

# KBQA: An Online Template Based Question Answering System over Freebase

Wanyun Cui, Yanghua Xiao\*, Wei Wang

Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University  
wanyuncui1@gmail.com, shawyh@fudan.edu.cn, weiwang1@fudan.edu.cn

## Abstract

Question answering (QA) has become a popular way for humans to access billion-scale knowledge bases. QA systems over knowledge bases produce accurate and concise answers. The key of QA over knowledge bases is to map the question to a certain substructure in the knowledge base. To do this, KBQA (Question Answering over Knowledge Bases) uses a new kind of question representation: **templates**, learned from a million scale QA corpora. For example, for questions about a city's population, KBQA learns templates such as What's the population of \$city?, How many people are there in \$city?. It learns overall 1171303 templates for 4690 relations. Based on these templates, KBQA effectively and efficiently supports binary factoid questions or complex questions.

## 1 Introduction

Question Answering (QA) has drawn a lot of research interests. A QA system is designed to answer a particular type of questions [2]. One of the most important types of questions is the factoid question (FQ), which asks about objective facts of an entity. A particular type of FQ, known as the binary factoid question (BFQ) [1], asks about a property of an entity. For example, @how many people are there in Honolulu? In this paper, we focus on answering these BFQs over Freebase.

KBQA understands a question by **templates**. As an example, How many people are there in \$city? is the template for @. No matter \$city refers to Honolulu or other cities, the template always asks about population of the question. KBQA learns the templates' corresponding predicate through Yahoo! Answers, a large scale QA corpora consisting of millions of qa pairs.

\*Correspondence author. This work was supported by Xiaoi Research, by Shanghai Internet Big Data Engineering and Technology Center, by the National NSFC(No.61472085, 61171132, 61033010), by National Key Basic Research Program of China under No.2015CB358800, by Shanghai Municipal Science and Technology Commission foundation key project under No.15JC1400900.

We show the number of predicates and templates KBQA learned in Table 1, with comparison to *bootstrapping* [4], which uses BOA patterns to represent questions. From the result, KBQA finds significantly more templates and predicates despite that the corpus size of bootstrapping is larger. This implies that KBQA is more effective: (1) the large number of templates ensures that KBQA understands diverse questions; (2) the large number of predicates ensures that KBQA understands diverse question intents.

	KBQA	Bootstrapping
Corpus	41M QA pairs	256M sentences
Templates	<b>1,171,303</b>	471,920
Predicates (BOA patterns)	<b>4690</b>	283

Table 1: Coverage of Templates

## 2 Anatomy

**Architecture** Figure 1 shows the architecture of KBQA. The **offline procedure** learns the mapping from templates to predicates in the *Template Extraction* module by using maximum likelihood (ML) estimator. Such estimator is trained by the result of *Entity & Value Identification*. Furthermore, KBQA understands complex predicate forms of the knowledge base in the *Predicate Expansion* module. When a question comes into the **online procedure**, KBQA first parses it into one or several BFQs. For each BFQ, KBQA extracts its template and lookups the predicates from the template repository. Finally KBQA returns the value of the entity and predicate retrieved in the knowledge base as the answer.

**Question Parsing** The core of the online procedure is to convert a question into **templates**. To do this, KBQA identifies the question's entity by Stanford NER [3], and replaces the entity by its concepts using conceptualization [5]. The conceptualization mechanism is based on a large semantic network (Probase [6]) that consists of millions of entities and concepts, so that KBQA has enough granularity to represent diverse questions.

We elaborate the **Template Extraction** component. KBQA learns templates and their mappings to knowledge base predicates by Yahoo! Answers. First, for each qa pair in Yahoo! Answers, KBQA extracts the entity in the question and the corresponding value in the answer. Then, KBQA

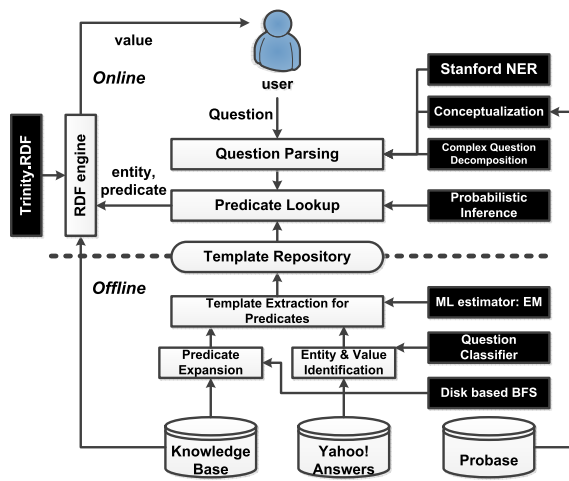


Figure 1: System Architecture

lookups the predicate between the entity and value in the knowledge base. The basic idea is, KBQA uses the qa pairs as the training data. For each template, if most of its question instances in the QA corpora share the same predicate, KBQA maps the template to this predicate.

### 3 Demonstration

As shown in Figure 2, KBQA provides a simple interface for users on the web page, which consists of three parts: (1) the QA part displays the main result of the question; (2) the feedback part provides a way for users to vote for the answer; (3) the explanation part displays how the answer is extracted and why KBQA can extract the answer.

**Diverse Question Types** KBQA actually satisfies users' different requirements and supports different question types.

- **Simple BFQ** Users can ask simple BFQs, i.e., questions about a property of an entity. Figure 2 shows the question asking about Shakespeare's birthday.
- **Questions relying on expanded predicates** The "predicate expansion" module enables KBQA to understand expanded predicates. For example, Freebase represents "spouse" relation by the expanded predicate "spouse\_s→spouse→name".
- **Complex question** The "question parsing" module decomposes a complex question into several BFQs. This enables KBQA to understand and answer complex questions. Figure 3 shows a complex question.

**User Feedback** In the feedback part, KBQA enables users to vote for the answer. The feedbacks of KBQA are routed back as inputs and help improve the system.

### References

[1] E. Agichtein, S. Cucerzan, and E. Brill. Analysis of factoid questions for effective relation extraction. In *SIGIR*, pages 567–568, 2005.

[2] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C.-Y. Lin, S. Maiorano, G. Miller,

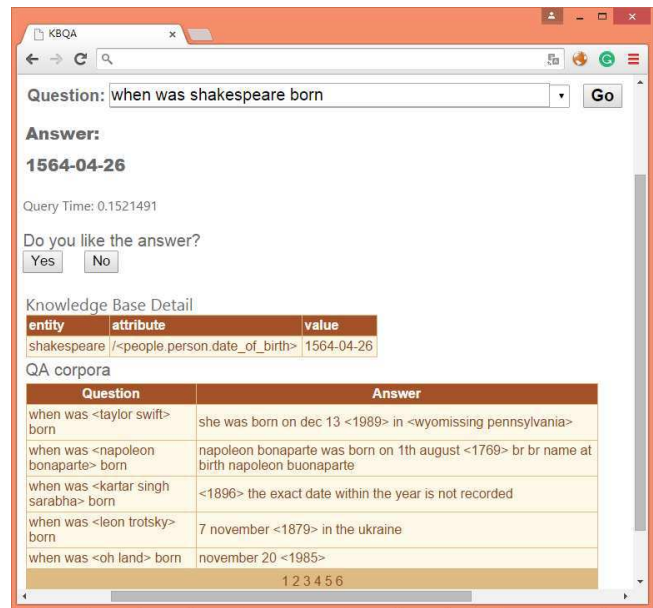


Figure 2: KBQA for simple BFQ

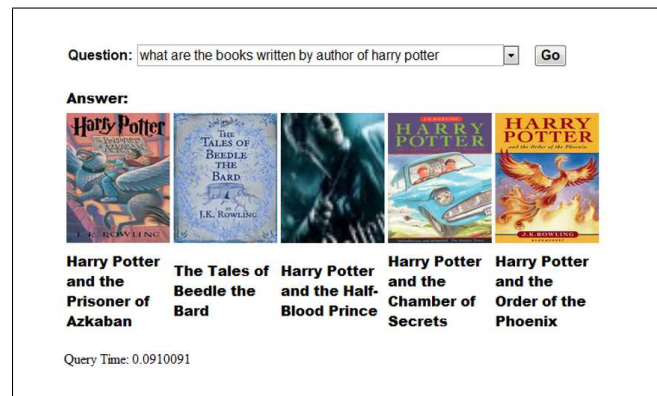


Figure 3: KBQA for complex questions

et al. Issues, tasks and program structures to roadmap research in question & answering (q&a). In *DUC Roadmapping Documents*, 2001.

[3] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, pages 363–370, 2005.

[4] D. Gerber and A. Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction ISWC*, 2011.

[5] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJ-CAI*, pages 2330–2336, 2011.

[6] W. Wu, H. Li, H. Wang, and K. Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492, 2012.