Pay More Attention to Relation Exploration for Knowledge Base Question Answering

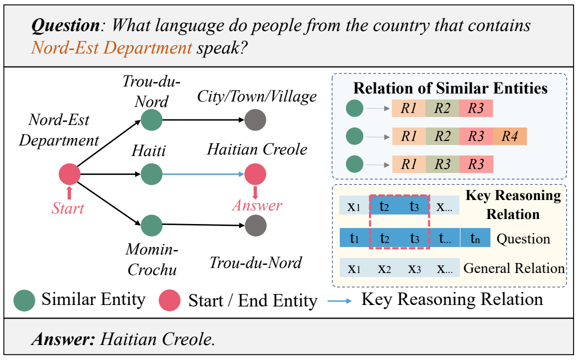
Yong Cao[1], **Xianzhi Li**[1][†], **Huiwen Liu**[2], **Wen Dai**[2], **Shuai Chen**[2],
Bin Wang[2], **Min Chen**[3], **and Daniel Hershcovich**[4]
[1] Huazhong University of Science and Technology [2]Xiaomi AI Lab, China.
[3]School of Computer Science and Engineering, South China University of Technology
[4]Department of Computer Science, University of Copenhagen
{yongcao_epic,xzli}@hust.edu.cn, minchen@ieee.org, dh@di.ku.dk
{liuhuiwen, daiwen, chenshuai3, wangbin11}@xiaomi.com

Abstract

Knowledge base question answering (KBQA) is a challenging task that aims to retrieve correct answers from large-scale knowledge bases. Existing attempts primarily focus on entity representation and final answer reasoning, which results in limited supervision for this task. Moreover, the relations, which empirically determine the reasoning path selection, are not fully considered in recent advancements. In this study, we propose a novel framework, RE-KBQA, that utilizes relations in the knowledge base to enhance entity representation and introduce additional supervision. We explore guidance from relations in three aspects, including (1) distinguishing similar entities by employing a variational graph auto-encoder to learn relation importance; (2) exploring extra supervision by predicting relation distributions as soft labels with a multi-task scheme; (3) designing a relation-guided re-ranking algorithm for post-processing. Experimental results on two benchmark datasets demonstrate the effectiveness and superiority of our framework, improving the F1 score by 5.8% from 40.5 to 46.3 on CWQ and 5.7% from 62.8 to 68.5 on WebQSP, better or on par with state-of-the-art methods.

1 Introduction

Given a question expressed in natural language, knowledge base question answering (KBQA) aims to find the correct answers from a large-scale knowledge base (KB), such as Freebase (Bollacker et al., 2008), Wikipedia (Vrandečić and Krötzsch, 2014), DBpeidia (Auer et al., 2007), etc. For example, the question "*Who is Emma Stone's father?*" can be answered by the fact of "(*Jeff Stone, person.parents, Emma Stone*)". The deployment of KBQA can significantly enhance a system's knowledge, improving performance for applications such as dialogue systems and search engines.


Question: *What language do people from the country that contains Nord-Est Department speak?*

Answer: *Haitian Creole.*

Figure 1: An example of KBQA process. The reasoning begins with the red node and passes through similar entities, which are defined as entities that have similar relations as shown in the upper right box. Besides, key reasoning relations whose tokens (x_i) hold overlap between given questions (t_i) are important for reasoning.

Early attempts on KBQA (Min et al., 2013; Zhang et al., 2018; Xu et al., 2019) mostly focus on transferring given questions into structured logic forms, which are strictly constrained by the consistent structure of parsed query and KB. To overcome the limitation of the incompleteness of KB, many approaches (Xiong et al., 2019; Deng et al., 2019; Lan et al., 2021) have been developed that aim to map questions and their related KB entities and relations into embeddings, and define the reasoning process as a similarity retrieval problem, which is called IR-based method. Additionally, some studies (Gao et al., 2022; Liu et al., 2023; Ge et al., 2022) have attempted to learn relation embeddings and then incorporate surrounding relations to represent entities, which successfully reduces the number of parameters needed for the model.

However, most of these works (Han et al., 2021) primarily focus on final answer reasoning and the representation of entities, while few explore the full utilization of relations in KB. Additionally, for answer reasoning, the supervision signal provided is also only from entities, while we believe that the relations also play an important role in determining the reasoning path and the answer choosing.

[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 2119–2136
July 9-14, 2023 ©2023 Association for Computational Linguistics

We propose a new framework, called Relation-Enhanced KBQA (RE-KBQA), to investigate the potential use of relations in KBQA by utilizing an embedding-fused framework. The proposed framework aims to study the role of relations in KBQA in the following three aspects:

Relations for entity representation. We find that similar entities with similar surrounding relations (e.g., the three green circles in the upper right of Figure 1) play an important role in reasoning. To distinguish them, we introduce QA-VGAE, a question-answering-oriented variational graph auto-encoder, which learns relation weights through global structure features and represents entities by integrating surrounding relations.

Relations for extra supervision. Multi-hop reasoning is often hindered by weak supervision, as models can only receive feedback from final answers (He et al., 2021). To overcome this limitation, we propose a multi-task scheme by predicting the relation distribution of the final answers as additional guidance, using the same reasoning architecture and mostly shared parameters. As illustrated in Figure 1, the proposed scheme requires the prediction of both the answer "Haitian Creole" and its surrounding relation distribution.

Relations for post-processing. We propose a stem-extraction re-ranking (SERR) algorithm to modify the confidence of candidates, motivated by the fact that relations parsed from given questions are empirically associated with strong reasoning paths. As depicted in the bottom of Figure 1, relations that overlap with a given question will be marked as key reasoning relations, and their confidence will be increased empirically. This allows for re-ranking and correction of the final answers.

In general, our contributions can be summarized as follows. (1) We propose a novel method named Relation Enhanced KBQA (RE-KBQA) by first presenting QA-VGAE for enhanced relation embedding. (2) We are the first to devise a multi-task scheme to implicitly exploit more supervised signals. (3) We design a simple yet effective post-processing algorithm to correct the final answers, which can be applied to any IR-based method. (4) Lastly, we conduct extensive experiments on two challenging benchmarks, WebQSP and CWQ to show the superiority of our RE-KBQA over other competitive methods. Our code and datasets are publicly available on Github[1].

2 Related Work

Knowledge Base Question Answering. Most existing research on KBQA can be categorized into two groups: a). Semantic Parsing (SP)-based methods (Abdelaziz et al., 2021; We et al., 2021; Cui et al., 2022), which transfer questions into logical form, e.g., SPARQL queries, by entity extraction, KB grounding, and structured query generation. b). Information Retrieval (IR)-based method (Ding et al., 2019; Chen et al., 2019; Wang et al., 2021; Feng et al., 2021; Zhang et al., 2022b), which applies retrieve-and-rank mechanism to reason and score all candidates of the subgraph with advancements in representation learning and ranking algorithms. Apart from the above approaches, recent studies (Xiong et al., 2019; Deng et al., 2019; Lan et al., 2021) also propose several alterations over the reasoning process, such as extra corpus exploration (Xiong et al., 2019), better semantic representation (Zhu et al., 2020; Ge et al., 2021), dynamic representation (Han et al., 2021), and intermediate supervised signals mining (Qiu et al., 2020; He et al., 2021). Aiming to tackle limited corpus, some works are devoted to utilizing external resources, such as using pre-trained language models (Unik-QA) (Oguz et al., 2022), retrieving similar documents (CBR-KBQA) (Das et al., 2021), extra corpus (KQA-Pro) (Cao et al., 2022), etc.

Multi-task Learning for KBQA. Multitask learning can boost the generalization capability on a primary task by learning additional auxiliary tasks (Liu et al., 2019) and sharing the learned parameters among tasks (Hwang et al., 2021; Xu et al., 2021). Many recent works have shown impressive results with the help of multi-task learning in many weak supervised tasks such as visual question answering (Liang et al., 2020; Rajani and Mooney, 2018), sequence labeling (Rei, 2017; Yu et al., 2021), text classification (Liu et al., 2017; Yu et al., 2019) and semantic parsing (Hershcovich et al., 2018). In KBQA, auxiliary information is often introduced in the form of artificial "tasks" relying on the same data as the main task (Hershcovich et al., 2018; Ansari et al., 2019; Gu et al., 2021), rather than independent tasks. This assists the reasoning process and proves to be more effective for the main task. To the best of our knowledge, we are the first to propose a multi-task to assist KBQA by using mostly shared parameters among tasks, for a balance of effectiveness and efficiency.

[1] github.com/yongcaoplus/RE-KBQA

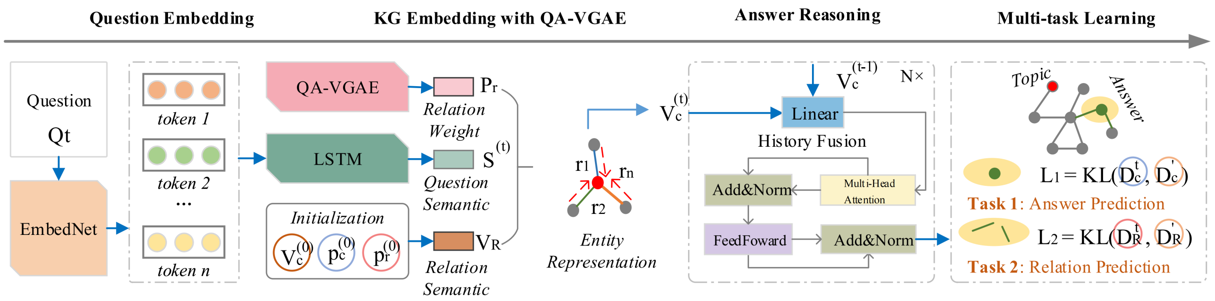


Figure 2: Framework of our proposed approach RE-KBQA. Given a question expressed in natural language, we first employ *question embedding* to encode semantic vectors. Then, we employ *QA-VGAE enhanced representation* module to learn candidate vectors $V_c^{(t)}$, aiming to identify similar entities and key reasoning paths while reasoning. At last, a *multi-task learning* module is proposed to promote training procedure.

3 Problem Formulation

Knowledge Base (KB). A knowledge base usually consists of a huge amount of triples: $\mathcal{G} = \{\langle e, r, e' \rangle | (e, e') \in \xi, r \in \mathcal{R}\}$, where $\langle e, r, e' \rangle$ denotes a triple with head entity e, relation r and tail entity e'. ξ and $\mathcal{R}$ mean the sets of all entities and relations, respectively. To apply the triples to downstream task, the entities and relations should be firstly embedded as d-dimensional vectors: $V = \{\langle V_e, V_r, V_{e'} \rangle | (V_e, V_{e'}) \in V_\xi, V_r \in V_\mathcal{R}\}$.

Knowledge Base Question Answering (KBQA). Our dataset is formed as question-answer pairs. Let Q represents the set of given questions and each question q is composed of separated tokens, where $Q = \{q \in Q | q = x_1, x_2, ..., x_n\}$. Let $\mathcal{A}$ $(\subseteq \xi)$ represents the correct answers of Q. Thus, the dataset is formulated as $\mathcal{D} = \{(Q, A) | (q_1, a_1), (q_2, a_2), ..., (q_m, a_m)\}$. To reduce the complexity of reasoning process, we extract question-related head entities e_h from q and generate an associated subgraph g_{sub} $(\in \mathcal{G}_{sub})$ within multi-hops walking from e_h. Thus, the goal of KBQA is transformed to reason the candidates c $(\subseteq \xi)$ of the highest confidence from g_{sub}, which can be formalized as:

$$c = \underset{\theta,\phi}{\arg\max}\, r_\phi(f_\theta(q, g_{sub})), \tag{1}$$

where $f_\theta(\cdot)$ and $r_\phi(\cdot)$ denote the representation and reasoning network, respectively.

4 Our Approach

As discussed in Section 1, we consider three aspects to further boost the performance of KBQA, including (i) the enhancement of the representation capability, especially for similar entities; (ii) a strategy of mining more supervision signals to guide the training; and (iii) a reasoning path correction algorithm to adjust the ranking results. Below, we shall elaborate on our network architecture (RE-KBQA) with our solutions to the above issues.

4.1 Architecture Overview

Inspired by the neighborhood aggregation strategy, we employ Neural State Machine (NSM) (He et al., 2021) as our backbone model, where entities are denoted by surrounding relations. We assume that the topic entities and the related subgraph are already achieved by preprocessing; see Section 5.2 for the details. Figure 2 shows the main pipeline of our RE-KBQA. Specifically, given a question q, we first employ a question embedding module to encode it into semantic vector. Here, for a fair comparison with NSM baseline, we follow (He et al., 2021) to adopt Glove (Pennington et al., 2014) to encode q into embeddings $\{V_q^j\}_{j=1}^n = \text{Glove}(x_1, x_2, ..., x_n)$, which is then mapped to hidden states by LSTM:

$$\{h', \{h_j\}_{j=1}^n\} = \text{LSTM}(V_q^1, V_q^2, ..., V_q^n), \tag{2}$$

where we set h' as the last hidden state of LSTM to denote question vector and $\{h_j\}_{j=1}^n$ denotes the vector of tokens. After obtaining h' and $\{h_j\}_{j=1}^n$, then we can calculate :

$$q^{(t)} = \psi(s^{(t-1)}, h'), \tag{3}$$

where $\psi(\cdot)$ denotes multi-layer percetron function. Then, the semantic vector $s^{(t)}$ at the t-th reasoning step of question q is obtained by:

$$s^{(t)} = \sum_{j=1}^{n} p(\psi(q^{(t)}, h_j)) \cdot h_j, \tag{4}$$

where $p(\cdot)$ denotes score function, and $s^{(0)}$ ($\in \mathbb{R}^{(|d|)}$) is initialized randomly.

Next, a *QA-VGAE enhanced representation* module is designed to represent KB elements under the guidance of $s^{(t)}$. Then, unlike previous works that directly predict final answer via a score function, we introduce a *multi-task learning-fused reasoning* module to further predict an auxiliary signal (i.e., relation distribution). Note that, though we adopt NSM framework to conduct KBQA task, we concentrate on the representation capability enhancement by identifying similar entities, as well as the multi-task learning via supervision signal mining. At last, to avoid ignoring strong reasoning paths, we further propose a *stem-extraction re-ranking* algorithm to post-process the predictions of our network. Below, we will present the details of three of our proposed contributed modules.

4.2 QA-VGAE Enhanced Representation

Similar entities are defined as entities that are connected mostly by the same edges, and only a small portion of edges are different. For example, as shown in Figure 1, the three nodes marked by dashed circles share almost the same edges, and only the node of "Haiti" holds the relation of "*Person.Spoken_language*" that is quite important for answering the question. Hence, distinguishing similar entities and identifying key reasoning paths are essential for embedding-fused information retrieval-based methods. Traditional methods like TransE (Bordes et al., 2013) can grasp local information from independent triples within a KB, but fail to capture the inter-relations between adjacent triple facts. Consequently, they tend to have difficulties in distinguishing similar entities.

To alleviate the above problem, we introduce Question Answering-oriented Variational Graph Auto-Encoder (QA-VGAE) module, as is shown in Figure 3, by assigning different weights to reasoning relations, where the weights are learned by VGAE (Kipf and Welling, 2016). Note that, compared with traditional methods like TransE (Bordes et al., 2013), TransR (Lin et al., 2015), and ComplEx (Trouillon et al., 2016), VGAE achieves superior performance in link prediction task. We thus adopt VGAE in our module to learn weights. The key insight of this module is to fully learn global structure features by executing graph reconstruction task and constraining the representation as normal distribution, thus promoting the relation representation to be more discriminating. Finally, by similarity evaluation of the learned representation, we can obtain the prior probability of relation (PPR) matrix, whose elements denote the conditional probability of relations.

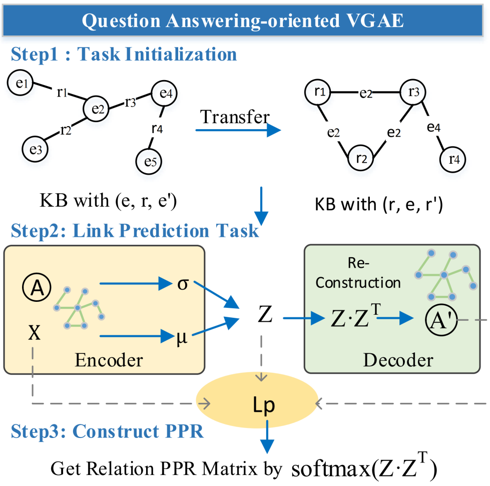


Figure 3: Illustration of training QA-VGAE, including a total of three steps. We adopt two-layers GCN as encoder formalized as GCN_σ and GCN_μ.

In detail, we first transfer the KB from $\langle e, r, e' \rangle$ (entity-oriented) to $\langle r, e, r' \rangle$ (relation-oriented). In this way, we can then learn PPR matrix via a link prediction task by unsupervised learning. Specifically, given the connection degrees X ($\in \mathbb{R}^{|n_r| \times |n_r|}$) of a relation and the adjacency A ($\in \mathbb{R}^{|n_r| \times |n_r|}$) between relation nodes, where n_r denotes the number of relations, we adopt two-layers GCN to learn the mean σ and variance μ of the relation importance distribution, and further compound the relation representation Z as :

$$Z = \mathrm{GCN}_\mu(X, A) \oplus \mathrm{GCN}_\sigma(X, A), \qquad (5)$$

where $\oplus$ is compound function. Then, PPR matrix $\mathcal{P}_r$ is obtained by distribution similarity evaluation:

$$\mathcal{P}_r = \mathrm{Softmax}\big(Z \cdot Z^\top\big), \qquad (6)$$

where $\mathcal{P}_r \in \mathbb{R}^{|n_r| \times |n_r|}$. Please refer to Appendix A.1 for loss function $\mathcal{L}_P$ of QA-VGAE. Next, we denote KB elements as d-dim vectors, $V_\mathcal{E}(\in \mathbb{R}^{|n_e| \times |d|})$ as entity vectors and $V_\mathcal{R}(\in \mathbb{R}^{|n_r| \times |d|})$ as relation vectors, where n_e is the number of enities. We denote candidate vectors $V_\mathcal{C}$ as:

$$V_\mathcal{C} = W_\mathcal{C} \cdot \mathcal{P}_r \cdot V_\mathcal{R}, \qquad (7)$$

where $W_\mathcal{C} \in \mathbb{R}^{|n_c| \times |n_r|}$ denotes the surrounding relation matrix of entities and n_c denotes number of candidates.

Then, to integrate semantic vectors $s^{(t)}$ of given question and the history vector, we update V_c as:

$$\hat{V}_c^{(t)} = \sigma([V_c^{(t-1)}; s^{(t)} \odot W_r \odot V_c]), \tag{8}$$

where $V_c^{(t)}$ ($\in V_{\mathcal{C}}$) is candidate vector at time step t, $\sigma(\cdot)$ is the linear layer, $[;]$ is the concatenation operation, $\odot$ is element-wise multiplication, and W_r ($\in \mathbb{R}^{|d|}$) is the matrix of learnable parameter.

4.3 Multi-task Learning-Fused Reasoning

The purpose of this module is to conduct answer reasoning from candidate vector $\hat{V}_c^{(t)}$. To this end, we jointly combine the reasoning paths implicitly among candidates by utilizing the Transformer (Vaswani et al., 2017), formalized as:

$$V_c^{(t)} = \text{Transformer}\left(\left[\hat{V}_{c_1}^{(t)}; \hat{V}_{c_2}^{(t)}; ...; \hat{V}_{c_l}^{(t)}\right]\right), \tag{9}$$

where $\{\hat{V}_{c_i}^{(t)}\}_{i=1}^{l}$ denotes all the candidate vectors.

However, like most existing works (Deng et al., 2019; Lange and Riedmiller, 2010), learning from the final answers as the feedback tends to make the model hard to train, due to the limited supervision. How to introduce extra supervision signals into network model is still an open question. In our method, we introduce a new multi-task to learn the distribution of candidates' surrounding relations, namely surrounding relations reasoning. The key idea is to leverage relations around final answer as extra supervisions to promote the performance, and also modify reasoning paths implicitly.

Specifically, motivated by weakly-supervised learning methods, we assume the reasoning process starts from topic entity's surrounding relations $S_R^{(0)}$ (initialized along with subgraph generation), and during reasoning, we can easily obtain next surrounding relations' distribution by:

$$S_R^{(t)} = \sigma\left(\left[(s^{(t)} \cdot V_{\mathcal{R}}^{\top(t)}; S_R^{(t-1)}\right]\right), \tag{10}$$

where $S_R^{(t)}$ denotes the surrounding relations of candidates at step t and $V_{\mathcal{R}}^{\top(t)}$ is the transpose of $V_{\mathcal{R}}$ at step t. Note that, introducing the multi-task will not increase the complexity of our method obviously, since the number of relations is far fewer than that of entities in most cases, and the multi-task shares most parameters with the main task.

In this way, there are two optimization goals of KBQA task, i.e., correct answer retrieving and surrounding relations prediction. We predict the final answers' possibilities by:

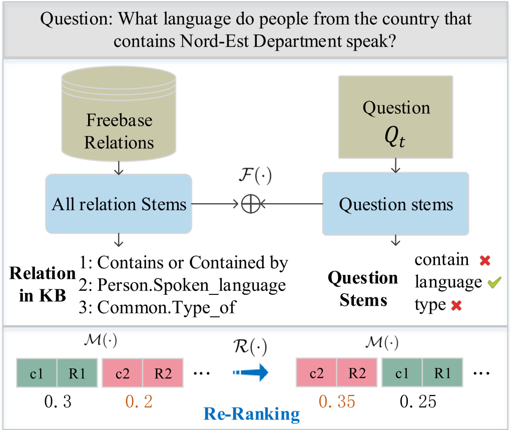


Figure 4: Illustration of SERR algorithm, where stem match mechanism is introduced between KB relations and given questions. If key reasoning relations exist, the rank of candidates will be increased.

$$p_c^{(t)} = \text{Softmax}\left(V_c^{(t)} \cdot W_c^{(t)}\right), \tag{11}$$

where $p_c^{(t)}$ is the confidence of predicted answers. Also, the relation distribution confidence $p_r^{(t)}$ is:

$$p_r^{(t)} = \text{Softmax}\left(S_R^{(t)} \cdot W_r^{(t)}\right), \tag{12}$$

where $W_c^{(t)}$ and $W_r^{(t)}$ are learnable parameters.

Then, the answer retriving loss $\mathcal{L}_c$ and the relation prediction loss $\mathcal{L}_r$ can be calculated by:

$$\begin{aligned} \mathcal{L}_c =& \text{KL}(p_c^{(t)}, p_c^{(*)}) \\ \mathcal{L}_r =& \text{KL}(p_r^{(t)}, p_r^{(*)}), \end{aligned} \tag{13}$$

where ${p_c}^{(*)}$ and ${p_r}^{(*)}$ denote the ground truths, KL is the KL divergence. Thus, the final total loss is:

$$\mathcal{L} = \lambda \mathcal{L}_c + (1 - \lambda)\mathcal{L}_r, \tag{14}$$

where λ denotes a hyper-parameter.

4.4 Stem-Extraction Re-Ranking

A limitation of embedding-fused KBQA methods is that the reasoning path is uncontrollable as the complete reasoning path is a blackbox in information retrieval-based methods. For example, in the question "*What is the Milwaukee Brewers mascot?*", the strongly related path "*education.mascot*" may be missed due to limited representation capability. However, this weakness can be easily addressed by semantic parsing-based methods by analyzing the semantic similarity of key elements of questions and relations and constraining the reasoning path. Inspired by this observation, we propose a stem-extraction re-ranking (SERR) algorithm for post-processing. The key idea is to stem-match and re-rank the candidates after obtaining candidates and their confidence from our network.

In detail, we design three operators to execute the re-ranking as shown in Algorithm 1: stemmer $\mathcal{F}(\cdot)$, modifier $\mathcal{M}(\cdot)$, and re-ranker $\mathcal{R}(\cdot)$. These operators are used to extract stems from relations or given questions, modify candidates' confidence, and then re-rank the candidates. As shown in Figure 4, given question and candidate predictions, we first use $\mathcal{F}(\cdot)$ to process all the relations of freebase relations and questions. Then, we generate a relation candidates pool by matching the stem pool of the question with the relation stems. This allows us to compare the subgraph of the given question with pseudo-facts produced by given topic entities and candidates, respectively. Finally, according to the comparison, $\mathcal{M}(\cdot)$ and $\mathcal{R}(\cdot)$ are employed to conduct the re-ranking process.

It is worth noting that, in our work, we directly use stem extraction method rather than similarity calculation to re-rank. The insight behind this choice is that, it is unnecessary to consider semantic features again, since we have already injected the question semantic information into our encoded semantic vector $s^{(t)}$, which means that the model is already equipped with semantic clustering capability. And obviously, stem extraction costs fewer computation resources, as proved in Appendix A.2. Also, our SERR can be migrated to other models as a plug-in and independent module.

5 Experiments and Results

5.1 Datasets

We conduct experiments on two popular benchmark datasets, including WebQuestionSP (Yih et al., 2015) and ComplexWebQuestions (Talmor and Berant, 2018). Specifically, WebQuestionSP (abbr. WebQSP) is composed of simple questions that can be answered within two hops reasoning, which is constructed based on Freebase (Bollacker et al., 2008). In contrast, ComplexWebQuestions (abbr. CWQ) is larger and more complicated, where the answers require multi-hop reasoning over several KB facts. The detailed statistics of the two datasets are summarized in Table 1.

Algorithm 1 Stem Extraction Re-Ranking

```
Input: natural language question $Q$, candidates $\mathcal{C}$, confidence $p_{\mathcal{C}}$, relation set $\mathcal{R}$.
Output: updated candidates $\mathcal{C}'$ and confidence $p'_{\mathcal{C}}$.
<* Step 1: Build Relation Trie $\mathcal{P}_s$ *>
$\emptyset \rightarrow \mathcal{P}_s$
for all $r$ in $\mathcal{R}$ do
    index $i$, stem $s = \mathcal{F}(r)$
    $\mathcal{P}_s.update(\langle i, s \rangle)$
end for
for all $\{q, c, p_{\mathcal{C}}\}$ in $\{Q, \mathcal{C}, p_{\mathcal{C}}\}$ do
    <* Step 2: Extract Stem of $Q$*>
    tokenize $q \rightarrow \mathcal{P}_q$
    $\mathcal{F}(\mathcal{P}_q) \rightarrow \mathcal{P}^e_{stem}$
    <* Step 3: Re-Ranking $c$ and $p_c$*>
    $r_c = \text{match}(\mathcal{P}^e_{stem}, \mathcal{P}_s)$
    generate $P = \langle e, \mathcal{P}_s(r_c), e' \rangle$
    generate $P' = \langle e, \mathcal{P}_s(r_c) \rangle \cup \langle e', \mathcal{P}_s(r_c) \rangle$
    for all $p$ in $P \cup P'$ do
        if $p$ in $g_{sub}$ and $p$ in $P$ then
            $\mathcal{M}(p_c, h_1)$
        end if
        if $p$ in $g_{sub}$ and $p$ in $P'$ then
            $\mathcal{M}(p_c, h_2)$
        end if
    end for
    $c' = \mathcal{R}(c)$ and $p'_c = \mathcal{R}(p_c)$
end for
```

Dataset	Train	Valid	Test	Entities	Relations
WebQSP	2,848	250	1,639	259,862	6,105
CWQ	27,639	3,519	3,531	598,564	6,649

Table 1: Statistics of WebQSP and CWQ datasets. Note that, *Entities* and *Relations* denote all the entities and relations covered in the subgraph respectively.

5.2 Experimental Setting

Basic setting. To make a fair comparison with other methods, we follow existing works (Sun et al., 2019, 2018; He et al., 2021) to process datasets, including candidates generation by PageRank-Nibble algorithm and subgraph construction within three-hops by retrieving from topic entities. We set the learning rate as $8e-4$ and decay it linearly throughout iterations on both datasets. We set the number of training epoch on WebQSP and CWQ as 200 and 100, respectively. For better reproducibility, we give all the parameter settings in Appendix A.3.

Baselines. We compare our method with multiple representative methods, including semantic pars-

Models	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
SP-Based Method				
SPARQA* (Sun et al., 2020)	-	-	31.6	-
QGG* (Lan and Jiang, 2020)	-	74.0	44.1	40.4
GNN-KBQA* (Hou et al., 2022)	68.5	68.9	-	-
IR-Based Method				
KV-Mem† (Miller et al., 2016)	46.6	34.5	18.4	15.7
EmbKGQA† (Saxena et al. 2020)	66.6	-	32.0	-
GraftNet† (Sun et al., 2018)	66.4	60.4	36.8	32.7
PullNet* (Sun et al. 2019)	68.1	-	45.9	-
ReTraCk* (Chen et al., 2021)	71.6	71.0	-	-
NSM† (He et al., 2021)	68.5	62.8	46.3	42.4
BiNSM* (He et al., 2021)	74.3	67.4	48.8	44.0
SR-KBQA* (Zhang et al., 2022a)	69.5	64.1	50.2	**47.1**
RNG-KBQA* (Ye et al., 2022)	-	**75.6**	-	-
Ours				
RE-KBQA$_b$	68.7	62.8	46.8	40.5
RE-KBQA	**74.6**	68.5	**50.3**	46.3

Table 2: Performance comparison over state-of-the-art IR-based approaches on WebQSP and CWQ datasets, where bold fonts denote the best scores, * denotes scores from original paper and † are from Zhang et al. (2022a).

ing (SP)-based methods and information retrieval (IR)-based methods. SPARQA (Sun et al., 2020) and QGG (Lan and Jiang, 2020) belong to the former category, which focuses on generating optimal query structures. Besides, KV-Mem (Miller et al., 2016), EmbedKGQA (Saxena et al., 2020), GraftNet (Sun et al., 2018), PullNet (Sun et al., 2019), ReTraCk (Chen et al., 2021) and BiNSM (He et al., 2021) are all IR-based methods, which are also the focus of our comparison.

Evaluation metrics. To fully evaluate KBQA performance, we should compare both the retrieved and ranked candidates with correct answers. To this end, we employ the commonly-used F1 score and Hit@1. F1 score measures whether the retrieved candidates are correct, while Hit@1 evaluates whether the ranked candidate of the highest confidence is in answer sets.

5.3 Comparison with Others

We first compare our RE-KBQA against the aforementioned baselines on two datasets and the results are reported in Table 2. Note that, RE-KBQA$_b$ indicates our backbone network without three modules, i.e., QA-VGAE, multi-task learning and SERR. Clearly, even using our backbone network, it already outperforms most baselines on two datasets, which is benefited from the semantic guidance of given questions and the reasoning mechanisms. Further, as shown in the bottom row, our full pipeline achieves the highest values on both datasets over both evaluation metrics.

Different cases	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
RE-KBQA$_b$	68.7	62.8	46.8	40.5
with QA-VGAE	73.4	67.7	48.2	45.0
	4.7 ↑	4.9 ↑	1.4 ↑	4.5 ↑
with AxLr	72.4	68.4	47.7	42.5
	3.7 ↑	5.6 ↑	0.9 ↑	2.0 ↑
with SERR	72.0	65.5	47.3	41.5
	3.3 ↑	2.7 ↑	0.5 ↑	1.0 ↑
RE-KBQA	**74.6**	**68.5**	**50.3**	**46.3**
	5.9 ↑	5.7 ↑	3.5 ↑	5.8 ↑

Table 3: Comparing our full pipeline (bottom row) with various cases in the ablation study. The cells with different background colors reveal the improvement over our backbone network RE-KBQA$_b$.

Particularly, compared with the results produced by RE-KBQA$_b$, our full method improves more on CWQ dataset, which has increased by 3.5 and 5.8 in terms of Hit@1 and F1, showing that our contributions can indeed boost the multi-hop reasoning process. Besides, RE-KBQA also obtains good results on simple questions (i.e., WebQSP dataset), especially a 5.7 increase in F1 score, which reveals that the model can recall more effective candidates.

As shown in Table 2, we can observe that the SP-based methods (i.e., SPARQA and QGG) show a good performance in WebQSP, but perform worse in complicated questions, which reveals that SP-based methods are still weak in multi-hop reasoning. Similarly, traditional embedding methods, i.e., KV-Mem, EmbedKGQA, and GraftNet, also perform better in simple questions than in complex ones. Though PullNet and BiNSM show good multi-hop reasoning capacity, the extra corpora analysis and bi-directional reasoning mechanism inevitably increase the complexity of these networks.

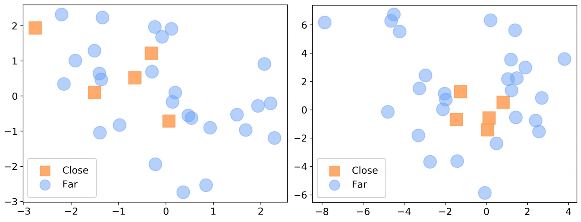
Apart from above methods, some attempts are conducted on utilizing additional resources for task enhancement recently. As shown in Table 2 *reference*, CBR-KBQA relies on expensive large-scale extra human annotations and Roberta pretrained model (PLM), Unik-QA tries to retrieve one-hundred extra context passages for relations in KB and T5-base (PLM), and KQA-Pro uses a large-scale dataset for pre-training with the help of explicit reasoning path annotation. While promising performance has been achieved through these methods, expensive human annotation costs and model efficiency also need to be concerned.

5.4 Network Component Analysis

To evaluate the effectiveness of each major component in our method, we conducted a comprehensive ablation study. In detail, similar to Section 5.3, we remove all three components and denote the backbone network as RE-KBQA$_b$. Then, we add QA-VGAE (Section 4.2), multi-task learning (Section 4.3), and SERR (Section 4.4) back on RE-KBQA$_b$, respectively. In this way, we constructed totally four network models and re-trained each model separately using the same settings of our RE-KBQA model. Table 3 shows the results. By comparing different cases with the bottom-most row (our full pipeline), we can see that each component contributes to improving the performance on both datasets. More ablation experiments can be found in Appendix. Below, we shall discuss the effect of each module separately.

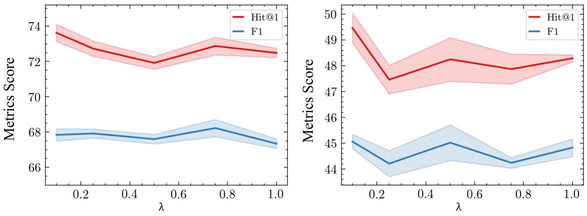
Effect of QA-VGAE. From the results of Table 3, we can observe that the improvements of using QA-VGAE are more remarkable than using the other two modules, demonstrating that the QA-VGAE is more helpful to boost the reasoning process for both simple and complex questions. Besides quantitative comparison, we also tried to reveal its effect in a visual manner. Here, we adopt T-SNE to visualize the relation vectors. Figure 5 shows a typical embedding distribution before and after QA-VGAE training. For a clear visualization, we randomly select some relations related to a case "*What is the capital of Austria?*". The orange nodes represent relations close to "*location*", such as "*location.country.capital*", "*location.country.first_level_divisions*", etc., and the blue nodes denote the relations that are not covered by the question subgraph, which we call far relations. Obviously, after using QA-VGAE, the related relations (orange nodes in (b)) tend to get closer and the other nodes get farther.

Effect of multi-task learning. As shown in Table 3, the multi-learning module shows better performance in simple questions (see WebQSP dataset), since the relation distribution is denser than candidates distribution, thus causing the prediction to be more complicated along with the increase of reasoning steps. To fully explore the effect of this module, we study different loss fusion weights and the results are shown in Figure 6, where a larger λ (range from 0.1 to 1.0, and we discard the setting of 0.0 for its bad performance) denotes a more weighted loss of main task. Clearly, only designing the primary task or auxiliary task is not optimal for KBQA, and the best setting of λ is 0.1 and 0.5 for the two datasets. An interesting observation is that the best Hit@1 is obtained with lower lambda while the best F1 score is obtained with higher lambda in each dataset. We claim that it is caused by the different goals of Hit@1 and F1 metrics, that is, Hit@1 shows whether the top one candidate is found while F1 score evaluates whether most candidates are found.



(a) Distribution-Before (b) Distribution-After

Figure 5: Relation vector visualization in the case of *"What is the capital of Austria?"* via T-SNE. Orange nodes indicate relations close to "*location*" and blue nodes indicate far relations.



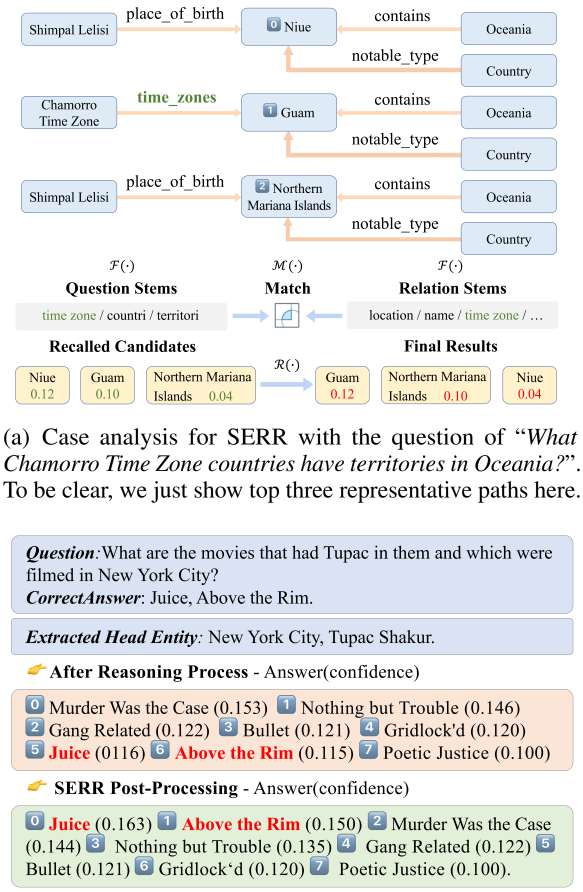
(a) WebQSP Dataset (b) CWQ Dataset

Figure 6: Analysis of using different loss fusion weights among two benchmark test sets in multi-task learning.

Effect of SERR. This module is lightweight (see Appendix A.2 for inference time) yet effective, especially for simple questions; see Table 3. Intuitively, the stem extraction for key paths is quite effective for questions that rely on direct-connected facts. In contrast, stem extraction for complex questions relies more on the startpoint and endpoint. Figure 7(a) further shows an example result of SERR module, which proves that it can effectively identify close connected facts of a given question and re-rank the candidates.

5.5 Case Study

At last, we show a case result produced by our RE-KBQA; see Figure 7(b). Given the question "What are the movies that had Tupac in them and which were filmed in New York City?", our method first



(a) Case analysis for SERR with the question of "*What Chamorro Time Zone countries have territories in Oceania?*". To be clear, we just show top three representative paths here.

Question: What are the movies that had Tupac in them and which were filmed in New York City?
CorrectAnswer: Juice, Above the Rim.

Extracted Head Entity: New York City, Tupac Shakur.

After Reasoning Process - Answer(confidence)

0 Murder Was the Case (0.153) 1 Nothing but Trouble (0.146) 2 Gang Related (0.122) 3 Bullet (0.121) 4 Gridlock'd (0.120) 5 **Juice** (0116) 6 **Above the Rim** (0.115) 7 Poetic Justice (0.100)

SERR Post-Processing - Answer(confidence)

0 **Juice** (0.163) 1 **Above the Rim** (0.150) 2 Murder Was the Case (0.144) 3 Nothing but Trouble (0.135) 4 Gang Related (0.122) 5 Bullet (0.121) 6 Gridlock'd (0.120) 7 Poetic Justice (0.100).

(b) Case analysis for RE-KBQA with proposed modules.

Figure 7: Case analysis of multi-hop reasoning process.

embedded the question into vectors and retrieve related subgraphs. Then, by utilizing the promotion of our proposed QA-VGAE and multi-task learning, we can use the trained model and obtain the candidates of "*Murder Was the Case*", "*Nothing but Trouble*", etc, and thanks to the SERR algorithm, our reasoning process can have a chance to re-rank the candidates, thus boosting its performance. Finally, we output *Juice* and *Above the Rim* as the correct answers. For similarity entity identification, SERR in other methods as a plug-in and more case results, please refer to Appendix A.2 and A.5.

6 Conclusion

In this paper, we proposed a novel framework, namely RE-KBQA, with three novel modules for knowledge base question answering, which are QA-VGAE to explore the relation promotion for entity representation, multi-task learning to exploit relations for more supervisions, and SERR to post-process relations to re-rank candidates. Extensive experiments validate the superior performance of our method compared with state-of-the-art IR-based approaches.

7 Limitations

While good performance has been achieved, there are still limitations in our work. First, though QA-VGAE extracts enhanced features and are fast to train, it is an independent module from the main framework. Second, as a post-processing step, the performance of SERR module on simple question is better than that of complex questions.

In the future, we would like to explore the possibility of fusing relation constraints into the representation module directly and inject strong facts identification mechanism as guidance signal of multi-hop reasoning process, aiming to integrate QA-VGAE and SERR into the main framework.

Acknowledgments

Thanks to the anonymous reviewers for their helpful feedback. We gratefully acknowledge the insightful suggestions from Zeqi Tan. This work is supported by the China National Natural Science Foundation No. 62202182. Yong Cao is supported by China Scholarship Council (No. 202206160052) and the Zhejiang Lab's International Talent Fund for Young Professionals.

References

Ibrahim Abdelaziz, Srinivas Ravishankar, Pavan Kapanipathi, Salim Roukos, and Alexander Gray. 2021. A semantic parsing and reasoning-based approach to knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15985–15987.

Ghulam Ahmed Ansari, Amrita Saha, Vishwajeet Kumar, Mohan Bhambhani, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2019. Neural program induction for kbqa without gold programs or query annotations. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence(IJCAI)*, pages 4890–4896.

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

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Shulin Cao, Jiaxin Shi, Zijun Yao, Xin Lv, Jifan Yu, Lei Hou, Juanzi Li, Zhiyuan Liu, and Jinghui Xiao. 2022. Program transfer for answering complex questions over knowledge bases. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8128–8140, Dublin, Ireland. Association for Computational Linguistics.

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

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

Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. 2022. Compositional generalization in multilingual semantic parsing over Wikidata. *Transactions of the Association for Computational Linguistics*, 10:937–955.

Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. 2021. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9594–9611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Nan Du, Wei Fan, Kai Lei, and Ying Shen. 2019. Multi-task learning with multi-view attention for answer selection and knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6318–6325.

Jiwei Ding, Wei Hu, Qixin Xu, and Yuzhong Qu. 2019. Leveraging frequent query substructures to generate formal queries for complex question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2614–2622, Hong Kong, China. Association for Computational Linguistics.

Yu Feng, Jing Zhang, Gaole He, Wayne Xin Zhao, Lemao Liu, Quan Liu, Cuiping Li, and Hong Chen. 2021. A pretraining numerical reasoning model for ordinal constrained question answering on knowledge base. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1852–1861, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yunjun Gao, Xiaoze Liu, Junyang Wu, Tianyi Li, Pengfei Wang, and Lu Chen. 2022. Clusterea: Scalable entity alignment with stochastic training and normalized mini-batch similarities. In *KDD*, pages 421–431.

Congcong Ge, Xiaoze Liu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2021. Make it easy: An effective end-to-end entity alignment framework. In *SIGIR*, pages 777–786.

Congcong Ge, Xiaoze Liu, Lu Chen, Baihua Zheng, and Yunjun Gao. 2022. Largeea: Aligning entities for large-scale knowledge graphs. *PVLDB*, 15(2):237–245.

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

Jiale Han, Bo Cheng, and Xu Wang. 2021. Two-phase hypergraph based reasoning with dynamic relations for multi-hop kbqa. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3615–3621.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 553–561.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 373–385, Melbourne, Australia. Association for Computational Linguistics.

Xia Hou, Jintao Luo, Junzhe Li, Liangguo Wang, and Hongbo Yang. 2022. A novel knowledge base question answering method based on graph convolutional network and optimized search space. *Electronics*, 11(23):3897.

Dasol Hwang, Jinyoung Park, Sunyoung Kwon, Kyung-Min Kim, Jung-Woo Ha, and Hyunwoo J Kim. 2021. Self-supervised auxiliary learning for graph neural networks via meta-learning. *arXiv preprint arXiv:2103.00771*.

Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv preprint arXiv:2105.11644*.

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

Sascha Lange and Martin Riedmiller. 2010. Deep autoencoder neural networks in reinforcement learning. In *The 2010 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Zujie Liang, Weitao Jiang, Haifeng Hu, and Jiaying Zhu. 2020. Learning to contrast the counterfactual samples for robust visual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3285–3292, Online. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*.

Shikun Liu, Andrew Davison, and Edward Johns. 2019. Self-supervised generalisation with meta auxiliary learning. *Advances in Neural Information Processing Systems*, 32.

Xiaoze Liu, Junyang Wu, Tianyi Li, Lu Chen, and Yunjun Gao. 2023. Unsupervised entity alignment for temporal knowledge graphs. In *WWW*.

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

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, Atlanta, Georgia. Association for Computational Linguistics.

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

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Yunqi Qiu, Yuanzhuo Wang, Xiaolong Jin, and Kun Zhang. 2020. Stepwise reasoning for multi-relation question answering over knowledge graph with weak supervision. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 474–482.

Nazneen Fatema Rajani and Raymond Mooney. 2018. Stacking with auxiliary features for visual question answering. 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 2217–2226, New Orleans, Louisiana. Association for Computational Linguistics.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130, Vancouver, Canada. Association for Computational Linguistics.

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

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen.

2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1803.06643*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2021. Retrieval, re-ranking and multi-task learning for knowledge-base question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 347–357, Online. Association for Computational Linguistics.

Peiyun We, Yunjie Wu, Linjuan Wu, Xiaowang Zhang, and Zhiyong Feng. 2021. Modeling global semantics for question answering over knowledge bases. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264, Florence, Italy. Association for Computational Linguistics.

Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019. Enhancing key-value memory neural networks for knowledge based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2937–2947, Minneapolis, Minnesota. Association for Computational Linguistics.

Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. 2021. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6984–6993.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043, Dublin, Ireland. Association for Computational Linguistics.

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

Jiaxin Yu, Wenyuan Liu, Yongjun He, and Chunyue Zhang. 2021. A mutually auxiliary multitask model with self-distillation for emotion-cause pair extraction. *IEEE Access*, 9:26811–26821.

Shanshan Yu, Jindian Su, and Da Luo. 2019. Improving bert-based text classification with auxiliary sentence and domain knowledge. *IEEE Access*, 7:176600–176612.

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

Jinhao Zhang, Lizong Zhang, Bei Hui, and Ling Tian. 2022b. Improving complex knowledge base question answering via structural information learning. *Knowledge-Based Systems*, page 108252.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Thirty-second AAAI conference on artificial intelligence*.

Shuguang Zhu, Xiang Cheng, and Sen Su. 2020. Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing*, 372:64–72.

A Appendix

A.1 QA-VGAE Training

In this section, we introduce the details of the QA-VGAE training procedure and demonstrate its effectiveness.

Training Goal. We adopt encoder-decoder models to conduct relation reconstruction tasks. Given the prepared adjacent matrix A, and feature matrix X, we use a two-layer GCN as a distribution learning model to estimate its mean and variance. The training loss function is formalized as:

$$\mathcal{L}_P = \mathbb{E}_{q(Z|X,A)}[\log p(A \mid Z)] - \mathrm{KL}(q(X, A), p(Z)) \tag{15}$$

where Z is calculated by Equation 5, KL is the Kullback-Leibler divergence, $q(\cdot)$ and $p(\cdot)$ denotes the encoder and decoder respectively, please refer to Kipf and Welling (2016) for more details.

Settings. Specifically, A is defined as the matrix of neighborhood relations between nodes, where we set $A(i, j)$ as 1 if there is a connection between relation r_i and r_j, and 0 for no connection. X is the feature matrix defined as the connectivity, which is accumulated as the number of edges between two nodes, aiming to show the importance of a relation. We set an empirical thresh of each element in the feature matrix to avoid extremely large values to hurt the model's training, such as the degrees of "*Common.type_of*" is quite huge, defined as:

$$X[i, j] = \begin{cases} \tau, & c \geq \tau, \\ c, & c < \tau. \end{cases} \tag{16}$$

where c is connectivity, τ is an emprical hyper-parameter, and we set τ as 2000 in our work.

A.2 SERR Algorithm

Complexity Analysis. Definitely, applying semantic similarity between relations and given questions is a more straightforward method to identify strong relations. However, the process of such a method is more complicated and time-consuming. To prove the efficiency of our method, we conduct a comparison experiment to reflect the complexity of the two methods. As is shown in Table 4, the top two rows denote semantic similarity method, and the last row denotes our method. Obviously, our method is more lightweight without extra pre-trained models and the dependence on GPU resources. For comparison, we adopt *Bert-base-uncased* model to conduct the semantic similarity process in this experiment, which can be downloaded in https://huggingface.co/bert-base-uncased.

Module	WebQSP			CWQ		
	Params	Time	GPU	Params	Time	GPU
Cosine Distance	420.10	28.8	√	420.10	53.5	√
Euclidean Distance	420.10	30.5	√	420.10	55.5	√
Stem Extraction	-	4.9	×	-	18.3	×

Table 4: Comparing SERR module with semantic similarity method, i.e., cosine distance and euclidean distance in terms of model parameters and computing resources. *Time* row denotes total handling time (*minutes*). *Params* row denotes model size (*MB*)

Performance Analysis. Besides, to demonstrate it can be plug-in and infer cases quickly, we further validate its accuracy and inference time, as is shown in Table 5, Note that, since SERR relies on traditional stem extraction rather than semantic understanding to identify the key paths, there is no training period for SERR, and it can be applied to any information-retrieval(IR)-based methods.

Finally, to demonstrate the plug-in attributes of the SERR module, we integrate this module into BiNSM network (He et al., 2021) and the results are shown in Table 6. The results show that SERR can indeed increase the Hit@1/F1 score from 74.3/67.4 to 74.8/68.0 in the WebQSP dataset, and from 48.8/44.0 to 49.5/45.3 in the CWQ dataset.

Factor	Webqsp	CWQ
Accuracy (%)	63.2	75.5
Infer Time (s)	0.18	0.32

Table 5: Performance of SERR algorithm in terms of accuracy score and inference time in two benchmark datasets. The accuracy score is calculated among recalled cases where close facts lie in its subgraph.

A.3 Hyper-parameter Setting.

In order to help reproduce RE-KBQA and its reasoning performance, as shown in Table 7, we list the hyper-parameters of the best results on two benchmark datasets. For the WebQSP dataset, the

Different cases	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
BiNSM	74.3	67.4	48.8	44.0
with SERR	74.8	68.0	49.5	45.3
	0.5 ↑	0.6 ↑	0.7 ↑	1.3 ↑

Table 6: Integrating SERR module into BiNSM network to demonstrate it can be a plug-in and independent module for any IR-based methods. The cells with different background colors reveal the improvement over our SERR module.

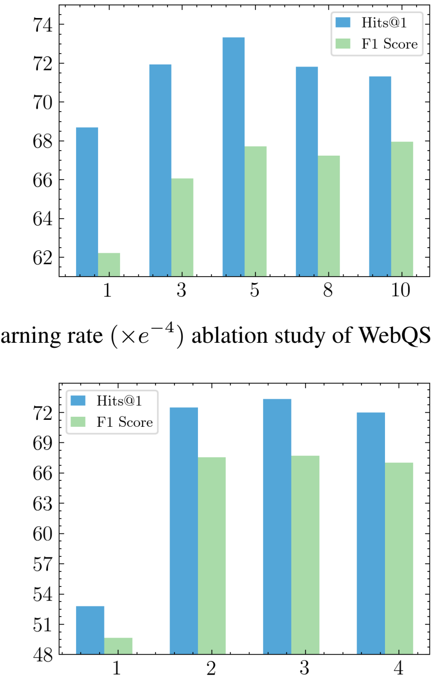
best results are obtained by using the initial learning rate of 0.0008, training batch size of 40, dropout rate of 0.30, reasoning step of 3, and max epoch size of 100. For the CWQ dataset, the best results are obtained by using the initial learning rate of 0.0008, training batch size of 100, dropout rate of 0.30, reasoning step of 3, and max epoch size of 200. For more experiment details, please refer to our code which will be published upon the publication of this work.

Parameter	WebQSP	CWQ
Learning rate	$8e^{-4}$	$8e^{-4}$
Batch size	40	100
Eps	0.95	0.95
Dropout	0.30	0.30
Num_step	3	3
Entity_dim	50	50
Word_dim	300	300
Num_epoch	200	100
Relations	6105	6649
Num_candidates	2000	2000

Table 7: The hyper-parameters of the best results on WebQSP and CWQ dataset for the KBQA task.

A.4 More Ablation Study

Reasoning Network. One minor modification of our work is that we adopt Transformer Encoder as a reasoning network, of its self-attention mechanism and superior capability of encoding information. As is shown in Table 8, compared with the backbone model (Linear layer), LSTM can acquire slight performance but with obviously longer training time, and Transformer Encoder can obtain promotion for KBQA task with tolerable extra training time. Therefore, different reasoning layers also affect the performance, and adopting Transformer Encoder can benefit a lot with three modules.



(a) Learning rate ($\times e^{-4}$) ablation study of WebQSP dataset.

(b) Performance comparison over different reasoning steps of WebQSP dataset.

Figure 8: More ablation analysis of reasoning process produced by RE-KBQA.

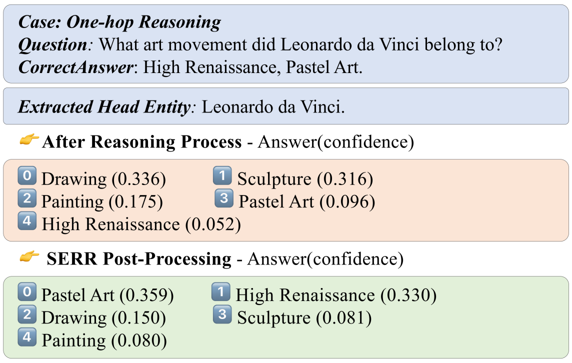
Training Settings. From Figure 8(a) and 8(b), we further study that $5e^{-4}$ and 3 is the best hyper-parameter setting for the learning rate and reasoning step. It is worth noting that, for embedding-fused methods, the more reasoning steps are not the determinant for network performance. We conduct our experiments on 2* V100 GPUs.

Models	WebQSP			CWQ		
	Hits@1	F1	Train	Hits@1	F1	Train
RE-KBQA$_b$	68.7	62.8	4.3	46.8	40.5	21.1
LSTM	70.9	66.6	6.5	47.6	41.5	27.0
	2.2 ↑	3.8 ↑	2.2 ↑	0.8 ↑	1.0 ↑	5.9 ↑
Transformer	71.0	66.1	4.5	47.1	42.7	21.5
	2.3 ↑	3.3 ↑	0.2 ↑	0.3 ↑	1.2 ↑	0.4 ↑

Table 8: Hit@1, F1 score and training time comparison of backbone model with different reasoning networks. *Train* row denotes training time in hours. The cells with different background colors reveal the extra training time over our backbone network RE-KBQA$_b$.

A.5 More Case Analysis

In this section, we deliver more case analysis on simple questions, similarity entity identification, and the intuitive reasoning process of our method.



Case: One-hop Reasoning
Question*:* What art movement did Leonardo da Vinci belong to?
CorrectAnswer: High Renaissance, Pastel Art.

Extracted Head Entity*:* Leonardo da Vinci.

After Reasoning Process - Answer(confidence)

0 Drawing (0.336)   1 Sculpture (0.316)
2 Painting (0.175)   3 Pastel Art (0.096)
4 High Renaissance (0.052)

SERR Post-Processing - Answer(confidence)

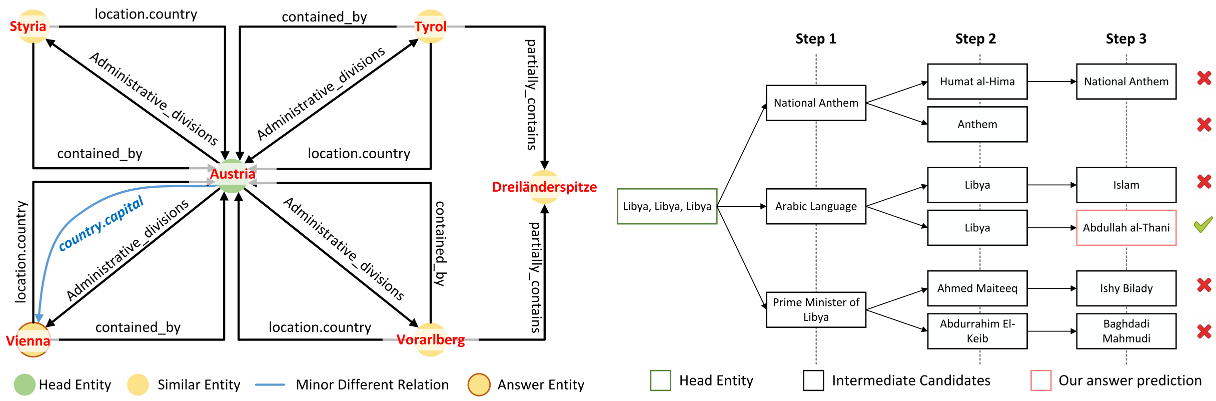
0 Pastel Art (0.359)   1 High Renaissance (0.330)
2 Drawing (0.150)   3 Sculpture (0.081)
4 Painting (0.080)

Figure 9: An example of one-hop reasoning process produced by our method.

Simple questions. As shown in Figure 9, we show a case of one-hop reasoning on the WebQSP dataset, which proved that RE-KBQA performs well in simple question answering, as the main network can recall correct candidates and the SERR module can effectively re-rank the candidates.

Similarity entity identification. To demonstrate our method can indeed distinguish similar entities, we choose a case that needs to reason across similar entities as is shown in Figure10(a). While most of the surrounding edges are the same among candidates of the first step, our method can still select the correct node as the final answer.

RE-KBQA reasoning process. Figure 10(b) shows a three-hop reasoning case of our method, to intuitively demonstrate that our method can effectively conduct a multi-hop reasoning process. Note that, the reasoning process of our method can be illustrated as the status transfer of the relation $V_r^{(t)}$ and candidate vectors $V_c^{(t)}$ from one distribution into another, which is not strictly consistent along the reasoning path, thus in some degree solve the problem of knowledge base incompleteness.



(a) Similar Entity Identification.

(b) Reasoning process of RE-KBQA

Figure 10: Cases Analysis of similar entity and thorough process of RE-KBQA compared with backbone network. Specifically, (a) is to demonstrate that our model can reason correct answers across similar entities that benefited from QA-VGAE in case *"What is the capital of Austria?"*. (b) aims to show the full pipeline of our proposed method in case *"Which man is the leader of the country that uses Libya, Libya, Libya as its national anthem?"*

.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☑ A2. Did you discuss any potential risks of your work?
Section 7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and section 1

☑ A4. Have you used AI writing assistants when working on this paper?
ChatGPT for some writing checking.

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5.2 and appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5.2 and appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.