

TIARA: Multi-grained Retrieval for Robust Question Answering over Large Knowledge Bases

Yiheng Shu^{♡*}, Zhiwei Yu[◇], Yuhan Li^{♣*}, Börje F. Karlsson[◇], Tingting Ma^{♣*},
Yuzhong Qu[♡] and Chin-Yew Lin[◇]

[♡] State Key Laboratory of Novel Software Technology, Nanjing University, China;

[◇] Microsoft Research; [♣] Nankai University; [♠] Harbin Institute of Technology
yhshu@smail.nju.edu.cn, yzqu@nju.edu.cn, yuhanli@mail.nankai.edu.cn

{zhiwei.yu, borje.karlsson, cyl}@microsoft.com

Abstract

Pre-trained language models (PLMs) have shown their effectiveness in multiple scenarios. However, KBQA remains challenging, especially regarding coverage and generalization settings. This is due to two main factors: i) understanding the semantics of both questions and relevant knowledge from the KB; ii) generating executable logical forms with both semantic and syntactic correctness. In this paper, we present a new KBQA model, **TIARA**, which addresses those issues by applying multi-grained retrieval to help the PLM focus on the most relevant KB contexts, viz., entities, exemplary logical forms, and schema items. Moreover, constrained decoding is used to control the output space and reduce generation errors. Experiments over important benchmarks demonstrate the effectiveness of our approach. **TIARA** outperforms previous SOTA, including those using PLMs or oracle entity annotations, by at least 4.1 and 1.1 F1 points on GrailQA and WebQuestionsSP, respectively. Specifically on GrailQA, **TIARA** outperforms previous models in all categories, with an improvement of 4.7 F1 points in zero-shot generalization.¹

1 Introduction

Knowledge base question answering (KBQA) has established itself as an important and promising research area as it greatly helps the accessibility and usability of existing large-scale knowledge bases (KBs). Such KBs contain abundant facts in a structured form, which can not only be accessed, but also reasoned over. KBQA bypasses the need for users to learn complex and burdensome formal query languages, and it allows users to leverage knowledge using natural language. While recent KBQA efforts have achieved interesting results,

most previous studies target or assume an underlying strong correspondence between the distributions of schema items in test questions and training data, i.e., the i.i.d. assumption. However, this assumption does not hold, especially for large-scale KBs, which typically contain very large numbers of entities and schema items (classes and relations).² This extensive space is a critical challenge as it both allows for a myriad of novel compositions of schema items (Keysers et al., 2020), i.e., requires compositional generalization. It also immensely increases the likelihood of user queries, including previously unseen items or domains (Gu et al., 2021), i.e., requires strong zero-shot generalization.

Meanwhile, pre-trained language models (PLMs), such as T5 (Raffel et al., 2020) and GPT-3 (Brown et al., 2020), have demonstrated notable success in many natural language processing (NLP) scenarios (Karpas et al., 2022). Many times they even show strong generalization capabilities. Inspired by the progress of PLMs on unstructured text-to-text tasks, researchers have explored semantic parsing of natural language to logical form utilizing PLMs (Poesia et al., 2022; Xie et al., 2022) to tackle the generalization challenges and improve system robustness. However, differently from unstructured data in the typical PLM pre-training phase, KBs represent rich semantics and complex structures, which lead to two challenges in using PLMs for KBQA: i) **KB Grounding**: given a KB, how to understand the semantics of both the question and the relevant knowledge from the KB? ii) **Logical Form Generation**: how to make sure syntax and semantics of generated logical forms conform to the KB specification and are executable (i.e., guarantee correctness)?

KB grounding requires linking a question to relevant KB items, but the large size of KBs makes it

¹Code is available at <https://github.com/microsoft/KC/tree/main/papers/TIARA>.

^{*}Work performed during their internships at Microsoft Research Asia.

²Our experiments involve more than 45M entities, 2K classes, and 6K relations. RDF Schema contains `rdfs:Class` (class) and `rdf:Property` (relation).

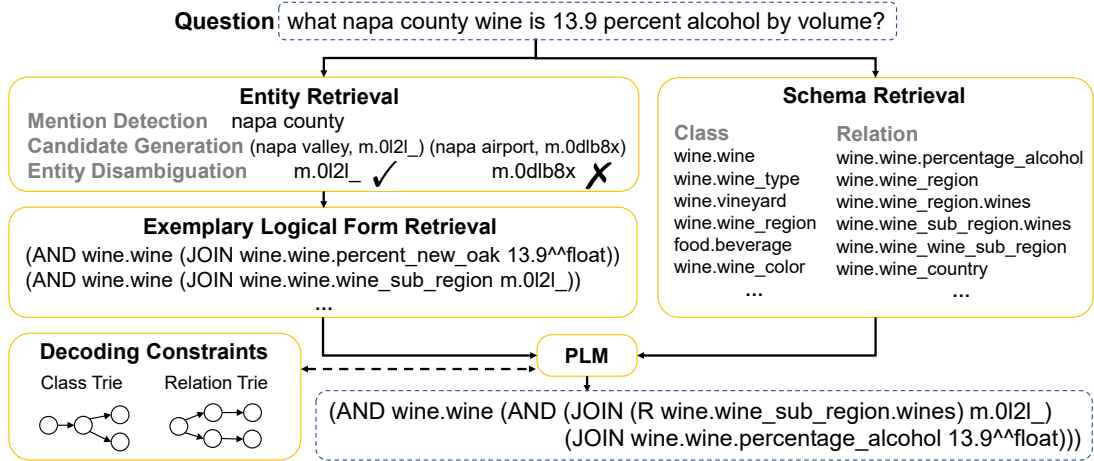


Figure 1: Overview of TIARA. 1) *Entity retrieval* grounds the mention to entity `m.0121_`. 2) *Exemplary logical form retrieval* enumerates logical forms starting from the entity `m.0121_` or the number 13.9, and ranks them. 3) *Schema retrieval* independently grounds the most related schema items. 4) Retrieved multi-grained contexts are then fed to the PLM for generation. 5) *Constrained decoding* controls the schema search space during logical form generation.

challenging. We consider that contextual retrieval can help the PLM focus on the most relevant contexts (Xie et al., 2022; Wei et al., 2022). Previous works propose different retrieval methods. Das et al. (2021) and Zan et al. (2022) retrieve questions with logical form annotations that are similar to the input question from the training set, which is only effective for i.i.d. scenarios. Ye et al. (2021) retrieve exemplary logical forms only within a two-hop range of linked entities. Chen et al. (2021) and Xie et al. (2022) retrieve schema items for each question without considering the connectivity on the KB and have poor zero-shot performance.

To make the best of structured KB contexts and be able to take advantage of recent PLM work, we analyze all stages in a KBQA system to identify and address their challenges, and we demonstrate how leveraging multi-grained retrieval augments PLM generation - especially regarding robustness in compositional and zero-shot settings: (i) **Entity retrieval**: finding topic entities is a key step, and we augment the mention detection method to improve entity retrieval (linking) performance in zero-shot cases; (ii) **Exemplary logical form retrieval**: logical forms provide the semantic and structural contexts from the KB, which assists the PLMs in KB grounding and in generating valid logical forms; and (iii) **Schema retrieval**: logical form enumeration alone cannot properly support questions with more than two hops or diverse function types. However, retrieving schema items is not subject to this constraint and it can be used as a semantic supplement to logical forms.

Although KB contexts are retrieved for the PLM, it may still generate invalid logical forms due to its unconstrained output space, e.g., generating schema items that do not exist in the KB. Similarly to how seq2seq models combined with rules (Liang et al., 2017; Chen et al., 2021) reduces syntax errors during logical form generation, we further introduce **constrained decoding** to alleviate this issue for the PLM using prefix trees constructed from the KB.

Here we propose a multi-grained Retrieval Augmented (TIARA) KBQA method (Section 3) that addresses the two mentioned challenges. As shown in Figure 1, we employ multi-grained retrieval to provide both semantic and syntactic references to a PLM. Then, target logical form generation is controlled by constrained decoding. Utilizing these mechanisms, TIARA achieves improved KBQA performance not only in i.i.d., but also in compositional and zero-shot generalization (Section 4). Experiments over two important benchmarks demonstrate the effectiveness of our approach. TIARA outperforms all previous SOTA methods, including those using PLMs or even oracle entity annotations, by at least 4.1 and 1.1 F1 points on GrailQA and WebQuestionsSP, respectively. Specifically on GrailQA, TIARA outperforms previous models in all generalization categories, with an improvement of 4.7 F1 points in zero-shot generalization. To further explore how TIARA strengthens the potential of PLMs for KBQA, we also analyze the impact of providing PLMs with different granularities of contexts on

different generalization levels and question types (Section 5).

2 Related Work

2.1 Knowledge Base Question Answering

KBQA endeavors to allow querying KBs through natural language questions. Existing KBQA methods are mainly either information retrieval-based (IR-based) or semantic parsing-based (SP-based) methods (Lan et al., 2021). IR-based methods (Shi et al., 2021; Zhang et al., 2022) construct a question-specific graph and rank entities or paths to get the top answers. SP-based methods aim at generating target logical forms, which makes them more interpretable. Such methods can be classified into feature-based ranking and seq2seq generation methods. Feature-based ranking methods search and rank query graphs in a step-wise manner. Yih et al. (2015) define a query graph and propose a staged generation method. Lan and Jiang (2020) incorporate constraints during the staged query graph generation. Hu et al. (2021) use rules to decompose questions and guide the search. Seq2seq generation methods (Liang et al., 2017; Chen et al., 2021) convert natural language sequences to logical form sequences. They are more flexible than the above ranking methods in generating logical forms with more hops or functions. In particular, seq2seq generation methods using PLMs (Oguz et al., 2020; Das et al., 2021; Ye et al., 2021; Cao et al., 2022a; Hu et al., 2022) show promising performance. Among them, the most relevant work to ours is RnG-KBQA (Ye et al., 2021), which also retrieves exemplary logical forms as important contexts. However, TIARA further explores the use of PLMs by improving zero-shot mention detection, matching entity-independent semantics with schema retriever, and reducing generation errors with constrained decoding.

2.2 Neural Semantic Parsing with PLMs

Several existing neural semantic parsing methods consider contextual retrieval when applying PLMs. Shi et al. (2022) use nearest neighbor retrieval to augment zero-shot inference. Xie et al. (2022) feed contexts to PLMs and conduct experiments on several semantic parsing tasks over structured data. Furthermore, other works also leverage constrained decoding for semantic parsing. Scholak et al. (2021) propose multiple levels of constrained decoding for the text-to-SQL task and improve per-

formance without additional training. Shin et al. (2021) illustrate PLMs with constrained decoding have few-shot semantic parsing ability. While, Poesia et al. (2022) propose that retrieval of related examples and constrained decoding can improve the performance of the code generation task without fine-tuning. In this paper, we introduce the idea of multi-grained contextual retrieval along with constrained decoding to bolster KBQA scenarios.

3 Method

To demonstrate how comprehensive KB context enhances the robustness of retrieval, a key component of our method, we retrieve KB contexts in a multi-grained manner and then generate target logical forms with constrained decoding, as shown in Figure 1. We introduce multi-grained retrieval in Section 3.2, target logical form generation in Section 3.3, and describe constrained decoding in Section 3.4.

3.1 Preliminaries

Here, a knowledge base is an RDF graph consisting of a collection of triples in the form (s, r, o) , where s is an entity, r is a relation, and o can be an entity, a literal, or a class. We use s-expressions as the logical form following Gu et al. (2021) and Ye et al. (2021), which can later be converted to SPARQL queries and executed over KBs. In this paper, **exemplary logical form** refers to the logical form that is input to the generative PLM as the context. It is obtained by enumeration and ranking on KBs. **Schema** refers to `rdfs:Class` (class) and `rdf:Property` (relation) together, and they are the necessary elements of logical forms. **Target logical form** refers to the logical form generated by the PLM. Examples of above concepts are shown in Figure 1.

3.2 Multi-grained Retriever

Entity Retriever We perform entity retrieval following a standard pipeline with three steps, i.e., mention detection, candidate generation, and entity disambiguation (Shen et al., 2021). To better detect zero-shot mentions, we regard mention detection as a span classification task (Jiang et al., 2021). In this way, we can deal well with sentences containing out-of-vocabulary entities (Fu et al., 2021) and achieve higher recall by adjusting the threshold of outputting a candidate mention (Yu et al., 2020). We obtain token representations from contextual-

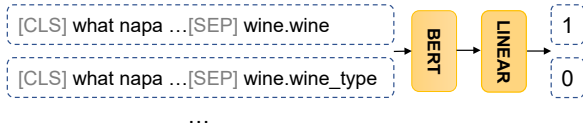


Figure 2: Schema retrieval learns if a question and a schema item are a match or not.

ized pre-trained BERT (Devlin et al., 2019) and enumerate all the possible spans in the question within a maximum length. For each span, representations of the start and the end token are concatenated together with the length embedding as the span representation.³ Span representations are fed into a classifier (i.e., softmax layer) to get the probability that each span is a mention. We refer to our mention detector as SpanMD in the rest of the paper. Then, we generate candidate entities for each mention using the alias mapping FACC1 (Gabrilovich et al., 2013). For entity disambiguation, we leverage a cross-encoder framework as Ye et al. (2021) to jointly encode the question and contexts of each candidate entity (i.e., the entity label and its linked relations) and obtain a matching score. The candidate with the highest score is kept as the entity for each mention.

Exemplary Logical Form Retriever The logical form retrieval includes enumeration and ranking. We follow the enumeration method proposed by Gu et al. (2021) and Ye et al. (2021). Starting from all potential entities, it searches their neighborhood up to two hops since it is too costly to enumerate beyond that due to the exponential increase in candidates. Each path is converted to an s-expression as an exemplary logical form. For ranking, each pair (x, c) of question x and candidate exemplary logical form c is concatenated by a [SEP] token and used as input to BERT. The [CLS] representation is fed into a linear layer to get the score $s(x, c)$ as follows:

$$s(x, c) = \text{LINEAR}(\text{BERTCLS}([x; c])) \quad (1)$$

To distinguish the target logical form from the enumeration results, the ranker is optimized to minimize the following loss function:

$$\mathcal{L}_{\text{ranker}} = -\frac{e^{s(x, y)}}{e^{s(x, y)} + \sum_{c \in C \wedge c \neq y} e^{s(x, c)}} \quad (2)$$

³Length embedding is obtained by a learnable look-up table.

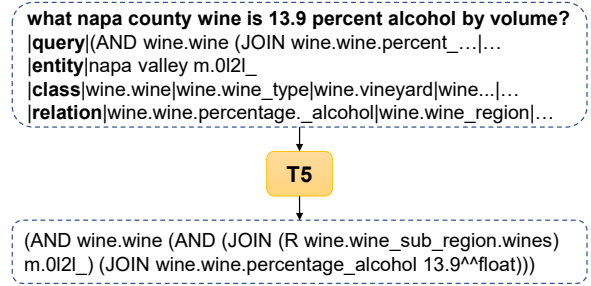


Figure 3: Given a set of retrieved contexts, T5 generates the target logical form.

where y is the target logical form or its semantic equivalent. Note that a question may not contain any entity, in which case no exemplary logical forms will be retrieved. Due to the hop limitation and the diversity of functions, the enumeration is not guaranteed to cover target logical forms.

Schema Retriever Independent of the above entity and exemplary logical form retrieval, we employ dense schema retrieval to find semantically relevant items (classes and relations), which are not restricted to the neighbors of entities and can be used as a semantic supplement to exemplary logical forms. As shown in Figure 2, schema retrieval is implemented by a cross-encoder. Compared to the bi-encoder (Chen et al., 2021), it learns the interaction representation between the question and schema items. The matching score $s(x, c)$ of a question x and a schema c is calculated as Equation 1. The objective is the same as in the sentence-pair classification task (Devlin et al., 2019). We select top-ranked classes and relations with the highest scores for each question. Class and relation retrievers are trained separately.

3.3 Target Logical Form Generation

The previously retrieved contexts are used to augment the PLM in target logical form generation. Generation is performed by a transformer-based seq2seq model - T5 (Raffel et al., 2020). As shown in Figure 3, the input sequence is a concatenation of the input question, retrieved entities, exemplary logical forms, and schema items. The entity is represented by its label and ID. The output sequence is the target logical form. The T5 model is fine-tuned to generate the target sequence with the cross-entropy objective as follows:

$$\mathcal{L}_{\text{gen}} = -\sum_{t=1}^n \log(p(y_t | y_{<t}, x, c)) \quad (3)$$

Method	Overall		I.I.D.		Compositional		Zero-shot	
	EM	F1	EM	F1	EM	F1	EM	F1
GloVe + TRANSDUCTION (Gu et al., 2021)	17.6	18.4	50.5	51.6	16.4	18.5	3.0	3.1
QGG (Lan and Jiang, 2020)	-	36.7	-	40.5	-	33.0	-	36.6
BERT + TRANSDUCTION (Gu et al., 2021)	33.3	36.8	51.8	53.9	31.0	36.0	25.7	29.3
GloVe + RANKING (Gu et al., 2021)	39.5	45.1	62.2	67.3	40.0	47.8	28.9	33.8
BERT + RANKING (Gu et al., 2021)	50.6	58.0	59.9	67.0	45.5	53.9	48.6	55.7
ReTraCk (Chen et al., 2021)	58.1	65.3	84.4	87.5	61.5	70.9	44.6	52.5
S ² QL (Zan et al., 2022)	57.5	66.2	65.1	72.9	54.7	64.7	55.1	63.6
ArcaneQA (Gu and Su, 2022)	63.8	73.7	85.6	88.9	65.8	75.3	52.9	66.0
RnG-KBQA (Ye et al., 2021)	68.8	74.4	86.2	89.0	63.8	71.2	63.0	69.2
TIARA (Ours)	73.0	78.5	87.8	90.6	69.2	76.5	68.0	73.9

Table 1: EM and F1 results (%) on the hidden test set of GrailQA. TIARA outperforms other methods with three levels of generalization settings in both EM and F1.

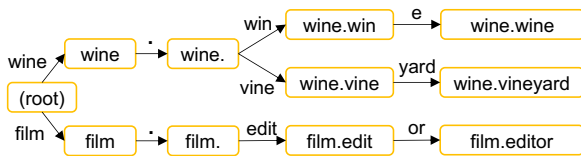


Figure 4: An example of a trie (prefix tree) that stores KB classes. Each edge represents a token that the PLM can select.

where x denotes the input question, c denotes retrieved contexts, y denotes the target sequence, and n is the length of the target sequence. During inference, the model uses beam search and retains the top- k sequences. Constrained decoding is performed during beam search (Section 3.4).

3.4 Constrained Decoding

Directly applying the beam search algorithm in the decoding process may generate invalid operators or schema items. The constrained decoding focuses on reducing generation errors on logical form operators and schema items. For operators, only those operator tokens allowed by the s-expression syntax are considered valid during the generation process. For schema items, their tokens are stored in the trie (prefix tree), as shown in Figure 4. KB classes and relations are stored in two separate tries. When a constraint rule finds that a schema item is currently being decoded, only the options in the trie are valid for the next token.

4 Experiments

4.1 Setup

Datasets We perform experiments on two important standard KBQA datasets.

GrailQA (Gu et al., 2021) is a large-scale KBQA

dataset on Freebase (Bollacker et al., 2008) with 64,331 questions annotated with logical forms. It is carefully split to evaluate three levels of generalization (i.i.d., compositional, and zero-shot). Questions contain up to 4-hop relations and optionally contain functions for counting, superlatives (ARGMIN, ARGMAX), and comparatives ($<$, \leq , $>$, \geq). **WebQuestionsSP** (WebQSP) (Yih et al., 2016) is a widely used semantic parsing dataset containing 4,937 questions from Google Suggest API. We follow the split of training and validation sets as Ye et al. (2021) since there is no official split.

Evaluation Metrics We use the official scripts to evaluate on both datasets. For GrailQA, we use exact match (EM) and F1 score (F1) as metrics. For WebQSP, we report F1 score and hits@1. It is worth noting that SP-based methods, including TIARA, generate logical forms that yield unordered answers, which are usually evaluated by F1, not hits@1. For a comprehensive comparison, we randomly select an answer for each question 100 times and calculate the average hits@1.⁴

Implementation Details Our experiments are done on a machine with an NVIDIA A100 GPU and 216GB of RAM. We implement our models utilizing PyTorch (Paszke et al., 2019) and Hugging Face.⁵ Due to the significant difference in dataset sizes, their experimental setups are different. For entity retriever, we use our proposed retriever (Section 3.2) for GrailQA with a maximum mention length of 15, and the off-the-shelf entity linker ELQ (Li et al., 2020) for WebQSP.⁶ For schema re-

⁴Previous SP-based methods calculate hits@1 by random selection with unknown times, so results may fluctuate.

⁵<https://huggingface.co/>

⁶Using ELQ is analogous to previous SOTA. The domain and mention length distribution between the two datasets is

triever, we fine-tune the BERT-base-uncased model for at most 3 epochs with early stopping. We randomly sample 50 negative candidates for each question. The top-10 classes and relations of the highest scores are kept respectively. As generator, we use the T5-base seq2seq generation model and fine-tune it with a learning rate of $3e-5$. The beam size is set to 10. We limit the maximum number of tokens to 1,000 for inputs and 128 for outputs. For GrailQA, we train the generation model for 10 epochs with a batch size of 8. For WebQSP, we train it for 20 epochs with a batch size of 2. After generation, we first check the logical forms generated by T5 in the beam search order by execution and exclude non-executable or invalid queries (with null answers). Then, if no generated logical form is valid, we further check if any exemplary logical form is valid. The first valid query is taken as the final prediction.

Baselines for Comparison We compare against models on the GrailQA leaderboard and previous SOTA methods for WebQSP.⁷ The results are taken from their published papers. RnG-KBQA is the previous SOTA on both GrailQA and WebQSP and uses a T5-base generator, the same as TIARA.

4.2 Overall Performance

TIARA results on the GrailQA hidden test set and WebQSP test set are presented in Table 1 and Table 2, respectively.

GrailQA Performance TIARA outperforms previous methods overall and in all three levels of generalization on both EM and F1. Although previous methods (Chen et al., 2021; Ye et al., 2021; Gu and Su, 2022) perform well in i.i.d., TIARA with multi-grained contexts gains an advantage over them in generalization scenarios. It increases the F1 by 5.3 and 4.7 points compared to the previous SOTA in compositional and zero-shot generalization, respectively. ArcaneQA (Gu and Su, 2022) is flexible in generating query structures with the help of dynamic program induction and has outperformed previous SOTA (Ye et al., 2021) in compositional generalization by a large margin. However, TIARA further boosts the performance in compositional generalization to 76.5 F1 points. Compared to the improvement in F1 mentioned above, the improvement in EM is more significant, indicating that

significant.

⁷<https://dki-lab.github.io/GrailQA/> on Jun. 3, 2022.

Method	F1	Hits@1
<i>IR-based methods</i>		
EmbedKGQA* (Saxena et al., 2020)	-	66.6
GRAFT-Net (Sun et al., 2018)	62.8	67.8
PullNet (Sun et al., 2019)	-	68.1
TransferNet [♡] (Shi et al., 2021)	-	71.4
Relation Learning ^{♡♣} (Yan et al., 2021)	64.5	72.9
NSM* [♡] (He et al., 2021)	67.4	74.3
Subgraph Retrieval* (Zhang et al., 2022)	74.5	83.2
<i>SP-based (feature-based ranking) methods</i>		
TextRay [♡] (Bhutani et al., 2019)	60.3	-
Topic Units [♡] (Lan et al., 2019)	67.9	-
UHop (Chen et al., 2019)	68.5	-
GrailQA RANKING ^{♡♣} (Gu et al., 2021)	70.0	-
STAGG [♡] (Yih et al., 2016)	71.7	-
QGG [♡] (Lan and Jiang, 2020)	74.0	-
<i>SP-based (seq2seq generation) methods</i>		
NSM [♡] (Liang et al., 2017)	69.0	-
ReTraCk (Chen et al., 2021)	71.0	71.6
CBR-KBQA (Das et al., 2021)	72.8	-
ArcaneQA (Gu and Su, 2022)	75.6	-
RnG-KBQA (Ye et al., 2021)	75.6	-
Program Transfer* [♣] (Cao et al., 2022b)	76.5	74.6
TIARA (Ours)	76.7	73.9
w/o Schema	76.4	73.7
w/o ELF	75.0	73.4
w/o ELF & Schema	73.2	71.1
TIARA*	78.9	75.2
w/o Schema	78.8	75.0
w/o ELF	76.2	74.5
w/o ELF & Schema	75.4	73.1

Table 2: F1 and hits@1 results (%) on WebQSP. * denotes using oracle entity linking annotations. ♡ denotes the assumption of a fixed number of hops. ♣ denotes pre-training on an auxiliary task or other KBQA datasets. For comparison, hits@1 on TIARA is obtained by randomly selecting one answer for each question 100 times.

TIARA performs better in understanding questions, e.g., case II in Section 5.4.

WebQSP Performance TIARA outperforms existing IR- and SP-based methods in F1, without assumptions of a fixed number of hops nor using oracle entity annotations. Zhang et al. (2022) propose sub-graph retrieval enhanced reasoning, which makes hits@1 substantially outperform other methods, but its F1 is still inferior to TIARA. Cao et al. (2022b) transfer the knowledge from KQA Pro (Cao et al., 2022a) to WebQSP with oracle entity annotations, which demonstrates the effect of pre-training on large KBQA datasets. TIARA reaches comparable results without such extra refinement. For further comparison, we also evaluate TIARA’s performance given oracle annotations, which boosts the F1 score to 78.9 points. Besides the methods listed in Table 2, Unik-QA (Oguz et al.,

Method	Overall		I.I.D.		Compositional		Zero-shot	
	EM	F1	EM	F1	EM	F1	EM	F1
RnG-KBQA (Ye et al., 2021)	71.4	76.8	86.7	89.0	61.7	68.9	68.8	74.7
ArcaneQA (Gu and Su, 2022)	69.5	76.9	86.1	89.2	65.5	73.9	64.0	72.8
TIARA (Ours)	75.3	81.9	88.4	91.2	66.4	74.8	73.3	80.7
w/o CD	75.1	81.5	88.4	91.2	66.4	74.6	72.9	80.2
w/o Schema	73.5	79.2	86.3	89.9	64.7	72.7	71.6	77.3
w/o Schema & CD	73.3	79.0	86.1	89.6	64.1	72.1	71.5	77.2
w/o ELF	62.3	65.0	87.7	88.9	60.4	66.0	51.9	54.2
w/o ELF & CD	60.2	62.6	87.3	88.6	59.2	64.4	48.8	50.6
w/o ELF & Schema	37.3	39.8	86.0	88.1	58.6	66.3	7.3	7.8
w/o ELF & Schema & CD	32.9	34.6	85.1	86.7	52.2	57.7	2.1	2.3
w/o SpanMD w/ BERT-NER	73.3	79.7	87.4	90.0	66.3	74.7	70.0	77.3
Exemplary Logical Forms	67.2	72.9	72.8	76.7	55.3	60.7	69.7	76.3

Table 3: Ablation experiments on the GrailQA validation set. *ELF* denotes retrieving exemplary logical forms as contexts. *Schema* denotes retrieving schema items as contexts. *CD* denotes constrained decoding in beam search. *Exemplary Logical Forms* means only using retrieved logical forms without generation.

2020) can achieve 76.7% Hits@1 on WebQSP by using T5-base to generate the answer text. However, while its results on WebQSP would be on par with TIARA, leveraging PLMs in an open-domain QA manner cannot generate all answers and does not fully solve KBQA, since the answer text may be ambiguous for entity linking (Freebase entity texts are not unique).

5 Analysis

5.1 Ablation Study

To verify the effect of the proposed multi-grained contextual retrieval and constrained decoding, we conduct ablation experiments using the GrailQA validation set, as shown in Table 3. The ablation experiment to remove the exemplary logical form retrieval (w/o ELF) or schema retrieval (w/o Schema) is to retrain and inference with different contexts.

Generation Alleviates Deficiencies of Retrieval

Using exemplary logical forms directly as prediction can achieve 72.9 F1 points, but they perform poorly in compositional generalization with complex structures (60.7 F1 points). This issue is well mitigated by generation, which achieves 72.1 F1 points w/o Schema & CD. TIARA with multi-grained contexts further gains 2.7 F1 points in compositional setting.

Exemplary Logical Forms as Critical Contexts

TIARA w/o ELF decreases performance by 26.5 F1 points in zero-shot setting, indicating the critical role of exemplary logical forms. PLMs have the transferable understanding capability but cannot perceive KB structure, which can be largely

alleviated by leveraging connected structures in exemplary logical forms.

Schema Retrieval as Necessary Supplement

Removing schema retrieval (w/o Schema) decreases F1 by 2.7 points overall, and even 25.2 points (w/o ELF & Schema compared to w/o ELF) when exemplary logical forms are not considered. It shows schema items effectively provide semantics. Besides, TIARA with no context other than entities (w/o ELF & Schema & CD) performs very poorly in zero-shot setting (F1 only 2.3 points), while schema retrieval remedies this issue to a large extent (50.6 F1 points w/o ELF & CD).

Constrained Decoding Reduces Errors

When the semantics and structure from exemplary logical forms and schema items are effective, constrained decoding contributes to 0.4 F1 points overall. The PLM performance drops dramatically when these contexts are unavailable, and TIARA w/o ELF & Schema improves F1 by 5.2 points compared to TIARA w/o ELF & Schema & CD. Constrained decoding without any additional retrieval or training can reduce generation errors, especially when no context is available for PLM augmentation.

Mention Detection as Key to Entity Recall

Mention detection (i.e., the first step of entity retrieval) is the key to entity recall since it determines the upper-bound recall of the entity retriever. In this step, we adopt SpanMD as mention detector (described in Section 3.2) instead of the typically used BERT-NER in previous works (Gu et al., 2021; Chen et al., 2021; Ye et al., 2021; Gu and Su, 2022). It shows that replacing BERT-NER with SpanMD

leads to 2.2 F1 points improvement overall. More specifically, SpanMD obtains a notable gain in the zero-shot scenario (F1 +3.4 points), demonstrating the superiority of SpanMD in tackling zero-shot cases. Please refer to Appendix A.1 for more discussion on mention detection and entity retrieval.

Function	None	Count	Comp.	Super.
RnG	77.5/81.8	73.0/77.5	55.1/76.0	13.8/22.3
TIARA	77.8/83.1	76.4/81.8	57.4/81.4	58.7/69.0
#relation	1	2	3	4
RnG	75.7/79.3	65.3/74.7	28.6/44.5	100.0/100.0
TIARA	81.2/85.6	64.7/75.8	29.3/48.5	50.0/83.3

Table 4: EM and F1 results (%) of different types of questions on the GrailQA validation set. *RnG* denotes RnG-KBQA (Ye et al., 2021). *None* denotes only operators AND and JOIN are in the s-expression. *Comp.* and *Super.* denotes comparative and superlative. *#relation* denotes the number of relations in the s-expression.

5.2 In-Depth Analysis

To show how TIARA performs on different logical form functions and structures, we compare it with previous SOTA (Ye et al., 2021) on the GrailQA validation set in more detail, as shown in Table 4.

Function TIARA performs better for all function types even with limited data, especially for superlative questions. While some superlative questions have no entities, which leads to no exemplary logical forms, TIARA does not rely exclusively on these logical forms and still has schema items as contexts. Thus, it reaches a 3x improvement on F1 score for this function type.

#relation TIARA also performs well when considering different numbers of relations, except only for the number of relations being 4 (due only to two questions, one of which is incorrect because of a missing numerical property).

5.3 Error Analysis

To further explore the limitations of TIARA, we randomly sample 50 questions whose predictions are different from the ground truth in the GrailQA validation set. The major errors can be classified as follows: **Entity retriever errors (46%)** is caused by mention detection failure or high ambiguity of mentions, e.g., TIARA misses “volt per metre” in the question “what is the unit of volt per metre?”. Entity retrieval remains a key challenge for KBQA.

Syntactic errors (26%) mostly occurs when questions involve 1) rare operators, e.g., ARGMIN, and ARGMAX (5.95% in the training corpus); 2) multiple constraints, whose logical forms are relatively complicated. **Semantic errors (12%)** where TIARA selects incorrect schema items when correct ones are provided. Of those errors of this type, 83.3% come from zero-shot instances. **False negatives (6%)** mainly occurs in comparative questions, i.e., “larger than” is annotated as \geq in some cases. **Miscellaneous (10%)** occurs when the question is semantically ambiguous, and predicted results are inconsistent with the oracle schema in the KB.

5.4 Case Study

To aid in visualizing TIARA’s capabilities, we select three case studies from the GrailQA validation set in Table 5. In Case I, the entity m.01p5ld (decimetre) is not connected to the relation “substance_units” predicted by **TIARA w/o ELF**. Using only discrete schema does not guarantee that generated logical forms are connected. The exemplary logical forms (ELF) ensure graph connectivity and provide better contexts in this case. Case II contains an entity and a literal connected by two relations, as well as a function *lt* (<). **TIARA w/o the Schema** chooses one of its exemplary logical forms with only one relation and no function. The enumeration of exemplary logic forms is limited regarding literals and functions, but schema retrieval contributes to a semantic supplement. Lastly, in Case III, **TIARA w/o CD** selects the phrase “resistance_unit” that, while somewhat semantically similar to the correct one, does not exist in the KB. It shows that constrained decoding helps PLM to generate logical forms that conform to the KB specification.

6 Conclusion

We present TIARA to empower PLMs on QA over large-scale KBs. Multi-grained retrieval transcends insufficient exemplary logical forms and isolated schema items, and its results provide both semantic and syntactic contexts for a given question.

In systematic stage-by-stage analysis, we show that: i) constrained decoding prunes invalid search and reduces generation errors (e.g., non-existent schema and illegal operators); ii) entity linking improvements address a key bottleneck for zero-shot generalization; iii) schema retrieval with flexible scope is an essential semantic supplement to sur-

<p>Case I Question name the system that has decimetre as a measurement unit. TIARA (AND measurement_unit.measurement_system (JOIN measurement_unit.measurement_system.length_units m.01p5ld)) (✓) TIARA w/o ELF (AND measurement_unit.measurement_system (JOIN measurement_unit.measurement_system.substance_units m.01p5ld)) (✗)</p>
<p>Case II Question which bipropellant rocket engine has a chamber pressure of less than 257.0 and uses an oxidizer of lox? TIARA (AND spaceflight.bipropellant_rocket_engine (AND (JOIN spaceflight.bipropellant_rocket_engine.oxidizer m.01tm_5) (lt spaceflight.bipropellant_rocket_engine.chamber_pressure 257.0^^float))) (✓) TIARA w/o Schema (AND spaceflight.bipropellant_rocket_engine (JOIN spaceflight.bipropellant_rocket_engine.chamber_pressure 257.0^^float)) (✗)</p>
<p>Case III Question find the smallest possible unit of resistivity. TIARA (ARGMIN measurement_unit.unit_of_resistivity measurement_unit.unit_of_resistivity.resistivity_in_ohm_meters) (✓) TIARA w/o CD (ARGMIN measurement_unit.unit_of_resistance_unit measurement_unit.unit_of_resistivity.resistivity_in_ohm_meters) (✗)</p>

Table 5: Case study of predicted logical forms by TIARA variants on the GrailQA validation set. The differences in the predictions are **bolded**. Errors are marked in **red**, and the corresponding correct parts are marked in **blue**.

pass constraints of previous systems⁸; and iv) the generalization capabilities of PLM can be further empowered by both the retrieval and constrained decoding for semantic parsing pipelines. All of which combined result in an effective and robust KBQA system.

Though we have started exploiting the potential of PLMs, there is still a gap between the pre-training of PLMs on natural language and the downstream semantic parsing task. In the future, it is worth exploring how to build a bridge between unstructured natural language and structured KB in the pre-training phase.

Limitations

Our method learns from logical form annotations, which require expensive and specialized crowd-sourcing work in data collection.

The retrieval efficiency of our method also deserves further optimization. Using figures from the evaluations over GrailQA, logical form enumeration alone takes more than 7 seconds per question when there is no cache, which would need improvement to meet requirements in certain practical scenarios. Schema retrieval takes 1.41 seconds per question, which is the major additional time consumption compared to RnG-KBQA. For comparison, BERT + RANKING (Gu et al., 2021), RnG-KBQA (Ye et al., 2021), and ArcaneQA (Gu and Su, 2022) take on average 115.5, 82.1 and 5.6 seconds per question respectively - as reported by

⁸E.g., limited to 2-hop relations; unable to deal with questions without entities.

Gu and Su (2022). ReTraCk (Chen et al., 2021) reports taking on average 1.62 seconds per question.

Lastly, while the semantic parsing approach in this paper applies well to RDF graphs, it is not fully applicable to other structured data, such as tables or unstructured data, which are important sources of knowledge in the wild.

Acknowledgments

We would like to thank all the anonymous reviewers for their constructive comments and useful suggestions. We thank Yu Gu for evaluating our submissions on the test set of the GrailQA benchmark. We also thank Yu Gu and Xi Ye for sharing pre-processed data on GrailQA and WebQuestionsSP.

References

- Nikita Bhutani, Xinyi Zheng, and H. V. Jagadish. 2019. [Learning to answer complex questions over knowledge bases with query composition](#). In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 739–748. ACM.
- 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.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss,

- Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyu 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.
- Shuang Chen, Qian Liu, Zhiwei Yu, Chin-Yew Lin, Jian-Guang Lou, and Feng Jiang. 2021. [ReTraCk: A flexible and efficient framework for knowledge base question answering](#). In *Proceedings of the 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.
- Zi-Yuan Chen, Chih-Hung Chang, Yi-Pei Chen, Jijnasa Nayak, and Lun-Wei Ku. 2019. [Uhop: An unrestricted-hop relation extraction framework for knowledge-based question answering](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 345–356. 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.
- Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. [Spanner: Named entity re-/recognition as span prediction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7183–7195. Association for Computational Linguistics.
- Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. [Facc1: Freebase annotation of cluweb corpora, version 1 \(release date 2013-06-26, format version 1, correction level 0\)](#).
- 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 *The Web Conference 2021*, pages 3477–3488. ACM / IW3C2.
- Yu Gu and Yu Su. 2022. [Arcaneqa: Dynamic program induction and contextualized encoding for knowledge base question answering](#). *CoRR*, abs/2204.08109.
- Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. [Improving multi-hop knowledge base question answering by learning intermediate supervision signals](#). In *WSDM 2021*, pages 553–561. ACM.
- 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.
- Huiqiang Jiang, Guoxin Wang, Weile Chen, Chengxi Zhang, and Börje F. Karlsson. 2021. [BoningKnife: Joint entity mention detection and typing for nested NER via prior boundary knowledge](#). *CoRR*, abs/2107.09429.
- Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Noam Rozen, Erez Schwartz, Gal Shachaf, Shai

- Shalev-Shwartz, Amnon Shashua, and Moshe Tenenholz. 2022. [MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning](#). *CoRR*, abs/2205.00445.
- 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. 2021. [Complex knowledge base question answering: A survey](#). *CoRR*, abs/2108.06688.
- Yunshi Lan and Jing Jiang. 2020. [Query graph generation for answering multi-hop complex questions from knowledge bases](#). In *Proceedings of the ACL 2020*, pages 969–974. Association for Computational Linguistics.
- Yunshi Lan, Shuohang Wang, and Jing Jiang. 2019. [Knowledge base question answering with topic units](#). In *Proceedings of the IJCAI 2019*, pages 5046–5052.
- Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. [Efficient one-pass end-to-end entity linking for questions](#). In *Proceedings of the EMNLP 2020*, pages 6433–6441. Association for Computational Linguistics.
- Chen Liang, Jonathan Berant, Quoc V. Le, Kenneth D. Forbus, and Ni Lao. 2017. [Neural symbolic machines: Learning semantic parsers on freebase with weak supervision](#). In *Proceedings of the ACL 2017*, pages 23–33. Association for Computational Linguistics.
- Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2020. [Unik-QA: Unified representations of structured and unstructured knowledge for open-domain question answering](#). *arXiv preprint arXiv:2012.14610*.
- 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.
- Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. [Synchromesh: Reliable code generation from pre-trained language models](#). *CoRR*, abs/2201.11227.
- 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.
- Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. 2020. [Improving multi-hop question answering over knowledge graphs using knowledge base embeddings](#). In *Proceedings of the ACL 2020*, pages 4498–4507. Association for Computational Linguistics.
- Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. [PICARD: parsing incrementally for constrained auto-regressive decoding from language models](#). In *Proceedings of the EMNLP 2021*, pages 9895–9901. Association for Computational Linguistics.
- Wei Shen, Yuhan Li, Yinan Liu, Jiawei Han, Jianyong Wang, and Xiaojie Yuan. 2021. [Entity linking meets deep learning: Techniques and solutions](#). *IEEE Transactions on Knowledge and Data Engineering*.
- Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 2021. [Transfernet: An effective and transparent framework for multi-hop question answering over relation graph](#). In *Proceedings of the EMNLP 2021*, pages 4149–4158. Association for Computational Linguistics.
- Weijia Shi, Julian Michael, Suchin Gururangan, and Luke Zettlemoyer. 2022. [Nearest neighbor zero-shot inference](#). *CoRR*, abs/2205.13792.
- Richard Shin, Christopher H. Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. [Constrained language models yield few-shot semantic parsers](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7699–7715. Association for Computational Linguistics.
- Haitian Sun, Tania Bedrax-Weiss, and William W. Cohen. 2019. [Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text](#). In *Proceedings of EMNLP-IJCNLP 2019*, pages 2380–2390. Association for Computational Linguistics.
- Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 2018. [Open domain question answering using early fusion of knowledge bases and text](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242. Association for Computational Linguistics.

- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *CoRR*, abs/2201.11903.
- Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. [Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models](#). *CoRR*, abs/2201.05966.
- Yuanmeng Yan, Rumei Li, Sirui Wang, Hongzhi Zhang, Zan Daoguang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Large-scale relation learning for question answering over knowledge bases with pre-trained language models. In *Proceedings of the EMNLP 2021*.
- Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2021. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the ACL 2022*.
- 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 ACL-IJCNLP 2015*, pages 1321–1331. The Association for Computer 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.
- Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. [Neural mention detection](#). In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 1–10. European Language Resources Association.
- Daoguang Zan, Wang Sirui, Zhang Hongzhi, Yan Yuanmeng, Wu Wei, Guan Bei, and Wang Yongji. 2022. [S²ql: Retrieval augmented zero-shot question answering over knowledge graph](#). In *PAKDD 2022*, volume 13282 of *Lecture Notes in Computer Science*, pages 223–236. Springer.
- Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the ACL 2022*.

A Retrieval Performance

A.1 Entity Retrieval Performance

Method	Overall	I.I.D.	Comp.	Zero.
BERT-NER	92.1	98.2	96.9	87.5
SpanMD (Ours)	94.3	98.3	95.1	92.2

Table 6: Recall (%) of mention detection on the GrailQA validation set. *Comp.* denotes compositional. *Zero.* denotes zero-shot.

Method	P	R	F1
GrailQA (Gu et al., 2021)	76.0	81.1	73.1
RnG-KBQA (Ye et al., 2021)	84.1	86.8	80.4
TIARA (Ours)	87.2	88.6	85.4

Table 7: Precision (P), recall (R) and F1 (%) of entity retrieval (%) on the GrailQA validation set.

We report the recall of mention detection on the GrailQA validation set based on two different mention detectors to show the effectiveness of SpanMD. As shown in Table 6, SpanMD substantially outperforms BERT-NER by 2.2 points on overall recall.⁹ Specifically, SpanMD achieves a significant recall gain (4.7 points) in a more challenging zero-shot setting compared to BERT-NER, demonstrating its superiority in tackling zero-shot cases.

In addition, we evaluate the overall performance of our entity retriever on the GrailQA validation set and report it in Table 7. Specifically, we compare against the following baselines: 1) **GrailQA** (Gu et al., 2021) which simply chooses the most popular entity according to the prior features provided by FACC1 (Gabrilovich et al., 2013). 2) **RnG-KBQA** (Ye et al., 2021) which utilizes the relation information linked with an entity to further enhance the semantics of the entity. While those and many previous works derive better entity retrieval results via focusing on how to improve the performance of disambiguation, TIARA also adopts an improved mention detector to boost entity retrieval.

From Table 7, it can be observed that our entity retriever significantly surpasses all the baseline models by at least 5.0 F1 points on the GrailQA validation set. Compared to the strongest baseline (Ye et al., 2021), we yield better results on both precision and recall, further demonstrating the effectiveness of the entity retriever.

⁹<https://github.com/kamalkraj/BERT-NER>.

Analogous to other previous SOTA (Ye et al., 2021), we use the off-the-shelf entity linker ELQ (Li et al., 2020) on WebQSP. ELQ is an end-to-end entity linking system that performs mention detection and entity disambiguation jointly on questions in one pass of BERT (Devlin et al., 2019). It is trained on both Wikipedia and WebQSP data and obtains SOTA performance on WebQSP. We refer readers to Li et al. (2020) for more details.

A.2 Schema Retrieval Performance

Class	Overall	I.I.D.	Comp.	Zero.
ReTraCk	94.3	98.1	97.5	91.3
TIARA	95.8	99.6	97.9	93.4
Relation	Overall	I.I.D.	Comp.	Zero.
ReTraCk	88.4	95.3	91.0	84.3
TIARA	92.0	97.9	93.7	88.7

Table 8: Recall (%) of top-10 schema retrieval on the GrailQA validation set.

We report the schema retrieval performance for GrailQA in Table 8. TIARA’s top-10 schema recall outperforms ReTraCk (Chen et al., 2021), the highest-ranked system in the GrailQA leaderboard to apply a dense retriever for schema retrieval, across all three levels of generalization. Note that ReTraCk uses 100 classes and 150 relations for each question, which does not result in higher QA performance, as shown in Table 1 and Table 2.

B Performance on the WebQSP Validation Set

Method	F1	Hits@1
TIARA	76.7	74.5
w/o Schema	75.4	73.5
w/o ELF	65.3	64.0
w/o ELF & Schema	63.9	61.5

Table 9: F1 and hits@1 results (%) on the WebQSP validation set.

We report the performance on the WebQSP validation set, as shown in Table 9. WebQSP does not contain the official validation set, and the dataset splitting is described in Section 4.1.

C Settings and Hyperparameters

Experimental settings are defined along the lines of previous works (Gu et al., 2021; Chen et al.,

2021; Ye et al., 2021) and empirically tuned non-exhaustively as significant results are obtained. We manually tune the hyperparameters based on the validation set results. The criterion used to select among them is F1 score. Specifically, we search for the batch size and learning rate of the generator from [2, 4, 6, 8] and [1e-5, 3e-5, 5e-5].

The number of parameters in our models is 110M for BERT-base-uncased (Devlin et al., 2019) and 220M for T5-base (Raffel et al., 2020).

D Evaluation Details

We use official scripts to evaluate on both datasets¹⁰, except for hits@1 on WebQSP, as mentioned in Section 4.1.

¹⁰GrailQA script: <https://worksheets.codalab.org/bundles/0x2d13989c17e44690ab62cc4edc0b900d/>, and WebQSP script: <https://www.microsoft.com/en-us/download/details.aspx?id=52763>.