ARCNN: A Semantic Enhanced Relation Detection Model for Knowledge Base Question Answering

Anonymous ACL submission

Abstract

Relation detection plays an important role in knowledge base question answering (KBQA), and it is critical for the final performance of KBQA systems. The previous works mainly focused on enriching the information representations of questions and relations, and neglected the interaction information of questions and relations and different tokens within the relation. In this paper, we propose a semantic enhanced relation detection model called ARCNN, which is carefully designed by combining BiGRU, multi-scale semantic extracted CNN, and different attention mechanisms in a seamless way. Moreover, we combine four levels of relation abstractions to ensure the integrity of relation information and hence to enrich the relation representation. The experimental results on two benchmarks show that our ARCNN model achieves new state-of-the-art accuracies of 96.42% for SimpleQuestions and 90.4% for WebQuestions. Moreover, it helps our KBQA system to yield the accuracy of 81.5% and the F_1 score of 72.0% on two benchmarks, respectively.

1 Introduction

Knowledge base question answering (KBQA) systems are developed with the growth of knowledge base such as WordNet (Miller, 1995), DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), and YAGO (Suchanek et al., 2007). A knowledge base usually contains a broad set of triples, where each triple is in the form of <*Subject, Predicate, Object*> and also called fact. KBQA systems enable users to answer questions more accurately and directly through operations such as question analysis, knowledge extraction, and knowledge reasoning on natural language questions (Deng et al., 2020). For example, there is a question "*What is Christina Gabrielle's profession?*", where *Christina Gabrielle* is the entity mention and the *profession* is the predicate. The KBQA system

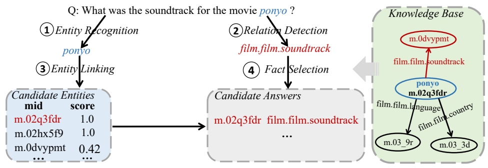

Figure 1: The process to answer the questions, and there are four mainly steps to obtain the answer.

utilizes phrases detection, resource mapping, semantic combination, and other methods to parse the question and then uses a fact <*Christina Gabrielle, people.people.profession, singer&writer*> in the knowledge base to answer the question. Therefore, *singer&writer* is the answer to the question.

The KBQA system is generally decomposed into several subtasks, among which relation detection is the most challenging one (Yu et al., 2018; Xu et al., 2018; Chen et al., 2019; Wu et al., 2019b). The previous research revealed that most of the wrong answers are caused by relation detection (He and Golub, 2016; Zhang et al., 2018). Consequently, the motivation of this paper is to improve the accuracy of relation detection and subsequently explore the contributions to the KBQA system. Figure 1 shows the main subtasks of the KBQA system.

At present, although the knowledge base relation detection (KBRD) has been developed rapidly, most studies still first enrich the semantic representations of questions and relations, and then calculate the similarity score between them, which neglects two kinds of interaction information: between questions and relations, and among different tokens within the relation. In addition, we find that the previous works on relation abstraction was incomplete, which may make the relation lose key information in representation. For example, Yu et al. (2017) utilized word-level and relation-level relation abstractions for relation representation. Yu et al. (2018) introduced the entity type information (type-level) for relation detection. Luo et al. (2020) only selected the whole relation (source-level) as input for relation detection. To address the aforementioned problems, this paper proposes a novel framework called ARCNN, which can enhance the semantic information representations of questions and relations with the help of its well-designed framework. Moreover, we merge the aforementioned four levels of relation abstractions to ensure the integrity of relation information.

Our ARCNN model combines bidirectional gated recurrent units (BiGRUs) (Cho et al., 2014) and different attention mechanisms (Vaswani et al., 2017) to fuse the semantic information of questions and relations. Furthermore, ARCNN utilizes the residual connection (He et al., 2016; Vaswani et al., 2017) to ensure the completeness of the semantic information representation of relations. Because convolutional neural networks (CNNs) can generate richer and more expressive feature representations (Nathani et al., 2019), ARCNN exploits multi-scale CNNs to extract hierarchical information by integrating local information, which is also the key to our success compared with previous methods (Yu et al., 2017; Wu et al., 2019a; Yu et al., 2018; Chen et al., 2019).

With the help of our ARCNN model, the accuracies of relation detection achieve 96.42% on SimpleQuestions and 90.4% on WebQuestions, while the previous state-of-the-art (SOTA) accuracies are 95.7% and 86.42% respectively. We further evaluate the contribution of the improvement of the relation detection to the KBQA system. The results show that the accuracy is 81.5% on SimpleQuestions and the F_1 score is 72.0% on WebQuestions while the previous SOTA results are 80.9% and 70.0% respectively. Therefore, the improvement of relation detection leads to an obvious performance boost of our KBQA system. The main contributions of our work can be summarized as follows:

- We design the ARCNN model for relation detection, which enhances the semantic information representations of questions and relations by attention-based BiGRUs and multi-scale CNNs.
- We enrich the relation representations by fusing four levels of relation abstractions to our model, and ensure the integrity of relation information.
- We perform extensive experiments on both SimpleQuestions and WebQuestions. The experiments show that our model results in new SOTA accuracies on relation detection. It is the first time that the accuracy of relation detection on WebQuestions exceeds 90% and the accuracy of relation detection on SimpleQuestions exceeds 96%.

2 Related Work

2.1 Knowledge Base Question Answering

KBQA systems understand and parse natural language questions and then utilize facts in the knowledge base to automatically answer natural language questions. The traditional methods for KBQA systems parse each natural language question into a logical expression such as Lambda-DCS (Liang et al., 2013) that can express the semantics of the question and then map the logical expression into a knowledge base supported structure queries such as SPARQL.

There are two main research directions related to KBQA systems. One is to implement the KBQA system in a pipeline manner. KBQA systems are usually divided into several subtasks (e.g., named entity recognition, entity linking, and relation detection). Some studies use deep learning to improve the performance of KBQA systems by improving a specific subtask (Yu et al., 2017; Petrochuk and Zettlemoyer, 2018; Wu et al., 2019a; Chen et al., 2019; Yu et al., 2018). The other is to implement the KBQA system in an end-to-end manner. Those methods exploit various neural networks to map the question and the candidate answers into dense vector representations respectively and calculate the semantic similarity scores (dot product) between them. By sorting the similarity scores between the candidate answers and the question, the candidates with the highest score will be selected as the answer to the question (Bordes et al., 2014; Dong et al., 2015; Hao et al., 2017; Lukovnikov et al., 2017).

2.2 Relation Detection for KBQA

KBRD is different from general relation detection in two aspects. On the one hand, the general relation detection is to extract relation from the text, and the number of relations is usually less than 100. However, there are always thousands of relations for KBRD (Bordes et al., 2015; Yu et al., 2017; Chen et al., 2019). Some methods viewed KBRD as a multi-classification task to implement the relation extraction by training a classifier (Yin et al., 2016; Petrochuk and Zettlemoyer, 2018; Mohammed et al., 2018). However, the performance of relation extraction did not achieve their expected

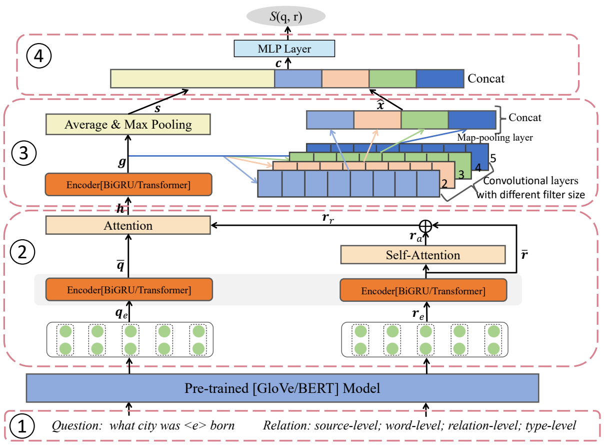

Figure 2: The overview of our ARCNN framework, which mainly includes four parts.

effect. On the other hand, KBRD always becomes a zero-shot learning task due to the unseen relations in training data (Yu et al., 2017). Those reasons make KBRD more challenging than general relation detection.

There are also some methods which map the question and the candidate relations into dense vectors respectively and then get the correct relation from candidate relations by calculating and comparing the semantic similarity scores between them. In order to enrich the semantic information representations of the questions and relation, various neural networks are used for relation detection (Yu et al., 2017; Zhang et al., 2018; Chen et al., 2019; Cui et al., 2021). Especially, pre-trained models in recent years provide better strategies to get more expressive representations and are widely used for KBRD (Lukovnikov et al., 2019; Chen and Li, 2020; Luo et al., 2020; Yan et al., 2021; Zhang et al., 2021; Kacupaj et al., 2021). Therefore, we test those strategies in our framework. In addition, the wide use of attention mechanism has also made a significant improvement in boosting semantic information representations of the questions and the relations (Qu et al., 2018; Nathani et al., 2019; Zhang et al., 2020). There are also other methods to construct various relation information representations (Yu et al., 2017; Zhang et al., 2018; Yu et al., 2018; Chen et al., 2019) to enrich the relation semantic. Therefore, in order to ensure the integrity of relation information, we merge four levels relation abstractions for relation detection.

3 Our Approach

3.1 Problem Definition

First, given the question $\mathbf{q}$ and the candidate relation set $R = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_l\}$, where l is the number of candidate relations. We then compare $\mathbf{q}$ with each candidate relation $\mathbf{r}_i$ from four levels of relation abstractions by our ARCNN and compute the semantic similarity score between them. Finally, we select the relation with the highest score as the predicted relation ($\hat{\mathbf{r}}^+$). The formula is as follows:

$$\hat{\mathbf{r}}^+ = \arg\max_{\mathbf{r}_i \in R} S(\mathbf{q}, \mathbf{r}_i)$$

3.2 Attention Mechanism

Given three inputs for scaled dot-product attention mechanism (Vaswani et al., 2017), which are the query sequence $\mathbf{Q}_{input} \in \mathbb{R}^{Q \times D}$, the key sequence $\mathbf{K}_{input} \in \mathbb{R}^{K \times D}$ and the value sequence $\mathbf{V}_{input} \in \mathbb{R}^{V \times D}$, where Q, K and V are the lengths of their respective sequences and D is the vector dimension. The attention is computed below:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK}^T}{\sqrt{D}})\mathbf{V}$$

$$\mathbf{Q} = \mathbf{Q}_{input}\boldsymbol{W}_q, \mathbf{K} = \mathbf{K}_{input}\boldsymbol{W}_k, \mathbf{V} = \mathbf{V}_{input}\boldsymbol{W}_v$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k$ and $\boldsymbol{W}_v \in \mathbb{R}^{D \times D}$ are learnable projection parameter matrices, $1/\sqrt{D}$ is the scaling factor, T denotes matrix transformation. If $\mathbf{Q}_{input} = \mathbf{K}_{input} = \mathbf{V}_{input}$, the attention is called self-attention.

3.3 Model Framework

3.3.1 Input Module

The question is represented as $\mathbf{q} = \{q_1, q_2, ..., q_m\}$, where m is the number of words in the question. For the question "*what city was* <e> *born*", which is obtained by replacing the entity mention "*John Santos*" in the original question "*what city was John Santos born*" with <e>, it has five words and hence $\mathbf{q} = \{q_1, q_2, q_3, q_4, q_5\}$. The relation can be represented as $\mathbf{r} = \{r_1, r_2, ..., r_n\}$, where $n = s + w + p + t$, s represents the number of source-level relation representations, w stands for the number of word-level relation representations, p denotes the number of relation-level relation representations, and t means the number of type-level relation representations. For the relation "*people.person.place_of_birth*", it has one source-level relation representation {*"people.person.place_of_birth"*}, three word-level relation representations {*"place", "of", "birth"*}, one relation-level relation representation {*"place_of_birth"*}, and two type-level relation representations {*"people", "person"*}. The distributed representations of the question and the relation are shown as follows.

$$\mathbf{q}_e = \{q_e^1, q_e^2, ..., q_e^m\}$$

$$\mathbf{r}_e = \{r_e^1, r_e^2, ..., r_e^n\}$$

where $\mathbf{q}_e \in \mathbb{R}^{m \times d}$, $\mathbf{r}_e \in \mathbb{R}^{n \times d}$, d is the vector dimension.

3.3.2 Semantic Representation and Fusion Module

Through this module, we can obtain the semantic information representation after the fusion of the question and the relation $\mathbf{h} \in \mathbb{R}^{n \times 2d}$, which mainly includes three steps. In the first step, our model exploits BiGRU to learn the semantic information of question $\bar{\mathbf{q}} = \{\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2, ..., \bar{\mathbf{q}}_m\}$ and relation $\bar{\mathbf{r}} = \{\bar{\mathbf{r}}_1, \bar{\mathbf{r}}_2, ..., \bar{\mathbf{r}}_n\}$, where $\bar{\mathbf{q}}_i$ and $\bar{\mathbf{r}}_j$ are processed as follows:

$$\bar{\mathbf{q}}_i = BiGRU(\mathbf{q}_e, i), \forall i \in [1, 2, ..., m]$$

$$\bar{\mathbf{r}}_j = BiGRU(\mathbf{r}_e, j), \forall j \in [1, 2, ..., n]$$

where $\bar{\mathbf{q}}_i = [\overrightarrow{\mathbf{q}_i}; \overleftarrow{\mathbf{q}_i}]$ is obtained by concatenating the forward hidden state sequence $\overrightarrow{\mathbf{q}_i} = \{\overrightarrow{q_1}, \overrightarrow{q_2}, ..., \overrightarrow{q_m}\}$ and the backward hidden state sequence $\overleftarrow{\mathbf{q}_i} = \{\overleftarrow{q_1}, \overleftarrow{q_2}, ..., \overleftarrow{q_m}\}$. So $\bar{\mathbf{q}}$ is the hidden state generated by $BiGRU$ at time i over the input sequence $\mathbf{q}_e$. $\bar{\mathbf{r}}_j$ has a similar processing and meaning as $\bar{\mathbf{q}}_i$. $[\cdot;\cdot]$ is the concatenation operator.

In the second step, ARCNN utilizes the self-attention mechanism to capture the internal correlation of different relation tokens. Using scaled dot-product attention, we obtain the attention result $\mathbf{r}_a$ by the following formula:

$$\mathbf{r}_a = Attention(\bar{\mathbf{r}}\boldsymbol{W}_1, \bar{\mathbf{r}}\boldsymbol{W}_2, \bar{\mathbf{r}}\boldsymbol{W}_3)$$

where $\mathbf{r}_a \in \mathbb{R}^{n \times 2d}$, $\bar{\mathbf{r}} \in \mathbb{R}^{n \times 2d}$. $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3 \in \mathbb{R}^{2d \times 2d}$ are learnable parameter matrices.

In order to ensure the completeness of the semantic information of relation, ARCNN performs residual connection between $\mathbf{r}_a$ and $\bar{\mathbf{r}}$, which is shown below:

$$\mathbf{r}_r = \mathbf{r}_a \bigoplus \bar{\mathbf{r}}$$

where $\mathbf{r}_r \in \mathbb{R}^{n \times 2d}$, $\bigoplus$ denotes the point-wise summation operator.

In the final step, we obtain the result of information fusion $\mathbf{h}$ by taking both enhanced relation semantic information representation $\mathbf{r}_r$ and question semantic representation $\bar{\mathbf{q}}$ as input to scaled dot-product attention as shown below:

$$\mathbf{h} = Attention(\mathbf{r}_r\boldsymbol{W}_4, \bar{\mathbf{q}}\boldsymbol{W}_5, \bar{\mathbf{q}}\boldsymbol{W}_6)$$

where $\bar{\mathbf{q}} \in \mathbb{R}^{m \times 2d}$. $\boldsymbol{W}_4$, $\boldsymbol{W}_5$, and $\boldsymbol{W}_6 \in \mathbb{R}^{2d \times 2d}$ are learnable parameter matrices.

3.3.3 Feature Extraction Module

We exploit another $BiGRU$ to learn the semantic information from $\mathbf{h}$, which is the output of information fusion from above module. We feed $\mathbf{h}$ into $BiGRU$ and then obtain the hidden state representation $\mathbf{g} \in \mathbb{R}^{n \times 2d}$ by the following formula:

$$\mathbf{g}_j = BiGRU(\mathbf{h}, j), \forall j \in [1, 2, ..., n] \quad (1)$$

$$\mathbf{g} = \{\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_n\} \quad (2)$$

After that, we employ two strategies to extract rich features of learned semantic information $\mathbf{g}$. One is that we utilize average-pooling and max-pooling operations to reduce our model parameters and prevent over-fitting when executing feature extraction. The aforementioned two pooling operations on $\mathbf{g}$ are shown in Eq.(3). We concatenate the vectors obtained from the above two pooling operations and output the semantic information representation $\mathbf{s}$ as shown in Eq.(4).

$$\mathbf{g}_a = \sum_{j=1}^{n} \frac{\mathbf{g}_j}{n}, \quad \mathbf{g}_m = \max_{j=1}^{n} \mathbf{g}_j \quad (3)$$

$$\mathbf{s} = [\mathbf{g}_a; \mathbf{g}_m] \quad (4)$$

where $\mathbf{g}_a \in \mathbb{R}^{2d}$, $\mathbf{g}_m \in \mathbb{R}^{2d}$, and $\mathbf{s} \in \mathbb{R}^{4d}$

The other is that we employ CNNs to extract features from $\mathbf{g}$. We first apply the 1D convolution filters $\mathbf{m} \in \mathbb{R}^{k(2d)}$ to capture the features of $\mathbf{g}$ with a window size k (k consecutive tokens), and obtain a new feature x_i according to the formula below:

$$x_i = f(\mathbf{m} \cdot \mathbf{g}_{i:i+k-1} + b)$$

where $\mathbf{g}_{i:i+k-1}$ means the window with k tokens, f is the non-liner activation function $ReLU$, $b \in \mathbb{R}$ is the bias term.

By sliding convolution filter on $\mathbf{g}$ with a certain step size, we can get a new feature map $\mathbf{x} = [x_1, x_2, ..., x_{n-k+1}]$. We then apply max-pooling operation on $\mathbf{x} \in \mathbb{R}^{n-k+1}$ and take the maximum value $\hat{x} = max\{\mathbf{x}\}$ as the feature corresponding to a particular filter. In order to enhance the semantic representation and capture the multi-scale features of $\mathbf{g}$, we use multiple filters with different windows sizes to extract features and get several features. Finally, we output a feature map $\hat{\mathbf{x}} = [\hat{x}_1; \hat{x}_2; ...; \hat{x}_s] \in \mathbb{R}^{4d}$ (s is the number of filters) by concatenating the features obtained by the max-pooling operations.

3.3.4 Output Module

In this module, we first construct a fixed-length vector $\mathbf{c} = [\hat{\mathbf{x}}; \mathbf{s}] \in \mathbb{R}^{8d}$ by concatenating the results of two feature extraction strategies. Then we exploit a multi-layer perceptron (MLP), which has three fully connected layers with *ReLU* activation function and dropout layer, to compute the semantic similarity score $S(\mathbf{q}, \mathbf{r})$ between the question and the relation. The formula is shown below:

$$S(\mathbf{q}, \mathbf{r}) = \delta(\mathbf{w}_3 \cdot \sigma(\mathbf{w}_2 \cdot \sigma(\mathbf{w}_1 \cdot \mathbf{c} + b_1) + b_2) + b_3)$$

where $\mathbf{w}_1, \mathbf{w}_2$ and $\mathbf{w}_3$ are the learnable weights of MLP layer, b_1, b_2 and b_3 are the bias terms, σ is the $ReLU$ activation function, and δ is the $sigmoid$ activation function.

During our ARCNN model training, we utilize the ranking loss as the training objective to maximize the margin between gold relation and negative relations. The ranking loss can be computed as follows:

$$L = max\{0, \gamma - S(\mathbf{q}, \mathbf{r}^+) + S(\mathbf{q}, \mathbf{r}^-)\}$$

where γ is the margin value and set to 0.5, $\mathbf{r}^-$ is the negative relation set of question $\mathbf{q}$, and $\mathbf{r}^+$ is the positive relation.

4 Experiments

4.1 Datasets

SimpleQuestions is constructed by (Bordes et al., 2015), which contains over 100,000 samples. Each question in SimpleQuestions has a corresponding fact from FB2M that provides the answer and explains this question, which is called the single-relation question. The FB2M is a Freebase[1] subset with 2M entities (Bordes et al., 2015). SimpleQuestions is split into the training set, validation set, and test set, which contain 75,722, 10,815, and 21,687 samples, respectively.

WebQuestions is proposed by Berant et al. (2013) for KBQA, which contains both single-relation samples (61%) and multi-relation samples (39%). It only has a training set with 3,116 samples and a test set with 1,649 samples. In our experiment, we divide the training set into a training set and a validation set at a ratio of 9:1, and use the test set for our model testing. The datasets[2] for relation detection are released by (Yin et al., 2016; Yu et al., 2017).

4.2 Experimental Details

We implement our model using PyTorch v1.8.1 and train it on a single Nvidia Titan RTX PCI-E GPU. All word embeddings are initialized by the pre-trained GloVe[3] with 300 dimensions and updated during training. The out-of-vocabulary words are randomly initialized by uniformly sampling from (-0.5,0.5). During the experiment, we set the initial learning rate to 0.001, the optimization strategy to $Adamax$, the batch size to 128, the loss margin to 0.5, the dropout rate to 0.35, and the size of hidden states for BiGRUs to 300. In our ARCNN model, its CNNs have four filters, whose sizes are $2, 3, 4$, and 5, respectively. For other parameter settings, please refer to the source code available at GitHub[4] for the details.

4.3 Results

This section reports the relation detection results on SimpleQuestions and WebQuestions, where *source*, *words*, *relation* and *types* correspond to the four levels of relation abstractions respectively. In Table 1, we compare the performance of our ARCNN model with other baselines on the test datasets of SimpleQuestions and WebQuestions (first block), i.e., AMPCNN (Yin et al., 2016), HR-BiLSTM (Yu et al., 2017), Multi-View (Yu et al., 2018), QURRD (Xu et al., 2018), MVA-MTQA-net (Deng et al., 2019), FOFE-net (Wu et al., 2019a), KRD (Chen et al., 2019), BERT (Lukovnikov et al., 2019), DAM (Chen and Li, 2020), and BiGRU (Cui et al., 2021).

[1] https://developers.google.com/freebase
[2] https://github.com/Gorov/KBQA_RE_data
[3] https://nlp.stanford.edu/projects/glove/
[4] https://github.com/example/ARCNN

Approach	Relation Input	SimpleQuestions	WebQuestions
AMPCNN (Yin et al., 2016)	words	91.3	-
HR-BiLSTM(Yu et al., 2017)	words + rel_names	93.3	82.53
Multi-View(Yu et al., 2018)	entity pair + relation + type	93.75	85.95
QURRD(Xu et al., 2018)	relation	94.2	**86.42**
MVA-MTQA-net(Deng et al., 2019)	word + knowledge	**95.7**	85.8
FOFE-net(Wu et al., 2019a)	relation	93.3	83.26
KRD(Chen et al., 2019)	words + relation	93.5	85.72
BERT(Lukovnikov et al., 2019)	relation	83.6	-
DAM(Chen and Li, 2020)	words	93.3	84.1
BiGRU(Cui et al., 2021)	relation	81.16	-
ARCNN	source + words + relation + types	**96.42** (+0.72)	**90.4** (+3.98)
w/o multi-scale semantic extracted CNN	source + words + relation + types	95.14	82.92
w/o residual connection	source + words + relation + types	95.54	84.62
w/o self-attention	source + words + relation + types	95.70	82.19
replacing attention with concatenation	source + words + relation + types	94.94	81.52
replacing BiGRU with Transformer for encode	source + words + relation + types	95.74	88.69
replacing pre-trained GloVe with BERT	source + words + relation + types	94.20	75.40
w/o source	words + relation + types	95.93	88.88
w/o source and types	words + relation	95.97	86.32
w/o source, relation, and types	words	95.90	85.90

Table 1: The accuracies of relation detection on SimpleQuestions and WebQuestions (test set) using different strategies, where w/o is the abbreviate of without. The first block shows the performance of baselines. The numbers in brackets represent the increasement of our model relative to the SOTA results.

As can be seen from Table 1, our model achieves the best accuracies of 96.42% for SimpleQuestions and 90.4% for WebQuestions, and outperforms the previous SOTA work on both benchmarks. Since the accuracies for SimpleQuestions are above 90%, the previous work (Yu et al., 2018) had predicted that there is still room for improvement for WebQuestions. Our ARCNN model achieves better performance than their expection by yielding the accuracy 90.4% for WebQuestions.

4.4 Ablation Tests of ARCNN

To evaluate the impacts of the different components of our model, we conduct the experiments with the following different strategies.

- **w/o multi-scale semantic extracted CNN** We remove the CNNs from *Feature Extraction Module*, and only use the average-pooling and max-pooling operations for feature extraction.

- **w/o residual connection** We eliminate the residual connection from *Semantic Representation and Fusion Module*, and directly use the self-attention results and question representation for semantic information fusion.

- **w/o self-attention** We get rid of the self-attention from ARCNN, and directly fuse the semantic information of the question and the relation after they encode respectively.

- **replacing attention with concatenation** We employ concatenation operation instead of attention for semantic information fusion of questions and relations.

- **replacing BiGRU with Transformer for encode** We utilize the Transformer encoder to replace the BiGRU for question and relation encode.

- **replacing pre-trained GloVe with BERT** We employ pre-trained BERT (BERT-Base, Uncased) to replace the pre-trained GloVe for tokens initialization of questions and relations.

- **different strategies for relation input** We perform some ablation experiments to compare the influence of different relation inputs on the accuracy of our ARCNN model.

By analyzing the ablation experimental results in Table 1, we can draw the following conclusions:

1. Without employing multi-scale CNNs as the feature extraction method, this strategy makes the accuracies dropped rapidly. It indicates that it is significant to use multi-scale CNNs to extract features, and multiple convolution filter sizes can extract the features of continuous tokens with different lengths and hence enrich the semantic information feature of the representation. Moreover, residual connection is also significant in ensuring the integrity of semantic information.

2. It is important to employ the self-attention mechanism to capture the mutual influences

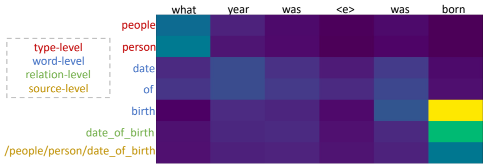

Figure 3: Visual display of the heat map of attention weight matrices. Different levels of relation abstractions are represented by different color fonts. The brighter the color, the greater the impact.

Error Samples	SQ	Rate(%)	WQ	Rate(%)
Question Ambiguity	361	46.5	91	57.6
Semantic Deficiency	90	11.6	7	4.4
Dataset Noise	158	20.4	6	3.8
Others	167	21.5	54	34.2
Sum	776	100	158	100

Table 2: The number and rate of error samples in different categorical errors in the relation detection results of SimpleQuestions and WebQuestions.

within the relation. Consequently, utilizing the attention mechanism to implement the fusion of semantic information between questions and relations is much more effective than using simple semantic information fusion (e.g., concatenation operation). We can also get this conclusion from Figure 3. For example, the *"birth"* in relation has the most significant impact on the *"born"* in the question, which is consistent with our basic intuition.

3. We utilize the new technique Transformer for tokens encoding and the famous pre-trained model BERT for tokens initialization of questions and relation. However, both methods fail to improve relation detection accuracy in our experiments. It shows that the Transformer encoder is not as good as BiGRU in capturing local information, and its shortcomings in obtaining position information make the Transformer encoder fail to achieve ideal results in relation detection. Moreover, the pre-trained BERT model may not be suitable for our ARCNN framework.

4. We use other strategies (e.g. "*words + relation + types*", "*words + relation*", and "*words*", etc) as input, and observe that the experimental results on both SimpleQuestions and WebQuestions have performance degradation. It suggests that our ARCNN has achieved obvious advantages by combining more levels of relation abstractions as model input, which can capture the comprehensive semantic information of relations.

4.5 Error Analysis

In this section, we analyze the errors of relation detection, which mainly contains the following three categories. Table 2 shows the number and rate of error samples in different categorical errors in the relation detection results of two benchmarks.

- **Question Ambiguity:** It makes the question ambiguity when we replace the entity mention in the question with "<e>". For example, through the question of "*which country uses <e>?*", it is difficult for us to understand the real meaning of the original question, which is "*which country uses ndali language?*".

- **Semantic Deficiency:** We get a short question pattern after replacing the entity mention in the question with "<e>". It does not make grammatical sense. For example, the templates corresponding to the question "*who wrote nocturnal pleasure?*" and "*who wrote love comes quickly?*" are both "*who wrote <e>?*". So it is difficult to understand the semantic information of question. And hence a detection error occurs.

- **Dataset Noise:** This kind of errors is very common, especially in SimpleQuestions. For example, through the question of "*where was <e> born?*", it is difficult to infer whether the question is about the nationality or birthplace of someone. So we may get the wrong relation for this question. Especially when there are a few training samples, it will be more common and more difficult to infer the correct relation.

- **Others:** We classify the rest error samples into the fourth category.

According to the statistics of the three categorical errors in Table 2, we find that a larger proportion of errors in SimpleQuestions and WebQuestions are caused by question ambiguity. Therefore, our future work is to make the question representations more expressive and to represent the questions more precisely. Moreover, we calculate the number of each category errors in 1-hop and 2-hop relations in WebQuestions. As we can see from Table 3, the accuracy of 2-hop (multi-relation) relations is 88.7%, and also exceeds the SOTA results (QURRD). It shows that our ARCNN model does

WQ	1-hop	2-hop	Sum
Original Samples	1046	603	1649
Error Samples	90	68	158
Acc(%)	91.4	88.7	90.4

Table 3: The number and ratio of different relation types (1-hop or 2-hops) in the original test samples and error samples in WebQuestions.

Dataset	SQ		WQ	
Relations	Seen	Unseen	Seen	Unseen
Original Data	21526	161 (0.7%)	1579	70 (4.2%)
Error Samples	754	18 (2.4%)	114	44 (27.8%)
Rate(%)	96.5	88.8	92.8	37.1

Table 4: Statistics on the relations has been seen/unseen in the training data for original and error samples in SimpleQuestions and WebQuestions. The numbers in brackets denote the proportion of unseen relations in total samples (original or error samples).

not give poor accuracies due to the complexity of the relation, which indicates that our model is still effective in predicting complex relations.

From another perspective, the previous approaches can obtain a high accuracy for samples whose relations have been seen in the training data, while the performance will drop rapidly for unseen relations (Wu et al., 2019b). Therefore, we have collected statistics on the relations that have been seen/unseen in the training data for error samples in SimpleQuestions and WebQuestions. As can be seen from Table 4, the performance of ARCNN does not drop too much for the unseen relations in SimpleQuestions (the accuracy drops from 96.5% to 88.8%) but falls rapidly for the unseen relations in WebQuestions (the accuracy drops from 92.8% to 37.1%). For SimpleQuestions, since the proportion of unseen relation samples in the original dataset is very small (0.7%), it can not significantly contribute to the performance of relation detection to pay much more attention to the research on unseen relations. However, for the WebQuestions, it is meaningful to improve the relation detection on the unseen relations (e.g., zero-shot learning), because unseen relations have a larger proportion in WebQuestions (4.2%) and there are more error samples caused by unseen relations (27.8%). Therefore, in future work, we will pay more attention to zero-shot learning for WebQuestions and focus more on enhancing the semantic representations of questions and relations for SimpleQuestions.

4.6 KBQA Results

In order to evaluate how the new SOTA accuracy of relation detection could benefit the KBQA system, we continue to complete the subsequent experiments of the KBQA system, including entity recognition, entity linking, and fact selection. The experimental results of our KBQA system are reported in Table 5, where we compare the performance of our method with other baselines, i.e., MemNN (Bordes et al., 2015), HR-BiLSTM (Yu et al., 2017), BiLSTM-CRF&BiLSTM (Petrochuk and Zettlemoyer, 2018), PR+JFS (Hao et al., 2018), FOFE-net (Wu et al., 2019b), BERT (Lukovnikov et al., 2019), BERT-based (Luo et al., 2020), DAM (Chen and Li, 2020), and BiGRU-CRF&BiGRU (Cui et al., 2021). As shown in Table 5, the F_1 score achieves 72.0% on WebQuestions, which exceeds the SOTA work (DAM) by 2.0%. On SimpleQuestions, the accuracy of our KBQA system reaches 81.5%, which exceeds the SOTA work (BERT-based) by 0.6%. It suggests that relation detection plays a critical role in the KBQA system, and hence directly contributes to the performance of the KBQA system.

Approach	SQ	WQ
MemNNs (Bordes et al., 2015)	63.9	42.2
HR-BiLSTM (Yu et al., 2017)	78.7	63.9
BiLSTM-CRF&BiLSTM (Petrochuk and Zettlemoyer, 2018)	78.1	-
PR+JFS (Hao et al., 2018)	80.2	-
FOFE-net (Wu et al., 2019a)	77.3	67.6
BERT (Lukovnikov et al., 2019)	77.3	-
BERT-based (Luo et al., 2020)	**80.9**	-
DAM (Chen and Li, 2020)	-	**70.0**
BiGRU-CRF&BiGRU (Cui et al., 2021)	80.37	-
Our method	**81.5**	**72.0**

Table 5: The performance of KBQA systems on SimpleQuestions and WebQuestions using different methods.

5 Conclusion

This paper introduces a new neural network framework called ARCNN to achieve more precise matching between questions and relations in a semantic space. Our ARCNN model combines BiGRU, multi-scale semantic extracted CNN, and different attention mechanisms to enhance the semantic information interaction and the semantic information representations of questions and relations. To construct richer relation representations, ARCNN merges four levels of relation abstractions to capture semantic and literal relevance information, which makes our model match questions and relations more precisely. The experimental results show that our approach is more competitive than others in the relation detection on both SimpleQuestions and WebQuestions.

References

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

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

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, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615–620.

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

Daoyuan Chen, Min Yang, Hai-Tao Zheng, Yaliang Li, and Ying Shen. 2019. Answer-enhanced path-aware relation detection over knowledge base. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1021–1024.

Yongrui Chen and Huiying Li. 2020. Dam: Transformer-based relation detection for question answering over knowledge base. *Knowledge-Based Systems*, 201:106077.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Hai Cui, Tao Peng, Lizhou Feng, Tie Bao, and Lu Liu. 2021. Simple question answering over knowledge graph enhanced by question pattern classification. *Knowledge and Information Systems*, 63(10):2741–2761.

Chaoyu Deng, Guangfu Zeng, Zhiping Cai, and Xiaoqiang Xiao. 2020. A survey of knowledge based question answering with deep learning. *Journal of Artificial Intelligence*, 2(4):157–166.

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

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over freebase with multi-column convolutional neural networks. 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 260–269.

Yanchao Hao, Hao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Pattern-revising enhanced simple question answering over knowledge bases. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3272–3282.

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 221–231.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Xiaodong He and David Golub. 2016. Character-level question answering with attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1598–1607.

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 850–862.

Percy Liang, Michael I Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

Denis Lukovnikov, Asja Fischer, and Jens Lehmann. 2019. Pretrained transformers for simple question answering over knowledge graphs. In *International Semantic Web Conference*, pages 470–486. Springer.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, pages 1211–1220.

Da Luo, Jindian Su, and Shanshan Yu. 2020. A bert-based approach with relation-aware attention for knowledge base question answering. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723.

Michael Petrochuk and Luke Zettlemoyer. 2018. Simplequestions nearly solved: A new upperbound and baseline approach. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 554–558.

Yingqi Qu, Jie Liu, Liangyi Kang, Qinfeng Shi, and Dan Ye. 2018. Question answering over freebase via attentive rnn with similarity matrix based cnn. *arXiv preprint arXiv:1804.03317*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Dekun Wu, Nana Nosirova, Hui Jiang, and Mingbin Xu. 2019a. A general fofe-net framework for simple and effective question answering over knowledge bases. *arXiv preprint arXiv:1903.12356*.

Peng Wu, Shujian Huang, Rongxiang Weng, Zaixiang Zheng, Jianbing Zhang, Xiaohui Yan, and Jiajun Chen. 2019b. Learning representation mapping for relation detection in knowledge base question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6130–6139.

Zihan Xu, Hai-Tao Zheng, Zuoyou Fu, and Wei Wang. 2018. Enhancing question understanding and representation for knowledge base relation detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1362–1367. IEEE.

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

Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple question answering by attentive convolutional neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1746–1756.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved neural relation detection for knowledge base question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–581.

Yang Yu, Kazi Saidul Hasan, Mo Yu, Wei Zhang, and Zhiguo Wang. 2018. Knowledge base relation detection via multi-view matching. In *European Conference on Advances in Databases and Information Systems*, pages 286–294. Springer.

Hongzhi Zhang, Guandong Xu, Xiao Liang, Guangluan Xu, Feng Li, Kun Fu, Lei Wang, and Tinglei Huang. 2018. An attention-based word-level interaction model for knowledge base relation detection. *IEEE Access*, 6:75429–75441.

Linhai Zhang, Deyu Zhou, Chao Lin, and Yulan He. 2021. A multi-label multi-hop relation detection model based on relation-aware sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4713–4719, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanan Zhang, Guangluan Xu, Xingyu Fu, Li Jin, and Tinglei Huang. 2020. Adversarial training improved multi-path multi-scale relation detector for knowledge base question answering. *IEEE Access*, 8:63310–63319.