Thou Shalt ASQFor And Shalt Receive The Semantic Answer*

Muhammad Rizwan Saeed, Charalampos Chelmis, Viktor K. Prasanna
University of Southern California, Los Angeles, California
{saeedm,chelmis,prasanna}@usc.edu

Abstract

The combination of data, semantics, and the Web has led to an ever growing and increasingly complex body of semantic data. Accessing such structured data requires learning formal query languages, such as SPARQL, which poses significant difficulties for non-expert users. Many existing interfaces for querying Ontologies are based on approaches that rely on predefined templates and require expensive customization. To avoid the pitfalls of existing approaches, while at the same time retaining the ability to capture users' complex information needs, we have developed a simple keyword-based search interface to the Semantic Web. In this demonstration, we will present ASQFor, a systematic framework for automated SPARQL query formulation and execution over RDF repository using simple concept-based search primitives. Allowing end-users to express simple queries based on a list of "key-value" pairs that are then translated on-the-fly into SPARQL queries is a hard problem. In this demonstration, we will discuss the challenges that we have addressed to bring ASQFor to real practice, and also the difficult problems that remain to be solved in future work. During our demonstration, we will show how ASQFor can be used for decision support as well as an intelligent Q/A System.

1 Introduction

As more and more semantic data become available on the Web, the question of how end users can access this body of knowledge becomes of crucial importance. Tools for creating, editing, and querying Ontologies have been widely developed [Kaufmann and Bernstein, 2010]. However accessing semantic data requires intimate familiarity with existing formal query languages such as SPARQL[1] and knowledge about underlying schema Ontologies of the semantic repository. SPARQL allows users to write queries against data repositories that follow the RDF[2] specification of the World Wide Web Consortium (W3C) by creating queries that consist of triples, conjunctions, disjunctions, and optional patterns. To automatically generate such SPARQL queries, a system would have to (i) separate the input into syntactic markers and "meaningful" tokens, (ii) map tokens to concepts in the Ontology, (iii) link identified concepts based on relationships in the Ontology, and (iv) issue the query to collect the results.

In our demonstration, we take a <key,value> approach to the problem of querying a semantic data repository which is similar to the way arguments are passed to functions using programming languages. We present our *Automatic SPARQL Query Formulation* (ASQFor) framework which is a reusable, extendable and domain independent approach that requires virtually no end-user training to facilitate semantic querying over a knowledgebase represented by RDF. We implicitly assume that a schema for the knowledgebase is also specified in RDF or OWL[3], and can be queried.

2 Challenges

Our goal is to enable end-users to formulate semantic queries over structured data in terms of classes and properties while being oblivious to the actual structure of the data. To do so, the user provided keywords need to be mapped to the concepts and attributes in the schema Ontology. The next step is to find the smallest subgraph that connects all these mapped entities related to the query. For a generic graph, finding such a subgraph pattern is a hard problem (Steiner Tree Problem). In order to provide at least a subset of the rich semantic web functionality to users who are unfamiliar with either RDF or SPARQL, we have limited our approach to Ontologies that have tree structure, as multiple proposed Ontologies modeling real-world domains (such as [Kharlamov *et al.*, 2014]) are organized in this way.

3 Automatic SPARQL Query Formulation

ASQFor generates SPARQL queries in three steps. First, the user provided keywords are mapped to concepts and attributes in the Ontology. Second, the semantic relations between concepts in the query are extracted, based on which, the semantic

*This work is supported by Chevron U.S.A. Inc. under the joint project, Center for Interactive Smart Oilfield Technologies (CiSoft), at the University of Southern California.

[1] http://www.w3.org/TR/rdf-sparql-query/

[2] http://www.w3.org/RDF/

[3] https://www.w3.org/2001/sw/wiki/OWL

Q_1	Name, birthplace, gender and marital status of all people on active military duty.
Q_2	Occupations in different industries.
Q_3	Names of people who attended private school.
Q_4	All attributes for people born in California.

Table 1: Possible Demonstration Queries

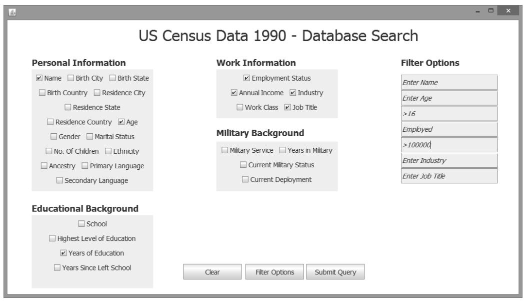

Figure 1: ASQFor Human-Computer Interactive Semantic Querying Software

query graph is built which models the user query intention. The SPARQL statements are generated while traversing this subgraph by populating statements that correspond to semantic relations and nodes at each step. Finally, the SPARQL query is executed on the semantic repository and results are returned to the user.

4 Demonstration Details

We will show how to use ASQFor in real-life by asking participants to query 1990 US census data stored in RDF without prior knowledge of RDF or SPARQL. Our sample knowledgebase contains 68 attributes for $2,458,285$ individuals in total. For the purpose of this demonstration, we will show how to formulate four queries (ranging from simple to complex queries, see Table 1) using a simple user interface that we have developed. Even though these queries are "simple", such queries can be hard to implement by users with few or no skills in using querying languages such as SPARQL.

We have already performed a limited user study to measure the usability of our ASQFor interface shown in Figure 1. Our study included 10 users with varying expertise level in Semantic Web (average expertise level was 1.9/5 with 1.3 SD, 5 being the expert). Minimal description of the underlying schema and virtually no instructions about how to use the ASQFor interface was provided. The users were asked to obtain results by issuing the four queries (Table 1) using the tool, and report on the ease of use. We also asked the users to create a query of their own devise. All users were able to issue the first three queries using the interface and get the intended results without any clarifications. For Q_4, 9/10 users were able to get all results. One user misinterpreted the query and selected fewer attributes and got different result. All users were able to complete the survey in under 6 minutes. In terms of "Easy to Use", the users gave the query interface an astounding 4.6/5 rating (SD 0.48, 5 being extremely easy to use). Our objective is to repeat this limited user study as part of our demonstration to quantify how easily end-users with virtually zero knowledge about semantic web technologies can query semantic data with the assistance of ASQFor. We will show that the overhead introduced by ASQFor is negligible for this dataset and discuss the challenges of scaling ASQFor with the size of the knowledgebase. Finally, we will discuss the current limitations of ASQFor as compared to the full capabilities of SPARQL and explain the challenges associated with relaxing the existing constraints of ASQFor to support more complex queries such as aggregation queries. Our objective is to show that providing even a small subset of the rich semantic web functionality to end-users who are unfamiliar with semantic web technologies, for example in the long tail of science, has the potential to accelerate the consumption of Semantic Web and Linked Open Data.

5 Significance of Work being Demonstrated

Search engines have become popular because they require minimum specified keywords for identifying the correct web documents. For example, a search comprising of keywords "Artist Mona Lisa" will show the same results as the syntactically correct English sentence "Who painted Mona Lisa?". In our demonstration, we are building on the same simple conceptual model by taking a <key,value> approach to the problem of querying a semantic data repository. For example, the equivalent of the keyword-based for this example would be <Artist,*>, <Painting, "Mona Lisa">, which is similar to the way arguments are passed to functions in programming languages such as Java. ASQFor algorithm traverses the Ontology dynamically to formulate queries. This gives it superiority over approaches that require pre-configured templates or pre-defined rules for formulating queries, since such approaches rely on involvement of an IT expert to add or modify the templates or rules. No information about the Ontologies is stored internally, which makes our approach portable across domains without any modification.

ASQFor enables end-users to easily answer complex semantic queries over structured repositories through an intuitive interface that matches the simplicity of keyword-based search engines. Formulating semantic queries that match users' information needs without requiring users to be intimately familiar with the underlying data representation or semantic web technologies is crucial for promoting the ubiquitous use of semantic web data.

References

[Kaufmann and Bernstein, 2010] Esther Kaufmann and Abraham Bernstein. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):377–393, 2010.

[Kharlamov *et al.*, 2014] Evgeny Kharlamov, Nina Solomakhina, Özgür Lütfü Özçep, Dmitriy Zheleznyakov, Thomas Hubauer, Steffen Lamparter, Mikhail Roshchin, Ahmet Soylu, and Stuart Watson. How semantic technologies can enhance data access at siemens energy. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 601–619, 2014.