Multi-Query Optimization in RDF Q/A System

Jie Jiao, Shujun Wang, 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. In this paper, we present an optimization to answer a question with multiple queries by detecting all common subqueries of that question. Moreover, we apply mutual information in reducing ambiguity of queries of a question to improve the quality of common subqueries. Finally, to improve the accuracy of SPARQL query generated, we evaluate the semantic importance of each word in a question via TF-IDF during the generating queries process. Experiments show that our proposal ourperforms those single query execution.

1 Introduction

SPARQL is a standard way to access RDF data, it remains tedious for users. Hence, qustion/answering(Q/A) based on Knowledge Graph has received wide attention in both natural language processing and database areas.

Generally, there are two significant challenges in RDF Q/A systems:***question understanding***(Question → SPARQL) and ***query evaluation***. A great deal of research has been done to address these two challenges. For example, Zou et al[2] proposed two frameworks to build the semantic query graph. To overcome the second challenge, single-machine RDF systems, like gStore[5] and many distributed SPARQL query engines have been introduced.

Phrase Linking is a huge challenge in Question Understanding stage. It refers to a word w_i in a question may have several meanings, e.g. the word "*Paul Anderson* in question "*Which movies are directed by Paul Anderson*" can map to three items(⟨Paul_S._Anderson⟩ ,⟨Paul_Anderson⟩ and ⟨Paul_W._S._Anderson⟩) in RDF dataset. However, these matches are not all we need. Therefore, we need a way to remove as many failed matches as possible.

The ambiguity in the *Phrase Linking* stage cannot be completely eliminated, therefore, a question may still correspond to multiple SPARQL queries. The current solution is to execute SPARQL queries one by one, but this method is not efficient enough.

In addition, picking which words in the question to generate a SPARQL query is also an important challenge. At present, the method based on statistical information is adopted, while the information provided by statistical methods is relatively limited.

2 Overview of Our Approach

Mutual Information Disambiguation Consider a question "*Which movies are directed by Paul Anderson?*". When we do phrase linking to the word "*Paul Anderson*", we may see that the RDF dataset contains a large number of entities about the word "*Paul Anderson*".

⟨Paul_Anderson⟩ and ⟨Paul_W_S_Anderson⟩ may be directors, but ⟨Paul_S_Anderson⟩ is a teacher. According to the semantics of question, we can actually know that although there are three "*Paul Anderson*" in RDF dataset, Movie-related ⟨Paul_Anderson⟩ and ⟨Paul_W._S._Anderson⟩ are what we really need. Hence, we can delete ⟨Paul_S_Anderson⟩.

The above example illustrates the intuition of our approach, we can deal with ambiguity by counting the number of predicates between entities in advance. However, there are too many entities in RDF graph. In this paper, we point out that collecting two types of lightweight information in RDF graphs:

1. Count the number of relationships between different types of entities.

2. Count the number of relationships between specific entities and different types of entities.

Word Core Measurement In the challenge of Question Understanding, we try to use the SPARQL Q to accurately express the semantics of the question N. In fact, the different words in N are different in the importance of generating Q. However, there is currently no way to measure the importance of each word in N for generating Q. Question Understanding stage can be expressed by the formula:$f(N) \rightarrow Q$.

The natural language question N is composed of the word w_i, and the S-PARQL is composed of triple pattern p_j, hence, we can convert the $f(N) \rightarrow Q$ into $f(w_1, w_2, \cdots, w_n) \rightarrow (p_1, p_2, \cdots, p_m)$, and then by vectorizing w_i and p_j we can get the following formula:

$$f(\overrightarrow{w_1}, \overrightarrow{w_2}, \cdots, \overrightarrow{w_n}) \rightarrow (\overrightarrow{p_1}, \overrightarrow{p_2}, \cdots, \overrightarrow{p_m}) \tag{1}$$

we defined a loss function as follows.

$$L = \sum (\sum_{j=1}^{m} \overrightarrow{p_j} - \sum_{i=1}^{n} \overrightarrow{w_i}) \tag{2}$$

We choose transH[4] to vectorize triple patterns in SPARQL queries. By formula 2, we make the overall $\langle N_i, Q_j \rangle$ in the dataset as equal as possible.

3 Multi-Query Optimization

We can divide all phrases in N into two categories:

(1) "The United States" in Figure 1 (c) only corresponds to the entity ⟨United_States⟩ in Figure 1(d). We use symbol w^o to represent this type of phrases.

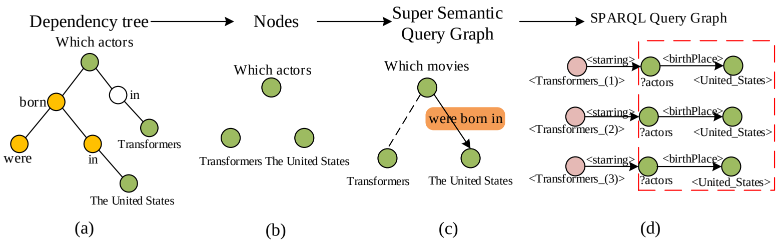

Fig. 1. An example of Question Understanding.

(2) The phrase "Transformers" in Figure 1 (c) corresponds to multiple entities ⟨Transformers_(1,2,3)⟩ in Figure 1(d). We use symbol w^m to represent these ambiguous phrases.

From Figure 1, we can see that subquery $\{?actors\ \langle birthPlace\rangle\ \langle United\_States\rangle\}$ is a common subquery among all SPARQL queries. Hence, we can conclude that the SPARQL queries generated by phrases without ambiguity in a question are unique and common.

Besides, due to the large amount of data in RDF Graph, it is very likely that a phrase in N corresponds to many entities. In this case, too many SPARQL queries are generated. Hence, we present a scoring mechanism for words in question. Pay more attention to phrases that have important implications.

We introduce TF-IDF to measure the semantic importance of different words in a question:

$$Score(w_i) = TF(\frac{|w_i|}{|w|}) \cdot IDF(\frac{|N|}{|N(w_i)|}) \tag{3}$$

$|w_i|$ indicates the number of occurrences of the word w_i, $|w|$ denotes the total number of words. $|N|$ denotes the number of questions in the corpus, $|N(w_i)|$ denotes the number of questions that contain the word w_i.

If the importance of a word w_i is not high, but w_i corresponds to a lot of items in RDF graph. We can ignore w_i. In this way, we can slightly reduce the accuracy, but in return for a great increase in efficiency.

4 Experiments and Evaluation

From the data shown in Table 1, we can see that the accuracy of Our Approach is slightly higher than the original work gAnswer[2] because of the better selection of words w_i in natural language question N.

As can be seen from Figure 2, the efficiency of processing multiple SPARQL queries can be improved by finding common structures between SPARQL queries. Because it avoids redundant execution of common subquery. On this basis, our method can further improve the execution efficiency of multiple SPARQL queries. Because our method can determine the common subquery in the Question Understanding phase. That is, all phrases in the question that

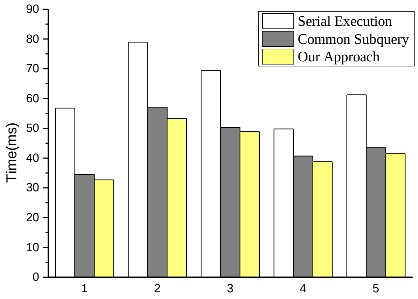

Fig. 2. Question Processing Time

	Processed	Right
Our Approach	100	70
gAnswer	100	68
RFF	100	40
KWGAnswer	100	52
Aqqu	100	36

Table 1. Evaluating QALD Testing Questions

do not contain ambiguity will generate a common subquery after the question understanding stage.

5 Conclusion

In this paper, we addressed the issue of multiple query optimization in RDF Q/A system. We introduce machine learning algorithm to select the useful part of question for SPARQL query generation. At the same time, we give a specific application of multiple SPARQL queries optimization. We hope that our work can inspire other RDF system designers to apply machine learning more in system design.

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. Bidoit N., Herschel M., Tzompanaki A.: Efficient computation of polynomial explanations of why-not questions. In *Proc. of CIKM 2015*, pp.713–722.
2. Hu S., Zou L., Yu J.X., Wang H., Zhao D.: Answering natural language questions by subgraph matching over knowledge graphs (extended abstract). In *Proc. of ICDE 2018*, pp.1815–1816.
3. Ren X., Wang J.: Multi-query optimization for subgraph isomorphism search. *PVLDB*,10(3):121–132 (2016).
4. Wang Z., Zhang J., Feng J., Chen Z.: Knowledge graph embedding by translating on hyperplanes. In *Proc. of AAAI 2014*, pp.1112–1119.
5. Zou L., Özsu M.T., Chen L., Shen X., Huang R., Zhao D.: gStore: A graph-based SPARQL query engine. *VLDB J.*, 23(4):565–590 (2014).