Proof based maintenance of linked data

Kanghao Lu
kennyluck@csail.mit.edu

September 25, 2009

Abstract

With the recent uptake on the linked open data on the Web, there are many interlinked data sets on the Web today. Interesting results that might not be obvious from this *raw* data itself, can be derived by computing answers to rules using the data and essentially connecting the dots to see the big picture. We describe a novel method to crawl the Web of data and make inferences that satisfy rules of a particular domain and to re-inject the derived conclusions, along with the traces of computation, back into the read-write Web. We have developed a "Computation Agent" to do this using rules written in a formal Semantic Web rule language, and a user interface for viewing the data which includes the derived conclusions along with the justification.

1 Introduction

The linked open data cloud has seen a steady growth since DBPedia [?] project started in 2007. The DBpedia knowledge base alone currently describes more than 2.6 million things, and the knowledge base consists of 274 million pieces of information expressed in RDF triples [?]. Some Weblog hosting sites and social networking sites have already begun exporting user profiles to RDF [?] using the Friend-Of-A-Friend (FOAF) vocabulary [?]. Many other data sets have linked up to this nucleus of DBpedia data set, and to each other, effectively increasing the intelligence of the Web and supporting large scale data integration. This allows people to use data browsers such as Tabulator [?] to surf across silos of data that have been converted into linked data.

Although much of the linked data cloud is interlinked, there are many data sets that have no explicit semantic associations with each other even though these relationships might be obvious after a simple reasoning process. Once these relationships are exposed through some mechanism, the newly found relationships should be contributed back to the linked data cloud. This will allow reuse of these derived facts, without having to reason over the entire graph of data again. For someone who may be interested in knowing how a particular fact is derived, a proof tree detailing the derivation steps could be attached with the fact to enable provenance [1].

For example, in Fig **??**, we have a simple linked data graph. There are several different nodes (encoded A - J), and different kinds of relations between these nodes and beyond. (One could assume that node A corresponds to the WebID of Alice, node B corresponds to the WebID of Bob, and the edge between A and B, i.e. 'rel:AB' corresponds to 'foaf:knows'. Likewise, other nodes and edges in this graph can be any other entity and relationship exposed as linked data). By merely looking at this graph of data, the relationship between the triple $A \rightarrow E$ and $H \rightarrow K$ may not be obvious at the first glance. But after the application of a specific rule, we might discover that there is some relationship between these two triples. Assume that the derivation path of this fact is as outlined in the red arrows where new relationships between triples are created. It would be useful to make these new relationships on the linked open data cloud persistent, so that somebody else might be able to reuse this fact later. However, to reinforce the trustworthiness of the fact that $A \rightarrow E$ derives $H \rightarrow K$ we attach the derivation steps of the proof to it.

2 Motivating Scenario

Explain the scenario

3 Implementation

4 Future Work and Conclusion

[1] The impatient reader is encouraged to glance through [**?**] or to take a look at the implementation of the User Interface given in Section **??**.