

AQUA: Approximation Enabled Query Rewriting for Linked Open Data (LOD)

Nophadol Jekjantuk and Jeff Z. Pan

Department of Computing Science
University of Aberdeen, Aberdeen AB24 3UE, UK

Abstract. Inference query answering over large dataset such as Linked Open Data(LOD) is very difficult since there is no tableaux reasoner can handle a large amount of data. Many ontology repositories are able to load a large dataset. However, most of them can only partial support inference query answering and the inference model need to compute and store in repositories in advanced. Therefore, when the ontologies have been changed, and one has to recompute the inference model which is not suitable for large-scale deployment. In this paper, we purpose the approximation enabled query rewriting for LOD by using the semantic approximation technique since most of LOD's TBox is available on-line. This will enable us to perform inference query answering over LOD through selected repository via SPARQL endpoint.

1 Introduction

Query answering for Linked Open Data(LOD) is a well known problem. Currently, most triple stores are able to load a large data set. However, they can only partially support query answering by either compute the inference ontology and store in repositories in advanced or define a rule for each ontology. The problem arises when the data have been changed, and one has to recompute the inference model which is not suitable for large-scale deployment. The idea of query rewriting is to rewrite an input query based on the target ontology. However, the query rewriting approaches can be only apply to tractable languages like OWL 2 QL or OWL 2 EL in order to obtain polynomial time computation. Since OWL DL ontologies are used in the LOD, such as the Music Ontology, proper query rewriting techniques are required even for intractable ontology languages, such as OWL DL and OWL 2 DL.

1.1 Approximation

The approximation approach is an approach that allow us to retrieve an ontology which is described in a less expressive DL language from the given more expressive language. In the other hand, the result ontology still maintain a semantic of the original ontology and it is acceptable for query answering.

1.2 Reusing SPARQL Endpoints

Currently, there are a lot of SPARQL endpoint that stored a lot useful data including LOD are available on the internet and we can freely use those endpoint. However, those data are read only data which is not allow us to perform any kind of reasoning apart from their services. Therefore, the query rewriting play an important role for inference query answering. Furthermore, we can have different TBox from the original dataset.

Another benefit of query rewriting approaches is that with the same dataset and query but different TBox then the result will different too. Therefore, we can reuse the dataset by modifying its TBox for the different purpose.

In this paper, we propose the query rewriting over LOD by using the semantic approximation. With semantic approximation, the TBox of LOD will be approximate into OWL 2 QL, which is designed for query answering and can deliver more complete result. Input queries can then be rewritten by the approximated OWL 2 QL ontology. The resulting queries can then be sent to SPARQL endpoints for data retrieval. In the rest of paper, we first introduces the features of the AQUA query rewriting engine in section 2. The design decisions of AQUA System are then further explained in Section 3. Finally we give some discussion and conclusion in section 4.

2 Features

In this section, we provide some of the features that we use in the AQUA system.

2.1 Knowledge Compilation

We refer to [2] for the definition of knowledge compilation for OWL ontologies. Knowledge compilation aims at preprocessing of the ontology (knowledge base) offline, i.e. before run-time reasoning is performed. Run-time or online reasoning performs different inferences like consistency checking, subsumption or concept satisfiability checking. The run-time reasoning depends on the particular application.

The off line reasoning is performed in advance and aims at reducing complexity of run-time reasoning. The derived facts from off line reasoning are added to the knowledge base. For instance all implicit subsumptions are expressed in an explicit class hierarchy.

2.2 Approximation

This section gives an outline of syntactic and semantic approximation of OWL languages. The knowledge base (KB) consists of the TBox and ABox, which is represented in the underlying language which is OWL FA in this chapter. The approximation can be applied to the whole KB, or separately to the TBox and ABox.

The following general approximation techniques are described in [2]:

- **Language Weakening** exploits the notion of weakening the logical language to a weaker and less expressive language which is more tractable for reasoning applications than the original logical language. This kind of approximation can also be realized in the other direction by starting with the weak language and successively strengthen the theory.
- **Approximate Deduction** is the weakening of the logical entailment as another approximation. This approximation is sound but not necessarily complete.
- **Preprocessing** of the knowledge using offline reasoning in advance is another technique that is already described knowledge compilation (section 2.1).

Syntactic Approximation We consider the Approximate Deduction as a syntactic approximation. These approximation is focused in [2] and [4]. The following example from [2] demonstrates weakening of the logical entailment for a concept subsumption. The knowledge base contains for instance the subclass axiom $C \sqsubseteq D$. The entailment is weakened if the concept C is replaced by the weaker (less specific) concept D . Another method of weakening is the forgetting of subclass relations of a superclass or replacing the subclass by a weaker subclass.

These syntactic approximations aims at improving efficiency of reasoning and querying. The more tractable reasoning and querying properties for a complex knowledge base are achieved by either approximating concepts in the knowledge base or within the query. This is realized by using less complex expressions which approximate the more complex concepts.

Semantic Approximation This section outlines semantic approximation [3] which guarantees in contrast to the syntactic approximation soundness. The descriptions and definitions of this section refer to [3].

For a given ontology O_1 described in a DL language \mathcal{L}_1 the result of the semantic approximation is an ontology O_2 which is described in a less expressive DL language \mathcal{L}_2 and the condition $O_1 \models O_2$ holds.

The entailment set of an ontology O_1 with respect to the DL language \mathcal{L}_2 is the set of all axioms which are entailed from ontology O_1 and are described with the vocabulary from \mathcal{L}_2 . The entailment set is described with $ES(O_1, \mathcal{L}_2)$.

2.3 Query Rewriting

The concept of query rewriting has been widely adopted for cases where multiple data sources exist, such as data integration of contexts involving inconsistent databases. Query rewriting is used to access these heterogeneous data sources via a consistent data schema, viz. a view. However, the aims of query rewriting for retrieving ontology inference contents differ from those of data integration. The main difference is that the proposed query rewriting focuses on extending a query to retrieve implicit contents of an ontology, other than retrieving data from multiple datasets. In this paper, we use the well known query rewriting technique from [1].

3 AQUA

In this section, we are briefly explain about a design of the system architecture. Figure 1 shown a design of the system architecture.

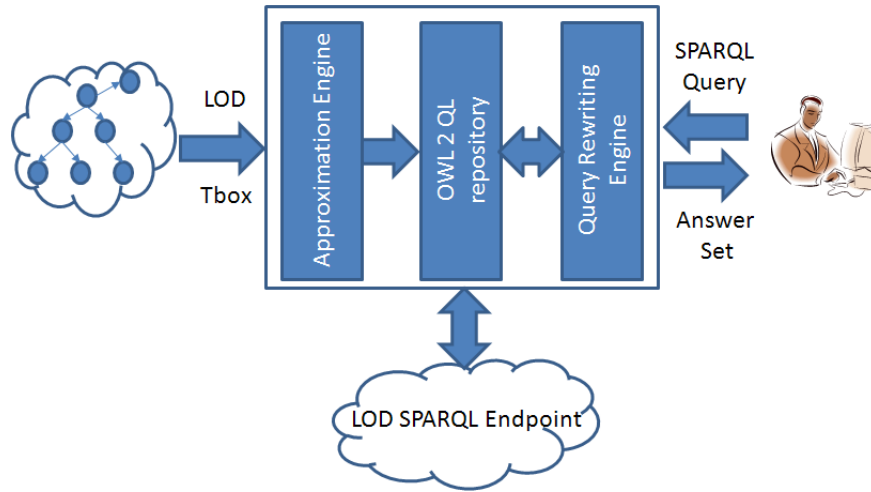


Fig. 1. System Architecture

3.1 Approximation Engine

The Approximation Engine will take the LOD's TBox as an input and approximate it into OWL 2 QL and store it on repositories. The Approximation Engine use the existing OWL2 DL reasoner for approximate step. The output is a saturated OWL2 QL TBox which dose not require any further reasoning at the query rewriting step. User can also submit a whole ontology for a small ontology to the approximation engine. However, AQUA will only store only an ontology's TBox.

3.2 Query Rewriting Engine

The Query Rewriting Engine will take the approximated TBox that stored on the repositories and SPARQL Query to perform a query rewriting bit. The Query Rewriting Engine will be execute following step:

- Firstly, the system will translate SPARQL query into Conjunctive Query(CQ) form and store it separately between classes and properties. The concept can be detect from a query body in the pattern like `?x rdf:type yago:Publication106589574` and for properties can be detect in the pattern like `?s dbpedia:name "The Lord of the Rings"@en`.

- Secondly, the query rewriting engine will load the approximated ontology TBox and CQ and compute the set of CQ according to the selected ontology's TBox.
- Finally, system will translate CQ back in SPARQL query and submit a set of SPARQL query to the selected SPARQL Endpoint and forward the result to the user.

3.3 Reusing SPARQL Endpoints

The same dataset and query but different TBox can give a different result set. Thus, this will enable user to retrieve a data according to user preference by modify the ontology's TBox.

3.4 Web Interface

The web interface is shown in Figure 2. From the web interface, user can submit a url of LOD's TBox or any ontology's TBox that user wish to use for their data retrieval. However, this url should link to an OWL, RDF or RDFS file and it must be consistent TBox. There is a section for user to submit the SPARQL endpoint that the data set are stored. For example, the SPARQL endpoint for DBpedia is <http://dbpedia.org/sparql>.

To use the AQUA, user can select the TBox and SPARQL endpoint from list box and enter a SPARQL query into a query box. Then click on a submit button. The result will appear on the screen.

4 Discussion and Conclusion

The AQUA system will enable user to perform the inference query answering over the large datasets without pre-compute and store the inference axiom which difficult and expensive to maintain and AQUA does not require any defined rule which is not reusable for another ontology. Further more, it allow user to retrieve a different answer set based on their preference TBox.

The main idea of this system is to combine the existing technology such as semantic approximation and query rewriting into practical solution for a real world application.

There are some limitations on the system. Currently, we support only SELECT query and we can only rewriting the basic graph pattern in a SPARQL query. In the future, we hope we can support another query type and provide a evaluation of the system.

References

1. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, Riccardo Rosati, and Guido Vetere. DL-lite: Practical reasoning for rich dls. In *Proc. of the 17th Int. Workshop on Description Logic (DL 2004)*, volume 104 of *CEUR Electronic Workshop Proceedings*, <http://ceur-ws.org/>, 2004.

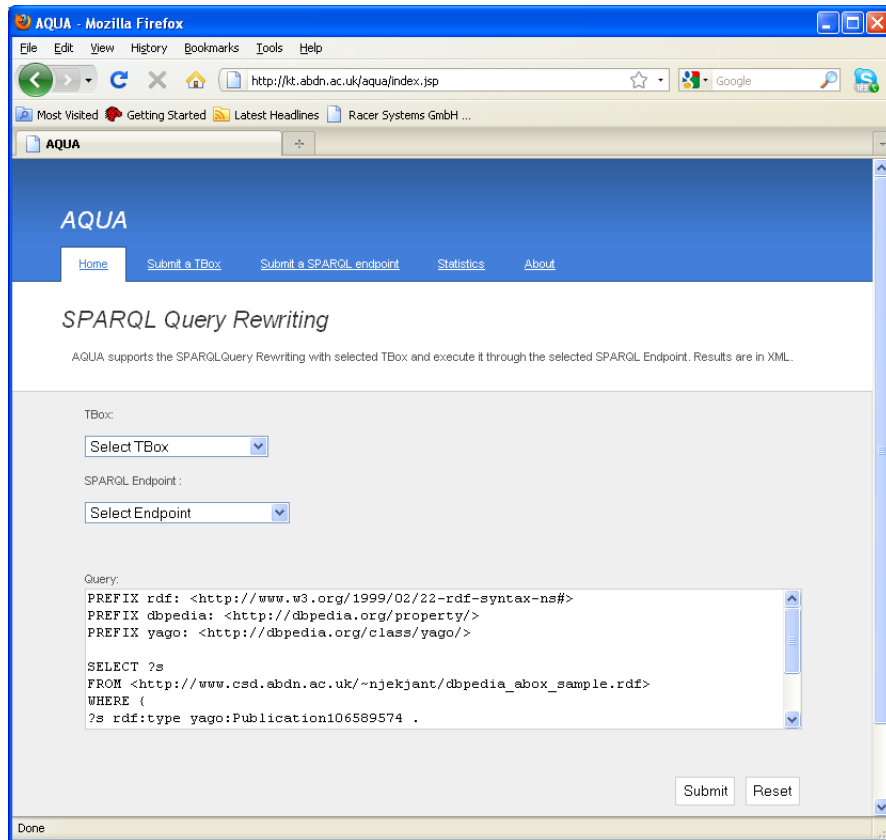


Fig. 2. AQUA

2. P. Groot and H. Stuckenschmidt. Approximating Description Logic Classification for Semantic Web Reasoning. In *Proc. of ESWC*, volume 3532, pages 318–332. LNCS, 2005.
3. J.Z. Pan and E. Thomas. Approximating OWL-DL Ontologies. In *Proceedings of AAAI*, 2007.
4. M. Schaerf and M. Cadoli. Tractable Reasoning via Approximation. *Artificial Intelligence*, 74:249–310, 1995.