OKBQA Framework towards an open collaboration for development of natural language question-answering systems over knowledge bases

Jin-Dong Kim[1,2], Christina Unger[3], Axel-Cyrille Ngonga Ngomo[4], André Freitas[5], Young-gyun Hahm[2], Jiseong Kim[2], Sangha Nam[2], Gyu-Hyun Choi[2], Jeong-uk Kim[2], Ricardo Usbeck[4], Myoung-Gu Kang[6], and Key-Sun Choi[2]

[1] DBCLS, 178-4-4 Wakashiba, Kashiwa-shi, Chiba, JAPAN
[2] KAIST, 291 Daehak-ro, Guseong-dong, Yuseong-gu, Daejeon, Korea
[3] University of Bielefeld, Universittsstrae 25, 33615 Bielefeld, Germany
[4] University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
[5] University of Passau, Innstrae 41, 94032 Passau, Germany
[6] Young Plus Soft Corp., 71 Karak-ro, Songpa-gu, Seoul, Korea

1 Introduction

Due to recent advances in Semantic Web (SW), the amount of Linked Data (LD) available particularly in Resource Description Framework (RDF) increases rapidly (http://lod-cloud.net). However, LD is still used mostly by SW experts. There are two main obstacles to making LD accessible for common Web users: (1) the need to learn the query language, SPARQL, and (2) the need to know the schemas underlying various datasets to be queried. Approaches to ease the access to LD include graphical query interfaces [7], agent-based systems [1], and natural language (NL) interfaces [2, 5, 8, 4, 6, 3]. Among them, NL interfaces are receiving increasing interest due to high expressive power and low learning cost.

Typically, a natural language question-answering (NLQA) system takes natural language queries as input. The queries are then converted in a structured query language, e.g. SPARQL, which will be used to consult a knowledge base (KB), e.g., a SPARQL endpoint, or KBs. While there are a number of relevant previous works, it is widely understood that the development of a NLQA system requires expertise in various technologies, e.g., natural language processing (NLP), database schema analysis, inference, and so on. There is thus a natural call for collaboration among interested parties.

With the goal to provide a platform of open collaboration for development of NLQA systems, the OKBQA framework has been developed. Recently, it has reached a milestone: (1) core module categories for NLQA systems are figured out and their APIs are documented, (2) a repository of OKBQA-compatible modules is implemented, and 24 modules are registered, and (3) a prototype demo system is implemented and two workflows for QA in English and Korean have been set up. This manuscript presents a summary of OKBQA Framework, and the

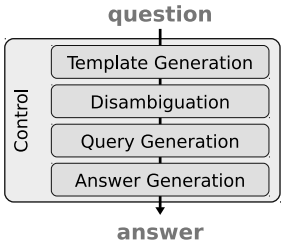

Fig. 1. OKBQA modules in a model flow

demo presentation will show how the system works to support collaboration for development of NLQA systems.

2 OKBQA Framework

Figure 1 shows an overview of the OKBQA framework. It has a modular architecture, which currently defines 4 categories of core modules: *Template Generation Module (TGM)*, *Disambiguation Module (DM)*, *Query Generation Module (QGM)*, and *Answer Generation Module (AGM)*. A *Control Module (CM)* is responsible for making and executing a workflow of QA by connecting several core modules. The input of a workflow is supposed to be a natural language query in character string, and the output to be a list of URIs or literals.

Two design choices were made to ease collaboration among different groups: (1) each module needs to be accessible as a REST service, and (2) the input and output of each module are represented in JSON. Due to the design, a module can be implemented in any programming language, and it can be deployed to any location in the net. A workflow is then defined as a sequence of REST services, which makes it easy to compose a workflow using modules distributed in the net. For more details about the the OKBQA framework, readers are referred to the online documentation at http://doc.okbqa.org.

3 Repository

An OKBQA repository is implemented and maintained to provide a venue for sharing information about modules developed for the OKBQA framework (http://repository.okbqa.org). The registration of modules is open to anybody. At the time of writing, there are 24 modules registered to the repository.

4 Demonstration

A prototype demo system is developed and maintained as a public service (http://ws.okbqa.org/wui-2016/), (1) to demonstrate how workflows in OKBQA

actually work, and (2) to support development of modules for the framework. Currently, two workflows have been set-up for QA in English and Korean. Users can choose a workflow and try it with natural language queries.

Note that the performance of currently available workflows may not yet be competitive, because they are composed by connecting modules developed by different groups without much tuning for harmonization. To improve the performance, further development of the modules is required, and the interface of the OKBQA demo system is designed to support it.

Firstly, the interface allows users to modify the workflows. Note that a modules for the OKBQA framework is required to be a REST service, and a workflow is defined as a sequence of URIs (of the REST services). This means that anyone can develop a module to replace one in a predefined workflow. Suppose one has developed a new DM, either by improving an existing one or by newly implementing it. Once it is deployed as a REST service, the new DM can be tested in a workflow by simply specifying (the URL of) the DM as the DM component of the workflow.

Secondly, the interface allows users to inspect the input and output of each module during execution of a workflow. Figure 2 shows a screen-shot of the interface, which shows the execution of the English workflow with the example query *Which river flows in Seoul?*. The left pane shows the progress of the workflow, and the right pane shows the input and output of each module. With the design, the OKBQA framework may be thought as an SDK system for development of NLQA systems

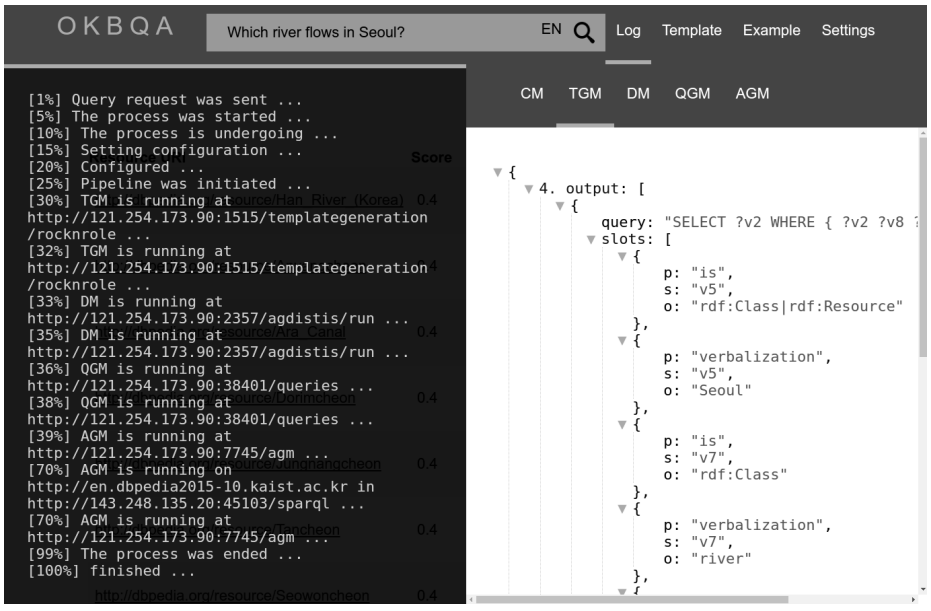

Fig. 2. A screenshot of the OKBQA prototype demo system

5 Conclusion

The OKBQA framework is developed as a platform of open collaboration for the development of NLQA systems. The modules developed for the framework can be found at the OKBQA repository, and they are all freely available as open-source projects. The prototype demo system is maintained as a public service to support distributed, voluntary development of modules for the framework.

There is a large room for improvement in the framework. For example, the composition of a workflow is not yet sufficiently flexible, and the performance of current reference workflows is not yet competitive. Nevertheless, we believe it is a significant milestone that such a framework has begun to work to organize distributed contributions. We hope this presentation to be an opportunity to receive feedback from interested parties and also to invite potential collaborators.

Acknowledgments

The development of OKBQA Framework is supported by the ExoBrain project (http://exobrain.kr/). JDK is supported by the Life Science Database Integration Project funded by National Bioscience Database Center (NBDC) of Japan Science and Technology Agency (JST). ACNN and RU are supported by the H2020 project HOBBIT (GA no. 688227) and the EuroStars projects DIESEL (01QE1512C) and QAMEL (01QE1549C). The authors thank to all the participants in OKBQA hackathon so far for their contribution.

References

1. Cimiano, P., Kopp, S.: Accessing the Web of Data through embodied virtual characters. Semantic Web 1(1, 2), 83–88 (2010)
2. Freitas, A., Oliveira, J.G., Curry, E., Carlos, J., Silva, P.: Treo: Combining entity-search, spreading activation and semantic relatedness for querying linked data. In: In: 1st Workshop on Question Answering over Linked Data (QALD-1) Workshop at 8th Extended Semantic Web Conference (ESWC) (2011)
3. Hamon, T., Grabar, N., Mougin, F.: Querying biomedical linked data with natural language questions. Semantic Web 8(4), 581–599 (2017)
4. Kim, J.D., Cohen, K.B.: Natural language query processing for SPARQL generation: A prototype system for SNOMED-CT. In: Proceedings of BioLink SIG meeting 2013 (2013)
5. Lopez, V., Fernández, M., Motta, E., Stieler, N.: Poweraqua: Supporting Users in Querying and Exploring the Semantic Web. Semantic web 3(3), 249–265 (Aug 2012)
6. Rozinajová, V., Macko, P.: Using Natural Language to Search Linked Data. In: Semantic Keyword-based Search on Structured Data Sources. pp. 179–189. Springer (2016)
7. Russell, A., Smart, P.: NITELIGHT: A Graphical Editor for SPARQL Queries. In: 7th International Semantic Web Conference (ISWC 2008) (2008)
8. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.C., Gerber, D., Cimiano, P.: Template-based Question Answering over RDF Data. In: Proceedings of the 21st International Conference on World Wide Web. pp. 639–648. WWW '12, ACM, New York, NY, USA (2012)