The Open Framework for Developing Knowledge Base And Question Answering System

Jiseong Kim, GyuHyeon Choi, Jung-Uk Kim, Eun-Kyung Kim, Key-Sun Choi
Machine Reading Laboratory
Semantic Web Research Center
Department of Computer Science
KAIST, Daejeon, Korea
{jiseong, wiany11, prismriver, kekeeo, kschoi}@kaist.ac.kr

Abstract

Developing a question answering (QA) system is a task of implementing and integrating modules of different technologies and evaluating an integrated whole system, which inevitably goes with a collaboration among experts of different domains. For supporting a easy collaboration, this demonstration presents the open framework that aims to support developing a QA system in collaborative and intuitive ways. The demonstration also shows the QA system developed by our novel framework.

1 Introduction

Recently, a system of a question answering capability, so-called question answering (QA) system, is on the rise by being applied on diverse domains, e.g., quiz show (IBM Watson), personal assistant (Apple Siri, Microsoft Cortana), home device (Amazon Echo), and so on.

Developing a QA system is a work of implementing and integrating modules of different technologies (e.g., natural language processing, disambiguation, graph manipulation), and then evaluating an integrated whole system. Each module is developed by different groups of specialists with such different technologies. All modules must be linked into one integrated system to reach a QA capability, which results in needs of a framework for collaboration.

It is not an easy work to collaborate among experts of different domains to build an integrated working system together. To ease the integration, for developers and researchers, a framework that supports logging I/O, exception handling, flexible system configuration, and so on is of need.

There has been many studies on a QA system over the past years. However, the integration environment for supporting collaborative developments is still lacking.

In this context, open knowledge base and question answering (OKBQA)[1] community has been developing a OKBQA framework for exchanging and harmonizing resources developed by different groups scattered over the world to promote an effective and efficient open collaboration for developing a QA system.

In this demonstration, we introduce the OKBQA framework with a state-of-the-arts-based novel QA pipeline, and the interfaces of the OKBQA framework that supports a development of a QA system in collaborative and intuitive ways.

In Section 2, we introduce the architecture and modules of the OKBQA framework in detail. In Section 4, we demonstrate interfaces of the OKBQA framework. Lastly, in Section 5, we conclude.

2 Architecture of The OKBQA Framework

The architecture of the OKBQA framework comprises a pipeline of OKBQA modules based on the state-of-the-art researches: template generation (Unger et al., 2012), graph search-based named entity disambiguation (Usbeck et al., 2014), SPARQL query generation (Jindong and Cohen, 2014), NLQ50 benchmark[2], and so on. Figure 1 shows the overall pipeline of the OKBQA framework that executes modules in a waterfall manner. The role of each module is detailed in the following section.

[1] http://www.okbqa.org
[2] http://2015.okbqa.org/nlq

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 161–165, Osaka, Japan, December 11-17 2016.

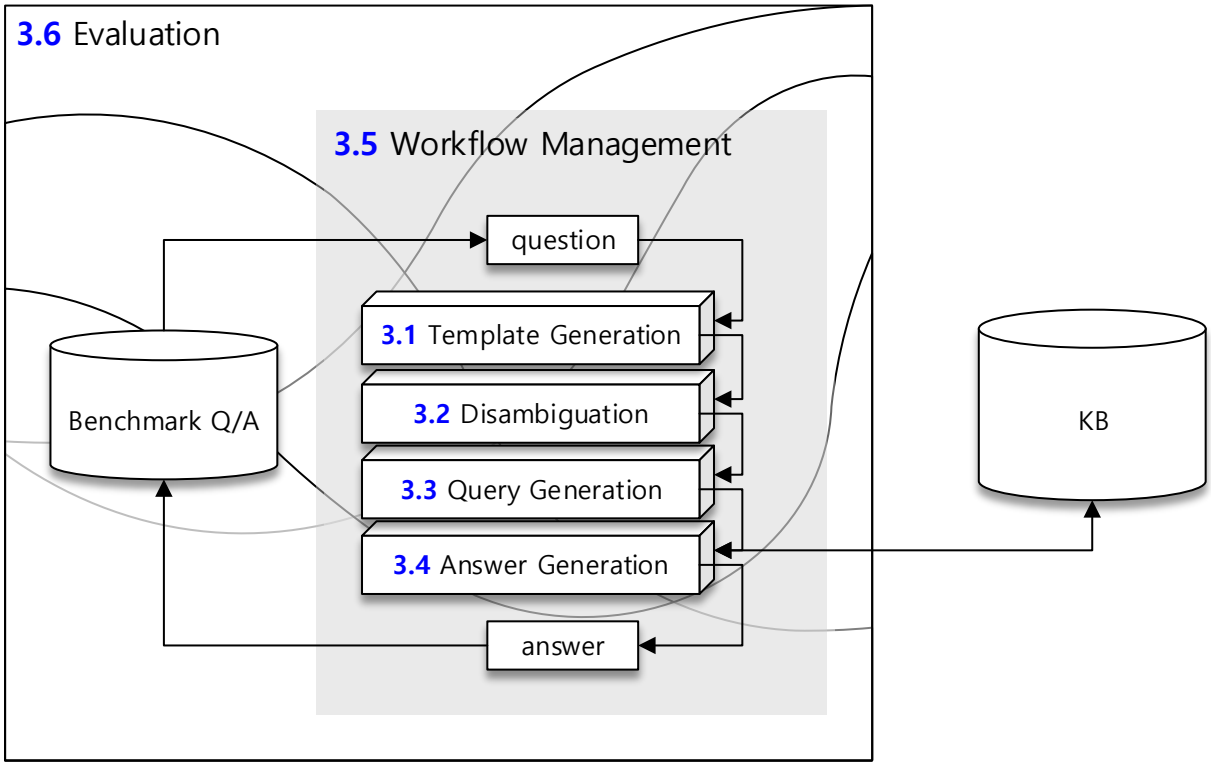

Figure 1: The architecture of the OKBQA framework: Each module is described in detail at each subsection of Section 3

3 OKBQA Modules

3.1 Template Generation

A template generation module (TGM) is for constructing a SPARQL query template from a question expressed in natural language. A *template* comprises *pseudo SPARQL query* and set of *slots*, which of a *pseudo SPARQL query* is a SPARQL query with unbounded variables for resources, classes, and properties specified in a KB, and a *slot* is description of variable. For example, the following shows an example of a template for a natural language question.

- Question: Which rivers flow through Seoul?
- Template:
  - Pseudo query: SELECT ?v4 WHERE { ?v4 ?v2 ?v6 ; ?v7 ?v3 . }
  - Slots (description of variables):
    1. v7 is bound to <http://lodqa.org/vocabulary/sort_of>.
    2. v6 is either a resource or a literal value.
    3. v6 is verbalized into “Seoul”.
    4. v2 is a property.
    5. v2 is verbalized into “flow”.
    6. v3 is a class.
    7. v3 is verbalized into “rivers”.

A template is generated from a question by analyzing a semantic structure of a question from lexical entries and syntactic structure of a question (Unger et al., 2012).

3.2 Disambiguation

A disambiguation module (DM) is for identifying a URI (Uniform Resource Identifier) specified in a KB from verbalization of lexical entries on a question. For example, the following shows an example of disambiguated results.

- Question: Which rivers flow through Seoul?
- Disambiguated results:
  - The lexical entry "Seoul" means the entity <http://dbpedia.org/ontology/Seoul> defined in a KB.
  - The lexical entry "flow" means the entity <http://dbpedia.org/ontology/city> defined in a KB.
  - The lexical entry "rivers" means the entity <http://dbpedia.org/ontology/River> defined in a KB.

3.3 Query Generation

A query generation module (QGM) is for generating and ranking a candidate SPARQL query for a question using results of TGM and DM. For example, the following shows the most top-ranked candidate SPARQL query generated by QGM.

- Question: Which rivers flow through Seoul?
- SPARQL query:
- SELECT ?v4 WHERE {
  ?v4 <http://dbpedia.org/ontology/city> <http://dbpedia.org/resource/Seoul> .
  ?v4 ?v7 <http://dbpedia.org/ontology/River> .
  }

The above example query means retrive all entities can be bound to v4 that is a river and located in city "Seoul".

3.4 Answer Generation

After candidate SPARQL queries for an input question are generated, they are filtered and selected by scores to get the most right answers from a KB. An answer generation module (AGM) is for filtering and selecting final SPARQL queries from candidate SPARQL queries and retrieving answers for an input question from RDF (Resource Description Framework) KB.

3.5 Workflow Management

A control module (CM) supports a function of workflow management to link all of modules in collaborative and intuitive ways. CM constructs an integrated system of the QA capability. To ease collaboration, CM provides the functions of diverse pipeline configuration, inter-module linking, logging I/O flow and exceptional messages, and so on. Examples of the functions are shown in Section 4.

3.6 Evaluation

After a QA system is integrated, the pipeline of a QA system is evaluated to qualify the QA capability. An evaluation module (EM) is for evaluating an integrated whole QA system to measure an accuracy of a QA capability. To measure the accuracy, EM uses the NLQ50 benchmark dataset [3] to qualifying a QA system.

[3] http://2015.okbqa.org/nlq

4 Demonstration

The OKBQA framework supports Web-based interfaces for workflow management and evaluation. Developers can build and evaluate their own QA system with the interfaces in intuitive and collaborative ways. In the followings, we show the demonstration of the interfaces and QA results of a QA system developed by the OKBQA framework.

4.1 Web-based interface for workflow management

The Web-based CM interface[4] supports developers to build a custom QA pipeline with different configurations. The figure 2 shows a configuration page of the CM interface. Currently, the interface supports configuration of module address, execution sequence of modules, and the limit of the execution time of each module.

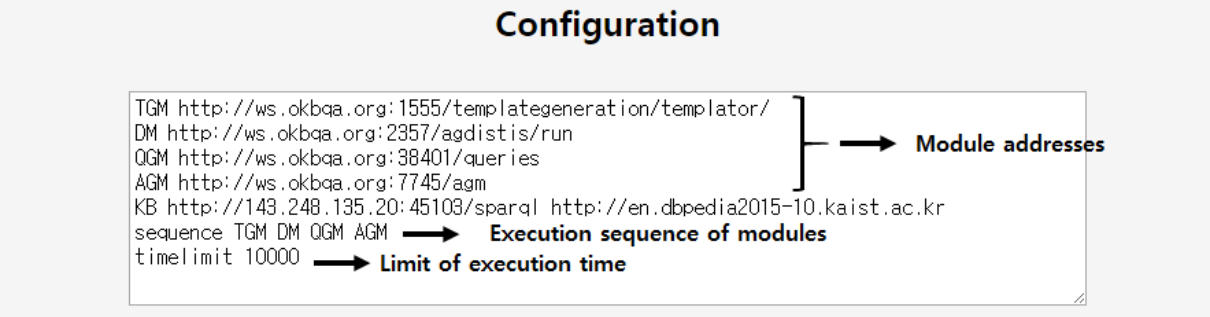

Figure 2: A configuration page of the interface of CM: The first field of each line is an configuration item to be configured and the remaining fields are configuration values for an item.

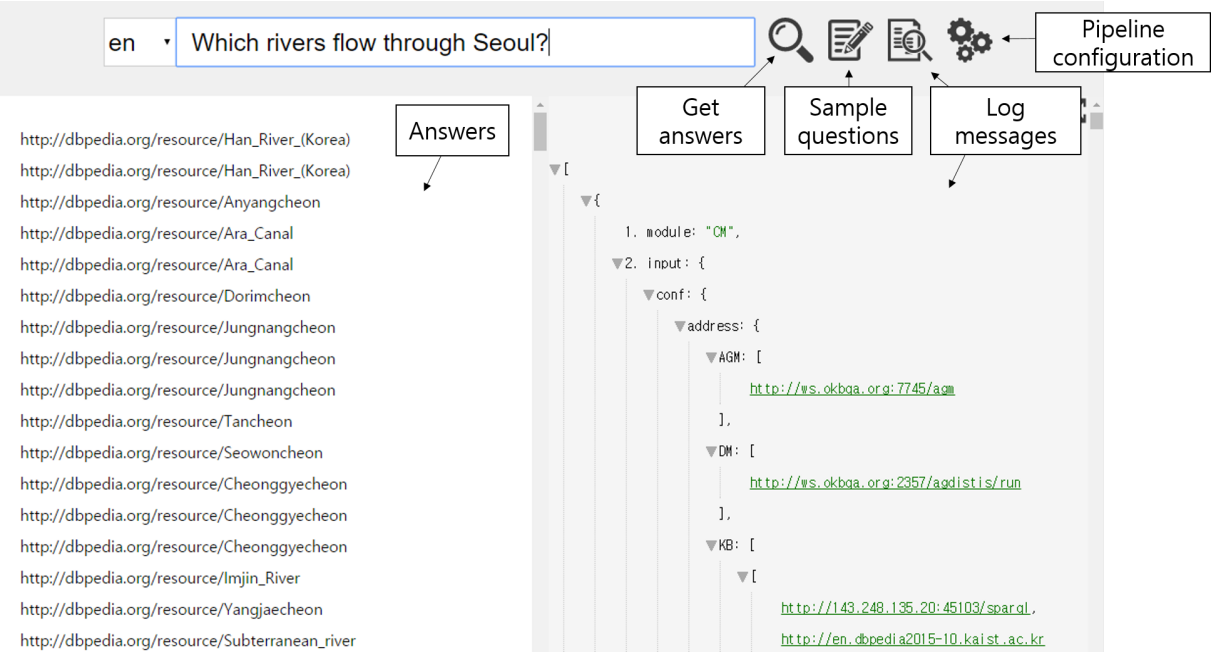

Figure 3: Results of a QA system developed by the OKBQA framework: Answers (left-side) of an input question with informative log messages (right-side) for collborative development

[4] http://ws.okbqa.org/web_interface

4.2 Web-based interface for evaluation

The Web-based EM interface supports measuring an accuracy of arbitrary QA pipelines. The interface uses CM to build a pipeline and evaluate with the NLQ50 benchmark dataset. The interface supports simple and intuitive design that can be checked on the Web page[5].

5 Conclusion

We showed the OKBQA framework with a state-of-the-arts-based novel QA pipeline and intuitive user interfaces for supporting a collaborative development and evaluation of a KB-based QA system. With our open framework, any developers can join the development of their own QA system with open collaboration. We argue that with our framework, a QA system can be built based on the sate-of-the-art researches and already implemented modules with the reduced efforts by decreased trial and error. Our already implemented modules are available on our repository[6], which can be a good starting point for beginners. We are persistently going to enhancing and qualifying supports of the framework with hoping to build a QA system of the qualified QA capability and the beyond.

Acknowledgements

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). And also this work was supported by the Bio & Medical Technology Development Program of the NRF funded by the Korean government, MSIP(2015M3A9A7029735).

References

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

Usbeck, Ricardo and Ngomo, Axel-Cyrille Ngonga and Röder, Michael and Gerber, Daniel and Coelho, Sandro Athaide and Auer, Sören and Both, Andreas. 2014. *AGDISTIS-graph-based disambiguation of named entities using linked data. International Semantic Web Conference* (pp.457-471). Springer.

Kim, J. D. and Cohen, K.. 2014. *Triple pattern variation operations for flexible graph search. Workshop on Natural Language Interfaces for Web of Data.*

[5] http://ws.okbqa.org/web_evaluation

[6] http://repository.okbqa.org