Answering Questions over RDF by Neural Machine Translating

Shujun Wang, Jie Jiao, Yuhan Li, Xiaowang Zhang*, and Zhiyong Feng

College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
* Corresponding author:{xiaowangzhang}@tju.edu.cn

Abstract. Question Answering over Knowledge Bases (KBQA) is a task that a natural language question can be accurately answered over a knowledge base. Unlike previous methods for KBQA use a pipelined approach, which focuses on entity linking and relation path ranking. In this paper, we present a translation-based approach to translate natural language questions into SPARQL queries. Specifically, this paper contributes to filling the gap between natural language question and SPARQL by utilizing multiple Neural Machine Translation(NMT) models such as RNN, CNN, and Transformer. More importantly, we bridge the gap between the NMT model and existing KBQA by combining the entity linking and relation linking technologies in KBQA with the NMT model. Based on which, we design four novel question translation approach for any NTM model, i.e., "Pure NMT", "NMT+Entity Linking", "NMT + Relation Linking" and "NMT + Entity Linking + Relation linking". Compared to the traditional KBQA system using a state-of-the-art semantic parser, our method achieves a precision measure of 67.9% on the QALD-9 dataset and win the first place.

1 Introduction

Knowledgebase question answering (KBQA) is an important task in NLP that has many real-world applications, such as in search engines and decision support systems. Most existing methods for KBQA use a pipelined approach: First, given a question q, an entity linking step is used to find KB entities mentioned in q. Next, relations or relation paths in the KB linked to the topic entities are ranked such that the best relation or relation path matching q is selected as the one that leads to the answer entities.

In the view of the success of Neural Machine Translation (NMT) approaches, it comes as a surprise that very few such models utilized to address the question translating challenge(Question→SPARQL) in KBQA. Although some NMT-based KBQA works have been proposed for answering questions over RDF.

However, these methods did not utilize the latest transformer model; more importantly, they did not try to associate the NMT model with critical technologies in traditional KBQA.

In order to utilize NMT models in the KBQA area, this paper presents a large-scale comparison of three distinct neural network architectures (Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and the Transformer model). Further, we bridge the gap between the NMT model and traditional KBQA technologies, and we combine NMT models with the key technology(entity linking and relation linking) of traditional KBQA to form four NMT-based KBQA approaches.

2 Overview

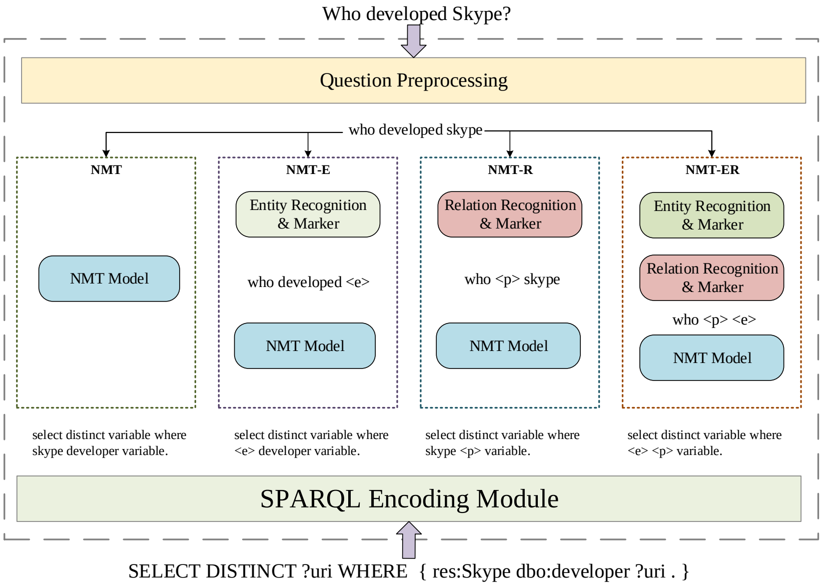
Who developed Skype?
Question Preprocessing
who developed skype
NMT
NMT-E
NMT-R
NMT-ER
Entity Recognition & Marker
Relation Recognition & Marker
Entity Recognition & Marker
Relation Recognition & Marker
NMT Model
who developed <e>
who <p> skype
who <p> <e>
NMT Model
NMT Model
NMT Model
select distinct variable where skype developer variable.
select distinct variable where <e> developer variable.
select distinct variable where skype <p> variable.
select distinct variable where <e> <p> variable.
SPARQL Encoding Module
SELECT DISTINCT ?uri WHERE { res:Skype dbo:developer ?uri . }

Fig. 1. Four NMT-based KBQA Model

As shown in Figure 1, we divide these four models into two categories, namely Pure Translation(NMT) and Template-based Translation(NMT-E,NMT-R and NMT-ER).

- **Pure Translation:** In this case, NMT model directly trains questions and SPARQL query sequences.
- **Template-based Translation:** In this case, NMT model trains questions template and SPARQL template sequences.

3 Methodology

3.1 SPARQL Encoding

Unlike natural language that can be easily tokenized, SPARQL queries are internally structured, combining elements of the query language with elements from the KBs and variables. Thus, SPARQL Encoding Module is first employed to encode each query as a sequence. Specifically, we ignore the prefixes of URIs. Brackets, wildcards, and dots are replaced by their verbal description. SPARQL operators are lower-cased and represented by a specified number of tokens. These operations can be implemented as a set of replacements, and applying them turns an original SPARQL query to a final sequence that contains tokens that are only formed of characters. An example has shown in Figure 1.

3.2 Tested NMT models

Neural Machine Translation (NMT) models are widely used in intelligent translation, which achieved excellent performance. We use NMT models to translate the English language into the SPARQL. Firstly, we encode the English language questions or templates and SPARQL queries into embedding representations. Then, we fed them to the NMT models for training. Finally, we can convert any English input question into its corresponding SPARQL query.

In this poster, we compare three types of network architectures, RNN-based, CNN-based, and self-attention models, since those represented the best performing NMT architectures in the field at the time of the experiment without considering hybrid and ensemble methods. Encoded SPARQL queries and natural language questions are fed to the network on a word-level.

3.3 Template-based Translating

Considering SPARQL as a foreign language is a novel and direct method in the KBQA task, which turns a question into a SPARQL query with machine translation. However, it would fail to accurately translate the entities and predicates of the question when the entity mentions or relation mentions that have not occurred in the trained set previously.

We consider learning the structure information and local semantic information in question and SPARQL query without entities and predicates, which is translating question template into the SPARQL query template, called Template-based Translation. Since no specific entity is involved and only the location information is learned, we can get better universality and performance.

Template Construction: There are three main ways to preprocess the data for constructing the templates: substitute entities, substitute predicates and substitute both entities and predicates, which has shown in Figure 1. In this step, we rely on existing entity linking tools[6] to recognize, mask, and replace entities in the question with $\langle e_i \rangle$. For the relation mention in the question, we directly recognize the verbs and adjectives in the questions as relations and replace them with $\langle p_i \rangle$.

4 Experiment and Evaluation

4.1 Datasets and Metrics

Our method are evaluated on two well-known public datasets, the Monument dataset, and QALD-9. For training, validation, and testing, we split the datasets randomly by 8:1:1.

Accuracy (Acc). Acc is a metric for evaluating the query results, which is computed as followed:

$$ACC = \frac{the\ number\ of\ right\ answers}{the\ number\ of\ query\ answers} \tag{1}$$

4.2 Evaluation

Table 1. Results on QALD-9

	NMT		NMT-E		NMT-R		NMT-ER	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
CNN-based	0.6607	0.5536	0.7679	0.6429	0.5582	0.5921	0.7500	0.6071
LSTM	0.2500	0.1607	0.3214	0.4821	0.5668	0.6169	0.3036	0.2679
Transformer	0.6786	0.5000	0.7143	**0.6786**	0.6051	0.6255	0.6071	0.5714

As shown in Table 1, "Transformer+NMT-E" beats all other combinations and win the first place, which acc is 0.6786, while the best result in QALD-9 competition is that gAnswer gets ACC = 0.293.

Table 2. Results on Monument

	NMT		NMT-E		NMT-R		NMT-ER	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
CNN-based	0.9851	**0.9876**	0.8830	0.8736	0.9531	0.9659	0.9675	0.9723
LSTM	0.9703	0.9655	0.9155	0.9175	0.9766	0.9703	0.9804	0.9872
Transformer	0.9642	0.9757	0.8830	0.8736	0.9631	0.9652	0.9675	0.9723

As shown in Table 1, "CNN-based+NMT-ER" beats all other combinations and win the first place, which acc is 0.9876. Through the experimental results of two datasets, we can see that it is feasible to translate questions into SPARQL queries by NMT alone. However, its accuracy can be further improved by combining entity recognition and relation recognition.

5 Conclusion

Using natural language questions to query knowledge graphs provides an easy and natural way for common users to acquire useful knowledge. Most traditional approaches for semantic parsing via recognizing entities and relations of the question and assemble them to a semantic query graph; however, it is very time-consuming. Thus, in this poster, we propose a question translation-based method translate natural language questions to SPARQLs. Extensive empirical evaluations over several benchmarks demonstrate that our proposed way is very useful and promising.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2017YFC0908401) and the National Natural Science Foundation of China (61972455,61672377). Xiaowang Zhang is supported by the Peiyang Young Scholars in Tianjin University (2019XRX-0032).

References

1. R. Cai, B. Xu, Z. Zhang, X. Yang., Z. Li, Z. Liang: An Encoder-Decoder Framework Translating Natural Language to Database Queries In *Proc. of* IJ*CAI 2018*, pp. 3977–3983.
2. L. Dong, M. Lapata.: Language to Logical Form with Neural Attention. In *Proc. of ACL 2016*, pp. 33–43.
3. J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin.: Convolutional Sequence to Sequence Learning. In *Proc. of ICML 2017*, pp. 1243–1252.
4. M.T. Luong, H. Pham, C.D. Manning: Effective Approaches to Attention-based Neural Machine Translation. In *Proc. of EMNLP 2015*, pp. 1412–1421.
5. T. Soru, E. Marx, D. Moussallem, G. Publio, A. Valdestilhas, D. Esteves, C.B. Neto: SPARQL as a Foreign Language. SEMANTICS Posters&Demos 2017.
6. Y. Yang and M. Chang Smart: Novel tree-based structured learning algorithms applied to tweet entity linking. In *Proc. of ACL 2015*, pp. 504–513.