A DBpedia-based Benchmark for Ontology-mediated Query Answering

Suxue Ma[1], Zhe Wang[2], and Kewen Wang[2*]

[1] College of Intelligence and Computing, Tianjin University, China
[2] School of Information and Communication Technology, Griffith University, Australia

Abstract. Ontology-mediated query answering (OMQA) is a framework for querying data with a background ontology. Detailed evaluation of OMQA systems remains a challenge due to limitations in existing benchmarks. In this paper, we propose a new benchmark for OMQA based on natural language questions over DBpedia. In particular, the data are sampled from DBpedia with adjustable volumes and can easily reach a scale that is difficult for existing OMQA systems to handle. Logical rules are automatically extracted from DBpedia using a rule learner, and the queries come from real-life natural language questions over DBpedia. We evaluated two state-of-the-art systems under various settings, to demonstrate the potential of our benchmark in benchmarking and analyzing the behavior of OMQA systems.

1 Introduction

Ontology-mediated query answering (OMQA) is a framework for querying data with a background ontology, a collection of logical rules. A prominent approach for OMQA is query rewriting [7], which transforms a query with relevant rules into another query that can be processed by conventional database management systems. Several OMQA systems have been developed aiming at scalable query answering over complex ontologies and large datasets. Yet comprehensive evaluations of these systems remains a challenge due to limitation in benchmarks.

To analyse the behavior of various OMQA systems, it is desirable for a benchmark to possess the following properties: (P1) the volume of data is adjustable and large enough to test the limits of existing systems; (P2) the complexity of the ontology can be fine-tuned, in terms of the number and the lengths of rules, as well as the rewriting depths [2]; and (P3) the benchmark comes from real-life applications [3]. The existing benchmarks in the OMQA literature include LUBM [3] and its variants, ChaseBench [2], and various real-life ontologies.

*Corresponding Author

LUBM comes with a data generator that satisfies (P1), and thus has been widely used for OMQA benchmarking, but its ontology is relatively small and does not satisfy (P2). ChaseBench is recently developed to achieve both (P1) and (P2), yet its rules are synthesised and do not satisfy (P3). Benchmarks with real-life ontologies satisfy (P3), but not (P1) or (P2). Also, the existing benchmarks may not be significantly challenging to systems like Graal [1] and Drewer [7, 8].

In this paper, we propose a new benchmark for OMQA based on natural language questions over DBpedia [4], with properties (P1) – (P3). In particular, the data are sampled from DBpedia with adjustable volumes and can easily reach a scale that is difficult for existing OMQA systems to handle. Logical rules are automatically extracted from DBpedia using a rule learner, which allows the configuration of the lengths of learned rules and their head predicates. By iteratively learning rules with specified head predicates, the rewriting depths can also be configured. Furthermore, the queries come from real-life natural language questions and are converted into conjunctive queries with predicates in DBpedia. We evaluated Graal and Drewer on their time and memory efficiency under various settings, to demonstrate the potential of our benchmark in benchmarking and analysing the behavior of OMQA systems. Our benchmark is available at https://github.com/bohemianc/benchmarking.

2 Our Benchmark

DBpedia [4] contains structured and multilingual knowledge extracted from Wikipedia, and is the backbone of many Semantic Web applications. Our benchmark include queries, rules and data obtained from natural language questions about DBpedia, and the ontology and data from DBpedia, as shown in Figure 1.

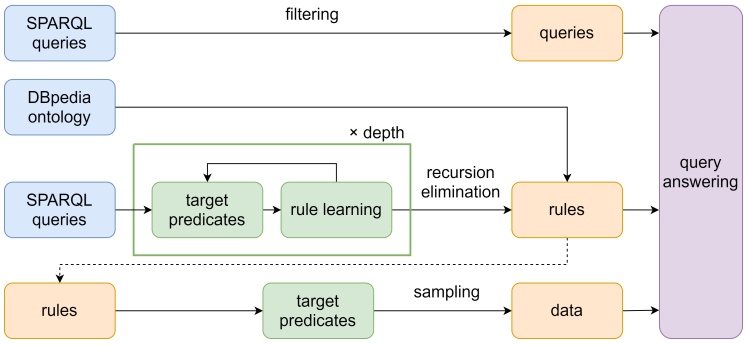

Fig. 1. Architecture of the benchmark construction

2.1 Queries

The queries come from question collection LC-QuAD [6] about DBpedia, which consists of 5K questions in natural language together with their corresponding

SPARQL queries. For example, the question "*Name the scientist whose supervisor was Ernest Rutherford and had a doctoral students named Charles Drummond Ellis?*" has the corrsponding SPARQL query

```
SELECT DISTINCT ?uri
WHERE {?uri dbo:doctoralAdvisor dbr:Ernest_Rutherford .
       ?uri dbp:doctoralStudents dbr:Charles_Drummond_Ellis .
       ?uri rdf:type dbo:Scientist }.
```

We converted the SPARQL queries into conjunctive queries in the DLGP format proposed by Graal [1]. Among SELECT, ASK, and COUNT queries, only SELECT queries can be directly converted. Also, some special characters are not supported by Graal. After filtering, 1961 SPARQL queries were converted into conjunctive queries, among which 1264 have more than one atoms.

2.2 Ontologies

An ontology in our benchmark is constructed in two ways. As DBpedia provides an ontology with a large number of rules which we can use for our benchmark. Yet such rules are relatively simple and have bounded lengths. On the other hand, to answer a given query q, the time and memory efficiency of OMQA systems are largely impacted by three factors related to the complexity of the ontology: the number of applicable rules, their lengths, and the rewriting depths. A rule is *applicable* if it is used in the rewriting of q, and the *length* of a rule refers to the the number of atoms in it. The *depth* of rewriting q is the largest number $n \geq 0$ with a sequence of queries $q_0, \ldots, q_n$ such that $q_0 = q$ and q_{i+1} is a result of rewriting q_i for $0 \leq i \leq n$. For example, consider the following rules:

$$\mathsf{nationality}(X, Y) \leftarrow \mathsf{birthPlace}(X, Z) \wedge \mathsf{country}(Z, Y), \tag{1}$$

$$\mathsf{birthPlace}(X, Y) \leftarrow \mathsf{parent}(X, Z) \wedge \mathsf{liveIn}(Z, Y). \tag{2}$$

Both rules have a length of 3. A query $\mathsf{nationality}(\mathsf{Bill}, X)$ can be rewritten by rule (1) and then by rule (2) into $\exists Y, Z.\mathsf{parent}(\mathsf{Bill}, Y) \wedge \mathsf{liveIn}(Y, Z) \wedge \mathsf{country}(Z, X)$. The rewriting depth is 2. In general, for an ontology of l rules with the maximum rule length m and rewriting depth n, the rewriting is bounded by $l^{n+1} \cdot \big((m-1) \cdot n + 1\big)$.

Thus, we also use the embedding-based rule learner R-Linker [9] to extract rules with configurable rule lengths and rewriting depths based on DBpedia data, as a supplement to the DBpedia ontology. Another reason of choosing R-Linker is that it allows us to specify *target predicates* and extracts rules with the specified predicates in their heads, such as nationality in rule (1) and birthPlace in rule (2). Other rule learners, such as RLvLR [5], can also be used. Given a query q, to extract rules with rewriting depth n, our method runs in n iterations. In the 1st iteration, the target predicates are the predicates in q. Then, in the i-th iteration for $2 \leq i \leq n$, the target predicates are those occurring in the bodies

of the rules extracted in the previous iteration (i.e., iteration $i-1$) that have not been target predicates before.

For example, let q be

SELECT DISTINCT ?uri
WHERE {?uri dbo : doctoralAdvisor dbr : Ernest_Rutherford },

the rule length be 2, and the rewriting depth be 2. Then, after 2 iterations the rule learner can get rules as

$$\mathsf{dbp{:}doctoralAdvisor}(X,Y) \leftarrow \mathsf{dbp{:}doctoralStudents}(Y,X),$$
$$\mathsf{dbp{:}doctoralStudents}(X,Y) \leftarrow \mathsf{dbo{:}influencedBy}(X,Y).$$

The extracted rules may be *recursive*, that is when a predicate occurs both in the head and the body of the rule, which cannot be handled by some rewriting-based OMQA systems. Such rules are eliminated from the ontology.

2.3 Datasets

DBpedia contains a huge amount of data, which cannot be handled by existing OMQA systems. Hence, our method samples subsets of the data with various sizes for evaluation. At the same time, a significant portion of the sampled dataset should be relevant to the queries and the applicable rules in the ontology. Thus, datasets are sampled according to the predicates occurring in the queries and the applicable rules. For instance, if the above rule (1) is applicable, then our method adds to the sample dataset those retrieved with the SPARQL query `SELECT ?x P ?z WHERE {?x P ?z}`, where `P` is nationality, birthPlace, and country, respectively.

3 Evaluation

We evaluated two state-of-the-art OMQA systems, Graal [1] and Drewer [7, 8], on their time and memory efficiency under various settings using our benchmark. For each setting, we used 5 queries with 2 - 3 atoms. For the complexity of the ontology, it is easy to control the rule lengths (Len., 2 or 3) and rewriting depths (Dep., 1 or 5 or 10), but it is relatively difficult to fix the exact numbers of applicable rules (#R) which is dynamically determined. We kept the applicable rule in a range of 75 - 200 by adjusting the total numbers of rules used for rewriting. The sizes of sampled datasets (#F) range from 5M to 20M. A '-' means the system exceeded the 10 minutes time limit.

From Table 1, in general, the processing times of both systems are impacted by the rule lengths, the rewriting depths, the numbers of applicable rules, and the data sizes, while the impacts may also depend on other factors such as the exact rules applied. On the other hand, their impact on the memory consumption is less obvious. Finally, Graal failed to complete when the data size was increased

Table 1. Evaluation results under various settings

Len.	Dep.	#R	#F	Time (sec)		Memory (MB)	
				Graal	Drewer	Graal	Drewer
2	1	88	5M	37.4	15.9	4006	102
2	5	102	5M	37.3	15.5	4061	110
2	10	119	5M	41.2	18.0	3404	110
3	1	78	5M	39.9	28.1	2968	102
3	5	152	5M	39.4	15.5	3377	110
3	10	98	5M	38.0	16.0	3648	106
3	10	98	10M	80.3	28.7	2127	110
3	10	98	20M	-	52.2	-	110
3	10	181	10M	81.1	-	2109	-

to 20M, whereas Drewer had difficulty when the applicable rules are increased to 181. It may suggest Graal is more sensitive to data sizes while Drewer is more impacted by the numbers of rules.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China under grant 61976153.

References

1. Baget, J.F., Leclère, M., Mugnier, M.L., Rocher, S., Sipieter, C.: Graal: A toolkit for query answering with existential rules. In: Proc. of RuleML. pp. 328–344 (2015)
2. Benedikt, M., Konstantinidis, G., Mecca, G., Motik, B., Papotti, P., Santoro, D., Tsamoura, E.: Benchmarking the chase. In: Proc. of SIGMOD. pp. 37–52 (2017)
3. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for owl knowledge base systems. J. Web Semant. **3**(2-3), 158–182 (2005)
4. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web **6**(2), 167–195 (2015)
5. Omran, P.G., Wang, K., Wang, Z.: An embedding-based approach to rule learning in knowledge graphs. IEEE Transactions on Knowledge and Data Engineering (2019)
6. Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J.: LC-QuAD: A corpus for complex question answering over knowledge graphs. In: Proc. of ISWC. pp. 210–218 (2017)
7. Wang, Z., Xiao, P., Wang, K., Zhuang, Z., Wan, H.: Query answering for existential rules via efficient datalog rewriting. In: Proc. of IJCAI. pp. 1933–1939 (2020)
8. Wang, Z., Xiao, P., Wang, K., Zhuang, Z., Wan, H.: Efficient datalog rewriting for query answering in TGD ontologies. IEEE Transactions on Knowledge and Data Engineering (2021)
9. Wu, H., Wang, Z., Zhang, X., Omran, P.G., Feng, Z., Wang, K.: A system for reasoning-based link prediction in large knowledge graphs. In: Proc. of ISWC Satellites. pp. 121–124 (2019)