SRDF: Extracting Lexical Knowledge Graph for Preserving Sentence Meaning

Sangha Nam, GyuHyeon Choi, Younggyun Hahm, and **Key-Sun Choi**
Machine Reading Lab, School of Computing,
Korea Advanced Institute of Science and Technology (KAIST),
Daejeon, Republic of Korea
{nam.sangha, wiany11, hahmyg, kschoi}@kaist.ac.kr

Abstract

In this paper, we present an open information extraction system so-called SRDF that generates lexical knowledge graphs from unstructured texts. In semantic web, knowledge is expressed in the RDF triple form but the natural language text consist of multiple relations between arguments. For this reason, we combine open information extraction with the reification for the full text extraction to preserve the meaning of sentences in our knowledge graph. And also our knowledge graph is designed to adapt for many existing semantic web applications. At the end of this paper, we introduce the result of an experiment and a Korean template generation module developed using SRDF.

1 Introduction

The web contains enormous information in the form of unstructured text. In recent years, Open Information Extraction (IE) based on self-supervised learning has become more strongly suggested to overcome limitations of traditional IE system, and it is now possible to process massive text corpora. However, early Open IE systems fall short of representing multiple relations between arguments within a sentence since they are designed to focus on binary extractions. This causes incomplete and insufficient extraction. To overcome this limitations, Kraken(Akbik and Löser, 2012), OLLIE(Mausam et al., 2012) and ClausIE(Del Corro and Gemulla, 2013) are designed to extract a set of arguments using dependency parsing and then represent the extracted knowledge as ternary or N-ary form.

Consider, for example, the sentence "*Marsel was established by the British government with the help of American policymakers in 1971 as the nation's first research oriented science institution.*". Current Open IE systems focus on extracting triples; (*Marsel, was established by, the British government*) and (*Marsel, was established in, 1971*). Even if these systems extract multiple triples, there is still missing information. The arguments '*help of American policymakers*' and '*the nation's first research oriented science institution*' are also important and necessary information for a question answering system when a question becomes more complicated. Furthermore, it is important to represent extracted knowledge for applying to existing semantic web applications.

In this paper, we propose an Open Information Extraction system so-called SRDF that generates lexical knowledge graph from unstructured texts. SRDF differs from other Open IE systems in terms of full sentence extraction and knowledge representation in reified triple form. In semantic web, knowledge is commonly expressed in RDF triple form that consists of subject, predicate and object. However, there are a lot of cases that multiple relations between arguments are associated within a sentence. The purpose of SRDF is to make a bridge between text and triple by the lexical knowledge graph. Not only does SRDF knowledge graph (KG) reflect the dependency structure of a sentence but can also be used in a variety of semantic web applications such as question answering.

2 What is SRDF?

Proceedings of the Open Knowledge Base and Question Answering (OKBQA) Workshop,
pages 77–81, Osaka, Japan, December 11 2016.

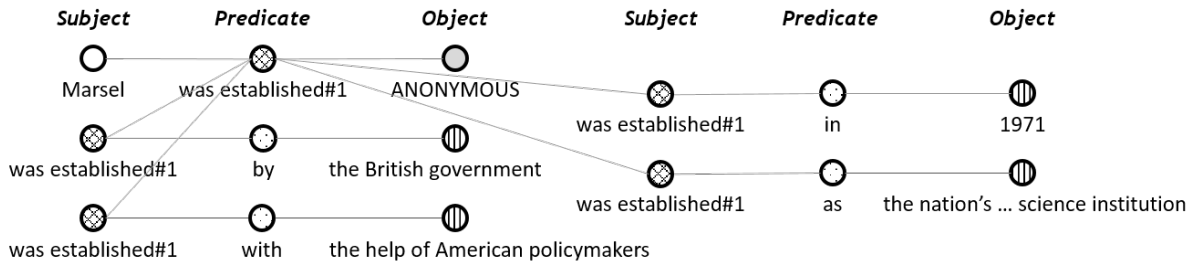

Figure 1: Example of SRDF knowledge graph

Extracting ontological triples directly from the text needs many steps such as entity linking, disambiguation and predicate linking, and also needs many resources like *Wordnet*. Nevertheless, usual performance is still unsatisfiable because the whole process is complicated and errors of each step propagating to the next step. That is why Open IE has been researched.

The purpose of SRDF generates a lexical knowledge graph as a bridge between text and triple. SRDF means sentence-based lexical knowledge graph structure. SRDF structure serves three purposes. First, it translates an input sentence to reified triple form with simple and concise rules and reflects the dependency structure of the sentence. Second, it supports handling both entity-centric and event-centric facts. Third, it is designed to be used in various semantic web applications.

As mentioned earlier, multiple arguments and relations are presented within a sentence, so we design our structure using *Singleton Property*(Nguyen et al., 2014) - the new method of reification. The main idea of *Singleton Property* is that every relationship is universally unique, so the predicate between two particular entities can be a key for any triple.

Figure 1 is an example of SRDF a knowledge graph from the sentence described in the introduction. As shown in Figure 1, SRDF follows a triple form. The reason for taking triple form is a versatility. Triple is the simplest semantic web representation form, moreover many semantic web applications such as knowledge base and question answering systems take the triple form. Therefore we extract the knowledge in triple form for integration with existing applications easily. As a knowledge graph, SRDF also consists of triples but of different properties followed by:

- **Subject** can only be the subject of the sentence or reified predicate.
- **Predicate** can only be the verb group of the sentence or pre/postposition of its objects.
- **Object** can only be the noun group or ANONYMOUS.

3 Overview and Workflow Description

SRDF system simply receives input as a text and outputs an extracted set of reified triples. Our system operates through three steps of procedure in total that are Preprocessor, Basic Skeleton Tree (BST) generator, and SRDF generator. Detailed explanations are following as shown in Figure 2.

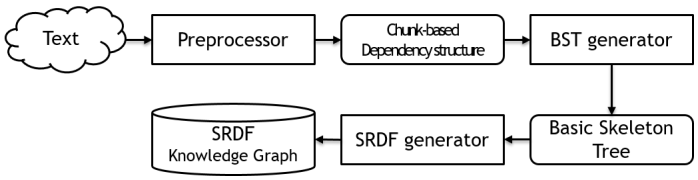

Figure 2: System architecture of SRDF

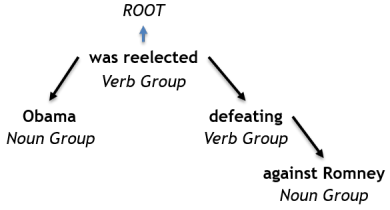

Figure 3: Example of chunk-based dependency structure

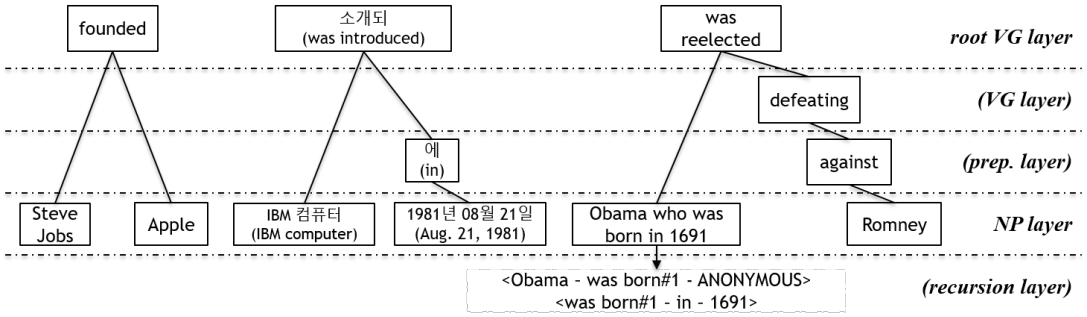

Figure 4: Three examples of Basic Skeleton Tree in SRDF

Figure 5: Example of SRDF reified triple generation

3.1 Preprocessor

Preprocessor consists of three sub-modules which are Sentence segmentor, Chunker, and Dependency parser. Sentence segmentor divides a sentence into its component sentences and attaches the subject to divided sentences. Chunker returns only noun phrases and verb groups. Noun phrases can contain adnominal phrase and verb groups could contain adverb phrase. Then, dependency parser outputs a chunk-based dependency structure like figure 3.

3.2 Basic Skeleton Tree Generator

BST generator takes an input as a chunk-based dependency structure and outputs a BST. Chunk-based dependency structure strongly depends on charasteristics of languages. For example, dependency structure of English, Korean and Chinese are different from one another. Therefore we make an intermediate structure between chunk-based dependency structure and SRDF. BST would be almost the same structure for any language could be adjusted to SRDF generation rules as well.

Figure 4 is examples of the Basic Skeleton Tree. BST has five layers; root-VG, VG, NP, preposition and recursion layer. Root-VG layer is the top layer and has only one node that is root verb group on dependency structure. NP layer contains all noun phrases including subject of the sentence. VG layer is placed between the root-VG and the NP layer. There could be more than one VG layers relying on depth of corresponding verb groups in chunk-based dependency structure. Preposition layer contains only preposition of its noun phrase and be placed over the NP layer. Recursion layer decomposes noun phrase with more detail when it contains adnominal phrase.

3.3 SRDF Generator

SRDF generator takes an input as a BST and outputs a lexical knowledge graph as reified triple form using our simple and concise algorithm as shown in Algorithm 1. It takes a graph *G*, a subject of sentence *sbj*, a root verb group *pred*, and child nodes of root verb group *objQueue* as input and returns *G*. For each *obj* in *objQueue*, check whether it is in NP layer or not. If the *obj* is in NP layer, make a triple and insert it to graph *G* (Line 4). If not, make an *ANON* triple and insert it to *G* and then change *sbj* and *pred* respectively for reification (Line 6 to 8). And then, call *generateSRDF* function recursively (Line 9). Figure 5 is an example of our algorithm about the third BST in Figure 4.

Algorithm 1 SRDF reified triple generation algorithm

```
1: procedure GENERATESRDF(G, SBJ, PRED, OBJQUEUE)
2:     for obj in objQueue do
3:         if obj is in NP layer then
4:             G ← G ∪ {<sbj,pred,obj>} ▷ Overwrite ANON object with the same sbj and pred
5:         else
6:             G ← G ∪ {<sbj,pred,ANON>}
7:             sbj ← pred
8:             pred ← obj
9:             generateSRDF(G, sbj, pred, obj.children)
10:    return G
```

Table 1: Results of experiments.

Precision	Recall	Completeness
0.74	0.75	0.93
301/407	251/336	128/137

4 Experiments and Application

4.1 Experiments

The performance of SRDF system has been evaluated with randomly sampled sentences from featured article in *Korean Wikipedia*. The evaluation results have been assessed by two human evaluators based on the precision, recall and the number of extractions. As shown in Table 1, our system extracted 407 triples from 137 sentences. The precision is 74% and the recall is 75%. The completeness means if all the information is extracted as triples from an input sentence or not. Overall completeness is 93%. We found that the 7% of incomplete extractions is caused by the Korean Analyzer, especially a problem of correctly finding the subject of a given sentence. In our experimental results, the precision and recall are similar to recent open information extraction systems' but our system can extract all information from the input sentence. Through the results, we assume that SRDF could be useful in QA task for a relatively long question.

4.2 SenTGM in OKBQA platform

Open Knowledge Base and Question Answering (OKBQA) is a community and a hackathon to make advanced technology for developing a question answering system. The virtue of OKBQA is open collaboration that harmonize resources developed by different groups scattered over the world. We made SenTGM for the first step of OKBQA called Template Generation using our SRDF system. SenTGM takes a Korean natural language question and produces a pseudo query defined in Templator (Unger et al., 2012) and it is now working for the Korean natural language question in OKBQA framework properly. The architecture and example of SenTGM is shown in Figure 6.

5 Conclusion

In this paper, we proposed a new open information extraction system called SRDF. Our approach is a novel method of combining Open IE with the singleton property technique for the full text extraction. Furthermore SRDF represents extracted knowledge as reified triple form for usability in many existing semantic web applications. And also we demonstrated that our approach can be used in the OKBQA framework for question answering. In the future, we will research a question answering approach over SRDF knowledge graph using the synonym such as Wordnet and word embedding to resolve the ambiguity.

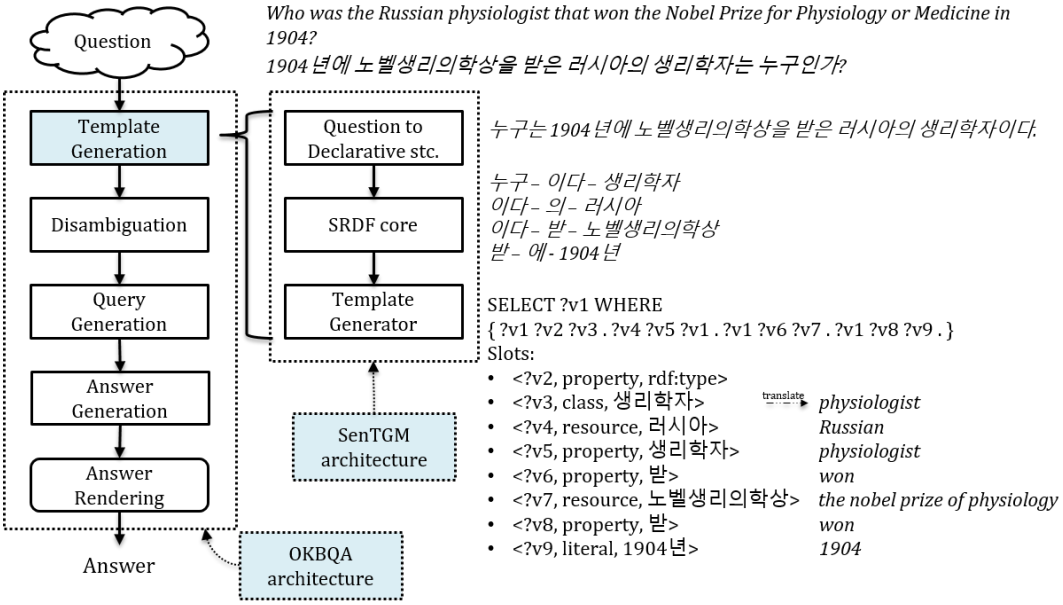

Figure 6: Architecture and Example of SenTGM

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. R0101-16-0054, WiseKB: Big data based self-evolving knowledge base and reasoning platform). This work was supported by the Bio & Medical Technology Development Program of the NRF funded by the Korean government, MSIP(2015M3A9A7029735)

References

Alan Akbik and Alexander Löser. 2012. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pages 52–56, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 355–366, New York, NY, USA. ACM.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. 2014. Don't like rdf reification?: making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770. ACM.

Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648. ACM.