Design and Implementation of an Evaluator for Building a Good Knowledge Base in Question Answering

Sangha Nam, Younggyun Hahm, Sejin Nam, and Key-Sun Choi

Semantic Web Research Center, KAIST, Korea
{nam.sangha, hahmyg, namsejin, kschoi}@kaist.ac.kr

Abstract. In order to design a good knowledge base in a question answering perspective, it is effective to evaluate whether the knowledge base has the answer to the question. For this, it is a common process to translate natural language question to SPARQL form. However, its performance varies depending on the language, structure and length of the question. In this paper, we propose a new evaluation method that translates natural language question to the form of triple. Through this, we can assess whether the knowledge base has the triple required to answer the certain question. Moreover, we can also use the triple as training data for building a good knowledge base. In other words, we can learn how to build a good knowledge base from the question through our new evaluation method. In order to demonstrate our evaluation method, we have developed a KB evaluation program called KB-Evaluator, and conducted an experiment to check the coverage of some knowledge bases.

Keywords: knowledge base design for question answering, knowledge base evaluation, question answering, linked data

1 Introduction

There is a large amount of unstructured information on the web. Open IE [1, 2] has been researched to build a knowledge base by converting information on the web to structured form such as RDF [3]. The reason for building a knowledge base is to answer the user's questions better compared to traditional keyword search. Thus, qualification of a good knowledge base can be found in questions. For example, if the questions are focused on medicine, the knowledge base should have the facts related to medical science. Therefore, the question answering is an essential factor in both building and assessing knowledge base. For designing a good knowledge base in a question answering perspective, there is a gap between the question and the knowledge base. Generally, the question is expressed in natural language and the knowledge base can only process a query expressed in SPARQL [4] form. Consequently, translating natural language question to SPARQL query has been researched in many communities such as OKBQA [5], QALD [6], and LODQA [7]. Template-based SPARQL generation [5, 8] is a common process in recent translation of NLQ to SPARQL. However, its performance varies depending on the language, structure and length of the question. QALD is an evaluation campaign on question answering over linked data, with multilingual and hybrid approaches using information from both structured and unstructured data. LODQA is an open source project to develop a system to generate SPARQL queries from natural language question with a combination technique of relations between two entities [9].
However, these researches may be a way to evaluate a knowledge base, but they cannot provide the guidelines of building a knowledge base through question answering.

In this paper, we propose a new evaluation method for building a good knowledge base in a question answering perspective. Our method does not translate natural language question to SPARQL query, but rather translates natural language question to triple form. With this, we evaluate the knowledge base to measure how much of triples are included in the given knowledge base. Thus, our method evaluates the knowledge base more easily than before, and also the triple made through our method can be used as training data or seed knowledge to build a good knowledge base in a question answering perspective. The new evaluation method we propose consists of 3-phases of data preparation, 4-types of validation basis, and a particular scoring method for Type hierarchy. In order to demonstrate our method, we will introduce how we have developed a knowledge base evaluation program called KB-Evaluator, while also present the experimental results.

2 Data Preparation

2.1 Question Filtering

It is important to filter and select questions based on the feature of target knowledge base. This phase focuses on filtering out unsuitable types of questions. For example, the question below is of an inadequate type since it has to infer the answer from three clues. For example:

**Question.** What do the following bring to mind? Primera Liga, Established in 1899, Lionel Messi.

2.2 Question Declaration

We can make a declarative sentence by simply using a question and its answer. The method of question declaration is placing the answer to the subject of the sentence, and then describing the rest of the contents followed by the subject. If we make a declarative sentence of a question, we can reduce an extra effort to translate natural language query to SPARQL. For example:

**Question.** What is the international organization established by Pierre de Coubertin and Demetrius Vikelas?

**Answer.** International Olympic Committee (IOC)

**Declarative Sentence.** International Olympic Committee is the international organization established by Pierre de Coubertin and Demetrius Vikelas.
### 2.3 Question Triple Generation

Representative open knowledge bases such as DBpedia [10] and YAGO [11] represent knowledge in the form of triples. In order to verify whether a knowledge base includes essential knowledge demanded by the declarative sentence to answer the question, we have to convert the essential knowledge to triple form. For example:

**Declarative Sentence.** International Olympic Committee is the international organization established by Pierre de Coubertin and Demetrius Vikelas.

**Question Triples.**
- `<International Olympic Committee, TYPE, Organization>`
- `<International Olympic Committee, establishedBy, Pierre_de_Coubertin>`
- `<International Olympic Committee, establishedBy, Demetrius_Vikelas>`

**Justification.** When converting a declarative sentence to triple form, we should go through an objective justification of the process and a subjective justification of the result. Firstly, the objective justification is a point of view that considers whether the principle and/or algorithm used is fair, whereas the subjective justification is a point of view that considers whether the triple made by multiple annotators is agreed by more than half of the annotators. The following is a list of principles of question triple generation that we have developed for the objective justification of our knowledge base evaluation method.

**Principles of question triple generation.**
- Only those expressions appeared in the declarative sentence must be used.
- The subject must be a URI presented in the target knowledge base.
- The URI of the subject must have at least one or more Type.
- The object must be either URI or Literal such as String, Integer, etc.

### 3 Validation Basis

Validation basis for evaluating knowledge base with question triples having granted justification can be divided into 4-types.

**Types of validation basis.**
- A-type: `<S, P, O>` in question triple is equal to `<S, P, O>` in knowledge base.
- Ai-type: `<S, P, O>` in question triple is equal to `<S, P, O>` in knowledge base.
- B-type: It is equal to A-type, but property (P) can be anything.
- Bi-type: It is equal to Ai-type, but property (P) can be anything.

The B-type and Bi-type do not compare “property” because they take into account the case in which the property in knowledge base is not well-structured. For example,
there are a lot of cases in English DBpedia that are applied to B-type and Bi-type. As shown below, the birth date of Leonardo da Vinci is represented in three different forms. In this case, all of these triples indicate the same meaning. In this light, B-type and Bi-type can be useful validation basis depending on the knowledge base.

\[
\langle \text{dbpedia-ko:Leonardo\_da\_Vinci}, \text{dbpprop:dateOfBirth}, 1452-04-15(\text{xsd:date}) \rangle \\
\langle \text{dbpedia-ko:Leonardo\_da\_Vinci}, \text{dbpprop:birthDate}, 1452-04-15(\text{xsd:date}) \rangle \\
\langle \text{dbpedia-ko:Leonardo\_da\_Vinci}, \text{dbpedia-owl:birthDate}, 1452-04-15(\text{xsd:date}) \rangle
\]

**Type Scoring.** Schema is a core element of knowledge base. Hence evaluating whether the Type is given in the URI in the knowledge base is needed to verify the value of knowledge base. Aforementioned validation basis gives score 1 if correct, and 0 if incorrect. Unlike this, due to the fact that Types have a hierarchy, a particular scoring method (1) is necessary.

\[
\text{Score} = 1 - |\text{type depth of QT} - \text{type depth of KB}| / \text{type depth of QT} \quad (1)
\]

For example, the Types of Leonardo da Vinci in English DBpedia are defined to have a hierarchical structure as follows; owl:Thing, dbpedia-owl:Agent, and dbpedia-owl:Person. In this case, if evaluating whether Leonardo da Vinci is a painter, the score is 0.66 based on the formula (1) because dbpedia-owl:Painter is a subclass of dbpedia-owl:Person.

### 4 Implementation and Experiment

#### 4.1 Implementation of KB Evaluator

According to the knowledge base evaluation method that we have explained so far, a KB Evaluator has been developed. The workflow of the KB Evaluator is shown in Fig. 1. The question triples made during the course of data preparation are converted to A & Ai-type SPARQL and B & Bi-type SPARQL through SPARQL Generator Module. Each type of SPARQL is created by a way of putting a value to a variable in the simple templates below.
A & Ai-Type SPARQL

ASK WHERE {
  { %SUBJECT% %PROPERTY% %OBJECT%. }
  UNION
  { %OBJECT% %PROPERTY% %SUBJECT%. } } 

B & Bi-Type SPARQL

SELECT DISTINCT ?p WHERE {
  { %SUBJECT% ?p %OBJECT%. }
  UNION
  { %OBJECT% ?p %SUBJECT%. } } 

Each of the SPARQL goes through SPARQL Executor on the end point of knowledge base, and then each result is processed by Triple Scoring Module and Type Scoring Module. Finally, Output Selector prints the coverage.

4.2 Experiments

In order to demonstrate our new knowledge base evaluation method, we have performed an experiment with KB Evaluator, K-Box, L-Box, and NLQ400. K-Box is a knowledge base that expands Korean DBpedia 2014 through enriching the triples from DBpedia category and assigning more Type of each entity represented in DBpedia. Besides, L-Box is a knowledge base that has triples converted using NIF ontology [12] and results of natural language processing from the full text of Korean Wikipedia [13, 14]. NLQ400 is a gold standard set of a total number of 384 pairs of natural language questions and answers that we have prepared to use as testing data. In accordance with our evaluation method, we have conducted question filtering first and selected 201 questions from NLQ400. We could then generate 1282 question triples as a result of

Fig. 2. Coverage of Knowledge Base
the question triple generation process and, among them, 826 question triples were selected once again after subjective justification.

In the experiment, three different knowledge bases were evaluated with regard to question triple coverage and question coverage, and the more detailed experimental results are shown in Fig. 2. K-Box has more triples than Korean DBpedia 2014 through making the relation between the entities presented in DBpedia, so the coverage of question triple has increased about 34.6%. In contrast, L-Box has triples as lexical relation form that does not exist in DBpedia, so the coverage of question triple has increased about 8.8%. Therefore, if we use lexical relation or various forms of relation represented in question as training data or seed knowledge to build a knowledge base, a good knowledge base that can be answered a lot of questions can be made. Consequently, our new knowledge base evaluation method can be the guidelines of building the knowledge base through question answering.

5 Conclusion

We have proposed a new evaluation method for building a good knowledge base in a question answering perspective. Our method does not translate natural language question to SPARQL query, but rather translates natural language question to triple form. Through this, we can assess whether the knowledge base has the triple required to answer the certain question. Thus, our method evaluates the knowledge base more easily than before, and also the triple made through our method can be used as training data or seed knowledge to build a good knowledge base. We have also developed the KB Evaluator Program, and introduced the experimental results for K-Box, L-Box and NLQ400.

In the future, we will aim to develop an automatic program for data preparation that involves question filtering, question declaration and question triple generation. Not only this, we will carry out further researches to overcome the current limitation of the schema of property as well.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. R0101-15-0054, WiseKB: Big data based self-evolving knowledge base and reasoning platform)

References