An Embedding-based Approach to Completing Question Semantics

Haixin Zhou[1] and Xiaowang Zhang[2]

College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
{haixinzhou,xiaowangzhang}@tju.edu.cn**

Abstract. A question can be correctly answered, which has complete semantics; that is, it contains all basic semantic elements. In practical, questions are not always complete due to users' ambiguous representation of their intents. Unfortunately, there is few research work on this problem. In this paper, we present an embedding-based approach to completing question semantics by inspiring from knowledge graph completion based on our proposed representation of a complete basic question as unique type and subject and multiple possible constraints.

1 Introduction

Question answering (QA) is a task that a natural language question can be accurately and concisely answered over a structured database of knowledge (as a knowledge base) or information or even an unstructured collection of natural language documents by understanding the intention of the question [3]. QA techniques have been widely used in many fields of NLP, such as chatbot, intelligent search, and recommendation [5]. In practical, questions asked by users are not always complete due to users' ambiguous representation of their intents. For instance, "*the last Paris metro*", "*Tokyo resident population*", and "*actors of the movie Green Book*" are incomplete questions. Hence it becomes very interesting to complete questions for further querying accurately and concisely in QA. Unfortunately, there is few research work on this problem.

In this paper, we present a novel embedding-based approach to extracting the semantics of incomplete questions based on some core techniques of knowledge graph completion (KGC) [1] so that those questions can still be answered accurately and concisely. Experiential results on the datasets revised from benchmark datasets demonstrate that our approach can effectively extract semantics lost in capturing intents.

2 Our Approach

In this section, we introduce our approach in technique in four parts: namely, *question formalization*, *question representation*, *question completion*, and *question generation*, and its overview framework is shown in Figure 1.

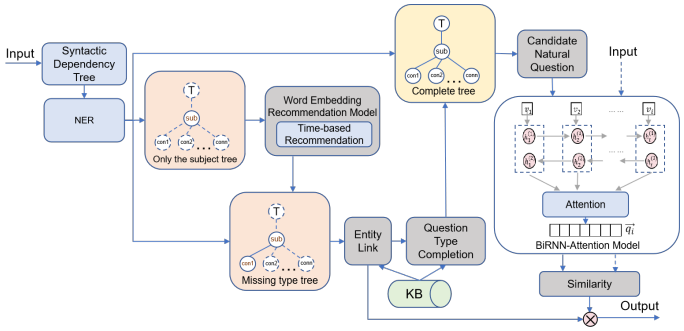

Fig. 1: The Overview Framework of Our Approach

– **Question Formalization** Let Σ be a set of words. A finite sequenece $(w_1, \ldots, w_n)$ of words with $w_i \in \Sigma$ $(i = 1, 2, \ldots, n)$ is a *sentence* over Σ. We use Σ^* to denote a set of sentences over Σ. A *phrase* is a group of words. Let V_N, V_E, V_T be three subsets of Σ^*.

Definition 1 (Structure of Question). *Let* $V = V_T \cup V_N \cup V_E$. *Let* q *be a question. A* syntactic structure *(*structure*, for short) of* q *over* V *as a labeled tree* $T_q = (N, E, \mathcal{L}, \lambda, \delta)$ *whose children is either a labelled tree named* substructure *of* T_q *or a leaf where*
$\mathcal{L} : \text{roots}(T_q) \to V_T$*: an injective function mapping it to one phrase;*
$\lambda : N \to V_N$*: a function mapping each node to a phrase;*
$\delta : E \to V_E$*: a function mapping each edge to a phrase;*
where $\text{roots}(T_q)$ *is a collection of non-leaf nodes in* T_q.

– **Question Representation** constructs a tree-structure (as a representation) of questions for each sequence of words (as incomplete questions) input. In this module, we use a parsing-dependency-tree-based method to construct the representation of all incomplete questions. Intuitively, a complete question contains at least three elements: type, subject, and constraint. See Figure 2, Figure 3.

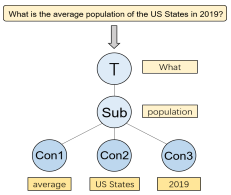

Fig. 2: Complete

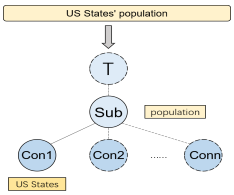

Fig. 3: Incomplete

– **Question Completion** builds complete tree-structures of incomplete questions after inputting the primary tree-structure of questions constructed in the question representation module together with incomplete questions.

– **Question Generator** generates all possible nature questions as candidates by learning stopwords via word2vec and enumerating all orders of constraints. Note that the structure of questions depends on non-stop words which have already been deleted as initialization.

3 Experiments and Evaluation

In our experiments, we select DBpeida[1], as our knowledge base. We also select three representative datasets of Webquestions(3778 QA pairs), QALD-7(100 QA pairs), and TrecQA-8(3000 QA pairs) to evaluate our approach.

Effectiveness of Question Completion To evaluate the accuracy of our question completion, we build QA datasets consisting of totally incomplete questions obtained from WebQuestions, TrecQA, and QALD via a random erasing tactic, denoted by WebQuestions*, TrecQA*, and QALD*, respectively. In the tactics of constructing incomplete questions, we apply the Fisher-Yates Shuffle algorithm [2] (a popular random ranking) in deleting words randomly to enure generating incomplete questions fair. Besides, we employ gAnswer [2] for answering. Indeed, this experiment can be conducted by any KBQA systems since our approach employs the query execution module of gAnswer.The experimental results are shown in Table **??**, where *QC-Precision* means the precision of answering after completing the question by our approach while *Precision* denotes the precision of answering without completing. By Table **??**, Precision over three datasets are zero, that is, any incomplete question cannot be answered. At the meantime, our QC-Precision are 0.5060%, 0.3322% and 0.2877%, respectively. That is, it verifies that our question completion actually achieves effectiveness. Moreover, to quantify the effectiveness of question completion, we introduce a new metric named *completion rate* defined as follows: QC-R $= \frac{\text{QC-Precision}}{\text{O-Precision}}$. Here *O-Precision* denotes the precision of answering over original datasets. We can show that the completion rate of our approach in the three datasets is over 0.80. This experiment verifies that our approach can significantly complete the semantics of questions effectively.

Applications of Question Completion Finally, we perform another experiment to exhibit applications of our QC approach, where QC preprocesses datasets to be tested KBQA systems. We select the three systems NFF [2], RFF [2], Aqqu [4] where NFF and RFF are non-template systems with the highest scores of QA-Task in the latest QALD [3] and Aqqu is a classical KBQA system. The experimental results w.r.t. F1-score on WebQuestions and QALD are showed in Table 2. By Table 2, all baselines with QC preprocessing achieve 0.0568∼0.0773 over WebQuestions and 0.0141∼ 0.141 over QALD. Therefore, the experiment demonstrates that QC is useful to improve the accuracy of off-the-shelf QA systems.

[1] https://wiki.dbpedia.org/
[2] http://hdl.handle.net/2440/10701/
[3] https://project-hobbit.eu/challenges/qald-9-challenge/

Table 1: Precision of Question Completion

	Precision	QC-Precision	QC-Rate
QALD*	0	0.5060	0.8432
TrecQA*	0	0.3322	0.8215
WebQuestions*	0	0.2877	0.8029

Table 2: QC Improving QA Baselines

	F1			
	WebQuestions		QALD	
	Baseline	Baseline+QC	Baseline	Baseline+QC
NFF	0.4847	**0.5415**	0.7751	**0.7892**
RFF	0.3052	**0.3825**	0.5341	**0.5513**
Aqqu	0.4944	**0.5532**	0.3741	**0.4882**

4 Conclusions

In this paper, we introduce a QA task named question completion and, inspired by knowledge graph completion, and we present an embedding-based approach for completing questions disabled in answering over any KB. Different from question generation, our QC can portrait the intent of an incomplete question, which widely applies to the improvement of QA. In future work, we are interested in question completion of some important domains with considering more domain knowledge.

5 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. Alberto, G., Sebastijan, D., Mathias, N. (2018). Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Prof. of EMNLP*: 4816–4821.
2. Hu, S., Zou, L., Yu, J., Wang, H., Zhao, D. (2018). Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. *IEEE Trans. Knowl. Data Eng.*, 30(5): 824–837.
3. Huang, X., Zhang, J., Li, D., Li, P. (2019). Knowledge Graph Embedding Based Question Answering. In *Prof. of WSDM*: 105–113.
4. Hannah, B., Elmar, H. (2015). More Accurate Question Answering on Freebase. In *Prof. of CIKM*: 1431-1440.
5. Zhang, W., Liu, T., Qin, B., Zhang, Y. (2017). Benben: A Chinese Intelligent Conversational Robot. In *Prof. of ACL*: 13–18.