A Configurable Natural Language Question Benchmark for Knowledge Bases

Yu Su[1], Huan Sun[1], Brian Sadler[2], Mudhakar Srivatsa[3], Izzeddin Gur[1], Zenghui Yan[1] and Xifeng Yan[1]
{ysu, huansun, izzeddingur, zyan, xyan}@cs.ucsb.edu, brian.m.sadler6.civ@mail.mil, msrivats@us.ibm.com
[1]University of California, Santa Barbara, [2]U.S. Army Research Lab, [3]IBM Research

***Abstract*—Supporting natural language interface is crucial for accessing knowledge bases (KBs) due to the complexity of their schema. Many natural language interfaces to KBs (NLIKBs) have been built to simplify query formulation and facilitate advanced applications like personal intelligent assistant. Unfortunately, there is a lack of a comprehensive natural language question benchmark that is able to fairly evaluate different NLIKBs in terms of accuracy and runtime. In this paper, we propose a framework to construct comprehensive question benchmarks, where an array of question characteristics, including query size, commonness, modifier, answer cardinality, and paraphrasing, are made configurable. The benchmark constructed in this way delivers unprecedentedly fine-grained performance profiling, broken down by each question characteristic, which is important for performance comparison and future improvement.**

I. Introduction

Recent years have witnessed the fast growth of large knowledge bases (KBs) such as DBpedia, Freebase and YAGO2, which contain enormous knowledge about real-world entities and their relations. Plenty of applications have benefited from KBs, for example, web search engines like Google Search and Microsoft Bing, personal assistants like Apple Siri, Google Now, and natural language understanding. Querying KBs is a challenging task because of the extremely high complexity of their schemas (e.g., thousands of entity and relation types). Therefore, natural language interfaces to knowledge bases (NLIKBs), or KB-based question answering, have drawn tremendous attention.

Benchmarks are indispensable for the rapid development of a research area. While the need is evident, from the benchmarks available for NLIKBs [4], [1], [5], [3], we found that their questions are in general simple. For example, around 85% of the questions in WEBQUESTIONS [1] and all the questions in SIMPLEQUESTIONS [3] can be answered directly by a single relation, e.g., *"what country was Barack Obama from?"* can be answered by (`BarackObama`, `nationality`, `UnitedStates`). A good question benchmark shall cover various *question characteristics*, i.e., dimensions along which the difficulty of answering questions varies. Some questions are difficult due to their complex semantic structure, e.g., *"who was the coach when Michael Jordan stopped playing for the Chicago Bulls?"* while some others may be difficult because they require a precise quantitative analysis (e.g., aggregation) over the answer space, e.g., *"how many launch sites does NASA have?"* Many other characteristics shall be considered too, e.g., what topic a question is about (questions about popular topics may be easier to answer) and how many answers a question has (it is harder to achieve high recall in case of multiple answers). Worse still, due to the flexibility of natural language, different users often describe the same question in quite different ways, i.e., paraphrasing; many systems cannot handle paraphrased questions elegantly.

Our major contribution is a framework, GQUEBEC (Graph-based Question Benchmark Construction), to construct natural language question benchmarks with configurable characteristics, where the following characteristics, among others, are formalized: (1) *Query size*, the number of relations involved in a question, (2) *Commonness*, how common a question is, e.g., "*where was Obama born?*" is more common than "*what is the tilt of axis of Polestar?*", (3) *Modifier*, additional functions such as aggregation, superlatives, and comparatives, (4) *Answer Cardinality*, the number of answers, (5) *Paraphrasing*, different natural language expressions of the same question. Next we will introduce our benchmark construction framework (Section II) and briefly show the benchmark results of several state-of-the-art NLIKB systems (Section III).

II. Framework Design

Our benchmark construction framework is illustrated with a running example in Figure 1. Due to the complexity of parsing natural language questions, it is hard to precisely describe question characteristics in the natural language space. Therefore, instead of directly working with natural language, we employ an intermediate graph query representation, where question characteristics can be easily formalized. Our framework consists of three phases: graph query generation, query refinement, and natural language question formulation.

Graph Query Generation. We introduce a top-down approach for graph query generation, proceeding from the ontology to the model. Query templates (e.g., Figure 1(b)), consisting of solely classes and relations, are first generated by walking in the ontology with a random starting class. Some nodes in a query template (e.g., `Datetime` and `CauseOfDeath`) will be randomly selected to be grounded with individuals, resulting in a raw graph query (Figure 1(c)). Techniques are then developed to refine the generated raw graph queries. Question characteristics such as query size, answer cardinality and modifiers can be configured in this stage.

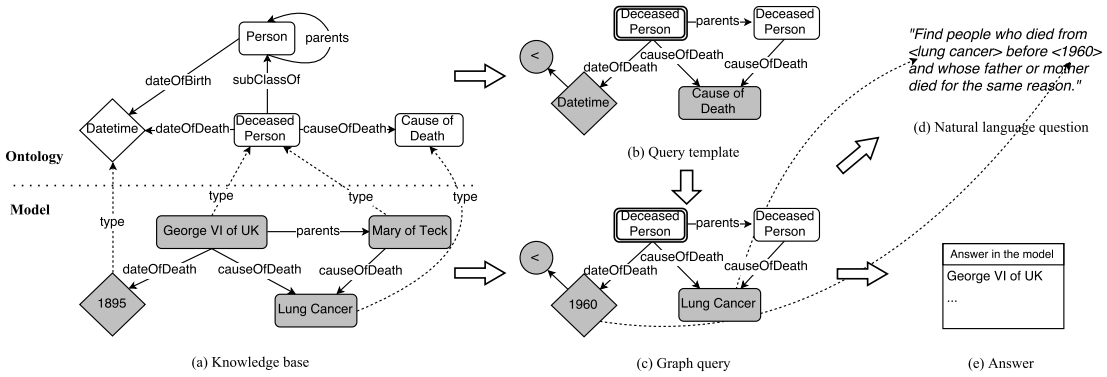

Fig. 1. Framework and running example.

Query Minimization. A randomly generated raw graph query likely contains redundant components which do not contribute to the answer. We propose a greedy algorithm to remove redundant components from graph queries, which only needs a single scan over the components of a graph query, and get *minimal* queries as a result. We strictly prove the resulted graph query is minimal and has the same answer with the original graph query.

Commonness Checking. A randomly generated query is not necessarily realistic. Certainly, we do not want a benchmark full of very rare questions that users never ask. We investigate what kind of graph queries more likely correspond to realistic questions, and find that *commonness*, i.e., how often people talk about things, is a good indicator. We harvest statistical information about query commonness from the Web. We estimate the frequency of entities, classes, and relations using a large-scale entity linking dataset, which contains over 10 billion entity mentions in 1 billion web documents, and derive query commonness from its components.

Natural Language Question Formulation. Crowdsourcing is employed to translate graph queries into natural language questions. Two levels of paraphrasing are provided: At the sentence level, multiple paraphrases of a question are provided via crowdsourcing; at the entity level, different lexical forms of the topic entities are mined from the Web. As a result, tens of paraphrased questions can be produced for a single graph query. The answer of questions is automatically obtained by executing the corresponding graph query in the KB.

EXAMPLE. *Consider the example in Figure 1. A query template as in Figure 1(b) is first extracted from the ontology. Two nodes,* `CauseOfDeath` *and* `Datetime` *are selected and grounded with compatible entity* `LungCancer` *and literal* `1960` *from the model to generate the graph query in Figure 1(c). A crowdsourcing annotator (among others) translates the graph query into the natural language question in Figure 1(d). Note the angle brackets that enclose topic entities. Substituting an enclosed entity with its other lexical forms (e.g. "lung tumor") gives us more question paraphrases. Figure 1(e) shows the corresponding answers identified in the knowledge base.*

III. Experiments and Conclusion

Based on the proposed framework, we construct a question benchmark using Freebase, a widely-adopted knowledge base. Our benchmark contains 505 high-quality graph queries 3,213 natural language questions. We extensively evaluate several state-of-the-art NLIKB systems whose code is publicly available, including SEMPRE [1], PARASEMPRE [2], and JACANA [7]. Equipped with our unprecedentedly comprehensive benchmark, we are able to draw many interesting observations that have not been revealed before. For example, we find that the widely held conclusion [7], [6], that information extraction methodology (e.g., JACANA) can achieve comparable performance with semantic parsing methodology (e.g., SEMPRE and PARASEMPRE) in question answering, does not hold any more. Information extraction methodology only works for simple questions, and fails to handle more complex ones. Breaking down system performance by question characteristics further enables deep system inspection. Take query commonness as an example. Intuitively one would expect that common questions should be easier to answer. However, this intuition is not always right. Our results show that the most common questions can be harder to answer. A probable reason is as follows: The common entities are often featured by a high degree in KBs. For example, `UnitedStatesOfAmerica`, the most common entity in Freebase, is connected with over 1 million of other entities. Such high degree dramatically increases the size of the candidate answer space, which makes it harder to identify the right answer.

References

[1] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*, 2013.
[2] J. Berant and P. Liang. Semantic parsing via paraphrasing. In *ACL*, 2014.
[3] A. Bordes, N. Usunier, S. Chopra, and J. Weston. Large-scale simple question answering with memory networks. *arXiv:1506.02075*, 2015.
[4] Q. Cai and A. Yates. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL*, 2013.
[5] V. Lopez, C. Unger, P. Cimiano, and E. Motta. Evaluating question answering over linked data. *Journal of Web Semantics*, 21:3–13, 2013.
[6] X. Yao, J. Berant, and B. Van Durme. Freebase QA: Information extraction or semantic parsing? In *ACL*, 2014.
[7] X. Yao and B. Van Durme. Information extraction over structured data: Question answering with Freebase. In *ACL*, 2014.