

# OKBQA Framework for collaboration on developing natural language question answering systems

Prototype System Demonstration

Jin-Dong Kim  
DBCLS and KAIST  
jdkim@dbcls.rois.ac.jp

Christina Unger  
University of Bielefeld  
cunger@cit-ec.uni-bielefeld.de

Axel-Cyrille Ngonga Ngomo  
University of Paderborn  
axel.ngonga@upb.de

André Freitas  
University of Passau  
andrenfreitas@gmail.com

Young-gyun Hahm  
KAIST  
hahmyg@kaist.ac.kr

Jiseong Kim  
KAIST  
jiseong@kaist.ac.kr

Sangha Nam  
KAIST  
nam.sangha@kaist.ac.kr

Gyu-Hyun Choi  
KAIST  
wiany11@kaist.ac.kr

Jeong-uk Kim  
KAIST  
kju0627@gmail.com

Ricardo Usbeck  
University of Paderborn  
ricardousbeck@gmail.com

Myoung-Gu Kang  
Young Plus Soft Corp.  
mgkaki@youngplussoft.com

Key-Sun Choi  
KAIST  
kschoi@kaist.ac.kr

## ABSTRACT

The OKBQA Framework is established to facilitate open collaboration for development of natural language question-answering systems. It defines necessary modules with their API, so that OKBQA-conformant modules can inter-operate with each other. The OKBQA repository (<http://repository.okbqa.org>) is where those modules are registered, and the OKBQA demo system (<http://ws.okbqa.org/wui-2016/>) allows composition and execution of workflows using the modules.

## KEYWORDS

question-answering, natural language processing, workflow

## 1 INTRODUCTION

The OKBQA framework (<http://www.okbqa.org>) has been developed as a platform for open collaboration on development of natural language (NL) question-answering (QA) systems. With the goal to facilitate collaboration through distributed voluntary contributions, activities around the framework include (1) identifying and defining modules necessary for NLQA, and their APIs, (2) implementing them, and (3) developing and maintaining a public service whereon workflows of QA can be composed and executed. Recently, the development has reached a milestone: a demo system has begun to work. On the system, workflows for NLQA can be easily composed and executed, using modules which are deployed as REST services. To demonstrate the functionality of the framework, two workflows for QA in English and Korean have been set up on the system.

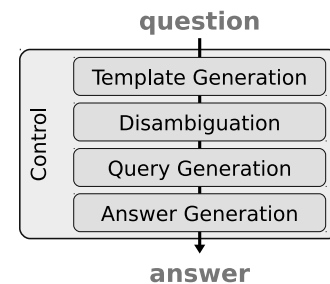


Figure 1: OKBQA modules in a model flow

## 2 OKBQA FRAMEWORK

Figure 1 gives an overview of the OKBQA framework. In its modular architecture, it currently defines four core categories of modules: *Template Generation Module (TGM)*, *Disambiguation Module (DM)*, *Query Generation Module (QGM)*, and *Answer Generation Module (AGM)*. A *Controller Module (CM)* is supposed to make a workflow of QA by connecting several core modules. The input of a core 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.

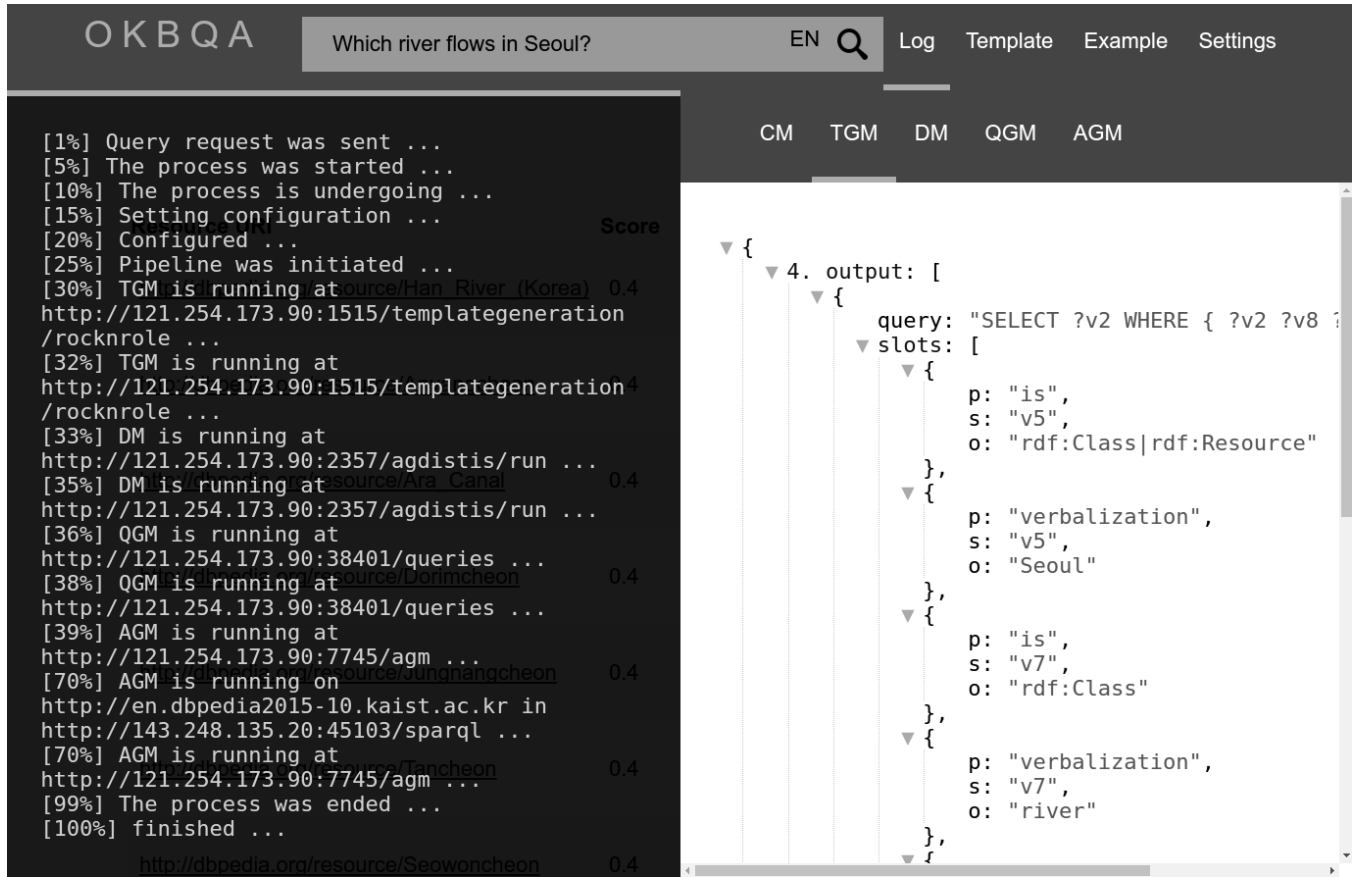


Figure 2: A screenshot of the OKBQA prototype demo system

### 3 REPOSITORY

The OKBQA repository is maintained to provide a venue for storing information about modules developed for the OKBQA framework (<http://repository.okbqa.org>). At the time of writing, there are 24 modules registered to the repository.

### 4 DEMO SYSTEM

A prototype demo system is developed and maintained as a public service (<http://ws.okbqa.org/wui-2016/>), to demonstrate how workflows in OKBQA actually work, and also to support development of new modules. Currently, two workflows have been set-up for QA in English and Korean. Users can choose a workflow and try it with NL queries. An important point here is that the system will show not only the final results but also the output of each module of the workflow works. For those who are new to the framework, such an interface may give a chance to figure out how an OKBQA workflow works. More importantly, The system allows users to freely modify a workflow by replacing a module with a new one. A newly developed module, once it is deployed as a REST service, can

be plugged-in to a workflow. By inspecting its IO in the workflow, the developer may also be able to figure out how it works in the workflow. In this way, the prototype demo system is designed to support the development of modules for OKBQA.

### 5 CONCLUSION

For those who are interested in developing NLQA systems, we believe the resources of OKBQA to provide a good starting point. As the system just began to work, there is of course 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.