Distribution Shifts Are Bottlenecks: Extensive Evaluation for Grounding Language Models to Knowledge Bases

Yiheng Shu*
The Ohio State University
shu.251@osu.edu

Zhiwei Yu
Microsoft
zhiwyu@microsoft.com

Abstract

Grounding language models (LMs) to knowledge bases (KBs) helps to obtain rich and accurate facts. However, it remains challenging because of the enormous size, complex structure, and partial observability of KBs. One reason is that current benchmarks fail to reflect robustness challenges and fairly evaluate models. This paper analyzes whether these robustness challenges arise from distribution shifts, including environmental, linguistic, and modal aspects. This affects the ability of LMs to cope with unseen schema, adapt to language variations, and perform few-shot learning. Thus, the paper proposes extensive evaluation protocols and conducts experiments to demonstrate that, despite utilizing our proposed data augmentation method, both advanced small and large language models exhibit poor robustness in these aspects. We conclude that current LMs are too fragile to navigate in complex environments due to distribution shifts. This underscores the need for future research focusing on data collection, evaluation protocols, and learning paradigms.[1]

1 Introduction

Language models (LMs), such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), and the GPT series (Ouyang et al., 2022; OpenAI, 2023), have demonstrated impressive capabilities in understanding and generating languages, highlighting the potential for artificial general intelligence (AGI). However, a major obstacle to achieving this goal is that LMs mainly built on natural languages are not yet well-grounded to real-world environments, such as knowledge base (KB), an environment of enormous size, complex structure, and only partially observable to LM.

Though LMs are highly skilled at natural language question answering (QA) today, the task of Knowledge Base Question Answering (KBQA) aims to parse natural language queries into formal queries on KBs, such as Freebase (Bollacker et al., 2008) and Wikidata (Vrandecic and Krötzsch, 2014). The significance of this task lies in building language agents on complex environments (Su, 2023), rather than merely recalling answers from the LM's stored knowledge.

Now, numerous LM-driven models (Das et al., 2021; Hu et al., 2022) continue to achieve higher F1/Hits@1 scores on KBQA benchmarks. However, achieving higher scores does not necessarily guarantee the development of robust and dependable models. We still need to ask whether such improvements apply to extensive scenarios, as benchmarks almost always create questions via crowdsourcing and evaluate with simplistic metrics (Table 1). These benchmarks may not fully represent the diverse scenarios encountered in real-world applications, which raises concerns about the robustness of LM-driven models. Thus, our research aims to bridge this gap by exploring the limitations of current KBQA benchmarks and proposing more comprehensive evaluation protocols.

To achieve this goal, we need to grasp the key factors in robustness. For modern deep learning systems, the amount of training data could be extremely rich, but *robustness is closely related to data distribution* (Hendrycks et al., 2020). In the general area of natural language processing (NLP), large-scale corpora have been collected and used for effective training (Touvron et al., 2023). However, real-world environments are rarely so accommodating, e.g., large KBs contain complex structures and schema items, and building a large-scale and representative corpus is quite challenging. The problem of inconsistent data distribution during training and inference, i.e., **distribution shifts** as shown in Figure 1, may negatively impact the per-

* Work performed when the author was graduate student at Nanjing University.

[1]Code and data are available at https://github.com/yhshu/Distribution-Shifts-for-KBQA.

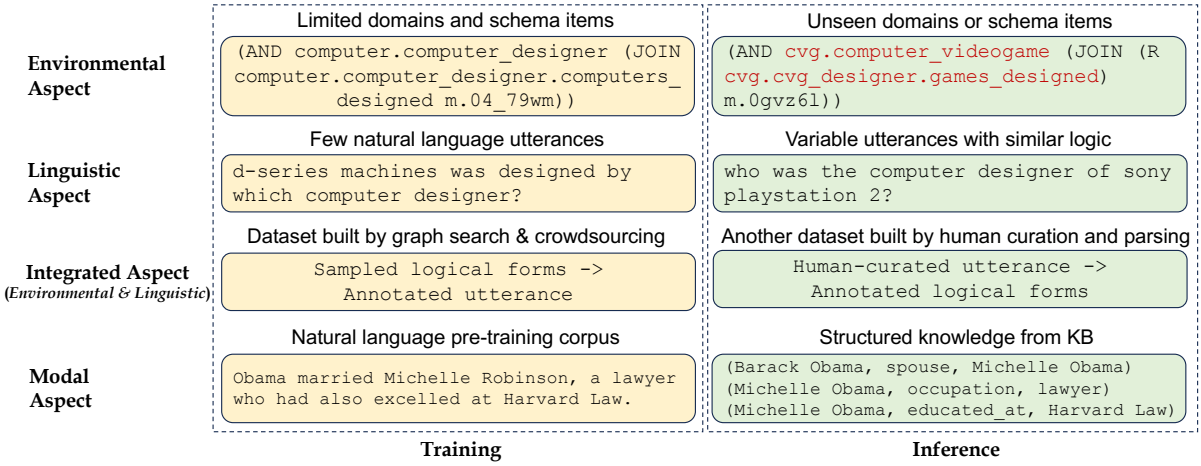

Figure 1: Distribution shifts during grounding in the case of KBQA. Training and inference using LM face completely different data distributions. We call for these shifts to be reflected in the evaluation protocols.

formance and robustness of grounded LMs.

This paper analyzes the distribution shifts from multiple aspects to understand both KBQA benchmarking and LM grounding better. We review existing works and identify several challenges. 1) **Environmental aspect**: generalization to unseen domains at the schema[2] level (Gu et al., 2021); 2) **Linguistic aspect**: adaptation to paraphrases featuring diverse language variations (Su et al., 2016); 3) **Integrated aspect**: transferability across datasets with both novel schema items and linguistic styles (Cao et al., 2022b); 4) **Modal aspect**: few-shot learning ability of large language models (LLMs) for the unfamiliar KB modality (Li et al., 2023). These challenges guide us to build more rigorous evaluation protocols for multiple aspects (§4).

To measure the impact of distribution shifts, we conduct extensive experiments under the proposed evaluation protocols. To present a fair evaluation under such rigorous protocols, we implement a data augmentation method for evaluated LMs and a retrieval augmentation method for evaluated LLMs (§5.1). Our findings reveal that even when employed with such methods and the highest EM scores are achieved on the GrailQA benchmark (Gu et al., 2021), *advanced small and large LMs still fall short of effectively tackling the majority of these challenges*. A striking example is the large difference between the best practice without WebQSP (Yih et al., 2016) fine-tuning (F1 43.0%) compared to the fine-tuned state-of-the-art (F1 79.6%) (§6), suggesting the weak robustness of LM-driven KBQA models on an unseen dataset. Such negative observations highlight an urgent need for future research in data collection methodologies and LM learning paradigms. Meanwhile, we expect our evaluation protocols to provide a reference for future benchmark construction, developing metrics that consider robustness.

Our contributions include 1) A systematic **analysis** of the robustness challenges of grounded LMs and a strong advocacy of a multi-aspect **evaluation protocol** for KBQA. 2) Extensive **experiments** quantitatively unveil the existence of these challenges and the vulnerability of both small and large LMs. 3) **Insights** into improving the robustness of grounded LMs, including data collection and learning paradigms.

2 Related Work

Compared to existing research, the necessity of this study is threefold. First, while LMs are commonly evaluated on natural language tasks (Hupkes et al., 2022; Patel et al., 2022), the complexity increases significantly when these models are applied to environments like KBs, where data is structured rather than purely unstructured natural language (Liu et al., 2023). Second, the evaluation protocols used in KBQA benchmarks tend to be uniform, leading to an insufficient evaluation of model robustness (Gu et al., 2021). Finally, recent reviews in this field (Lan et al., 2022; Gu et al., 2022b) have largely overlooked the advancements in the development and application of LMs, particularly LLMs.

[2]Schema denotes rdfs:Class (class) and rdf:Property (relation) here.

Benchmark	KB	Size	LF	Generalization	Paraphrases	Metrics
WebQuestions (Berant et al., 2013)	Freebase	5,810	N/A	i.i.d.	✗	F1
SimpleQuestions (Bordes et al., 2015)	Freebase	108,442	N/A	i.i.d.	✗	Acc
WebQuestionsSP (Yih et al., 2016)	Freebase	4,737	SPARQL	i.i.d.	✗	F1
GraphQuestions (Su et al., 2016)	Freebase	5,166	Graph query	comp.+zero	✓	F1
LC-QuAD (Trivedi et al., 2017)	DBpedia	5,000	SPARQL	i.i.d.	✗	F1
CWQ (Talmor and Berant, 2018)	Freebase	34,689	SPARQL	i.i.d.	✗	Hits
LC-QuAD 2.0 (Dubey et al., 2019)	Wikidata	30,000	SPARQL	i.i.d.	✓	F1
SQB (Wu et al., 2019)	Freebase	108,443	N/A	i.i.d.+zero	✗	Acc
CFQ (Keysers et al., 2020)	Freebase	239,357	SPARQL	comp	✗	Acc
GrailQA (Gu et al., 2021)	Freebase	64,331	S-expression	i.i.d.+comp.+zero	✓	EM, F1
KQA Pro (Cao et al., 2022a)	Wikidata	117,970	KoPL	i.i.d.	✗	Acc
QALD series (Perevalov et al., 2022)	DBpedia	558	SPARQL	comp.	✗	F1

Table 1: Selected KBQA benchmarks. LF denotes logical forms. Generalization settings follow Gu et al. (2021). *i.i.d.* denotes that the schema distribution in the test set is the same as the training set. *comp.* and *zero* denote compositional and zero-shot generalization, respectively. Paraphrases are questions containing the same semantics (machine-generated paraphrases are not included). *Acc* denotes accuracy. *EM* denotes exact match.

3 Challenges from Distribution Shifts

In this paper, the **robustness** of a KBQA model refers to its ability to adapt to various natural language inputs and maintain consistent performance when data distribution shifts. Due to the distribution shifts between the training corpus of LMs and KB environments, grounding LMs face robustness challenges from environmental, linguistic, and modal aspects.

3.1 Environmental Aspect

A primary environmental challenge is **schema-level generalization**. The RDF Schema provides a data-modeling vocabulary crucial for querying a KB. Table 1 indicates most KBQA benchmarks assume a consistent schema distribution between training and testing. However, this often does not hold in large KBs with numerous schema items. Few benchmarks, like the reconfigured SimpleQuestions-Balance dataset (Wu et al., 2019) and GrailQA (Gu et al., 2021), address non-i.i.d. schema items and varying levels of schema-level generalization. GraphQuestions (Su et al., 2016) provides a stringent test with seldom-seen schema items in training. Given real-world non-i.i.d. complexities, these datasets better represent the practical generalization challenges. Yet, despite progress (Shu et al., 2022; Gu et al., 2022a), compositional and zero-shot generalization are far from solved. We stress that this challenge applies to the overall KBQA semantic parsing process, especially to important subtasks such as relation linking.

3.2 Linguistic Aspect

Various natural language expressions make question understanding challenging for KBQA models. One common way this variety shows up is through paraphrasing. In this paper, a paraphrase set denotes different ways to express the same logical form, as illustrated in Table 14. **Paraphrase adaptation** is an intuitive form of the ability to comprehend variable language expressions. It could be measured by whether a model is able to accurately answer paraphrased questions that this model has already answered correctly before. Unfortunately, as shown in Table 1, many KBQA benchmarks do not account for paraphrasing with only one utterance for each logical form. Exceptionally, some datasets (Su et al., 2016; Dubey et al., 2019; Gu et al., 2021) are based on automatically generated logical forms and include multiple natural language expressions for the same logical form (template). These data characteristics highlight the difficulties in adapting to paraphrased questions.

3.3 Integrated Aspect

Evaluating KBQA benchmarks often hinges on a single dataset, thereby complicating the task of ascertaining the model performance consistency across novel scenarios. This form of robustness, termed as **cross-dataset transfer** in this paper, combines both the environmental and linguistic aspects discussed earlier and is more difficult to achieve. This is because construction methods vary across datasets, as do schema distributions and natural language expressions. Specifically, KBQA dataset construction generally falls into two dis-

tinct categories: 1) Graph Search and Crowdsourcing: in this approach, logical forms or triples are initially extracted from a KB, where structures or operators of logical form are usually finite. Subsequently, they are converted into natural language utterances through crowdsourcing methods (Bordes et al., 2015; Trivedi et al., 2017). 2) Human Curation and Parsing: logical forms are labeled directly from human-provided utterances (Berant et al., 2013; Perevalov et al., 2022). Existing works (Gu et al., 2021; Cao et al., 2022b) suggest that models pre-trained on large-scale datasets can adapt reasonably well to other target datasets, such as WebQSP (Yih et al., 2016). However, the necessity for fine-tuning these pre-trained models on the intended target dataset remains imperative for achieving optimal performance. Despite the advantages offered by pre-training on expansive KBQA datasets, models still encounter challenges in transferring directly to previously unseen target datasets while sustaining high performance.

3.4 Modal Aspect

Aside from considering environmental and linguistic factors, focusing on the modal aspect is also crucial. Recently, LLMs like GPT series (OpenAI, 2023) have demonstrated exceptional capabilities across a variety of tasks, outperforming smaller yet potent LMs such as BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020). Despite these advancements, these LLMs face substantial challenges when interacting with environments. One notable issue is they predominantly rely on an in-context learning paradigm as opposed to fine-tuning, as a trade-off between computational cost and model efficiency. In comparison to fine-tuning, in-context learning offers the advantage of reduced training costs but at the expense of being forced to reason over the **unfamiliar modality**. Distribution shifts between natural language pre-training and reasoning over structured knowledge contexts could lead to poor performance. For instance, a discernible performance gap exists between KBQA models that employ in-context learning with Codex (Chen et al., 2021a) and those built on fine-tuned LMs (Gu et al., 2022a; Li et al., 2023). However, the empirical specifics of this difference are not yet clear, leaving us with an inadequate understanding of the limitations of in-context learning and ways to improve grounding with LLMs.

4 Evaluation Protocols

Regarding these challenges, we introduce extensive protocols for evaluating LMs in several aspects overlooked by current benchmarks.

4.1 Evaluating Environmental Aspect

To set the environmental schema-level generalization scenario, we use GrailQA (Gu et al., 2021) and GraphQuestions (Su et al., 2016) datasets. GrailQA contains three generalization levels: i.i.d. (25%), compositional (25%), and zero-shot (50%). GraphQuestions has no seen relations in the test set. We also use SimpleQuestions-Balance (SQB) (Wu et al., 2019) for the relation linking task (an important KBQA sub-task), where 50% of the samples contain unseen relations.

4.2 Evaluating Linguistic Aspect

To set a paraphrase adaptation scenario, we use GrailQA (Gu et al., 2021) and GraphQuestions (Su et al., 2016) datasets. To evaluate adaptability to paraphrases (§3.2), we propose a new metric, the standard deviation (std) of EM/F1 for questions of each logical form template. As shown in Equation 1, suppose there are n sets of paraphrases in the dataset, each set of paraphrases corresponds to a logical form template with m natural language expressions, and the F1 score obtained by the KBQA model on the j-th question of the i-th set of paraphrases is $F1_{i,j}$. The metric Std_{F1} first calculates the standard deviation of the F1 scores obtained by the model on the m questions for each set of paraphrases and then calculates the average of the n standard deviations. This metric is used to measure the robustness of the model to different representations of the same semantics, i.e., whether it can cope with diverse natural language expressions. A lower standard deviation indicates that the model is more adaptive to different expressions. Std_{EM} is calculated in the same way.

$$Std_{F1} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(\frac{\sum_{j=1}^{m}(F1_{i,j} - \bar{F1}_i)^2}{m}\right)} \quad (1)$$

4.3 Evaluating Integrated Aspect

To emulate a real-world scenario with unknown schema and linguistic distribution for the integrated aspect, we evaluate the performance of pre-trained models on the unseen human-curated WebQSP

(Yih et al., 2016) dataset, where the questions are derived from search logs and are more realistic. This is a scenario where the distribution changes significantly, as most benchmarks create questions by sampling logical forms and annotating natural language questions via crowdsourcing (Lan et al., 2022), where questions are confined to sampled logical forms.

4.4 Evaluating Modal Aspect

To test the capability of the in-context learning paradigm to inference from the KB modality rather than from pure texts, we retrieve structured KB contexts as prompt to evaluate the LLM without particular fine-tuning on KB (§5.1.2). KBs are structured and expansive, but in this case, LLM can only encode a portion of linearized KB contexts.

5 Experiments

5.1 Augmentation Approach

To ensure fair evaluation and fully harness the capabilities of LMs under our extensive and rigorous evaluation protocols, we suggest two strategies to counteract distribution shifts: data augmentation and retrieval augmentation.

5.1.1 Data Augmentation for LMs

Off-the-shelf datasets of limited size may make LM easily overfitted and not adaptable to large KBs. To address the problem that many domains in the KB are often not collected as training data, we propose a data augmentation method named **G**raph se**A**rch and quest**I**on generatio**N** (**GAIN**). Some data augmentation or question generation models (Bi et al., 2020; Guo et al., 2022) are only evaluated by the qualify of generated sentences rather than evaluated by the QA task, but GAIN directly serves our KBQA evaluations. Besides, compared to the previous work (Hu et al., 2019) that only considers generating questions for triples to help KBQA, GAIN applies to KBQA corresponding to both logical forms and triples. GAIN scales data volume and distribution through four steps: 1) Graph search: Sampling logical forms or triples from arbitrary domains in the KB without being restricted to any particular KBQA dataset. 2) Training question generator: learning to convert logical forms or triples into natural language questions on existing KBQA datasets. 3) Verbalization: Using the question generator from step 2 to verbalize sampled logical forms or triples from step 1, thus creating synthetic questions. 4) Training data expansion: Before fine-tuning any neural models on KBQA datasets, GAIN-synthetic data can be used to train these models or to expand the corpus of in-context samples for LLMs. That is, as a data augmentation method, GAIN is not a KBQA model, but it is used to augment a base KBQA model.

5.1.2 Retrieval Augmentation for LLMs

As the trade-off between cost and effectiveness, we experiment with the prevalent in-context learning paradigm but attempt to improve the quality of in-context samples. We use advanced retrieval methods based on smaller LMs as plug-ins to augment the LLM, similar to the SuperICL approach (Xu et al., 2023). Specifically, our steps to generate an LLM prompt for each question include the following. 1) Given an input question, we retrieve k questions (k-shot) with BM25 (Robertson et al., 2009) from the corpus (the combination of KBQA training set and the GAIN-synthetic dataset). 2) The role of retrieval augmentation for KB environments has been shown by fine-tuned LMs (Shu et al., 2022). To assist with grounding LLM, we retrieve KB contexts with off-the-shelf retrievers for k samples and the input question.[3]

5.2 Setup

Data All experiments use S-expression (Gu et al., 2021) as the logical form due to its clear and concise structure. Entity linking results are taken from TIARA (Shu et al., 2022) for GrailQA and WebQSP, and ArcaneQA (Gu and Su, 2022) for GraphQuestions, because of their public availability and performance.

Model Compared models are mainly selected from the leaderboard.[4] The performances are taken from their papers. For the relation linking task on SQB, we use BERT (Devlin et al., 2019) as the base model for GAIN. For KBQA tasks, we use the open-source advanced model TIARA (Shu et al., 2022) as the base model for GAIN, due to its strong performance on zero-shot schema items.[5] TIARA is composed of multi-grained retrievers and a generator, with the retrievers providing KB contexts[6] for the generator. The term "TIARA+GAIN" represents a model (both the retrievers and the generator)

[3]The prompt example is demonstrated in Appendix A.

[4]https://dki-lab.github.io/GrailQA/

[5]Pangu (Gu et al., 2022a) also uses entity linking results from TIARA.

[6]Entities, exemplary logical forms, and schema items are retrieved.

that is first tuned using GAIN synthetic data and subsequently fine-tuned on a target dataset. For LLM evaluation in the modal aspect, we use the `gpt-3.5-turbo-0613`[7] model, and the few-shot contexts are retrieved from the combination of GrailQA training set and synthetic dataset using the TIARA+GAIN retrievers.

Metrics Following previous works, we use Exact Match (EM), F1, and Hits@1 to measure the performance of KBQA models. We also use the std of EM/F1 to measure the adaptability to paraphrases (§4.2).

5.3 Implementation Details

We use a machine with an NVIDIA A100 GPU and up to 504GB of RAM. Models are implemented by PyTorch (Paszke et al., 2019) and Hugging Face.[8] TIARA+GAIN (T5-3B) takes about 100 hours to train the logical form generator on the synthetic dataset.

Model Training 1) For question generation, we fine-tune the T5-base model (Raffel et al., 2020) to convert S-expression or triple to natural language questions. We set the beam size to 10, the learning rate to 3e-5, the number of epochs to 10, and the batch size to 8. 2) The training of the TIARA model (Shu et al., 2022) follows its original settings, including the setting of hyperparameters and the calculation of metrics. Note that Hits@1 on TIARA is obtained by randomly selecting one answer for each question 100 times. Both the schema retriever and generator of TIARA are pre-trained on synthetic data and then fine-tuned on KBQA datasets. Since GraphQuestions has no official training-valid split, we randomly take 200 questions from the original training set as the valid set. 3) We use BERT-base-uncased (Devlin et al., 2019) to rank candidate relations for SQB, and the input form is the same as the schema retriever of TIARA. We set the learning rate to 3e-5, the batch size to 256, and the max number of epochs to 3 with early stopping.

Data Augmentation The statistics of GAIN-synthetic datasets for both logical forms and triples are shown in Table 11 and 12.[9] Note that the sampling of the GAIN method is not limited to the scale of the synthetic data we use here.

[7] https://platform.openai.com/docs/models

[8] https://huggingface.co/

[9] Details of synthetic data are shown in Appendix B.

6 Analysis

We report and analyze the experimental results in this section for each aspect.

6.1 Analysis of Environmental Aspect

Effectiveness of Synthesis and Scaling Up As shown in Tables 2 and 3, the models perform significantly better on i.i.d. than compositional and zero-shot generalization, with the zero-shot partition being the most challenging. TIARA+GAIN (T5-base) improves 2.5 zero-shot F1 points compared to TIARA (T5-base). Besides, an increased number of model parameters, combined with richer data from GAIN, significantly enhance the generalization capabilities of T5 models. TIARA+GAIN (T5-3B) further improves 1.4 zero-shot F1 points compared to its T5-base version. TIARA+GAIN achieves the highest EM scores, including that on zero-shot scenes. It demonstrates promising ideas for further improving LM generalization capabilities, i.e., the positive effect of synthetic data and parametric scales on training LMs.

Fine-tuning Better Than Few-shot Learning However, it is important to note that fine-tuned models consistently outperform few-shot learning models, regardless of whether the schema is seen or not. Given the training and inference costs of LLMs, their performance has yet to show any superiority in this task.

6.2 Analysis of Linguistic Aspect

Improvements Are Linguistic Biased We calculate the standard deviation (std) of EM or F1 in the dev/test set, as shown in Equation 1. For GrailQA, the std of EM and F1 decreases with the application of GAIN or an increase in model size, i.e., F1/EM and the std of F1/EM are both better, as shown in Table 2 and 5. However, in the case of more challenging GraphQuestions, GAIN significantly improves the F1 by 8.3 points but also results in a larger std (0.170 compared to 0.157), as shown in Table 3. It suggests that improving paraphrase adaptation using GAIN is more difficult when the base model (TIARA, T5-base, with only 37.9% F1) still struggles to address most of the dataset. Consequently, the performance gains observed on the KBQA benchmark may not necessarily reflect a deeper understanding of linguistic complexities, but they could simply render the model more sensitive to specific phrases. Strategies for deeper decomposition and understanding (Hu et al., 2021;

Model on GrailQA Test Set	Overall		I.I.D.		Compositional		Zero-shot	
	EM	F1	EM	F1	EM	F1	EM	F1
Fine-tuned Models								
BERT + Ranking (Gu et al., 2021)	50.6	58.0	59.9	67.0	45.5	53.9	48.6	55.7
RnG-KBQA (Ye et al., 2022)	68.8	74.4	86.2	89.0	63.8	71.2	63.0	69.2
TIARA (T5-base) (Shu et al., 2022)	73.0	78.5	87.8	90.6	69.2	76.5	68.0	73.9
DecAF (FiD-3B) (Yu et al., 2022)	68.4	78.8	84.8	89.9	73.4	81.8	58.6	72.3
Pangu (BERT-base) (Gu et al., 2022a)	73.7	79.9	82.6	87.1	74.9	81.2	69.1	76.1
Pangu (T5-large) (Gu et al., 2022a)	74.8	81.4	82.5	87.3	**75.2**	**82.2**	71.0	78.4
Pangu (T5-3B) (Gu et al., 2022a)	75.4	**81.7**	84.4	88.8	74.6	81.5	71.6	**78.5**
Codex-driven Models								
KB-BINDER (6)-R (Li et al., 2023)	53.2	58.5	72.5	77.4	51.8	58.3	45.0	49.9
Pangu (Codex) (Gu et al., 2022a)	56.4	65.0	67.5	73.7	58.2	64.9	50.7	61.1
GAIN-augmented Models								
TIARA + **GAIN** (T5-base)	75.1	80.6	88.3	91.0	73.0	79.6	69.9	76.4
TIARA + **GAIN** (T5-3B)	**76.3**	81.5	**88.5**	**91.2**	73.7	80.0	**71.8**	77.8
GPT-3.5-turbo (5-shot)	66.6	71.4	82.7	85.3	60.5	66.3	61.9	67.2

Table 2: EM and F1 scores (%) on the hidden test set of GrailQA.

Model on GraphQuestions	F1(↑)	Std(↓)
GraphQuestions on Freebase 2013-07		
UDepLambda (Reddy et al., 2017)	17.7	-
PARA4QA (Dong et al., 2017)	20.4	-
SPARQA (Sun et al., 2020)	21.5	-
BERT + Ranking (Gu et al., 2021)	25.0	-
ArcaneQA (Gu and Su, 2022)	31.8	-
TIARA♣ (T5-base) (Shu et al., 2022)	37.9	**0.141**
KB-BINDER (6) (Li et al., 2023)	39.5	-
TIARA + **GAIN** (T5-base)	45.5	0.153
TIARA + **GAIN** (T5-3B)	**48.7**	0.180
GraphQuestions on Freebase 2015-08-09		
BERT + Ranking (Gu et al., 2021)	27.0	-
ArcaneQA (Gu and Su, 2022)	34.3	-
TIARA♣ (T5-base) (Shu et al., 2022)	41.2	**0.157**
Pangu (Codex) (Gu et al., 2022a)	44.3	-
Pangu (T5-3B) (Gu et al., 2022a)	**62.2**	-
TIARA + **GAIN** (T5-base)	49.5	0.170
TIARA + **GAIN** (T5-3B)	53.0	0.200

Table 3: F1 scores (%) and average standard deviation (std) of F1 scores for each set of paraphrases on the test set of GraphQuestions. The setting for Freebase 2015-08-09 is described by Gu and Su (2022). ♣ denotes our replication results.

Huang et al., 2023) of the questions may be needed to mitigate this challenge.

6.3 Analysis of Integrated Aspect

Hard Transfer Across Datasets We evaluate the performance of pre-trained models on the human-curated WebQSP dataset without fine-tuning, as shown in Table 6. BERT+Ranking (Gu et al., 2021) and TIARA+GAIN (Shu et al., 2022) are trained on the large-scale GrailQA dataset. We compare these results to the state-of-the-art Pangu (Gu et al., 2022a), which is fine-tuned on WebQSP and achieves an F1 score of 79.6% Although we recognize that GAIN and large models offer few advantages, the performance of these pre-trained models without fine-tuning is considerably lower than Pangu's.

Causes from Data Collection We attribute this to the significant differences between training and test data, as shown in Table 8. The question length, the difficulty of entity/relation linking[10], and the proportion of unseen schema vary dramatically across KBQA datasets. These discrepancies arise from the dataset construction process: WebQSP is an annotation of search logs, whereas the remaining datasets are derived from graph search and crowdsourcing. To further enhance robustness in cross-dataset transfer, we believe that better data collection methods are required to obtain diverse and balanced training data. Additionally, the representation of the logical form increases the transfer difficulty, as the S-expression used in the GrailQA dataset cannot express all queries in WebQSP.

6.4 Analysis of Modal Aspect

Context Alone Is Insufficient We evaluate the performance of GPT-3.5 using retrieved KB contexts (§5.1.2) and in-context learning on the GrailQA dataset. The prompts for the model include the task description and the few-shot KB contexts. As illustrated in Table 7, when provided with contexts from the TIARA+GAIN retrievers, GPT-3.5 outperforms two compared models but

[10]Measured by literal similarity: https://anhaidgroup.github.io/py_stringmatching/v0.3.x/PartialRatio.

Model on SimpleQuestions-Balance	Overall			Seen			Unseen		
	1	5	10	1	5	10	1	5	10
HR-BiLSTM (Wu et al., 2019)	63.3	-	-	**93.5**	-	-	33.0	-	-
Adversarial-Adapter (Wu et al., 2019)	84.9	-	-	92.6	-	-	77.1	-	-
BERT-base	83.7	95.0	96.9	85.8	95.0	96.0	81.5	95.1	97.8
BERT-base + **GAIN**	**88.4**	**96.0**	**97.3**	87.8	95.4	96.3	**89.1**	**96.7**	**98.4**

Table 4: Hits@k (1, 5, 10) scores (%) for relation linking on the test set of SimpleQuestions-Balance, including seen and unseen relations.

Model on GrailQA Valid Set	Std Overall		Std I.I.D.		Std Compositional		Std Zero-shot	
	EM(↓)	F1(↓)	EM	F1	EM	F1	EM	F1
TIARA (T5-base) (Shu et al., 2022)	0.079	0.066	0.021	0.017	0.211	0.203	0.222	0.181
TIARA + **GAIN** (T5-base)	0.077	0.061	0.020	0.016	0.215	0.198	0.218	0.160
TIARA + **GAIN** (T5-3B)	**0.075**	**0.058**	**0.020**	**0.016**	**0.196**	**0.180**	**0.212**	**0.155**
GPT-3.5-turbo (5-shot)	0.093	0.091	0.027	0.023	0.272	0.281	0.251	0.247

Table 5: Average **standard deviation** of EM and F1 scores for each set of paraphrases on the GrailQA valid set.

Model on WebQSP	F1	Hits@1
TIARA♣ (T5-base) (Shu et al., 2022)	28.5	27.6
TIARA*♣ (T5-base) (Shu et al., 2022)	33.5	31.5
BERT + Ranking* (Gu et al., 2021)	**43.0**	-
TIARA + **GAIN** (T5-base)	29.1	28.2
TIARA + **GAIN** (T5-3B)	29.8	28.7
TIARA* + **GAIN** (T5-base)	33.9	31.8
TIARA* + **GAIN** (T5-3B)	34.5	32.3

Table 6: F1 and Hits@1 scores (%) on WebQSP without fine-tuning on it. All models are trained on large-scale GrailQA. * denotes using oracle entity annotations. ♣ denotes our replication results.

falls short compared to TIARA+GAIN. Among the GPT-3.5 predictions, 79.62% come directly from the substring of the corresponding prompts, achieving an average F1 score of 86.19% for this portion. However, the remaining predictions are not part of their prompts and are entirely new predictions generated by GPT-3.5, with an average F1 score of merely 30.29%. Although a baseline level is attained, these results suggest that GPT-3.5 cannot be accurately grounded to the KB environment when it does not copy the retrievers' contexts. It also shows the modal severance of natural language pre-training and KB contexts for the LLM. LLMs are not able to utilize KB contexts as proficiently as they understand natural language, and the faithfulness and controllability of grounded LLMs are not yet guaranteed under the current approach (Gu et al., 2022a). To mitigate this problem, alternative paradigms should be explored, such as tool learning (Schick et al., 2023) and multi-step planning (Liu et al., 2023) with Chain-of-Thought (Wei et al., 2022), which enables more refined access and control over environments and reduces modal differences during the reasoning process.

7 Conclusion

Despite the recent progress of LM-driven models, robustness challenges posed by distribution shifts for the KBQA task are rarely discussed. Our analyses call for further research into better evaluation protocols for grounding LMs to KBs and enhancing the robustness of environmental, linguistic, and modal aspects. Notably, the experiments reveal that LLMs sometimes simply copy the provided prompt. It indicates that the existing methodologies for grounding LLMs are yet to prove their efficacy and superiority. Future research issues include collecting more balanced environment-specific corpora and improving the LLM learning paradigms. For the corpora, our experiments show that the data augmentation techniques deserve further research.

Limitations

1) For question generation, the verbalization process of the GAIN method relies heavily on large-scale KBQA annotations. The training data influence the style of generated questions, and overly complex logical forms (e.g., with three or more hops) are difficult to convert into natural language questions. Besides, synthetic data is less diverse and natural than human annotations, though it improves generalization performance. 2) Multilingual KBQA is a problem that lacks attention in

Model on GrailQA Valid Set	Overall		I.I.D.		Compositional		Zero-shot	
	EM	F1	EM	F1	EM	F1	EM	F1
BERT + Ranking (Gu et al., 2021)	51.0	58.4	58.6	66.1	40.9	48.1	51.8	59.2
TIARA ELF only (Shu et al., 2022)	67.2	72.9	72.8	76.7	55.3	60.7	69.7	76.3
RnG-KBQA (Ye et al., 2022)	71.4	76.8	86.7	89.0	61.7	68.9	68.8	74.7
DecAF (FiD-3B) (Yu et al., 2022)	-	81.4	-	89.7	-	**80.1**	-	78.4
TIARA (T5-base) (Shu et al., 2022)	75.3	81.9	88.4	91.2	66.4	74.8	73.3	80.7
Pangu (T5-3B) (Gu et al., 2022a)	75.8	83.4	-	-	-	-	-	-
TIARA + **GAIN** (T5-base)	77.1	83.5	89.0	91.9	68.6	75.5	75.4	83.2
TIARA + **GAIN** (T5-3B)	**77.1**	**83.8**	**89.0**	**92.1**	**68.8**	76.1	**75.4**	**83.4**
GPT-3.5-turbo (5-shot)	69.7	74.8	83.0	85.5	58.7	64.6	68.6	74.4

Table 7: EM and F1 scores (%) on the GrailQA valid set. ELF denotes exemplary logical form (Shu et al., 2022).

	GrailQA	GraphQ	WebQSP	SQB
Train size	44,337	2,381	3,097	75,819
Valid size	6,763	-	-	11,141
Test size	13,231	2,395	1,638	21,483
Length	62.96	54.62	35.93	42.16
# of entities	0.903	1.028	1.112	1.000
# of relations	1.358	1.434	1.464	1.000
Similarity between questions and KB items				
Entity	0.999	1.000	0.921	0.985
Class	0.547	0.457	-	-
Relation	0.470	0.389	0.300	0.779
Unseen ratio (%)				
Schema	16.90	86.78	20.44	32.67
Question	54.06	98.25	4.03	49.18

Table 8: KBQA dataset statistics. *Length* denotes the average number of question characters. *# of entities/relations* denotes the average number of entities/relations in the logical form. *Unseen Schema* is the ratio of unseen schema items in the dev/test set. *Unseen Question* is the ratio of questions containing unseen schema.

the KBQA research and is also a linguistic-aspect challenge. However, since most KBQA datasets are based on English and do not discuss other languages at all, this paper leaves the evaluation of this problem for future work.

Ethics Statement

The proposed data augmentation method GAIN could be used on any KB. The Freebase (Bollacker et al., 2008) used in this work is a KB that has been publicly released and manually reviewed. For uncensored KBs, if harmful information is collected, it could make synthetic data contain harmful information and make LMs generate harmful answers.

Acknowledgments

We extend our sincere gratitude to all the anonymous reviewers for their insightful suggestions. Special thanks to Xiao Xu, Xiang Huang, and Sitao Cheng for their valuable discussions on this paper. We appreciate Yu Gu's efforts in evaluating our submissions on the GrailQA benchmark test set.

References

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.

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Yongzhen Wang, and Guilin Qi. 2020. Knowledge-enriched, type-constrained and grammar-guided question generation over knowledge bases. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2776–2786. International Committee on Computational Linguistics.

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.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. 2022a. KQA Pro: A dataset with explicit compositional programs for complex question answering over knowledge base. 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 6101–6119. Association for Computational Linguistics.

Shulin Cao, Jiaxin Shi, Zijun Yao, Xin Lv, Jifan Yu, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jinghui Xiao. 2022b. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8128–8140. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 2021b. ReTraCk: A flexible and efficient framework for knowledge base question answering. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL 2021 - System Demonstrations, Online, August 1-6, 2021*, pages 325–336. Association for Computational Linguistics.

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

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 875–886. Association for Computational Linguistics.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. LC-QuAD 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 69–78. Springer.

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

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

Yu Gu, Vardaan Pahuja, Gong Cheng, and Yu Su. 2022b. Knowledge base question answering: A semantic parsing perspective. *arXiv preprint arXiv:2209.04994*.

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.

Shasha Guo, Jing Zhang, Yanling Wang, Qianyi Zhang, Cuiping Li, and Hong Chen. 2022. DSM: Question generation over knowledge base via modeling diverse subgraphs with meta-learner.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. 2020. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329.

Sen Hu, Lei Zou, and Zhanxing Zhu. 2019. How question generation can help question answering over knowledge base. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 80–92. Springer.

Xixin Hu, Yiheng Shu, Xiang Huang, and Yuzhong Qu. 2021. EDG-based question decomposition for complex question answering over knowledge bases. In *Proceedings of the ISWC 2021*, volume 12922 of *Lecture Notes in Computer Science*, pages 128–145. Springer.

Xixin Hu, Xuan Wu, Yiheng Shu, and Yuzhong Qu. 2022. Logical form generation via multi-task learning for complex question answering over knowledge

bases. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1687–1696, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiang Huang, Sitao Cheng, Yiheng Shu, Yuheng Bao, and Yuzhong Qu. 2023. Question decomposition tree for answering complex questions over knowledge bases. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 12924–12932. AAAI Press.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2022. State-of-the-art generalisation research in NLP: a taxonomy and review. *CoRR*, abs/2210.03050.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. 2020. Measuring compositional generalization: A comprehensive method on realistic data. In *ICLR 2020*. OpenReview.net.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20.

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

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Yuxian Gu, Hangliang Ding, Kai Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Shengqi Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint 2308.03688*.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Arkil Patel, Satwik Bhattamishra, Phil Blunsom, and Navin Goyal. 2022. Revisiting the compositional generalization abilities of neural sequence models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 424–434. Association for Computational Linguistics.

Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 2022. QALD-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In *16th IEEE International Conference on Semantic Computing, ICSC 2022, Laguna Hills, CA, USA, January 26-28, 2022*, pages 229–234. IEEE.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark. Association for Computational Linguistics.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer:

Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

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.

Yu Su. 2023. Language agents: a critical evolutionary step of artificial intelligence. *yusu.substack.com*.

Yu Su, Huan Sun, Brian M. Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for QA evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 562–572. The Association for Computational Linguistics.

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

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

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu 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. *arXiv preprint 2307.09288*.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QuAD: A corpus for complex question answering over knowledge graphs. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, volume 10588 of *Lecture Notes in Computer Science*, pages 210–218. Springer.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Peng Wu, Shujian Huang, Rongxiang Weng, Zaixiang Zheng, Jianbing Zhang, Xiaohui Yan, and Jiajun Chen. 2019. Learning representation mapping for relation detection in knowledge base question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6130–6139. Association for Computational Linguistics.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2023. Small models are valuable plug-ins for large language models. *arXiv preprint arXiv:2305.08848*.

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, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*. The Association for Computer Linguistics.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Wang, Zhiguo Wang, and Bing Xiang. 2022. DecAF: Joint decoding of answers and logical forms for question answering over knowledge bases. *arXiv preprint arXiv:2210.00063*.

Model of Class Retrieval	All	I.I.D.	Comp.	Zero.
ReTraCk (Chen et al., 2021b)	94.3	98.1	97.5	91.3
TIARA (Shu et al., 2022)	95.8	**99.6**	97.9	93.4
TIARA + **GAIN**	**96.1**	99.6	**98.1**	**93.8**
Model of Relation Retrieval	**All**	**I.I.D.**	**Comp.**	**Zero.**
ReTraCk (Chen et al., 2021b)	88.4	95.3	91.0	84.3
TIARA (Shu et al., 2022)	92.0	97.9	93.7	88.7
TIARA + **GAIN**	**93.0**	**99.2**	**94.1**	**89.8**

Table 9: Recall (%) of top-10 retrieved schema items on the GrailQA valid set. *comp.* and *zero.* denote compositional and zero-shot generalization, respectively. Note that ReTraCk uses 100 classes and 150 relations for each question, while TIARA uses 10 classes and 10 relations.

Model on GraphQuestions	Class	Relation
TIARA♣ (Shu et al., 2022)	81.5	67.1
TIARA + **GAIN**	**83.3**	**74.3**

Table 10: Recall (%) of top-10 retrieved schema items on the GraphQuestions test set (Freebase 2015-08-09 version). TIARA uses 10 classes and 10 relations for each question. ♣ denotes our replication results.

A Example of LLM Prompt

We present an example of an excerpted prompt, which is shown in Figures 2 and 3. In particular, Figure 2 illustrates the task instruction and teaching example segment, while Figure 3 displays the input query segment.

B Details of Synthetic Data

The graph search process of GAIN is slightly different for logical forms and triples.

Searching Logical Forms GAIN employs a graph search approach similar to GraphQuestions (Su et al., 2016) to collect logical forms derived from graph queries. The graph query construction process consists of four steps: 1) query template construction, 2) aligning template nodes, 3) configuring functions, and 4) verification by execution. Query templates, obtained through random graph searching, contain nodes that represent entity/literal types (not aligned to a value), referred to as *template nodes*. Each unaligned node in the query template is then aligned with a topic entity or literal to generate multiple aligned graph queries. To synthesize counting, comparative, and superlative questions and enhance the diversity of synthetic data, we add functions like `COUNT`, `ARGMIN/ARGMAX` (Gu et al., 2021) to graph queries. Since KBQA research typically assumes that questions can be answered by the KB, we execute all resulting graph queries after the above steps and remove those with a null[11] result.

#question	#one-hop	#two-hop	#domain
127,329	78,668	48,661	759
#none	**#count**	**#comparatives**	**#superlatives**
115,221	7,115	1,874	3,119
#class	**#relation**	**#entity**	
5,078	12,942	46,645	

Table 11: Statistics for the synthetic dataset of logical forms. *none* denotes no function.

#question	#relation	#subject	#domain
162,557	7,349	108,804	673

Table 12: Statistics for the synthetic dataset of triples. *Subject* denotes subject entities.

Searching Triples A single KB triple can be treated as a QA pair, where the head entity and relation together form the query, and the tail entity is the answer. The triple search process consists of two steps: 1) candidate relation selection, and 2) triple sampling. First, arbitrary relations $\mathcal{R}$ are selected from the KB, without being restricted to any particular KBQA dataset. Then, triples are collected from head entities $\mathcal{H}$, where entities in $\mathcal{H}$ are connected to relations in $\mathcal{R}$.

Examples of Synthetic Data We present some examples of synthetic data in Table 15, where the logical form contains a variety of functions.

Statistics The statistics for KBQA datasets, including our synthetic dataset, are shown in Table 16. To calculate the number of domains in Table 11, 12, we take the domain of each Freebase class from its first segment, except for classes starting with "base", where we take the first two segments, e.g., domain "music" for the class "music.performance_venue", and domain "base.plants" for the class "base.plants.plant".

C Experimental Details

C.1 Question Generation

Training a question generation (QG) model is the second step of GAIN. Because logical forms in the

[11] `Null` for querying entities and `zero` for counting.

Metrics	GrailQA Dev	GraphQ Test	SQB Dev
BLEU-4	0.347	0.178	0.369
ROUGE-L	0.526	0.411	0.640

Table 13: The performance of the question generator on KBQA datasets. The generator for logical form is evaluated on the GrailQA valid set and GraphQuestions test set. The generator for triple is evaluated on the SQB valid set.

synthetic dataset do not have corresponding human-labeled natural language questions, evaluating QG on the synthetic dataset is difficult. Existing KBQA datasets contain questions written by crowdsourced workers for logical forms or triples, so we evaluate the question generator by generating questions on these datasets and calculate the BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin and Och, 2004) scores (Table 13).[12]

C.2 Retrieval Performance

The performance of schema retrieval on the GrailQA valid set and GraphQuestions test set is shown in Table 9 and 10, respectively. GAIN improves the performance of schema retrieval on both GrailQA and GraphQuestions. In particular, GAIN improves the relation recall@10 by 7.2 points on GraphQuestions.

C.3 Performance on Various Logical Forms

To show how TIARA+GAIN performs on different types of logical forms, we compare it with previous KBQA models on the GrailQA valid set, as shown in Table 17. TIARA+GAIN improves performance in nearly all these scenarios compared to TIARA.

C.4 Error Analysis

To analyze the QA errors of TIARA+GAIN (T5-3B), we randomly sample 50 questions where predicted logical forms are not the same as the ground truth in the GrailQA valid set. We follow Shu et al. (2022) in their classification of errors. Entity linking errors (missing or redundant entities), syntactic errors (generating the wrong logical form structure), semantic errors (generating the wrong schema item when correct contexts are provided), false negatives (flawed annotations), and miscellaneous (e.g., ambiguity in the question) account for 48%, 26%, 16%, 4%, and 6%, respectively. For entity linking errors, 62.5% of them are from the zero-shot level. For syntactic errors, the number of errors from zero-shot and compositional levels is almost the same. It means that entity linking for zero-shot domains and logical form generation for complex structures remain significant challenges.

[12]Calculated by Hugging Face Evaluate.

D Details of Scientific Artifacts

All datasets we use are publicly available. GrailQA[13] (Gu et al., 2021) uses CC BY-SA 4.0 license, and GraphQuestions[14] (Su et al., 2016) uses CC BY 4.0 license. WebQSP[15] (Yih et al., 2016) and SimpleQuestions-Balance[16] (Wu et al., 2019) are also downloaded from their official release channels. We have complied with their distribution rules. These datasets involve manual construction rather than fully automated construction, which includes the review process. They contain questions about some famous individual people, but the corresponding content is available on Freebase, which is a publicly released, human-reviewed knowledge base.

Although the training data for LLMs could be quite large, the test set annotation of GrailQA is not publicly available. In addition, our experimental results on the test set and the validation set show the same trend, so the impact of the data contamination problem on the experiments of this paper could be ignored.

[13]https://dki-lab.github.io/GrailQA

[14]https://github.com/ysu1989/GraphQuestions

[15]https://www.microsoft.com/en-us/download/details.aspx?id=52763

[16]https://github.com/wudapeng268/KBQA-Adapter/tree/master/Data/SQB

Logical Form (S-expression)	Question
1. (AND book.journal (JOIN book.periodical.editorial_staff (AND (JOIN book.editorial_tenure.editor m.012z2ncg) (JOIN book.editorial_tenure.title m.02h6676)))) (GrailQA valid set)	1. john oliver la gorce was the editor on the editor for what journal?
2. (AND book.journal (JOIN book.periodical.editorial_staff (AND (JOIN book.editorial_tenure.editor m.05ws_t6) (JOIN book.editorial_tenure.title m.02wk2cy)))) (GrailQA valid set)	2. with which journal did don slater serve as editor on the editor in chief?
All four S-expressions are (COUNT (AND book.reviewed_work (JOIN book.reviewed_work.reviews_of_this_work m.0240y2))) (GraphQuestions training set)	1. how many works did fresh air review? 2. how many works were reviewed by fresh air in total? 3. what is the total amount of works reviewed by fresh air? 4. fresh air has reviewed how many different works?

Table 14: Examples of paraphrases in GrailQA and GraphQuestions.

Sampled Logical Form	Synthetic Question
(COUNT (AND people.profession (JOIN people.profession.people_with_this_profession m.012d40))) lentity|m.012d40 jackie chan	how many professions does jackie chan have?
(AND food.beer (le food.beer.original_gravity 1.067^^float))	which beer has an original gravity less than or equal to 1.067?
(AND medicine.manufactured_drug_form (AND (lt medicine.manufactured_drug_form.size 10.0^^float) (JOIN medicine.manufactured_drug_form.fda_otc_part m.0h9yt7z))) lentity|m.0h9yt7z fda otc monograph part 348	which manufactured drug form has part fda otc monograph part 348 and has a size smaller than 10.0?
(ARGMAX (AND measurement_unit.power_unit (JOIN measurement_unit.power_unit.measurement_system m.07y37)) measurement_unit.power_unit.power_in_watts) lentity|m.07y37 us customary units	what is the largest power unit in the us customary units?
(AND music.release (AND (JOIN music.release.engineers m.011mbx12) (JOIN music.release.label m.0g12fn3))) lentity|m.011mbx12 raynard glass |m.0g12fn3 hostyle gospel ministries	what musical release is engineered by raynard glass and labelled hostyle gospel ministries?
Sampled Triple	**Synthetic Question**
D.W. Robertson, Jr. (m.09ggymq), people.person.place_of_birth, Washington, D.C. (m.0rh6k)	where was D. W. Robertson, Jr. born
Alfred Chao (m.046cmd8), computer.operating_system_developer.operating_systems_developed, pSOS (m.0lscq)	what operating system did Alfred Chao develop?

Table 15: Examples of synthetic data. The logical form is S-expression (Gu et al., 2021). The entity label is appended to the logical form.

Datasets	#question	#class	#relation	#entity
GrailQA (Gu et al., 2021)	64,331	1,534	3,720	32,585
GraphQuestions (Su et al., 2016)	5,166	506	596	376
WebQSP (Yih et al., 2016)	4,737	408	661	2,593
GAIN-synthetic	127,329	5,078	12,942	46,645

Table 16: Statistics of KBQA datasets and the GAIN-synthetic dataset.

Function	**None**	**Count**	**Comparative**	**Superlative**
ArcaneQA (Gu and Su, 2022)	70.8/77.8	62.5/68.2	54.5/75.7	70.5/**75.6**
RnG-KBQA (Ye et al., 2022)	77.5/81.8	73.0/77.5	55.1/76.0	13.8/22.3
TIARA (T5-base) (Shu et al., 2022)	77.8/83.1	76.4/81.8	57.4/81.4	58.7/69.0
TIARA + **GAIN** ELF only	76.8/81.7	73.9/80.0	0.0/25.3	0.0/8.3
TIARA + **GAIN** (T5-base)	**78.6**/84.6	**77.7/83.0**	61.7/82.3	69.9/73.2
TIARA + **GAIN** (T5-3B)	78.5/**84.8**	77.3/82.5	**63.0/84.5**	**70.7**/74.1
GPT-3.5-turbo (5-shot)	74.1/78.0	66.8/70.5	38.3/60.5	43.9/52.3
# of relations	**1**	**2**	**3**	**4**
RnG-KBQA (Ye et al., 2022)	75.7/79.3	65.3/74.7	28.6/44.5	**100.0/100.0**
TIARA (T5-base) (Shu et al., 2022)	81.2/85.6	64.7/75.8	29.3/48.5	50.0/83.3
TIARA + **GAIN** ELF only	74.0/77.8	56.6/67.9	9.9/31.0	0.0/33.3
TIARA + **GAIN** (T5-base)	**82.4/87.2**	67.0/78.0	**38.9**/49.8	50.0/83.3
TIARA + **GAIN** (T5-3B)	82.0/87.2	**68.8/79.0**	37.5/**51.3**	50.0/83.3
GPT-3.5-turbo (5-shot)	75.0/78.6	61.9/69.6	19.8/36.6	50.0/50.0
# of entities	**0**	**1**	**2**	
RnG-KBQA (Ye et al., 2022)	58.5/63.6	75.4/79.9	**55.6/73.5**	
TIARA (T5-base) (Shu et al., 2022)	77.5/83.1	76.6/82.6	49.9/68.0	
TIARA + **GAIN** ELF only	42.8/47.0	74.2/79.9	47.4/67.6	
TIARA + **GAIN** (T5-base)	**82.2**/86.5	77.6/83.7	53.4/71.4	
TIARA + **GAIN** (T5-3B)	82.0/**86.6**	**77.6/84.0**	55.6/73.0	
GPT-3.5-turbo (5-shot)	66.8/72.8	71.9/76.1	48.8/62.1	

Table 17: EM and F1 scores (%) for different types of logical forms on the GrailQA valid set. *None* denotes no function. *# of relations/entities* denotes the number of relations/entities in the S-expression. ELF denotes exemplary logical form (Shu et al., 2022).

```
Given a question and Freebase contexts, write a logical form that answers the question.

Question: dark sun: wake of the ravager was designed by what video game designer?
Candidate entities:
(A) [Dark Sun: Wake of the Ravager]
Exemplary Logical Forms:
(A) (AND cvg.cvg_designer (JOIN cvg.cvg_designer.games_designed [Dark Sun: Wake of the Ravager]))
(B) (AND cvg.cvg_designer (JOIN (R cvg.computer_videogame.designers) [Dark Sun: Wake of the Ravager]))
(C) (AND cvg.computer_videogame (JOIN (R cvg.cvg_designer.games_designed) (JOIN
cvg.cvg_designer.games_designed [Dark Sun: Wake of the Ravager])))
(D) (AND cvg.computer_videogame (JOIN cvg.computer_videogame.designers (JOIN
cvg.cvg_designer.games_designed [Dark Sun: Wake of the Ravager])))
(E) (AND base.wikipedia_infobox.video_game (JOIN base.wikipedia_infobox.video_game.developer (JOIN
cvg.cvg_designer.games_designed [Dark Sun: Wake of the Ravager])))
Candidate classes:
(A) cvg.cvg_designer
(B) cvg.game_performance
(C) cvg.musical_game
(D) cvg.game_character
(E) cvg.computer_game_engine_developer
(F) cvg.computer_videogame
(G) cvg.computer_game_performance_type
(H) cvg.game_version
(I) cvg.computer_game_subject
(J) cvg.computer_game_evaluation
Candidate relations:
(A) cvg.cvg_designer.games_designed
(B) cvg.computer_videogame.designers
(C) cvg.computer_videogame.prequel
(D) cvg.computer_videogame.sequel
(E) cvg.computer_videogame.mods
(F) cvg.computer_videogame.expansions
(G) cvg.computer_videogame.developer
(H) cvg.computer_videogame.characters
(I) cvg.game_version.game
(J) cvg.computer_game_mod.game_modded
Prediction: (AND cvg.cvg_designer (JOIN cvg.cvg_designer.games_designed [Dark Sun: Wake of the
Ravager]))
```

Figure 2: Example of LLM prompt (part 1): The task instruction and k teaching examples (only one is shown because of the length) with questions and their contexts.

```
Question: worldofwarcraft is the creation of which video game designer?
Candidate entities:
(A) [worldofwarcraft]
Exemplary Logical Forms:
(A) (AND cvg.cvg_designer (JOIN cvg.cvg_designer.games_designed [worldofwarcraft]))
(B) (AND cvg.cvg_designer (JOIN (R cvg.computer_videogame.designers) [worldofwarcraft]))
(C)       (AND       cvg.cvg_designer       (JOIN       cvg.cvg_designer.games_designed       (JOIN
cvg.computer_game_expansion.expansion_for [worldofwarcraft])))
(D)     (AND     cvg.cvg_designer     (JOIN     (R     cvg.computer_videogame.designers)     (JOIN
cvg.computer_game_expansion.expansion_for [worldofwarcraft])))
(E)    (AND    cvg.cvg_designer    (JOIN    (R    cvg.computer_videogame.designers)    (JOIN    (R
cvg.computer_videogame.expansions) [worldofwarcraft])))
Candidate classes:
(A) games.game_designer
(B) cvg.cvg_designer
(C) amusement_parks.ride_designer
(D) cvg.cvg_developer
(E) cvg.computer_videogame
(F) cvg.computer_game_engine_developer
(G) cvg.computer_game_engine
(H) cvg.computer_game_mod
(I) cvg.game_performance
(J) cvg.musical_game
Candidate relations:
(A) cvg.computer_videogame.designers
(B) cvg.cvg_designer.games_designed
(C) games.game_designer.games_designed
(D) games.game.designer
(E) cvg.computer_videogame.developer
(F) cvg.cvg_developer.games_developed
(G) cvg.computer_game_engine.developer
(H) cvg.computer_videogame.expansions
(I) cvg.computer_videogame.publisher
(J) cvg.game_version.developer
Prediction:
```

Figure 3: Example of LLM prompt (part 2): After k teaching examples, the input question and its contexts is given. The golden prediction is `(AND cvg.cvg_designer (JOIN cvg.cvg_designer.games_designed [worldofwarcraft]))`.