RDFKB: A Semantic Web Knowledge Base

James P. McGlothlin, Latifur Khan, Bhavani Thuraisingham
The University of Texas at Dallas
Richardson, Texas, USA
{jpmcglothlin, lkhan, bhavani.thuraisingham}@utdallas.edu

Abstract

There are many significant research projects focused on providing semantic web repositories that are scalable and efficient. However, the true value of the semantic web architecture is its ability to represent meaningful knowledge and not just data. Therefore, a semantic web knowledge base should do more than retrieve collections of triples. We propose RDFKB (Resource Description Knowledge Base), a complete semantic web knowledge case. RDFKB is a solution for managing, persisting and querying semantic web knowledge. Our experiments with real world and synthetic datasets demonstrate that RDFKB achieves superior query performance to other state-of-the-art solutions. The key features of RDFKB that differentiate it from other solutions are: 1) a simple and efficient process for data additions, deletions and updates that does not involve reprocessing the dataset; 2) materialization of inferred triples at addition time without performance degradation; 3) materialization of uncertain information and support for queries involving probabilities; 4) distributed inference across datasets; 5) ability to apply alignments to the dataset and perform queries against multiple sources using alignment. RDFKB allows more knowledge to be stored and retrieved; it is a repository not just for RDF datasets, but also for inferred triples, probability information, and lineage information. RDFKB provides a complete and efficient RDF data repository and knowledge base.
.

1 Introduction

The World Wide Web Consortium (W3C) specifies the standards that define the semantic web. RDF (Resource Description Framework) is the standard format for data. All RDF datasets can be represented as collections of triples, where each triple contains a subject URI, a property URI and an object URI or literal. RDFS (RDF schema) and OWL (Web Ontology Language) are ontology languages defined by W3C. These standards allow ontologies to specify description logic and define classes, relationships, concepts and inference rules for a dataset. The URW3-XG (http://www.w3.org/2005/Incubator/urw3/), an incubator group of W3C, reviewed additional ontology formats that are available for expressing uncertainty and probability.

Our goal is to provide an efficient semantic web repository that reasons with all of this information. We will be able to query the RDF triples, but the results will take into account all available knowledge. OWL has provable knowledge entailment, such that inferred triples can be derived with complete certainty. Therefore, our query results will include inferred knowledge. Probabilistic ontologies and uncertainty reasoners allow us to infer possible knowledge with probabilities. Therefore our query results can return uncertain knowledge and probabilities. The queries should rank results based on confidence values, and should enable selection based on probability conditionals.

Ontology alignment allows us to match related concepts and terminology between ontologies. The goal is to allow queries against the complete knowledge set. Our goal is to allow such alignments to then be applied to the datasets. The queries should be able to be specified using the terminologies from any of the ontologies, or from a new common global terminology. The queries should return all relevant knowledge, including inferred triples and triples that are specified using different, but corresponding, terminologies.

Furthermore, we propose to support addition, deletions and updates to the data, the inference rules, and the alignment matchings.

2 Proposed Solution

Our solution is to use forward chaining to infer all possible knowledge. We propose to materialize and persist all data including inferred information and uncertain information. To implement this, we provide an inference engine that allows inference rules to register, and an interface defining the methods the inference rules must provide. We have developed inference rules for all OWL and RDFS constructs and for several proposed uncertainty reasoners including BayesOWL[Zhang et al., 2009] and Pronto (http://pellet.owldl.com/pronto). However, any inference rule can be registered.

We utilize a data management schema to track all of this data. The goal of the data management schema is to provide a simple way to manage and materialize all of the information. The data management tables are not accessed during query processing, therefore they are designed for efficient information retrieval.

The data management schema includes a triples table which stores all the triples and their probabilities. It also includes 4 provenance tables (Users, Datasets, InferenceEvents and Dependencies) which are used to track a triple's lineage. This allows us to efficiently support automatic deleting of inferred triples and updating of probabilities.

Our query schema consists of two tables, POTable and PSTable. These bit vector tables include a bit for every possible combination of subject, property and object. For example, the POTable contains columns for property, object, and subjectBitVector. The subjectBitVector contains a bit for each known subject URI. If the bit is on, that indicates that a triple exists in the dataset with that subject, property and object. We can now query entire collections of triples by retrieving a single vector. We can implement joins using bit vector *and* and *or* operations. The POTable and PSTable also includes a column for bitCount. bitCount, the number of 1's in the bit vector, provides us valuable selectivity factor information useful for query planning.

This bit vector schema is ideal for materializing inferred triples because a triple can be added by simply turning a bit from 0 to 1. Thus, it is possible to forward chain inferred triples and persist them without increasing the size of the query tables. This allows for very efficient inference queries.

These tables also include a threshold column used to support queries of uncertain information. If the threshold is <1, the bit vectors have a 1 for each and every triple that appears with a probability >= threshold. This allows us to quickly query information based on probability criteria.

3 Progress

We first implemented our bit vector tables, inference engine and OWL inference rule. We published this as [McGlothlin and Khan, 2009]. In this paper, we demonstrated that we could support inference queries without a performance penality and that we could outperform vertical partitioning [Abadi et al., 2007]. At this time, our solution was still read only.

In [McGlothlin and Khan, 2010a], we published our solution for storing and querying uncertain information. We also provided a solution for adding and deleting inferred triples. We showed experimental results indicating that we outperformed 7 state-of-the-art repositories (including RDF-3X[Neumann and Weikum, 2008] over 26 benchmark queries. We also presented performance results for some probabilistic queries we created.

In [McGlothlin and Khan, 2010b], we proposed our provenance tables as a solution for complete data management. We also offered a solution for implementing trust factors.

In [McGlothlin and Khan, 2010c], we built on this technology to support applying alignments to the dataset. We added support for updating and deleting inference rules. We used probabilistic inference rules to apply alignment matches and manage similarity measures. We used ranking queries to retrieve the best answers to queries.

Most recently, we have implemented an ontology alignment algorithm and are performing more realistic and thorough experiments with applying ontology alignment.

4 Future Work

We plan to implement a solution for improving ontology alignment with supervised learning and user feedback [Feng et al., 2009]. We will utilize our ability to update inference rules. Also, we plan to use cloud computing technologies to create a distributed version of our repository. This will improve scalability.

References

[McGlothlin and Khan, 2009] James P. McGlothlin, Latifur Khan. RDFKB: efficient support for RDF inference queries and knowledge management. *In Proceedings of IDEAS*, pages 259-266, September 2009.

[McGlothlin and Khan, 2010a] James P. McGlothlin, Latifur R. Khan. Materializing Inferred and Uncertain Knowledge in RDF Datasets. *In Proceedings of AAAI*, pages 1405-1412, July 2010.

[McGlothlin and Khan, 2010b] James P. McGlothlin, Latifur Khan. Efficient RDF data management including provenance and uncertainty. *In Proceedings of IDEAS,* pages 193-198, August 2010.

[McGlothlin and Khan, 2010b] James P. McGlothlin, Latifur Khan. A Semantic Web Repository for Managing and Querying Aligned Knowledge. *In Proceedings of ISWC,* November 2010.

[Abadi et al., 2007] Daniel J. Abadi, Adam Marcus, Samuel Madden, Katherine J. Hollenbach. Scalable Semantic Web Data Management Using Vertical Partitioning. *In Proceedings of VLDB,* pages 411~422, September 2007.

[Zhang et al., 2009] Shenyong Zhang, Yi Sun, Yun Peng, Xiaopu Wang. BayesOWL: A Prototype System for Uncertainty in Semantic Web. *In Proceedings of IC-AI*, pages 678~684, July 2009.

[Neumann and Weikum, 2008] Thomas Neumann, Gerhard Weikum. RDF-3X: a RISC-style engine for RDF. *In Proc. of VLDB*, pages 647-659, September 2009.

[Feng et al., 2009] Feng Shi, Juanzi Li, Jie Tang, Guo Tong Xie, Hanyu Li. Actively Learning Ontology Matching via User Interaction. *In Proceedings of ISWC,* pages 585-600, November 2009.