 , 8"])) (JOIN (R [language , human language , countries spoken in]) [Portuguese Language]))

In this instruction, the keywords "logical form" and "question" emphasize critical information prompting the LLM to perform semantic parsing.

Entity Recognition and Alignment The EntRA subtask instruction $\mathcal{I}_{EntRA}$ guides the LLM to output the identified and aligned the entities in the KBs. Specifically, our designed $\mathcal{I}_{EntRA}$ is as:

Instruction:
Please identify the **entities** in the following question and output their **corresponding names aligned in the knowledge base**. **Question**: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*

Response:
[Portuguese Language]

The key words "entities" and "question" as well as the phrase "corresponding names aligned in the knowledge base" indicate the task information of EntRA that prompts the LLM to perform entity recognition and alignment.

Entity-Aware Relation Extraction The instruction $\mathcal{I}_{EARE}$ of this task guides the LLM to extract the relations existing among the entity candidates. Specifically, our designed $\mathcal{I}_{EARE}$ is as:

Instruction:
Given a question and some entities, please identify and output the **relations** among the **entities**. **Question**: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*
Entities: Portuguese Language

Response:
[location , statistical region , child labor percent]
[measurement unit , dated percentage , rate]
[language , human language , countries spoken in]

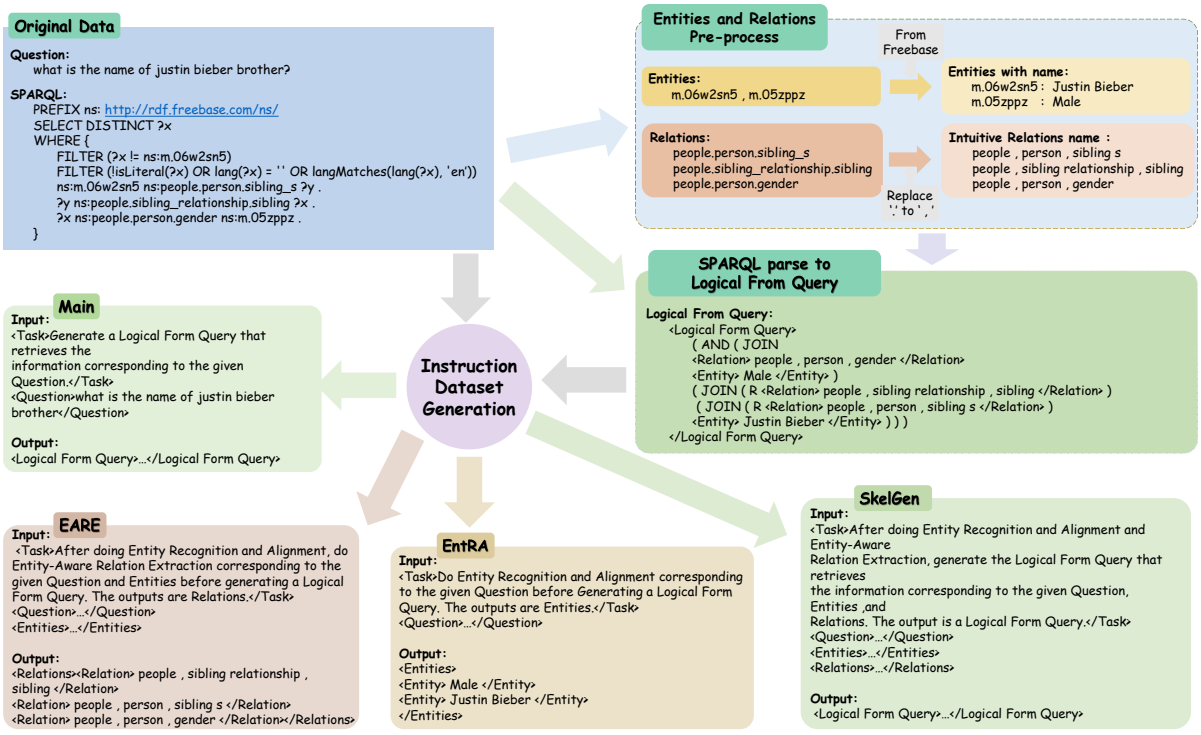

Figure 3: Illustration of the instruction construction process.

The key words "entities", "relations", and "question" convey the task information of EARE that prompts the LLM to perform entity-aware relation extraction.

Logical Form Skeleton Generation The instruction $\mathcal{I}_{SkelGen}$ of this task guides the LLM to generate the correct logical form of the question with the guidance of the given entities and relations. Specifically, $\mathcal{I}_{SkelGen}$ is designed as:

Instruction:
Please generate the **logical form** of the following question, based on the provided **entities** and **relations**.
Question: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*
Entities: Portuguese Language
Relations:
[location , statistical region , child labor percent]
[measurement unit , dated percentage , rate]
[language , human language , countries spoken in]

Response:
(AND (JOIN [location , statistical region , child labor percent] (JOIN [measurement unit , dated percentage , rate] ["1 , 8"])) (JOIN (R [language , human language , countries spoken in]) [Portuguese Language]))

The keywords "question", "logical form", "entity" and "relations" provide the task information of SkelGen for the LLM. Then the LLM is guided to compose the entity-relation components and analyze the logical structure, finally generating the correct logical form.

2.3 Training and Inference

Training We first construct all instructions of $\mathcal{I}_{SP}$, $\mathcal{I}_{EntRA}$, $\mathcal{I}_{EARE}$ and $\mathcal{I}_{SkelGen}$ for all training samples, as shown in Fig. 3. Then we randomly select α ratio of the instructions of $\mathcal{I}_{EntRA}$, $\mathcal{I}_{EARE}$ and $\mathcal{I}_{SkelGen}$ and merge them with all $\mathcal{I}_{SP}$ instructions, forming the final instruction-following fine-tuning data. We use the shuffled training data with batching strategy to fine-tune the LLM in the text-to-text generation form. The training objective is to minimize the negative log-likelihood for each instruction: $\mathcal{L} = -\sum_{n=1}^{N} \log p\left(y_n \mid y_{<n}, I\right)$. N is the length of the golden output sequence $y_1, ..., y_N$ and I denotes the current input instruction.

As illustrated in Fig. 2, the interdependencies of subtasks are explicitly modeled through the task-specific keywords in their instructions, establishing functional equivalence between their combined operation and the main task objective. Through joint instruction-tuning, the primary task semantic parsing undergoes the systematic enhancement from the three interconnected subtasks, because this explicit multi-task learning framework enables simultaneous optimization of the primary task, and subtasks as well as the modeling of their correlations during model training.

Method	SFT	Backbone	WebQSP			CWQ		
			F1	Hit@1	Acc	F1	Hit@1	Acc
ChatGPT(Wang et al., 2023)	×	GPT-3.5-Turbo	–	66.8	–	–	39.9	–
ChatGPT+CoT(Wang et al., 2023)	×	GPT-3.5-Turbo	–	75.6	–	–	48.9	–
KD-CoT-RtR(Wang et al., 2023)	×	GPT-3.5-Turbo	50.2	73.7	–	–	50.5	–
KD-CoT(Wang et al., 2023)	×	GPT-3.5-Turbo	52.5	68.6	–	–	55.7	–
ToG-R(Sun et al., 2024)	×	GPT-3.5-Turbo	–	75.8	–	–	58.9	–
ToG-R(Sun et al., 2024)	×	GPT-4	–	81.9	–	–	69.5	–
ToG(Sun et al., 2024)	×	GPT-3.5-Turbo	–	76.2	–	–	57.1	–
ToG(Sun et al., 2024)	×	GPT-4	–	82.6	–	–	67.6	–
GoG(Xu et al., 2024)	×	GPT-4	–	84.4	–	–	75.2	–
ARC-KBQA(Tian et al., 2024)	×	GPT-3.5-0125	75.6	–	–	–	–	–
Interactive-KBQA(Xiong et al., 2024)	×	GPT-4-Turbo	–	71.2	–	–	49.1	–
WebGLM(Dehghan et al., 2024)	×	WebGLM-10B	–	63.5	–	–	42.3	–
EWEK-QA(Dehghan et al., 2024)	✓	WebGLM-10B	–	71.3	–	–	52.5	–
CoQ(Peng et al., 2024)	✓	GPT-3.5-Turbo,T5	78.1	79.3	–	78.8	79.0	–
RoG(Luo et al., 2024b)	✓	LLaMA2-Chat-7B	70.8	**85.7**	–	56.2	62.6	–
ChatKBQA(Luo et al., 2024a)	✓	LLaMA2-7B	79.8	83.2	73.8	–	–	–
ChatKBQA(Luo et al., 2024a)	✓	LLaMA2-13B	–	–	–	77.8	82.7	73.3
HTML(Ours)	✓	LLaMA2-7B	81.0	84.4	74.9	77.4	81.5	73.7
HTML(Ours)	✓	LLaMA2-13B	**81.1**	84.1	**75.3**	**78.9**	**82.9**	**75.1**

Table 1: Performance comparison with different baselines on WebQSP and CWQ.

Inference In the inference stage, the initial step is to generate the logical form of the question using $\mathcal{I}_{SP}$. Subsequently, an entity and relation mapping phase is performed via KB retrieval (Luo et al., 2024a). Then the refined logical form is transformed into the executable SPARQL query. Finally, the query is used to retrieve and obtain the exact answer from the KB.

Note that we only adopt $\mathcal{I}_{SP}$ for inference, without introducing extra inference latency and cost. After the training stage, the respective functions of the subtasks have been captured and integrated into the LLM and the learned capabilities can effectively support semantic parsing.

3 Experiments

3.1 Settings

Datasets Following previous works (Sun et al., 2024; Luo et al., 2024a), we adopt ComplexWebQuestions 1.1 (CWQ) (Talmor and Berant, 2018) and WebQuestionsSP (WebQSP) (Yih et al., 2015) as our text bed to evaluate our HTML. The selected datasets are widely recognized within the field for their complexity and relevance. Specifically, CWQ extends WebQSP by incorporating more intricate query structures that demand a deeper understanding of the underlying knowledge bases. WebQSP includes 3098 and 1639 samples for training and testing, respectively. CWQ consists of 27639, 3531 and 3519 samples for training, testing and validation, respectively.

Implementation Details We choose the open-source 7B and 13B version of LLaMA2[1] as our LLM backbone. LoRA (Hu et al., 2022a) is adopted for parameter-efficient fine-tuning. All experiments are conducted on a NVIDIA A40 GPU equipped with 48GB of video memory. For fair comparison, we use the same SFT hyper-parameter setting as SOTA baselines (Luo et al., 2024a). The learning rate was set to 5e-5, the batch size was configured to 4, and gradient updates were scheduled every four steps. HTML[2] is trained for 100 epochs on the WebQSP dataset and 10 epochs on the CWQ dataset.

Evaluation Metrics In this study, we employ three conventional evaluation metrics: F1 score, Hits@1, and Accuracy (Acc). The F1 score serves as a comprehensive metric that evaluates the overall answer coverage by harmonizing the precision and recall rates of predicted answers. The Hits@1 metric specifically measures the accuracy rate of the top-ranked prediction among all candidate answers. Accuracy (Acc) is utilized to quantify the exact-match ratio, representing the proportion of questions that receive completely correct answers within the entire question set.

3.2 Main Results

This study compares nine state-of-the-art KBQA methods as baselines, which encompass two major technical paradigms: Training-Free and SFT

[1] https://huggingface.co/meta-llama

[2] https://github.com/XingBowen714/HTML

Method	Backbone	WebQSP			CWQ		
		F1	Hit@1	Acc	F1	Hit@1	Acc
Interactive-KBQA	GPT-4-Turbo	–	78.64	–	–	56.74	–
ChatKBQA	LLaMA2-7B	83.5	86.4	77.8	–	–	–
ChatKBQA	LLaMA2-13B	–	–	–	81.3	86.0	76.8
HTML(Ours)	LLaMA2-7B	83.5	86.6	77.7	81.6	85.6	77.9
HTML(Ours)	LLaMA2-13B	**84.1**	**87.1**	**78.3**	**82.8**	**86.5**	**79.0**

Table 2: Comparison of Results using Golden Entities as retrieval results.

Dataset	Ablation	Retri. Ent.			Gold. Ent.		
		F1	Hit@1	Acc	F1	Hit@1	Acc
WebQSP	HTML	81.0	84.4	74.9	83.5	86.6	77.7
	w/o EntRA	76.3	79.6	70.4	79.5	82.4	73.9
	w/o EARE	77.2	80.2	71.4	80.1	83.0	74.6
	w/o SkelGen	76.1	79.3	70.1	78.9	82.0	73.1
CWQ	HTML	77.4	81.5	73.7	81.6	85.6	77.9
	w/o EntRA	71.7	77.8	67.6	75.7	81.4	71.5
	w/o EARE	71.9	77.9	67.6	75.8	81.7	71.6
	w/o SkelGen	69.2	75.4	64.7	73.2	79.5	68.7

Table 3: Ablation results of subtasks.

(as detailed in Table 1). We compare our HTML with various SOTA approaches, and the results are shown in Table 1. We can find that our proposed HTML method achieves SOTA performance on both the WebQSP and CWQ datasets. Specifically, when using LLaMA2-13B as the backbone, HTML outperforms previous best SFT SOTA ChatKBQA by 1.5% and 1.8% in Acc on the WebQSP and CWQ datasets, respectively. Compared with previous best training-free SOTA GOG, we achieve 7.7% improvement in Hit@1 on the CWQ dataset. The significant improvements can be attributed to the fact that HTML can marry the hierarchical task decomposition and multi-task instruction-tuning to capture integrated hierarchical topology multi-task knowledge. Besides, generally SFT approaches can outperform training-free ones, especially on the CWQ dataset, which includes more complex and challenging scenarios. This indicates that fine-tuning smaller open-source LLMs remain an effective technical solution for complex KBQA.

Following the practices of some baselines (e.g., Interactive-KBQA and ChatKBQA), we also conduct a set of evaluations using Golden Entities instead of Retrieved Entities, The performance comparison is shown in Table 2. On the WebQSP dataset, HTML gains significant improvements of 2.2% to 3.0% in all metrics using Golden Entities compared to Retrieved Entities. On the CWQ dataset, the improvements are sharper, ranging from 3.6% to 4.2% percentage points. Compared with baselines, our HTML demonstrates consistent and significant superiorities, e.g., 2.2% improvement in Acc on CWQ.

3.3 Ablation Study of Subtasks

We conduct a group of ablation experiments to verify the effectiveness of each subtask decomposed from the main task semantic parsing. The results are shown in Table 3. We can observe that removing any subtask leads to obvious performance decreases on both datasets and all metrics. For instance, the removal of the EntRA task causes ~4% and ~6% Acc drops on WebQSP and CWQ datasets, respectively. The similar phenomenon can be observed from the results of w/o EARE. As for SkelGen, removing it leads to >4% and >9% drops in Acc on both datasets. These results demonstrate that the decomposed subtasks play a vital role in HTML via providing the model with respective capabilities for semantic parsing. We can find that the performance drops on CWQ are sharper than WebQSP, which is an easier benchmark. This can be attributed that our HTML unleashes more potential in the more challenging scenarios. Besides, w/o SkelGen brings more obvious performance drops than w/o EntRA and EARE. We suspect the reason is that SkelGen is the closest subtask to the main task, more responsible for the generated logical form quality.

3.4 Effect of Instruction Mixture Ratio α

To study the effect of the ratio α that controls the extent of subtask instructions, we conduct a group of experiments with different values of α in the range of [10%, 30%, 50%, 70%, 100%]. The experiment results are shown in Fig. 4. We can find that HTML's performance is relatively sensitive to the value of α. Specifically, for larger model scales, best performances are achieved when $\alpha = 50\%$ is adopted for the CWQ dataset while $\alpha = 70\%$ for WebQSP. For smaller model scales, 70% and 100% are chosen for α to achieve the best scores on CWQ and WebQSP, respectively.

3.5 Task Topology Analysis

This study explores 6 different settings of task topology and the results are shown in Table 4. Due to space limitation, the definitions of Only E, Only R, E&R, CoT, Mix in Appendix section *Definition and Instruction of Different Task Topologies in Sec*.

The performance of various framework enhancements to LLaMA2 models (13B and 7B) across the

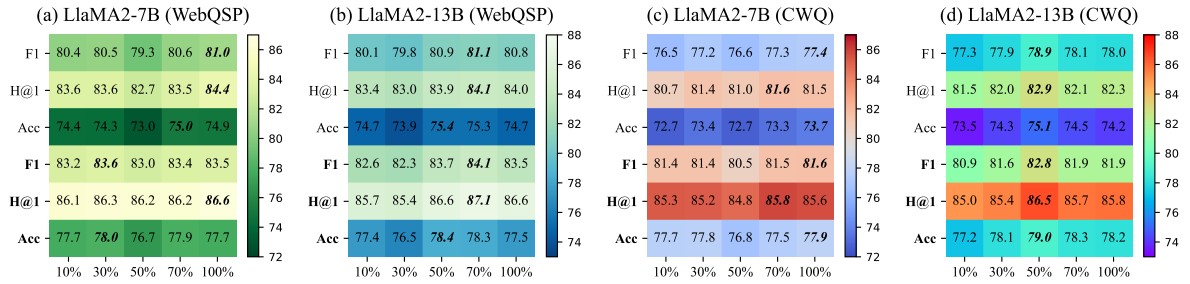

Figure 4: Experiment results on F1 using different value of α.

Dataset	Model	Method	Retri. Ent.			Gold. Ent.		
			F1	Hit@1	Acc	F1	Hit@1	Acc
WebQSP	LLaMA2 7B	Only E	80.1	83.2	74.2	83.5	86.5	77.8
		Only R	80.0	82.8	74.4	83.0	85.6	**77.9**
		E&R	80.5	83.6	74.3	82.8	85.8	76.9
		CoT	79.6	82.9	73.9	83.0	86.0	77.5
		Mix	80.0	83.0	74.0	82.9	85.6	77.4
		HTML	**81.0**	**84.4**	**74.9**	**83.5**	**86.6**	77.7
	LLaMA2 13B	Only E	80.8	83.9	74.7	83.7	86.8	77.9
		Only R	80.7	83.8	74.6	83.6	86.7	77.8
		E&R	81.0	84.0	75.0	83.5	86.5	77.5
		CoT	80.0	83.5	74.1	83.8	86.7	78.5
		Mix	80.2	83.7	74.2	83.7	86.5	78.2
		HTML	**81.1**	**84.1**	**75.3**	**84.1**	**87.1**	**78.3**
CWQ	LLaMA2 7B	Only E	76.8	80.8	73.1	80.8	84.7	77.1
		Only R	76.7	81.1	72.7	80.5	84.7	76.6
		E&R	76.6	80.5	73.3	81.1	84.8	77.7
		CoT	73.6	80.8	73.2	81.0	84.7	77.5
		Mix	**77.4**	**81.5**	**73.7**	**81.6**	85.4	**78.0**
		HTML	**77.4**	**81.5**	**73.7**	**81.6**	**85.6**	77.9
	LLaMA2 13B	Only E	78.1	82.3	74.2	82.1	86.1	78.4
		Only R	**79.0**	**83.4**	75.0	81.9	86.0	78.1
		E&R	78.1	82.4	74.3	82.0	86.0	78.4
		CoT	78.1	82.6	74.2	81.8	85.9	77.9
		Mix	78.6	82.8	74.7	81.8	85.7	78.5
		HTML	78.9	82.9	**75.1**	**82.8**	**86.5**	**79.0**

Table 4: Performances of different task topologies.

Dataset	Task Acc	ChatKBQA		HTML	
		Retrieve	Golden	Retrieve	Golden
WebQSP	Entity	81.8	81.8	83.3	83.3
	Relation	70.8	71.2	72.0	72.4
	Skeleton	82.5	82.6	82.4	82.5
	LF-EM	63.2	63.1	64.4	64.4
CWQ	Entity	85.7	86.7	86.1	87.4
	Relation	80.4	81.2	81.7	82.6
	Skeleton	74.1	74.6	75.3	76.0
	LF-EM	58.1	57.5	58.8	59.6

Table 5: HTML vs SOTA on different subtasks.

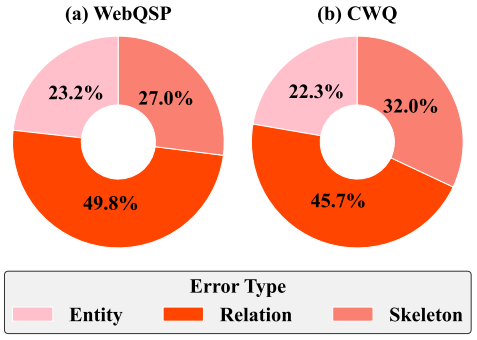

Figure 5: Error type distributions.

CWQ and WebQSP datasets. The evaluated methods include baseline (Main), entity name alignment (Only E), relation prediction (Only R), combined entity name alignment and relation prediction (E&R), mixed strategies (Mix), HTML (HTML), and Chain-of-Thought (CoT). Metrics focus on F1 scores , Hit@1 and accuracy for Entity Retrieval and Golden Entity tasks. Key findings indicate that HTML consistently achieves superior results, particularly for the 13B model on WebQSP (e.g., 81.1 F1 for Entity Retrieval). The 7B model benefits most from Mix on CWQ (77.4 F1). Overall, integrating structured data (e.g., HTML) and hybrid methods demonstrate robust performance improvements, highlighting the impact of architectural adaptations on entity-centric tasks (see fig.4).

3.6 Subtask-specific Evaluation

Subtask Accuracy We conduct a set of experiments to further analyze the quality of parsed logical forms on four metrics: entity Acc, relation Acc, skeleton Acc and logical form exact match (LF-EM) Acc. Due to space limitation, we put the detailed definition of the metrics in Appendix B. The comparisons of our HTML and ChatKBQA are shown in Table 5. On CWQ with retrieved entities our HTML achieves a 0.4% entity Acc improvement over the baseline, and in relation Acc the improvement is 1.3%. In skeleton Acc, our HTML shows a 1.2% improvement over the baseline, and for LF-EM Acc, the increase was 0.7%. On WebQSP, HTML achieves an improvement of 1.5% nd 1.2% on entity Acc and relation Acc, respectively. For LF-EM Acc, the improvement is 1.2%. These findings indicate that while HTML demonstrates consistent advantages on all aspects of semantic parsing, which thanks to the hierarchi-

Figure 6: Case Studies.

cal task discomposition and multi-task correlative learning achieved by HTML.

Error Analysis We count the logical form errors and categorize them into three types: entity error, relation, and skeleton error. Due to space limitation, we put the content of error definition and metric calculation in Appendix C. The error distribution is shown in Fig. 5. We can observe that the relation errors are the most prevalent, accounting for 49.8% on WebQSP and 45.7% on CWQ. This high error rate stems from the model's reliance on generative capabilities and associative reasoning to infer relationships among the entities. This indicates that while the model excels in entity recognition and logical form skeleton generation, more effort should be paid on relation extraction. Future work can follow this line to address the relation errors for comprehensive system optimization.

3.7 Case Study

In Figure 6, we present two detailed cases to further validate the effectiveness of our approach in practical applications. In Case A, with the natural language question *"What movie directed by Angelina Jolie that was released after 12-22-2011?"*, our method incorporates the main task semantic parsing with the hierarchical subtasks EntRA, EARE and SkelGen, thus successfully generating the correct logical form, avoiding the incorrect relation prediction made by ChatKBQA. In Case B, with the question *"Who is the wife of the president of the United States?"*, our HTML can generates the correct logical form. However, ChatKBQA generate a wrong logical form including an strange entity which is not related to the original question, posing a hallucination issue. This indicates that HTML can alleviate the hallucination issue thanks to the explicit modeling the subtasks which can provide the crucial and beneficial meta knowledge for semantic parsing.

4 Related Works

4.1 Neural Network based KBQA

Early KBQA systems employed modular architectures combining pre-trained language models (PLMs) like BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020) with specialized components. Graph Neural Networks (GNNs) (Schlichtkrull et al., 2018) and sequential models (e.g., LSTM(Wang et al., 2019) and CRF (Wang et al., 2019)) were typically integrated for structural reasoning (Jie and Lu, 2019), as seen in KagNet (Lin et al., 2019) which cascades PLM encoders with GCNs(Liu et al., 2021) and LSTMs for multi-hop reasoning. Besides, some innovations focuse on architectural adaptations: GMT-KBQA (Hu et al., 2022b) implemented weight-sharing in T5 for multi-task learning, while FC-KBQA (Zhang et al., 2023) ensemble multiple PLMs to enhance performance.

4.2 LLM based KBQA

The recent emergence of LLMs has introduced two primary paradigms for KBQA, namely training-free and SFT.

Training-free approaches emloy powerful commercial LLMs like ChatGPT or GPT4 and rely on multi-step prompting(Sun et al., 2024; Wang et al., 2023) or longer in-context-learning (Nie et al., 2024) to reason through queries (Sun et al., 2024) with multiple API calls. ToG enhances deep reasoning without additional training, while (Nie et al., 2024) proposes transforming logical forms into code generation to reduce formatting errors. However, the cost of commercial API calls is usually of high invocation costs, and training-free methods often under-perform SFT approaches in overall performance.

SFT approaches adopt the open-resource LLMs (e.g., LLaMA series (Touvron et al., 2023)) as the backbone and adopt parameter-efficient fine-tuning (e.g., LoRA (Hu et al., 2022a)) strategy (Lin et al., 2019; Hu et al., 2022b; Ye et al., 2022; Zhang et al., 2023; Jiang et al., 2023; Luo et al., 2024b; Peng et al., 2024; Luo et al., 2024a; Dehghan et al., 2024). ChatKBQA (Luo et al., 2024a) introduces an innovative framework that first generates logical forms followed by entity-relation substitution, leveraging pre-trained language models to significantly enhance the efficiency of knowledge retrieval and the accuracy of semantic parsing.

However, existing SFT methods suffer from meta knowledge deficiency and error propagation. To overcome the drawbacks of existing methods, we propose HTML, which also follows the SFT manner. Different from existing approaches, HTML is based a novel hierarchical task topology paradigm to allow the subtasks synergistically enhance the precise semantic parsing.

5 Conclusion

In this paper, we propose Hierarchical Topology Multi-task Learning (HTML), which is a novel framework discomposing the semantic parsing task into several interdependent subtasks. HTML effectively models the interdependencies among semantic parsing, entity recognition and alignment, entity-aware relation extraction, and logical form skeleton generation within a unified instruction-tuning framework. The subtasks can synergistically enhance semantic parsing by effectively improving the quality of logical form generation. Extensive experiments on public benchmark verify the superiority HTML, which achieves new state-of-the-art performance and demonstrates advantages.

Acknowledgment

This work was supported by the National Key Research and Development Project of China (No. 2022YFC3502303), Fundamental Research Funds for the Central Universities (No. FRF-TP-25-035).

Limitation

Although HTML achieves significant improvement on semantic parsing, it not yet focuses on the executing stage of logical forms. We actually find considerable executing errors in experiments even with correctly generated logical forms. In the future, we plan to extend HTML to handle logical form execution errors by introducing a new task. We also plan to explore HTML's potential in other tasks that require multi-step reasoning, such as task-oriented dialog systems (Xing and Tsang, 2022b, 2023, 2022a,c, 2024a,b; Xing et al., 2024).

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Debayan Banerjee, Pranav Ajit Nair, Ricardo Usbeck, and Chris Biemann. 2023. The role of output vocabulary in T2T lms for SPARQL semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12219–12228. Association for Computational Linguistics.

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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL.

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of*

Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008, pages 1247–1250. ACM.

Mohammad Dehghan, Mohammad Ali Alomrani, Sunyam Bagga, David Alfonso-Hermelo, Khalil Bibi, Abbas Ghaddar, Yingxue Zhang, Xiaoguang Li, Jianye Hao, Qun Liu, Jimmy Lin, Boxing Chen, Prasanna Parthasarathi, Mahdi Biparva, and Mehdi Rezagholizadeh. 2024. EWEK-QA : Enhanced web and efficient knowledge graph retrieval for citation-based question answering systems. 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 14169–14187. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ben Eyal, Moran Mahabi, Ophir Haroche, Amir Bachar, and Michael Elhadad. 2023. Semantic decomposition of question and SQL for text-to-sql parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 13629–13645. Association for Computational Linguistics.

Yu Gu, Sue Kase, Michelle Vanni, Brian M. Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond I.I.D.: three levels of generalization for question answering on knowledge bases. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3477–3488. ACM / IW3C2.

Stephen Harris and Nigel Shadbolt. 2005. SPARQL query processing with conventional relational database systems. In *Web Information Systems Engineering - WISE 2005 Workshops, WISE 2005 International Workshops, New York, NY, USA, November 20-22, 2005, Proceedings*, volume 3807 of *Lecture Notes in Computer Science*, pages 235–244. Springer.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1687–1696. International Committee on Computational Linguistics.

Binxuan Huang, Han Wang, Tong Wang, Yue Liu, and Yang Liu. 2020. Entity linking for short text using structured knowledge graph via multi-grained text matching. In *21st Annual Conference of the International Speech Communication Association, Interspeech 2020, Virtual Event, Shanghai, China, October 25-29, 2020*, pages 4178–4182. ISCA.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zhanming Jie and Wei Lu. 2019. Dependency-guided LSTM-CRF for named entity recognition. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3860–3870. Association for Computational Linguistics.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4483–4491. ijcai.org.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Complex knowledge base question answering: A survey. *IEEE Trans. Knowl. Data Eng.*, 35(11):11196–11215.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2829–2839. Association for Computational Linguistics.

Shuwen Liu, Bernardo Cuenca Grau, Ian Horrocks, and Egor V. Kostylev. 2021. INDIGO: gnn-based inductive knowledge graph completion using pair-wise encoding. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2034–2045.

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. 2024a. 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, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 2039–2056. Association for Computational Linguistics.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024b. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Stephen Mayhew, Nitish Gupta, and Dan Roth. 2020. Robust named entity recognition with truecasing pretraining. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8480–8487. AAAI Press.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. 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.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Yixing Peng, Quan Wang, Licheng Zhang, Yi Liu, and Zhendong Mao. 2024. Chain-of-question: A progressive question decomposition approach for complex knowledge base question answering. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 4763–4776. Association for Computational Linguistics.

Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. 2009. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3):16:1–16:45.

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.*, 21:140:1–140:67.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. 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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 641–651. 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, Miami, Florida, USA, November 12-16, 2024*, pages 11967–11977. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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

Jin Wang, Liang-Chih Yu, K. Robert Lai, and Xuejie Zhang. 2019. Investigating dynamic routing in tree-structured LSTM for sentiment analysis. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3430–3435. Association for Computational Linguistics.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-driven

cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *CoRR*, abs/2308.13259.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Bowen Xing, Lizi Liao, Minlie Huang, and Ivor Tsang. 2024. DC-instruct: An effective framework for generative multi-intent spoken language understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14520–14534, Miami, Florida, USA. Association for Computational Linguistics.

Bowen Xing and Ivor Tsang. 2022a. Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 159–169, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bowen Xing and Ivor Tsang. 2022b. DARER: Dual-task temporal relational recurrent reasoning network for joint dialog sentiment classification and act recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3611–3621, Dublin, Ireland. Association for Computational Linguistics.

Bowen Xing and Ivor Tsang. 2022c. Group is better than individual: Exploiting label topologies and label relations for joint multiple intent detection and slot filling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3964–3975, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bowen Xing and Ivor W Tsang. 2023. Relational temporal graph reasoning for dual-task dialogue language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13170–13184.

Bowen Xing and Ivor W. Tsang. 2024a. Co-guiding for multi-intent spoken language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2965–2980.

Bowen Xing and Ivor W. Tsang. 2024b. Hc^{2}2l: Hybrid and cooperative contrastive learning for cross-lingual spoken language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):8094–8105.

Guanming Xiong, Junwei Bao, and Wen Zhao. 2024. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. 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 10561–10582. Association for Computational Linguistics.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 18410–18430. Association for Computational Linguistics.

Yiyu Yao, Yi Zeng, Ning Zhong, and Xiangji Huang. 2007. Knowledge retrieval (KR). In *2007 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2007, 2-5 November 2007, Silicon Valley, CA, USA, Main Conference Proceedings*, pages 729–735. IEEE Computer Society.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6032–6043. 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.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1002–1017. Association for Computational Linguistics.

A Definition and Instruction of Different Task Topologies in Sec. 3.5

Only E In the Only E methodology, we employed the subtask prompt designated for Entity Recognition and Alignment, maintaining consistency with the notation I_{EntRA} as specified under the method selection criteria. Concurrently, the primary prompt was kept consistent with the notation

I_{SP} , also delineated within the method selection framework.

Only R In the Only R method, the subtask prompt for Relation Prediction/Extraction is formulated as follows, while the main prompt remains consistent with I_{SP} in the method selection.

Instruction:
Please do **Relation Prediction** before Generate a **Logical Form** query.
Question: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*
Entities: Portuguese Language

Response:
[location , statistical region , child labor percent]
[measurement unit , dated percentage , rate]
[language , human language , countries spoken in]

E&R In the E&R method, the prompt for Entity Recognition and Alignment remains consistent with I_{EntRA} in the method selection process, while the prompt for Relation Prediction/Extraction aligns with the Only R method. The combined task is formulated as follows:

Instruction:
Please generate a **Logical Form** query by using the result of **Entity Linking** and **Relation Prediction**.
Question: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*
Entities: Portuguese Language
Relations:
[location , statistical region , child labor percent]
[measurement unit , dated percentage , rate]
[language , human language , countries spoken in]

Response:
(AND (JOIN [location , statistical region , child labor percent] (JOIN [measurement unit , dated percentage , rate] ["1 , 8"])) (JOIN (R [language , human language , countries spoken in]) [Portuguese Language]))

CoT In CoT method, we only applied the prompt as the main prompt. The example is as follows:

Instruction:
After doing **Entity Linking** and doing **Relation Prediction**, generate a **Logical Form** query that retrieves the information corresponding to the given **question** , **Entity Linking** Result and **Relation Prediction** Result.
Question:*In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*

Response:
(AND (JOIN [location , statistical region , child labor percent] (JOIN [measurement unit , dated percentage , rate] ["1 , 8"])) (JOIN (R [language , human language , countries spoken in]) [Portuguese Language]))

Mix In Mix method, we applied the prompt as the subtask prompt, and the main prompt is the same as I_{SP} in the method selection. The example is as follows:

Instruction:
First Do **Entity Linking** and **Relation Prediction**, then generate a Logical Form query corresponding to the given **question**: *In which countries do the people speak Portuguese, where the child labor percentage was once 1.8?*

Response:
(AND (JOIN [location , statistical region , child labor percent] (JOIN [measurement unit , dated percentage , rate] ["1 , 8"])) (JOIN (R [language , human language , countries spoken in]) [Portuguese Language]))

B Calculation Metrics for Sec. 3.6

Let $\mathbf{T}$ denote the test set with its prediction result set $\mathbf{R}$ consisting of C instances. For each prediction result $r \in \mathbf{R}$, the accuracy Acc of Entity, Relation, and SkelGen, can be calculated as follows.

The accuracy Acc_{Entity} of Entity Recognition and Alignment is computed as eq. (1):

$$Acc_{\text{Entity}} = \frac{\sum_{r \in \mathbf{R}} C_{\text{r.ent}}}{\sum_{r \in \mathbf{R}} \tilde{C}_{\text{r.ent}}} \tag{1}$$

The accuracy Acc_{Relation} of Relation Prediction or Extraction is computed as eq. (2):

$$Acc_{\text{Relation}} = \frac{\sum_{r \in \mathbf{R}} C_{\text{r.rel}}}{\sum_{r \in \mathbf{R}} \tilde{C}_{\text{r.rel}}} \tag{2}$$

The accuracy Acc_{Skel} of Logical Form Skeleton Generation is computed as eq. (3):

$$Acc_{\text{Skel}} = \frac{\sum_{r \in \mathbf{R}} C_{\text{r.skel}}}{C} \tag{3}$$

The accuracy Acc_{LFEM} of Logical Form Full Match is computed as eq. (4):

$$Acc_{\text{LFEM}} = \frac{\sum_{r \in \mathbf{R}} C_{\text{r.lfem}}}{C} \tag{4}$$

Where $C_{\text{r.ent}}$ is the result r of correctly predicted entities, $\tilde{C}_{\text{r.ent}}$ is the golden entities in result$r$, $C_{\text{r.rel}}$ is the result r of correctly predicted relations, $\tilde{C}_{\text{r.rel}}$ is the golden relations in result$r$, C is the element count of $\mathbf{R}$, $C_{r.ent}$ is the total entity count of $r \in \mathbf{R}$ and $C_{r.rel}$ is the total entity count of $r \in \mathbf{R}$.

C Definition and Metric Calculation of Error Analysis in Sec. 3.6

In the context of the set $\mathbf{R}$, each element r may contain up to three types of errors:
1. Entity type errors, calculated as shown in eq. (5);
2. Relation type errors, calculated as shown in eq. (6);
3. Skeleton generation type errors, calculated as shown in eq. (7).

$$Err_{\text{r.ent.type}} = \begin{cases} 1, & Err_{\text{r.ent.c}} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$Err_{\text{r.rel.type}} = \begin{cases} 1, & Err_{\text{r.rel.c}} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$Err_{\text{r.Skel.type}} = \begin{cases} 1, & Err_{\text{r.Skel.c}} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Here, $Err_{\text{r.ent.c}}$ denotes the number of entity errors in r, $Err_{\text{r.rel.type}}$ represents the number of relation errors in r, and $Err_{\text{r.Skel.c}}$ indicates whether a logical form skeleton generation error occurs in r (recorded as 1 if an error is present, and 0 otherwise).

We conducted a statistical analysis of the error types committed by the model on the test sets of CWQ and WebQSP, selecting three types of logical form synthesis errors for our error analysis proportion. Specifically, we categorized and quantified these errors Err using the following definitions:

- $Err_{\text{Ent.C}}$: Total number of errors associated with Entity Recognition and Alignment;
- $Err_{\text{Rel.C}}$: Total number of errors related to Entity Attribute Relation Extraction;
- $Err_{\text{Skel.C}}$: Total number of errors pertaining to Logical Form Skeleton Generation.

$$Err_{\text{Ent.C}} = \sum_{r \in \mathbf{R}} Err_{\text{r.ent.c}}$$

$$Err_{\text{Rel.C}} = \sum_{r \in \mathbf{R}} Err_{\text{r.rel.type}}$$

$$Err_{\text{Skel.C}} = \sum_{r \in \mathbf{R}} Err_{\text{r.Skel.c}}$$

The formula for calculating the aggregate count of each error type is defined as follows:

$$Err_{\text{Total}} = Err_{\text{Ent.C}} Err_{\text{Rel.C}} Err_{\text{Skel.C}}$$

Then the percentage Ep of each error type is calculated as:

$$Ep_{\text{Ent}} = \frac{Err_{\text{Ent.C}}}{Err_{\text{Total}}}$$

$$Ep_{\text{Rel}} = \frac{Err_{\text{Rel.C}}}{Err_{\text{Total}}}$$

$$Ep_{\text{Skel}} = \frac{Err_{\text{Skel.C}}}{Err_{\text{Total}}}$$

D Ratios of Subtask Mixing

In table 6, we present the detailed results of the subtask mixing ratios for the CWQ and WebQSP datasets. Specifically, we examine the effects of adjusting the subtask coverage rates on model performance.

When using LLaMA2-13B, for the CWQ dataset, we observe that a 50% subtask coverage rate yields the best results, with an F1 score of 78.6%, a Hit@1 accuracy of 82.8%, and an Acc of 74.7%. On the WebQSP dataset, a 70% subtask coverage rate is optimal, resulting in an F1 score of 81.1%, a Hit@1 accuracy of 84.4%, and an Acc of 74.9%. These findings underscore the importance of adjusting subtask mixing proportions to optimize model performance, particularly in the context of Knowledge Base Question Answering (KBQA) tasks.

When using LLaMA2-7B, for the CWQ dataset, a 70% subtask coverage rate is optimal, yielding an F1 score of 77.4%, a Hit@1 accuracy of 81.5%, and an Acc of 73.7%. On the WebQSP dataset, a 100% subtask coverage rate is preferred, resulting in an F1 score of 81.0%, a Hit@1 accuracy of 84.4%, and an Acc of 74.9%. These results further highlight the significance of adjusting subtask mixing ratios to enhance model performance across different datasets and model scales.

E Explorations on Implementation Strategies

According to Table 7, the third set of parameters demonstrated optimal overall performance on the WebQSP dataset, achieving high scores in F1, Hit@1, and Acc. Consequently, the selected configuration includes employing float16 precision for the backbone network, float32 for Layer-Norm layers, and enabling flash attention to accelerate the training process. Specifically, the Epoch value is set to 100 for the WebQSP dataset, whereas for the CWQ dataset, which has approximately one-tenth the data volume of WebQSP, the Epoch value is configured to 10. This setup aims to optimize model performance while enhancing training efficiency.

Table 6: An Investigation into Subtask Mixing Proportions and Extended Issues Analysis in Hierarchical Topology Multi-task Learning.

LLaMA2-7B													
Dataset		WebQSP						CWQ					
Method	**Cover**	**Retrieved Entities**			**Golden Entities**			**Retrieved Entities**			**Golden Entities**		
		F1	Hit@1	Acc	F1	Hit@1	Acc	F1	Hit@1	Acc	F1	Hit@1	Acc
E&R	10%	79.3	82.2	73.2	81.9	84.7	76.1	**77.5**	**81.8**	**73.5**	**81.2**	**85.4**	77.4
	30%	80.2	83.3	**74.3**	**82.8**	85.7	**77.4**	76.4	80.6	72.6	80.7	84.5	77.1
	50%	79.4	82.4	73.6	82.5	85.2	76.9	76.6	80.6	72.8	80.9	84.5	77.2
	70%	79.3	82.4	73.3	82.5	85.4	76.6	76.3	80.4	72.7	80.2	84.3	76.6
	100%	**80.5**	**83.6**	74.3	82.8	**85.8**	76.9	76.6	80.5	73.3	81.1	84.8	**77.7**
Mix	10%	79.7	**83.0**	73.6	82.4	85.4	76.9	**77.4**	**81.5**	**73.7**	**81.6**	**85.4**	**78.0**
	30%	79.5	82.9	73.3	82.9	85.8	77.2	76.6	80.5	73.0	81.1	84.7	77.5
	50%	79.6	83.0	73.6	**83.0**	**85.9**	**77.6**	76.7	80.7	73.0	81.2	85.1	77.4
	70%	79.6	82.8	73.6	82.6	85.5	76.9	76.6	80.6	72.9	81.0	84.8	77.3
	100%	**80.0**	83.0	**74.0**	82.9	85.6	77.4	76.1	80.1	72.4	80.3	84.1	76.8
HTML	10%	80.4	83.6	74.4	83.2	86.1	77.7	76.5	80.7	72.7	81.4	85.3	77.7
	30%	80.5	83.6	74.3	**83.6**	86.3	**78.0**	77.2	81.4	73.4	81.4	85.2	77.8
	50%	79.3	82.7	73.0	83.0	86.2	76.7	76.6	81.0	72.7	80.5	84.8	76.8
	70%	80.6	83.5	**75.0**	83.4	86.2	77.9	77.3	**81.6**	73.3	81.5	**85.8**	77.5
	100%	**81.0**	**84.4**	74.9	83.5	**86.6**	77.7	**77.4**	81.5	**73.7**	**81.6**	85.6	**77.9**
LLaMA2-13B													
Dataset		WebQSP						CWQ					
Method	**Cover**	**Retrieved Entities**			**Golden Entities**			**Retrieved Entities**			**Golden Entities**		
		F1	Hit@1	Acc	F1	Hit@1	Acc	F1	Hit@1	Acc	F1	Hit@1	Acc
E&R	10%	79.9	82.9	74.0	82.7	85.4	76.9	**78.1**	**82.4**	74.1	81.5	85.7	77.5
	30%	**81.0**	**84.0**	**75.0**	83.4	86.4	**78.0**	78.1	82.4	**74.3**	**82.0**	**86.0**	**78.4**
	50%	80.1	83.0	74.4	83.2	85.8	77.6	77.9	81.9	74.3	81.7	85.4	78.2
	70%	79.9	83.1	74.0	83.3	86.1	77.3	78.1	82.3	74.3	82.0	86.0	78.2
	100%	81.0	84.0	75.0	**83.5**	**86.5**	77.5	77.7	81.7	74.3	81.8	85.7	78.3
Mix	10%	80.4	**83.7**	74.4	83.1	86.0	77.5	**78.6**	**82.8**	**74.7**	**81.8**	**85.7**	**78.5**
	30%	80.1	83.6	74.1	83.6	**86.6**	77.9	77.4	81.6	73.6	81.2	85.2	77.5
	50%	80.2	83.7	74.2	**83.7**	86.5	**78.2**	77.8	82.1	74.0	81.2	85.1	77.6
	70%	80.3	83.4	74.2	83.4	86.2	77.7	78.1	82.3	74.1	81.3	85.1	77.5
	100%	**80.7**	83.7	**74.6**	83.7	86.2	78.1	78.0	81.8	74.4	81.6	85.2	78.1
HTML	10%	80.1	83.4	74.7	82.6	85.7	77.4	77.3	81.5	73.5	80.9	85.0	77.2
	30%	79.8	83.0	73.9	82.3	85.4	76.5	77.9	82.0	74.3	81.6	85.4	78.1
	50%	80.9	83.9	**75.4**	83.7	86.6	**78.4**	**78.9**	**82.9**	**75.1**	**82.8**	**86.5**	**79.0**
	70%	**81.1**	**84.1**	75.3	**84.1**	**87.1**	78.3	78.1	82.1	74.5	81.9	85.7	78.3
	100%	80.8	84.0	74.7	83.5	86.6	77.5	78.0	82.3	74.2	81.9	85.8	78.2

Table 7: Performance on different implementation strategies.

Implementation Combination	**Strategies**				**Retrieved Entities**			**Golden Entities**		
	Epoch	**Compute**	**Layer-Norm**	**Flash-Attn**	F1	Hit@1	Acc	F1	Hit@1	Acc
1	100	fp16	fp32	–	80.1	83.1	74.3	83.1	86	77.8
2	100	fp16	fp16	–	78.9	81.8	73.3	82	84.5	76.8
3	**100**	**fp16**	**fp32**	✓	**80.1**	**83.2**	**74.3**	**83.3**	**86.0**	**78.2**
4	100	fp16	fp16	✓	79.4	82.7	73.1	82.7	85.8	76.9
5	100	bf16	bf16	–	79.9	83	73.9	83	85.8	77.5
6	100	bf16	bf16	✓	79.8	82.9	74	82.3	85.1	77.1
7	50	fp16	fp32	–	78.9	81.9	73	81.4	84.2	75.8
8	75	fp16	fp32	–	79.0	82.1	73.3	82.1	85.0	76.8