A Hybrid Similarity Computing Method for KBQA

Chunpei Wang, Xiaowang Zhang and Zhiyong Feng

College of Intelligence and Computing, Tianjin University, Tianjin 300350, China

Abstract. With the rapid growth of knowledge bases(KBs), knowledge-base question answering has drawn huge attention in recent years. Most existing KBQA methods translate questions into SPARQLs to help end-users access the knowledge base represented by RDF more naturally. However, a natural language question is always corresponding to multiple candidate SPARQLs due to the gap between the unstructured Question and the structural SPARQL query. To pick the best SPARQL query from the candidate SPARQL query set, in this Poster, we propose a hybrid similarity computing method to rank the SPARQL query. Firstly, we employ two attentive recurrent neural networks to capture the semantic Similarity between the SPARQL query and the Question. Secondly, we compute the string similarity between the SPARQL query and the Question by leveraging the convolutional neural network. Our method can capture the two-level Similarity between the Question and the SPARQL. Experiments show that our method can improve the effectiveness of KBQA.

Keywords: KBQA · Semantic Parsing · RDF · SPARQL.

1 Introduction

Generally, we will obtain multiple SPARQL queries after the semantic parsing stage since the ambiguity between the natural language question and the knowledge base. For instance, the entity mention *St. Lawrence* of the Question "*What body of water does St.Lawrence flow into?*" will be mapped to a set of semantic instances in the KB, e.g., $E = \{\langle Siant\ Lawrence\rangle$, or $\langle Siant\ Lawrence\ River\rangle\}$. Thus, the main challenge in the semantic parsing stage is how to pick the best SPARQL query in the candidate query set.

Most existing KBQA work maps the Question and the KB facts(triple) to a common embedding space. The Similarity between the question vector and the SPARQL vectors can be conveniently computed. However, these methods tend to lose original word interaction information. To preserve more original information, we propose a hybrid similarity computing method to pick the best SPARQL query from the candidate set. Both consider the semantic Similarity and the strong Similarity between the Question and the SPARQL query.

2 Our Approach

In this Poster, we present a hybrid ranking model to rank the SPARQL query, which considers both string similarity and Semantic Similarity. Given the natural language question N, for each query q_i in the candidate SPARQL set, we compute the similarity score $S(N, q_i)$ that represents the semantic Similarity between N and q_i. Finally, all candidate queries are ranked via their similarity scores with N.

Semantic-level Similarity. We construct an attentive recurrent neural network for computing semantic-level similarity between question N and the candidate query q_i. The model uses an encoder-compare framework which encodes the semantic information of N and q_i into high-dimensional embedding space and then estimates their similarity via multilayer perceptron(MLP).

- *Encoding.* Firstly, each elements in question N and query q_i is mapped to its corresponding embedding vectors $\{w_1, \ldots, w_L\}$, where L is the length of the question or query. And then all the embeddings will be input into a bidirectional GRUs neural network to learn the hidden representations $H_{1:L} = [h_1, \ldots, h_L]$, where h_i is the concatenation of forward and backward vectors learned at time i. Since each word contribute differently to the full sentence semantics, the model would pay different attention to each word and learn promising vectors to represent the question/query sentences. The self-attention model is used here to learn the weight of each word semantic for the input sentence. The semantic representation Y of sentence can be calculated as follows:

$$Y = \sum_{i=1}^{L} a_i h_i, \tag{1}$$

$$a = Attention(Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d^k}}\right)V \tag{2}$$

  where $\{Q, K, V\}$ are the shorthand for $\{$*query, key, value*$\}$, which are three matrices that mapped with the same input. K and V is a one-to-one correspondence with key-value relation. Q could be the hidden state to be processed, such as h_i. First, the dot product between Q and K is computed, which will be divided by a scale factor of $\sqrt{d^k}$ to prevent the result from being too large. And then, the result will be processed with *softmax* function to get normalized probability, which will be multiplied by V to get the weight. Finally, each hidden presentation h_i is multiplied by the attention weights and summed to get semantic representation Y.
- *Similarity estimation.* With the representations Y_p and Y_{q_i} of question p and query q_i, their similarity will be calculated by a MLP layer

$$z_1 = f\left(W^\top [Y_p; Y_{q_i}] + b\right) \tag{3}$$

  where W is the parameters to be learned, b is the deviation and $f(\cdot)$ is an activation function. The semantic information extracted from the two

sentences are spliced as the input of MLP hidden layer, which nonlinearly mapped the two sentences into their Similarity.

String-level Similarity. The Similarity over string-level is evaluated via a text-matching model. Some words or phrases with the same meaning may be expressed differently in the Question and query, i.e., the pair of words $(musical, music)$ have similar semantics. Since the high-level semantic embedding cannot preserve these words interaction information, we construct a similarity matrix whose elements represent the similarities between question words and query words, regarding it as a two-dimensional vector space to utilize the convolutional layer to capture the matching features.

- *Similarity matrix.* Firstly, we construct a similarity matrix M, where each element M_{ij} indicates the basic interaction. The M_{ij} can be calculated as:

$$M_{ij} = u_i \otimes v_i \tag{4}$$

  Where u_i and v_j denotes the i-th and the j-th word was embedding in Question and query, respectively. The operator $\otimes$ stands for a general operator to compute the Similarity. Here the matrix can obtain the Similarity of words with different expressions.
- *Convolution layer.* The different levels of matching patterns can be extracted by a convolutional kernel. The k-th kernel w^k scans over the similarity matrix M to generate a feature map g^k:

$$g_{i,j}^k = \sigma\left(\sum_{s=0}^{r_k-1}\sum_{t=0}^{r_k-1} w_{s,t}^k \cdot M_{i+s,j+t} + b^k\right) \tag{5}$$

  where r^k denotes the size of the k-th kernel.
- *Max pooling layer.* Two different pooling kernels are used on the top of feature map g^k:

$$y_i^{1,k} = \max_{0\le t<d_1} g_{i,t}^k, \tag{6}$$

$$y_j^{2,k} = \max_{0\le t<d_2} g_{i,j}^k. \tag{7}$$

  where d_1 and d_2 denote to the width and length of similarity matrix. The max-pooling layer retains the max matching feature from the perspective of both questions and queries.
- *Fully connected layer.* Similar to semantic Similarity, A MLP layer is used to produce the final feature z_2 and z_3.

Combination. With three feature (z_1, z_2, z_3) generated from two level similarity, we utilize a linear layer to learn their respective contribution for holistic similarity score:

$$S(p, q_i) = Sigmoid\left(W^\top [z_1; z_2; z_3] + b\right). \tag{8}$$

Finally, all candidate queries q_i will be sorted with the similarity scores $S(p, q_i)$.

3 Experiments and Evaluation

Table 1. Result on SimpleQuestions

	Accuracy
Our Method	**0.962**
BiCNN(Yih et al.)	0.900
AMPCNN (Wenpeng et al.)	0.913
HR-BiLSTM (Yu et al.)	0.933
Multiple View Matching (Yu et al.)	0.957

SimpleQuestions is a single-relation KBQA dataset. This dataset consists of questions annotated with a corresponding fact from Freebase that provides the answer. We report Accuracy as previous studies. We verify our proposed approach on the SimpleQuestion dataset. Table 1 summarizes the experimental results of different methods on answer selection and knowledge base question answering. Clearly, our method achieves the state-of-the-art results in SimpleQuestions, which confirms the effectiveness of our solution.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2017YFC0908401) and the National Natural Science Foundation of China (61972455). Xiaowang Zhang is supported by the Peiyang Young Scholars in Tianjin University (2019XRX-0032).

References

1. A. P. B. Veyseh, "Cross-lingual question answering using common semantic space," in *Proceedings of TextGraphs@NAACL-HLT 2016*, pp. 15–19, 2016.
2. L. Dong, F. Wei, M. Zhou, and K. Xu, "Question answering over Freebase with multi-column convolutional neural networks in ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pp. 260–269, 2015.
3. A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," in EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 615–620, 2014.
4. X. Yao and B. V. Durme, "Information extraction over structured data: Question answering with freebase," in ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, pp. 956–966, 2014.
5. X. Lu, S. Pramanik, R. S. Roy, A. Abujabal, Y. Wang, and G. Weikum, "Answering complex questions by joining multi-document evidence with quasi knowledge graphs," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.*, pp. 105–114, 2019.