Provenance Management for Evolving RDF Datasets

Argyro Avgoustaki
ICS-FORTH and
Computer Science
Department,University of
Crete, Greece
argiro@ics.forth.gr

Giorgos Flouris
ICS-FORTH, Greece
fgeo@ics.forth.gr

Irini Fundulaki
ICS-FORTH, Greece
fundul@ics.forth.gr

Dimitris Plexousakis
ICS-FORTH and
Computer Science
Department,University of
Crete,Greece
dp@ics.forth.gr

ABSTRACT

Tracking the provenance of information published on the Web is of crucial importance for effectively supporting trustworthiness, accountability and repeatability in the Web of Data. Although extensive work has been done on computing the provenance for SPARQL queries, little research has been conducted for the case of SPARQL updates. In our work, we propose a new provenance model that borrows properties from both *how* and *where* provenance models, and is suitable for capturing the triple and attribute level provenance of data introduced via SPARQL INSERT updates. To the best of our knowledge, this is the first model that deals with the provenance of SPARQL updates using algebraic expressions, in the spirit of the well-established model of *provenance semirings*.

Keywords

Data provenance, SPARQL Update, Reconstructability, How and Where provenance, RDF

1. PROVENANCE MANAGEMENT

During the last few years, we have witnessed an explosion in the volume of semantic data available on the Web. These data are usually published using the RDF data model[1], where information is represented using *triples* (in the form of subject, predicate, object), organized in *named graphs*, thereby forming *quadruples*. Querying and updating RDF data is performed using the W3C standards SPARQL[2] and SPARQL Update[3] respectively.

[1]http://www.w3.org/TR/rdf11-primer/
[2]http://www.w3.org/TR/sparql11-overview/
[3]http://www.w3.org/TR/sparql11-update/

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Nowadays semantic data is the most prominent example of large scale data where one could create new *datasets* (sets of quadruples) by integrating existing ones. In this setting, recording the *provenance* of such data, i.e., their *origin* or *source*, which describes from *where* [2] and *how* [6] the data was obtained, is of crucial importance for supporting trustworthiness, accountability and repeatability. This is necessary due to the open and unconstrained nature of the Web of Data and the growing tendency to populate scientific data warehouses through SPARQL updates offered by SPARQL endpoints.

In this work, we deal with the problem of *capturing and managing the provenance of quadruples constructed through SPARQL updates*. More specifically, we focus on SPARQL INSERT operations used to add newly created triples in a target named graph. The purpose of computing the provenance for such operations is to record from *where* and *how* each quadruple was constructed, thereby allowing us to determine the quadruples and the SPARQL operators that were used to produce it.

The problem of managing provenance information has received considerable attention [2, 4, 5, 6, 7, 8, 9, 10, 11], but most works deal with query provenance. Algebraic expressions have been used to capture (query) provenance in varying levels of detail [5, 6, 8]. In the RDF context, provenance is often represented using named graphs [3, 4, 8].

However, the unique requirements associated with SPARQL update provenance do not allow a direct reuse of such approaches.One problem is that the named graph component of a quadruple is defined by the user in the INSERT update, so triples with different origin may be added to the same named graph. Thus, the standard approach of capturing provenance through the named graph of a quadruple is not sufficient, and provenance should be defined for quadruples, rather than triples (as in most works).

In addition, quadruples created via INSERT updates could be the result of combining values found in different quadruples through different SPARQL operators (union, join). This creates a unique challenge, because each attribute of a quadruple may have a different provenance. Thus, fine-grained, *attribute level* provenance models are called for, and more expressive models that go beyond the named graphs approach are needed.

Another challenge stems from the persistence of a SPARQL update result, which implies that when a quadruple is accessed, the

SPARQL update that generated the quadruple is no longer available. This makes standard *how* provenance models unsuitable for recording provenance at a fine-grained level in this setting. As an example, standard how-provenance approaches will record that a join was used to generate a quadruple, but will not record the components of the quadruples that were joined to produce the result; even though this information is easily available during queries (via the SPARQL query), this is not the case for SPARQL updates (where the SPARQL update is not available). Recording the INSERT update is not an efficient remedy for the situation, because (a) the syntactic form of the actual INSERT update is irrelevant and (b) the INSERT update is no longer relevant, as the dataset has evolved.

Therefore, more fine-grained forms of how-provenance are called for. We define this more demanding form of how-provenance in an indirect manner, by introducing the notion of *reconstructability* [1], which refers to the ability of using the provenance information for *reconstructing* an INSERT update that is *compatible* with the INSERT update that generated this quadruple. By compatible, we mean that these two updates differ only in their variables' names and in the filters that the first update may employ.

We show that, to satisfy the requirement of reconstructability, the provenance of a quadruple should be expressive enough to identify: (a) the quadruples that contributed to its creation (*where provenance* [2]), and (b) how these quadruples were used (via joins and unions) to generate the new one (*how provenance* [6]), under the more demanding form of how-provenance explained above.

2. REFERENCES

[1] A. Avgoustaki, G. Flouris, I. Fundulaki, and D. Plexousakis. Provenance Management for Evolving RDF Datasets. In *Extended Semantic Web Conference*, 2016.

[2] P. Buneman, S. Khanna, and W.-C. Tan. Why and where: A characterization of data provenance. In *International Conference on Database Theory*, 2001.

[3] J. J. Carroll, C. Bizer, P. J. Hayes, and P. Stickler. Named graphs, provenance and trust. In *International Conference on World Wide Web*, 2005.

[4] G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides. Coloring RDF triples to capture provenance. In *International Semantic Web Conference*, 2009.

[5] F. Geerts, G. Karvounarakis, V. Christophides, and I. Fundulaki. Algebraic Structures for Capturing the Provenance of SPARQL Queries. In *International Conference on Database Theory*, 2013.

[6] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *Principles Of Database Systems*, 2007.

[7] H. Halpin and J. Cheney. Dynamic provenance for SPARQL updates. In *International Semantic Web Conference*, 2014.

[8] G. Karvounarakis, I. Fundulaki, and V. Christophides. Provenance for Linked Data. In *Search of Elegance in the Theory and Practice of Computation*. 2013.

[9] L. Moreau. The foundations for provenance on the web. *Foundations and Trends in Web Science*, 2(2-3), 2010.

[10] S. Vansummeren and J. Cheney. Recording Provenance for SQL Queries and Updates. *IEEE Data Engineering Bulletin*, 30(4), 2007.

[11] M. Wylot, P. Cudré-Mauroux, and P. T. Groth. Tripleprov: efficient processing of lineage queries in a native RDF store. In *International Conference on World Wide Web*, 2014.