A General Approach to Query the Web of Data

Xin Liu[1]

Department of Information Science and Engineering,
University of Trento, Trento, Italy
liu@disi.unitn.it

Abstract. With the development of the Semantic Web, an increasing amount of data with semantics has been published on the web according to the Linked Data principles and become ubiquitous. The requirement of utilizing the entire web of data to answer a query has arised since the desire for the application of such ubiquitous semantic data sources. However, limited research work has been made on the query above the web of data and there is not a formal way to describe the query processing procedure. In this paper, we propose a general query processing method on the web of data, which contains three steps: data inference configuration, data discovery, result generation and ranking. Finally, we briefly represent the work already done and the future work.

1 Introduction

Due to the development of the Semantic Web, an increasing amount of data with semantics has already been published on the web according to the Linked Data rules [1]. The core idea of Linked Data is to make links between different semantic data so that a person or machine can explore the web of data. The basic rules to publish such data require the identification of an entity with a URI that can be dereferenced into RDF data which is the description information of the entity with the standard format RDF[6]. In such data, links that point to other data sources make it possible to access the destination data. The entire web of data generates a single, distributed database that contains all kinds of information, where some pieces of the RDF data are interlinked through the links. With the increasing number of semantic data on the web, the requirement of efficient data query is inevitable.

RDF query languages (like RQL[5], SeRQL[2], SPARQL[3], etc.) and RDF storage tools (Jena[1], Sesame[2], 3store[3], etc.) provide the basis of the query of one single RDF storage. Furthermore, some research work has already been made on the query of multiple data sources. They can provide a transparent query access to multiple, distributed RDF data sources just as query to a single RDF storage. However, the precondition of these methods is that the required data sources to answer a query are known in advance. If we do not know the data sources in

[1] http://jena.sourceforge.net/
[2] http://www.openrdf.org/
[3] http://threestore.sourceforge.net/

advance, the available approaches or theories of querying upon the web of data is still missing .

In this paper, we try to fill the research gap by proposing a general approach to execute SPARQL queries on the entire web of data. The task of semantic web search engines (like Sindice, Falcns, etc) belongs to the information retrieval approaches while the work in this paper mainly derived from the execution of (SPARQL) queries on the semantic web. The proposed mechanism mainly includes three steps: data inference configuration, data discovery and extraction, result generation and ranking. Data inference configuration is used to configure a query processing procedure, that is whether allowed to use implicit information got from data reasoning to answer a query or not. The task of data discovery and extraction is to discover new data sources those are unknown in advance for the query processing according to the data inference configuration. The final results will be get and ranked in the result generation and ranking step.

The remainder of the paper is structured as follows: in section 2, I briefly introduce the state of the art of the data querying methods. In section 3, I propose a general approach to process a query on the web of data and finally, I summerize this proposal and briefly represent the future work in section 4.

2 Related Work

With the development of the semantic web, the query process on the semantic data is moving forward from one single data source to various, distributed data sources and even the entire web of data.

DARQ[4] is a query engine for federated SPARQL queries. It provides a transparent query access to multiple, distributed SPARQL endpoints like querying a single RDF graph. **SEWimQ**[7] is similar to the DARQ approach. It provides a middleware that enables the virtual access to distributed RDF data sources. It set up an RDF endpoint for each non-RDF format data source, make use of query parser, optimizer and engine to rewrite a SPARQL and federate the results. **Virtuoso**[5], a comprehensive data integration software developed by OpenLink Software, combines the functionality of a traditional RDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system. It enables the integration of numerous heterogeneous data from distributed data sources. However, all of the methods mentioned above have the precondition that the data sources to be used are already known before the query process.

The **Sematic Web Client Library**[6] presents the complete Semantic Web as a single RDF graph and support to query on the semantic web. It makes use of a pipeline based approach to dynamically retrieve information on the web by dereferencing HTTP URIs or query semantic web search engines during the process of a query. Paolo et al. present a formal description of query the web of

[4] `http://darq.sourceforge.net/`
[5] `http://virtuoso.openlinksw.com/`
[6] `http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/`

data in [8]. They propose a model of open collection of RDF graphs on the web and three different ways in which a query can be answered.

The approaches adopted in the two papers try to solve the problems to some extent on the data discovery during the query processing but they do not make a further research on the data relevance analysis and result ranking algorithms.

3 Proposed Methods and Preliminary Results

The semantic web languages such as OWL, RDF make open world assumption and assume incomplete information by default. One single ontology may not answer a query but a set of ontologies can do. So the query processing on the web of data requires data integration. On the current semantic web, there are significant amount of distributed ontologies. One single RDF graph may not include enough data to get answers for a query but may contribute to part of them, thus if one query is answerable on the web of data, then the integration of possible relevant RDF graphs can complete the answers.

The methodology of the query processing approach is based on the analysis of the query language. On the semantic web, SPARQL is a W3C recommendation language to query RDF data. A relational algebra for SPARQL[7] can be used to analyze and express a SPARQL query in relational algebra. In order to answer a SPARQL query, it is required to find all solutions [4] for each triple pattern in the query. The set of solutions for one triple pattern can be treated as an "relation" in relational algebra, called "RDF relation". And then the relational algebra operators, such as *selection, projection, rename, inner join*, etc. can be executed on such RDF relations to get the final answer. Therefore, the following discussion focus on the solutions for one triple pattern. As long as we get solutions for each triple pattern, we can get the final answers through relational operators.

The goal of global query processing is to use the entire (or as much as possible) web of data to execute SPARQL queries and generate answers. Many issues have arised due to the features of "openness" and "incompleteness" of the RDF graphs. **(1)**. The data can be published anywhere, we cannot find all the data to answer a query; **(2)**. People don't know the schema of each data source so that we cannot send a precise query to a specific RDF data source as we use SQL to query relational databases; **(3)**. The answer should include not only the explicit information represented in RDF data but also the implicit information which can be got through data inference. Based on the issues above, I propose a way with three steps to query the entire web of data using semantic web query language.

3.1 Data Inference Configuration

The first step to execute a query is to configure the data inference. The answer of a query may be various according to the function of data inference. Current available RDFS/OWL reasoners mainly include Jena, Pallet, FaCT++, etc. Some

[7] `http://www.hpl.hp.com/techreports/2005/HPL-2005-170.html`

rules in RDFS/OWL vocabularies used to generate implicit statements includes: *rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range, owl:equivalentClass, owl:equivalentProperty, owl:inverseOf, owl:tran sitiveProperty, owl:sameAs*, etc. Besides, user-defined rules can also be used in Jena reasoner, like "uncle_rule": (?x <`http://foo.com/rel/botherOf`>?y) (?y <`http://foo.com/rel/fatherOf`>?z) = (?x <`http://foo.com/rel/uncleOf`>?z). One or multiple inference rules can be used to complete the execution of a query. The strategy of data discovery will depend on the configuration result. As long as the inference configuration is fixed, we can further make a decision on the data discovery strategy.

3.2 Data Discovery

One of the main idea of the semantic web is to use information from anywhere on the web. When a query is executed, it is infeasible to locate every data source in advance. Therefore, in order to get answers from the web of data, the first step should be the data discovery. The task of data discovery is to collect as many data sources as possible those may be relevant to answer a query. Based on different features of current RDF data sources on the web, there are two ways to locate globally distributed sources, one is relying on the links between different graphs and the other is depending on the semantic web search engines.

In terms of the semantic data according to the Linked Data principle, we can rely on RDF links to locate other data sources. RDF links take the form of RDF triples, in which subject is a URI reference with the namespace of one data source and object is a URI reference with the namespace of the other one. When defining a global query, we should simultaneously set a starting point (an RDF graph or a merge of a set of graphs) from which we can crawl the web of data to collect more data sources. The staring point can be either dereferenced from the URIs in the query or provided by users. The process of collecting interlinked graph will follow the principle of web page collection of a web crawler.

Besides, graphs may be discovered through the semantic web search engines (e.g. Sindice[8], Sig.ma[9], Swoogle[10], etc.). They can be used to collect the graphs those include the same keywords or URIs. Due to the introduction of data inference, the data discovery through search engines is an iteration process. Different inference rules will generate different data discovery strategy. In general, it will stop until there is no new statements found. Data discovery task through search engines will collect two kinds of statements: one is the set of statements with the same triple pattern as the query, the other is the set of statements with available data inference rules.

[8] `http://sindice.com/`
[9] `http://sig.ma/`
[10] `http://swoogle.umbc.edu/`

3.3 Results Generation and Ranking

Through the last two steps, potentially relevant statements have already been discovered and collected. It is feasible to execute a query on such statements relying on available reasoners. There may be many answers found for a query, so it is necessary to return a ranking result according to the importance of each answer. Evaluating the weight of one answer mainly lies on two factors: the number of occurrences of one answer and the number of triples which can get this answer. The more number of occurrences of one answer is and the less number of triples is, the more weight of the answer is.

4 Conclusion and Future work

In this paper, we propose a general approach to complete the query processing upon the web of data, which mainly includes data infererce configuration, data discovery, result generation and ranking.

For the future, my research work will focus on the following points: **(1)**. The design and development of efficient and precise data discovery strategy. **(2)**. The design of result ranking algorithm. **(3)**. The implementation of this general approach. We can use some real world queries to evaluate this approach on performance and quality. **(4)**. The definitions of relevant sub-graph (part of an RDF graph) for a query. Currently, the granularity of relevance is a whole RDF graph, but the fact is that not every triple in a graph is relevant to a query. So the collection of relevant sub-graphs will make the query more efficient.

References

[1] *Tim Berners-Lee. Design Issues: Linked Data. Online, last change: June 2009, http://www.w3.org/DesignIssues/LinkedData.html.*

[2] Jeen Broekstra and Arjohn Kampman. Serql: An rdf query and transformation language. August 2004.

[3] Hewlett-Packard Laboratories Bristol Eric Prud'hommeaux, Andy Seaborne. *SPARQL Query Language for RDF*, January 2008. http://www.w3.org/TR/rdf-sparql-query/.

[4] Marcelo Arenas Jorge Perez and Claudio Gutierrez. Semantics of sparql. Technical report, May 2006.

[5] Greg Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. Rql: A declarative query language for rdf. pages 592–603. ACM Press.

[6] Graham Klyne and Jeremy J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax.* http://www.w3.org/TR/rdf-concepts/.

[7] Andreas Langegger, Wolfram Wöß, and Martin Blöchl. A Semantic Web Middleware for Virtual Data Integration on the Web. In *ESWC*, pages 493–507, 2008.

[8] Chiara Ghidini Paolo Bouquet and Luciano Serafini. Query the web of data: A formal approach. In *Inproceedings of Asian Semantic Web Conference 2009*, 2009.