QAF: Frame Semantics-based Question Interpretation

Younggyun Hahm, Sangha Nam, Key-Sun Choi
Machine Reading Lab
KAIST
Republic of Korea
`{hahmyg, nam.sangha, kschoi}@kaist.ac.kr`

Abstract

Natural language questions are interpreted to a sequence of patterns to be matched with instances of patterns in a knowledge base (KB) for answering. A natural language (NL) question answering (QA) system utilizes meaningful patterns matching the syntactic/lexical features between the NL questions and KB. In the most of KBs, there are only binary relations in triple form to represent relation between two entities or entity and a value using the domain specific ontology. However, the binary relation representation is not enough to cover complex information in questions, and the ontology vocabulary sometimes does not cover the lexical meaning in questions. Complex meaning needs a knowledge representation to link the binary relation-type triples in KB. In this paper, we propose a frame semantics-based semantic parsing approach as KB-independent question pre-processing. We will propose requirements of question interpretation in the KBQA perspective, and a query form representation based on our proposed format *QAF (Question Answering with the Frame Semantics)*, which is supposed to cover the requirements. In QAF, frame semantics roles as a model to represent complex information in questions and to disambiguate the lexical meaning in questions to match with the ontology vocabulary. Our system takes a question as an input and outputs QAF-query by the process which assigns semantic information in the question to its corresponding frame semantic structure using the semantic parsing rules.

1 Introduction

Nowadays, there are many ongoing researches to build a knowledge base question answering (KBQA) system with the growing interest of KBs such as Freebase (Bollacker *et al*., 2008), DBpedia (Auer et al., 2007) and YAGO2 (Hoffart *et al*., 2011). Most of KBs consist of structured data in triple form <`s, p, o`>, and SPARQL query is used to access triple data. However, for common users, it is required to learn query language and the schemas underlying KBs. Thus, providing intuitive interfaces for KBQA is an important task to help users access the massive amount of information in KB. Question interpretation is an essential task to generate the suitable query to answer natural language questions by translating it. And there are many research efforts such as QALD[1] and OKBQA[2] to address this problem.

Traditionally, to translate natural language question into machine readable query, there are two major approaches, the information extraction approach and the semantic parsing approach (Yao *et al*., 2014a). The information extraction (IE) approach learns meaningful patterns and rules by matching the syntactic structure of question with the schemas in KB, and the lexical features with the ontology vocabulary in KB (Yao *et al*., 2014b). This process is based on the traditional IE approach such as a distant supervision

[1] http://qald.sebastianwalter.org/
[2] http://www.okbqa.org/

Proceedings of the Open Knowledge Base and Question Answering (OKBQA) Workshop,
pages 82–90, Osaka, Japan, December 11 2016.

(Mintz *et al.*, 2009), and it generates KB specified query. For example, to answer the example question "Who was the first person reached the South Pole?" over the target KB, the IE approach extracts triples from the question based on the schema underlying target KB. Let's consider DBpedia as the target KB to answer the example question. In DBpedia, there is knowledge to answer the question that exists in the triple form `<dbr:Roald_Amundsen, dbo:knownFor, dbr:South_Pole>`. First, a IE-based system searches sentences which includes the entities, `dbr:Roald_Amundsen` and `dbr:South_Pole`. If a sentence "…Roald Amundsen was the first Norwegian explorer to reach the South Pole…" is discovered, then the system learns patterns syntactic/lexical features by matching the sentence and the triple. For instance, if there are three conditions in a sentence; 1) a word "reach" in a sentence, 2) a subject is PERSON, and 3) an object is LOCATION, a triple is generated by using a pattern rule; `<subject, dbo:knownFor, object>`.

This SPARQL would be supposed suitable to answer the example question over DBpedia;

```
SELECT ?x WHERE {
       ?x    rdf:type      dbo:Person          .
       ?x    dbo:knownFor  dbr:South_Pole      . }
```

In the above SPARQL query, `rdf:type` and `dbo:knownFor` are properties in ontology, and `dbo:Person` is a class in ontology. Expected query result is an entity `dbr:Roald_Amundsen`, which is matched with variable `?x` in the two triple patterns, `<?x, rdf:type, dbo:Person>` and `<?x dbo:knownFor dbr:South_Pole>`.

In the example question and its SPARQL, the interrogative word "who" is considered as a variable `?x` in query, and its type is expected to be `dbo:Person` in combination with the word "person" in question. By using the triple pattern `<?x, rdf:type, dbo:Person>`, SPARQL can represent the query intention: '*the expected answer would be a person*'. The IE-based approach extracts the triple pattern `<?x dbo:knownFor dbr:South_Pole>` from the example question by matching the syntactic/lexical features in the example question to DBpedia.

In this case, the property `dbo:knownFor` is used to represent the relationship between two entities; `?x` and `dbr:South_Pole`. The relation is extracted by using the syntactic features such as the grammatical role and named entity, and by using the lexical features such as the meaning of the verb, in this case, "reached". The word "reached" is used to disambiguate the relation and match it with the property `dbo:knownFor`. This IE-based question interpretation is a model with focusing target KB. It would be easy to learn patterns for the specified domain KBs.

However, the IE-based approach involves a limitation of the number of learnable rules because of not only the lack of the ontology vocabulary (Berant *et al.*, 2014) but also the way of expression of knowledge. First, the lack of the ontology vocabulary involves the lack of coverage for the scope of question interpretation. Especially, in our example, DBpedia is constructed under the its own schema, DBpedia Ontology. It is based on the Wikipedia and Infobox (Auer *et al.*, 2007), thus it is suitable to represent factual knowledge such as *NAME*, *JOB*, *POPULATION*, *HEIGHT*, and *NATIONALITY*, because of the characteristic of the Wikipedia as an encyclopedia. However, for the example question, there are irregular mappings between the word "reached" and the ontology vocabulary because of the absence of the proper property to represent the meaning of "reach". By this reason, the word "reach" sometimes would be mapped with the several properties such as `dbo:location`, `dbo:residence`, `dbo:knownFor`, and even `dbo:wikiPageExternalLink`. It is a reason why there is the limitation to interpret the question enough in the KB-dependent approach (Hahm *et al.*, 2014). Second, there is the gap between natural language and structured data in the perspective of the expressiveness. In other words, natural language represents complex information underlying its various syntactic/semantic structure, however, structured data represents information using its schema. In the RDF syntax, there are many ways to represent complex information, such as the attributes of the relation. In our SPARQL example, the variable `?x` has a relation `dbo:knownFor` with the entity `dbr:South_Pole`. To represent more information shown in the example question, the relation `dbo:knownFor` would have an attribute, for example, 'something ?x is known for the South Pole **as the first person** to reach'.

However, in most of KBs, there are only binary relations in triple form to represent relation between two entities. Therefore, for instance, even we have these two triples; `<dbr:Roald_Amundsen,`

`dbo:knownFor, dbr:South_Pole>` and `<dbr:Roald_Amundsen, isa, first_man>`, we don't know information for 'dbr:Roald_Amindsen is known for South_Pole as the first person' because of the absence of the relation between the property `dbo:knownFor` and the concept of "first man". Thus the specific KB-dependent approach has the limitation of the scope of representable knowledge, in our example, the attribute of the relation.

By contrast, the semantic parsing (SP) is considered the KB-independent approach to analyse user's intention and semantics of information in questions (Xu *et al*., 2014). The SP approach is not dependent on the specific KB, so that it is efficient on the open domain question answering (Yao *et al*., 2014a). In this paper, we propose SP approach based on the frame semantics in FrameNet (Baker *et al*., 1998) to interpret questions. FrameNet uses a PropBank-style predicate-argument structure to represent relations between each argument. Each relation evoked by *target* words, and each relation is disambiguated by assigning the target words to the frames. For instance, in our example question, the word "reached" roles as a *target* and evokes the frame `Arriving` (`frame:Arriving`), and the word "first" also evokes the frame `First_experience` (`frame:First_experience`). The frame is a lexicon to represent not only encyclopedia-like information similar to DBpedia Ontology, but also linguistic level semantics for various information such as *CAUSE & EFFECT*, *EMOTION*, *OPINION*, *MOTION*, *PROBLEM & SOLUTION* and so on. These frames would be used for bottom-up grounding of knowledge to interpret questions in the perspective of KBQA, and is used for the ontology vocabulary model for KB directly (Vossen *et al*., 2014; Rouces *et al*., 2015). In this paper, as an approach to interpret questions, our goal is to generate the model for machine readable query based on the frames, and our scope is to analyse the single sentence factoid Korean questions as the first step of KBQA system.

To achieve our goal, in Section 2 we will propose the requirements of question interpretation and define the logical form query, QAF, which is supposed to cover the requirements. We designed the frame-based semantic parsing rules for Koran questions in Section 3, and the evaluation result and discussion are described in Section 4.

2 Question Interpretation based on the Frame Semantics

In this section, we define QAF based on the frames for query which are interpreted from NL questions. QAF is designed to cover the requirements of question interpretation in KBQA system.

2.1 Requirements of Question Interpretation

To translate questions into a machine readable queries, there are some requirements which should be analysed. For example, the question:

> What was the naval warfare commanded by Admiral Yi Sun-sin at Myeongyang Strait in 1597?

The proper SPARQL query which is translated from the question to get answer from DBpedia would be:

```
SELECT ?x WHERE {
       ?x     rdf:type           dbo:MilitaryConflict       .
       ?x     dbo:commander      "Admiral Yi Sun-sin"       .
       ?x     dbo:place          "Myongyang Strait"         .
       ?x     dbo:date           "1597-00-00"               . }
```

Traditionally, KBQA considers the following three elements as the major things in the question interpretation task (Yao *et al*., 2014b).

(1) Expected answer type (in our example, dbo:MilitaryConflict)
(2) Question words (What)
(3) Clues of the question (who is commander, where occurred at, when occurred in)

In most of KBs, each entity is defined by using an ontology class (e.g. PERSON, LOCATION, EVENT, and so on), and it is useful to reduce the search space and to select the more disambiguated entities in the process of selecting answer candidates. Thus, in the question interpretation task, the process which identifies and disambiguates the expected answer type in requirement (1) is a major subtask. Also identification of question words in requirement (2) is used to figure out user's intention. The SPARQL query differs for each question words such as "how many", "what is the highest" and "who", in the different way to get answers (Unger *et al.*, 2012). The clues of questions in requirement (3) is written in a triple pattern, <?answer, p, o> in the SPARQL query, to find the variable ?answer in KB. In this paper, we define QAF as a model which covers these requirements, and we developed a question interpretation system which assigns the requirements in questions to the frame structure using the semantic parsing rules that we experimented for the Korean question.

2.2 QAF: Question Answering with the Frame Semantics

Before developing our question interpretation system, we examine the dataset so-called NLQ400 which is used for (Nam *et al*., 2015). NLQ400 consists of the 384 Korean questions which covers various domains, such as history, science, art, and so on. We choose 95 factoid questions which could be answered by using one single sentence in Wikipedia, and then choose 72 questions excepting multiple choice questions and O/X questions. And then we manually annotate frames for the 72 questions to figure out how to use frames for question answering. For our example question, the frame annotation result is:

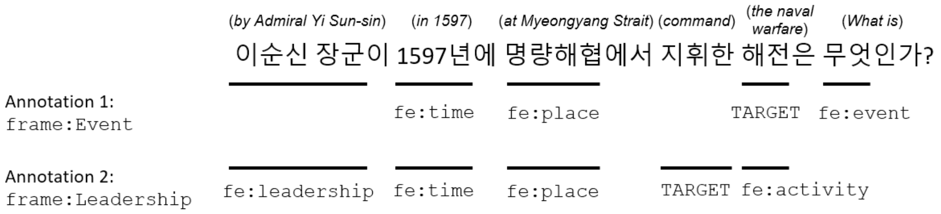

Figure 1 An Example of Frame Annotation for Korean Question

In Figure 1, for our example question "What was the naval warfare commanded by Admiral Yi Sun-sin at Myeongyang Strait in 1597?", the word "the naval warfare" is a *target* word which evokes the frame:Event, and the semantic role of its arguments are defined by using each frame element (FE) in the frame:Event, such as fe:time, fe:place. In this result, the question word "What" is annotated as a FE (i.e. fe:event). And, in the annotation 2, the word "command" evokes the frame:Leadership, and each FE is; leader:"Admiral Yi Sun-sin", time:"1597", place:"Myeongyang Strait", and activity: "the naval warfare". In this case the question word "What" does not annotated as a FE in the annotation 2.

By these annotations, we figure out (1) the expected answer type and (2) the question word are annotated in the annotation 1, and (3) the clues of the question is in the annotation 2. All of 72 questions is annotated in the case of annotation 1. And the word for identification of the expected answer type "the naval warfare" is a node which connects each annotation. Thus, in QAF, the case of annotation 1 would be a basic graph to represent questions in the structured format, and the other annotations are connected with the annotation 1 by using the word for the expected answer type.

We define some terms: the word for the connecting node "the naval warfare" as Q-frame, and the question word "What" as Q-FE, and the clues of questions as Sub-Frame, which is the frame:Leadership and its FEs in our example.

The resulting graph for Figure 1 is a representation for QAF (Question Answering with Frame Semantics) to satisfy the requirements. Section 3 is about developing QAF for Korean QA.

3 Frame-semantic Parsing of Question Sentence

3.1 Scope of development

To develop the Korean question interpretation system, we list up the several goals:

Use less amount of training data

English FrameNet[3] is a well-constructed lexicon in its long history, and there are many well-performing frame semantic parsers (Das *et al*., 2010) using 19,582 target words in 154,607 sample sentences and 3,256 training data sentences in FrameNet. For Korean, there is Korean FrameNet corpus which is constructed by (Park *et al*., 2014), which had 6,802 target words in 5,507 sentences. However, it is the insufficient amount to use for training, and, furthermore, there are a few number of frame annotation for questions in our best knowledge, in both of English and Korean. Thus our system is built by using existing NLP tools without training process.

Coverage for questions

In this paper, we choose the SP approach to interpret questions. To according with this, the system should deal with the various type of questions and analyse the requirements of the question interpretation task in KBQA.

Use standardized format

The system will be used for question interpretation module to generate machine readable query, SPARQL. To publish our system as an open-source, all of results is in JSON and RDF format for the convenience for the other users who want to use it for their KBQA system.

3.2 Q-frame and Q-FE Identification

In this section, the process, Q-frame/Q-FE identification is described.

We figure out that there are three type of questions.

Table 1 The Rules for Q-frame and Q-FE Identification based on the Question Type

Question Pattern	Question Type	Dependency of Q-frame	Root Node
What is the naval warfare … ?	1	NP_SBJ, dist=1	VNP
What the naval warfare … ?	1	NP_SBJ, dist=1	NP
Is the naval warfare … ?	2	NP_SBJ, dist=0	NP_SBJ
The naval warfare … ?	2	NP, 0	NP
Describe about the naval warfare … .	3	NP_OBJ, dist=0	VP

The tag `NP_SBJ` is for the noun phrase which roles as a subject in a sentence, and `NP_OBJ` roles as an object. The tag `VP` is for verb phrase, and `VNP` is for the verb phrase as the copula.

The type 1 is a typical factoid question, for instance, "What was the naval warfare…?". In the type 1, the question word, Q-FE, is represented within interrogative pronouns. The type 2 is a question without the interrogative pronouns. This case is well-shown in many Korean questions, such as "The naval warfare commanded by Admiral Yi Sun-sin?". The type 3 is a imperative sentence, for example, "Describe about the naval warfare which…". To cover three type of questions, our system is built by using the rules that we designed in Table 1. For our example question, "What is the naval warfare…?", our system finds the head node in the dependency structure and figure out its phrase tag, `NP`, and find its child nodes (dist=1) and its phrase tag, `NP_SBJ`. And then our system figures out the word "naval warfare" as a target word that evokes Q-frame, `frame:Event`. And then the system identifies Q-FE based on its question type. If the type is 2 or 3, the system makes a virtual node for Q-FE, and if the type is 1, the root node is considered as Q-FE. Figure 2 shows the result for our example question.

In Figure 2, the system identifies the word "the naval warfare" as a target of Q-frame, and the word "What" as Q-FE by using the rules. And then each word is assigned to frames by using the mapping table which consists of word-frame pair based on the 6,820 lexical units in Korean FrameNet[4]. In our example, the word "the naval warfare" is assigned to `frame:Event`, so that the expected answer type is considered as an Event.

[3] https://framenet.icsi.berkeley.edu/fndrupal/

[4] http://framenet.kaist.ac.kr

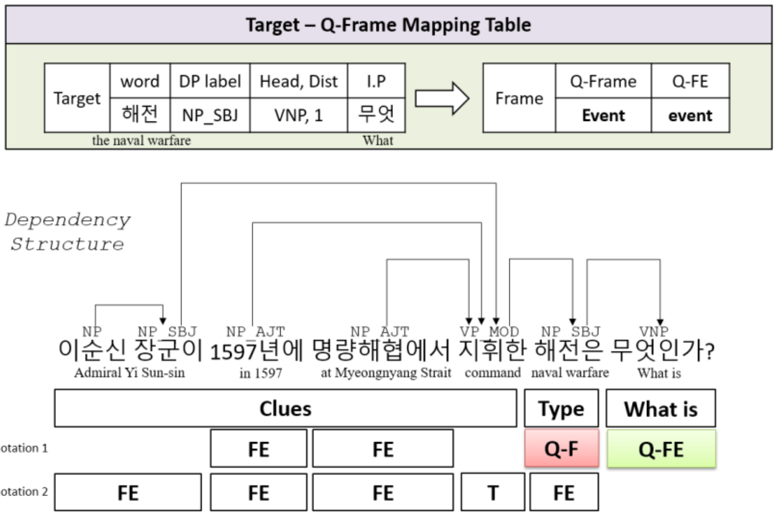

Figure 2 Q-frame and Q-FE Identification and Target-frame Mapping using Dependency

3.3 Sub-frame Identification

The purpose of Sub-frame is to include the clues of questions in query, for example, `<?x, p, o>` format triple patterns in SPARQL query, for information such as "commanded by Admiral Yi Sun-sin", "at Myeongyang Strait", and "in 1597" in our example question. To generate these triple patterns, the system uses the predicate-arguments structure based on the frames in a question. In this paper, we use the existing Korean SRL tool (Lim *et al*., 2014) to analyse predicate-arguments structure.

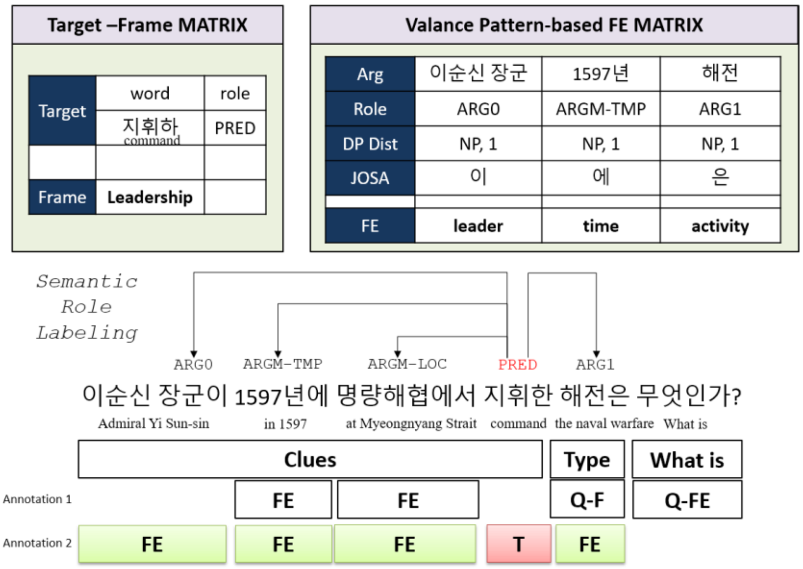

Figure 3 Sub-frame Identification using SRL

SRL tool identifies the target word of Sub-frame by using an identified predicate of sentence. Each FEs are identified by using arguments that identified by SRL also. And then each target word is assigned to the frames by using the mapping table which is used in the Q-frame identification process. The valence pattern is a grammatical condition of each FE. In Figure 3, the argument "Admiral Yi Sun-sin" is assigned to the FE tag `fe:leader` by combining the condition of josa and SRL tag.

However, sometimes SRL tool does not figure out the predicate-arguments structure for some questions, and several arguments are not identified also in some cases. Especially, the node Q-FE in QAF is used to connect each predicate-arguments graph, so that Q-FE should be identified. We developed several post-processing modules to handle these problems.

3.4 Post-processing

For the sentence without predicates

The PropBank-style SRL tools does not figure out the predicate-arguments structure for sentences without verbs. However, in the question “Who is the member of the Singanhoe?”, the word “member” implicits that the expected answer type is a PERSON, and the phrase “the member of the Singanhoe” includes the clues to answer the question. Thus, even though there are no predicate-arguments structure in the question, information of question should be represented in the query. Our system outputs this results as a clue of the question; `<Member, description, Member of the Singanhoe>`

Handling undetected arguments

The target words of Q-frame connect each frame graph in QAF. However, in some case, the target word of Q-frame is not identified in the other frame graphs as an argument. Thus, if a predicate-arguments graph failed to identify the target words as an argument, our system adds it as an argument for all of predicate-arguments graph which does not include it.

Connect each predicate-arguments graph

In SRL results, each identified predicate-arguments graph is in each independent annotation layer. Our system connects each graphs by matching the spans of each argument.

Phrase chunking

Existing Korean SLR tools identify only a last token (called *eojeol* in Korean) of a noun phrase as an argument. The phrase chunking module is developed for our system to identify noun phrases as arguments in predicate-arguments graphs. Conjunctive noun phrases are considered as arguments of the predicate, and *josa* (particles in Korean) is dropped out of arguments.

3.5 QAF result

As a result of our system, QAF is generated from a question based on RDF format. For our example question, “What was the naval warfare commanded by Admiral Yi Sun-sin at Myeongyang Strait in 1597?”, our system outputs;

```
frdf-event:해전#1                                     (the naval warfare)
        rdf:type      frame:Event         ;
        fe:name       ?answer             ;

frdf-event:지휘하#1                                   (command)
        fe:leader     “이순신 장군”         ;         (Admiral Yi Sun-sin)
        fe:time       “1597년”             ;         (1597)
        fe:place      “명량해협”            ;         (Myeongyang Strait)
        fe:activity   frdf-event:해전#1    .         (the naval warfare)
```

The target words, “the naval warfare” and “command”, are given URIs (Uniform Resource Identifiers) and role as a subject in triples, and the arguments role as an object in triples. The FE tags is used as properties. QAF does not represents binary relations, like DBpedia, but represents events and its elements in the RDF format by using the method that n-ary relation with creating a individual (in our example, frdf:event) to role as a subject and generating links to all arguments with the FE tags which role as properties. This event-centric representation would cover the complex information in questions based on the frame structure.

4 Evaluation and Discussion

4.1 Frame Identification

The evaluation is performed on the NLQ50[5] data in OKBQA. We use 45 questions excepting O/X questions and description question as our test data.

For 45 questions, our system identifies all target words of Q-frame for every question, and 51 target words of Sub-frames also. 58 frames are assigned for 96 target words, and all of frames are correctly assigned by manual evaluation. And our system identifies 36 FEs of Sub-frames. By manual evaluation, Sub-frame identification task is evaluated as Table 2.

Table 2 Evaluation of Frame Identification

Task	Precision	Recall	F1
Frame Identification	1.0	0.6041	0.7531
FE Identification	0.90	0.73	0.8137

4.2 Discussion

Frame identification

Our system uses the lexical units of Korean FrameNet to assign the frames to target words. However, the coverage of the lexical units is about 60%, so that it is required to increase the overall performance. As future work, we plan to develop the frame identification module based on the word embedding approach (Hermann *et al*., 2014) to increase the coverage.

For the multiple questions

In the scope of this paper, our system deals with only a single sentence question. Although it performs well, but it is required to handle multiple sentence questions, such as a complex sentence, O/X questions, and multiple choice questions. Especially we focus on the multiple sentence question as the future work.

Ontology mapping

QAF is a format based on the frames to represent information of questions in structured format with assuming that there is the imaginary KB. To develop a question interpretation system for existing KBs, it is required to map QAF with SPARQL underlying its ontological schemas.

5 Conclusion

In this paper we designed a format, QAF, to represent complex information of questions in the event-centric RDF format. The KB-dependent approach extracts only binary relations from questions, and it involves the limitation of coverage of question interpretation because of the incompleteness of KB. And the schemas in KBs does not cover the all of the lexical meaning in questions also. For this reason, we propose the semantic parsing approach based on the frame semantics to analyse complex information in questions. And then we developed the system which translates Korean questions into QAF. Handling multiple sentence questions and mapping QAF to existing KBs are remains as the future works.

Acknowledgments

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)

[5] http://3.okbqa.org/development/resources

Reference

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer Berlin Heidelberg, 722-735.

Collin F. Baker, Charles J. Fillmore, John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1* (pp. 86-90). Association for Computational Linguistics.

Jonathan Berant, Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *ACL* (1) (pp. 1415-1425).

Kurt Bollacker, Coling Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM.

Dipanjan Das, Nathan Schneider, Desai Chen, Noah A. Smith. 2010. SEMAFOR 1.0: A probabilistic frame-semantic parser. *Language Technologies Institute, School of Computer Science, Carnegie Mellon University*.

Younggyun Hahm, Youngsik Kim, Yousung Won, Jongsung Woo, Jiwoo Seo, Jiseong Kim, Seongbae Park, Dosam Hwang, Key-Sun Choi. 2014. Toward Matching the Relation Instantiation from DBpedia Ontology to Wikipedia Text: Fusing FrameNet to Korean, In *Proceedings of iSemantics 2014*.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, Kuzman Ganchev. 2014. zSemantic frame identification with distributed word representations. In *ACL*.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194, 28-61.

Soojong Lim, Changki Lee, Pum-Mo Ryu, Hyunki Kim, Sangkyu Park, Dongyul Ra. 2014. A Domain Adaptation Technique for Semantic Role Labeling with Structural Learning. In *ETRI Journal* vol. 36, no. 3, June. 2014, pp. 429-438.

Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky. 2009, A. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*: Volume 2-Volume 2 (pp. 1003-1011). Association for Computational Linguistics.

Sangha Nam, Younggyun Hahm, Sejin Nam, Key-Sun Choi. 2015. SRDF: Korean Open Information Extraction using Singleton Property. In *Proceedings of International Semantic Web Conference, ISWC, Posters & Demonstrations Track*.

Jungyeul Park, Sejin Nam, Youngsik Kim, Younggyun Hahm, Dosam Hwang, Key-Sun Choi, 2014. Frame Semantic Web: A Case Study for Korean, In *Proceedings of International Semantic Web Conference*, ISWC.

Jacobo Rouces, Gerard de Melo, Katja Hose. 2015. FrameBase: representing n-ary relations using semantic frames. In *European Semantic Web Conference* (pp. 505-521). Springer International Publishing.

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

Piek Vossen, German Rigau, Luciano Serafini, Pim Stouten, Francis Irving, Willem Van Hage. 2014. NewsReader: recording history from daily news streams. In *LREC* (pp. 2000-2007).

Kun Xu, Sheng Zhang, Yansong Feng, and Dongyan Zhao. 2014. Answering Natural Language Questions via Phrasal Semantic Parsing. In *Natural Language Processing and Chinese Computing* (pp. 333-344). Springer Berlin Heidelberg.

Xuchen Yao, Jonathan Berant, Benjamin Van Durme. 2014. Freebase QA: Information Extraction or Semantic Parsing?. In *ACL* 2014, 82. (a)

Xuchen Yao, Benjamin Van Durme. 2014. Information Extraction over Structured Data: Question Answering with Freebase. In *ACL* (1) (pp. 956-966). (b)