Integrating Linked Data through RDFS and OWL: Some Lessons Learnt

Aidan Hogan

Digital Enterprise Research Institute,
National University of Ireland Galway, Ireland

Abstract. In this paper, we summarise the lessons learnt from the PhD Thesis *Exploiting RDFS and OWL for Integrating Heterogeneous, Large-Scale, Linked Data Corpora* where we looked at three use-cases for reasoning over Linked Data: (i) translating data between different vocabulary terms; (ii) identifying and repairing noise in the form of inconsistency; and (iii) detecting and processing coreferent identifiers (identifiers which refer to the same thing). We summarise how we overcome the challenges of *scalability* and *robustness* faced when reasoning over Linked Data. We validate our methods against an open-domain corpus of 1.1 billion quadruples crawled from 4 million Linked Data documents, discussing the applicability and utility of our reasoning methods in such scenarios.

1 Introduction

The Linked Data community has encouraged many publishers to disseminate information on the Web using the Semantic Web standards [3]. Much of the success of Linked Data is perhaps attributable to their bottom-up approach to the Semantic Web, where higher levels of the Semantic Web stack—*ontologies*, *logic*, *proof*, *trust* and *cryptography*—are downplayed. However, many of the challenges originally envisaged for the traditional Semantic Web are now being realised for the "Web of Data". Applications sourcing a Linked Data corpus from numerous different domains will encounter challenges with respect to consuming and integrating it in a meaningful way.

First, in Linked Data, complete agreement upon a single URI for each possible resource of interest is infeasible. In fact, Linked Data principles encourage minting local, dereferenceable URIs. Further still, use of blank-nodes is prevalent (although expressly discouraged). Consequently, we propose that Linked Data needs methods for (i) *resolving coreferent identifiers*; (ii) *processing coreference* for consuming heterogeneous corpus as if (more) complete agreement on identifiers was present.

Second, Linked Data publishers may use different but analogous terms to describe their data: for example, choosing `foaf:maker` when its inverse `foaf:made` is more commonly used, or favouring the more specific `foaf:homepage` over `foaf:page`. Publishers may also adopt different *vocabularies*: for example, picking `foaf:maker` and not `dct:creator`. We thus propose that Linked Data needs some means of *translating between terminologies*, e.g., to aid querying.

Third, various forms of noise may exist in the data, some of which can be characterised as being formally inconsistent. Thus, we propose that Linked Data consumers may require methods which *detect and repair inconsistency*.

Notably, RDFS and OWL have seen good uptake in Linked Data. Various vocabularies have emerged as de facto choices; e.g., FOAF for personal information, DC for annotating documents, and so on [8]. Such vocabularies are described using subsets of the RDFS and OWL standards [9]; these descriptions often include, e.g., mappings between (possibly remote) terms, disjointness constraints useful for finding inconsistency, (inverse-)functional properties useful for resolving coreferent resources, and so on [9].

In the thesis *Exploiting RDFS and OWL for Integrating Heterogeneous, Large-Scale, Linked Data Corpora*, we looked at three use-cases for reasoning over Linked Data: (i) translating between data described using different vocabulary terms; (ii) identifying and repairing inconsistencies; and (iii) resolving and processing coreferent identifiers. Similar use-cases are motivated by, e.g., Auer & Lehmann [1] and Jain et al. [12].

To help ensure scale, all of our methods are distributed over a cluster of commodity hardware; to ensure robustness, our methods critically examine the source of data. We focus on application over static datasets; we evaluate all of our methods against a corpus of 1.118 g quadruples crawled from 3.985 m RDF/XML Web documents (965 m unique triples). We now summarise our results; for more details, please see [9].

2 Baseline Reasoning

Our first use-case establishes a baseline for reasoning, materialising translations of assertional data from one terminology to another based on RDFS/OWL mappings provided by Linked Data publishers. We perform rule-based reasoning, where we apply a *tailored* subset of the OWL 2 RL/RDF ruleset [6].

OWL 2 RL/RDF rules are cubic in nature, where for a given RDF graph G, OWL 2 RL/RDF can entail every triple representing all combinations of constants in G (and constants in the heads of the OWL 2 RL/RDF rules). It is not difficult to show that this cubic bound is tight, where the following two triples added to G:

```
owl:sameAs owl:sameAs rdf:type ; rdfs:domain owl:Thing .
```

will, through the OWL 2 RL/RDF rules for equality and domain, infer all possible triples for all available constants. Finally, many rules prescribe quadratic entailments, including, e.g., transitivity and rules supporting equality.

OWL 2 RL/RDF is thus not directly applicable for large-scale materialisation tasks. Our first optimisation is to separate terminological (aka. schema or ontological) data from instance data, based on the observation that for a sufficiently large crawl of Linked Data, such data represents $<0.1\%$ of the total volume and is the most commonly accessed during reasoning—assertional (aka. instance) data is much more numerous in such scenarios. (Similar optimisations have been justified in the recent scalable reasoning literature [10, 14, 17, 16, 15].) Thereafter, we select and apply a subset of OWL 2 RL/RDF rules which contain a maximum of one assertional pattern in the body: assuming that the terminological data is fixed, such rules enable linear-scale reasoning over the assertional data of the corpus. Further still, we introduce various optimisations possible through this separation; in particular, we ground the terminological patterns in the OWL 2 RL/RDF ruleset, generating a large-set of domain-specific rules. For example, consider the OWL 2 RL/RDF rule cax-sco (where the terminological pattern in underlined) and the following two terminological triples:

$$(?x, a, ?c2) \leftarrow (?c1, rdfs:subClassOf, ?c2), (?x, a, ?c1)$$
$$foaf:Person\ rdfs:subClassOf\ foaf:Agent\ .$$
$$foaf:Agent\ rdfs:subClassOf\ dc:Agent\ .$$

We *partially evaluate* the cax-sco rule to generate two new *T-ground* rules as follows:

$$(?x, a, foaf:Agent) \leftarrow (?x, a, foaf:Person)$$
$$(?x, a, dc:Agent) \leftarrow (?x, a, foaf:Agent)$$

Thereafter, we build a linked-rule index which returns rules which apply to a given triple, allowing for efficient reasoning by means of a simple scan of the data; such optimisations reduced the reasoning runtime by a factor of $\sim 5\times$ for our reasoning [9].

Our reduced fragment of OWL 2 RL/RDF rules is also amenable to distribution over a cluster of shared-nothing commodity hardware with no co-ordination required during the bulk of reasoning [17, 16, 11]. We evenly split the corpus over the nodes in the cluster, extract and merge the terminological data in parallel, perform reasoning over the terminology, generate a set of T-ground rules, and replicate these T-ground rules on all nodes in the cluster. Since our rules only contain one assertional pattern in the body, each T-ground rule will only contain one pattern in the body, and each node in the cluster can then perform reasoning independently over its segment of the data.

The next issue tackled was that of robustness: we found various examples of documents defining impudent RDFS and OWL axioms involving popular third-party terms where, e.g., one document defines nine local *properties* to be the domain of `rdf:type`.[1] Based on our separation of terminological data, we described our general notion of authoritative reasoning [4, 10], whereby the terminological data provided by a given Web document can only effect entailments over assertional data that uses terms which dereference to that document. Thus, e.g., only the FOAF vocabulary can specify superclasses of `foaf:Person`, but any vocabulary can declare local terms to be subclasses thereof. We refer the interested reader to [9] for details.

We analysed the competency of our approach wrt. the usage of RDFS and OWL in prominent Web vocabularies. We found that our scalable subset supported 99.3% of the terminological axioms in our corpus which would be supported by full OWL 2 RL/RDF; excluding non-authoritative axioms, we would support 81.7% of axioms (one document gave 13.4 pp of the non-authoritative axioms), and *fully* support 90.6% of all vocabularies. We also applied a PageRank algorithm over the documents in our corpus, where the summation of the ranks of vocabulary documents *fully* supported by our A-linear rules was 77% of the total, and the analagous percentage for authoritative reasoning over these rules was 70.3% of the total. We demonstrated that for memberships of the most common classes and properties, applying standard (non-authoritative) reasoning for our scalable subset of OWL 2 RL/RDF would increase materialised triples by an approx. factor of 55.46×, or 12.74× excluding inferences involving core terms like `rdf:type`, `rdfs:Resource`, `owl:Thing` (which are commonly "redefined").

Finally, using 9 machines (2.2GHz, 4GB ram) we ran authoritative reasoning for our OWL 2 RL/RDF subset over our 1.1 g quads Linked Data corpus : in 3.35 h, we derived 1.58 g raw inferred triples, of which 962 m were novel and unique.

[1] `http://www.eiao.net/rdf/1.0/`

In terms of lessons learnt, applying standard OWL 2 RL/RDF materialisation over large-scale Linked Data is impractical, but with (i) our carefully selected subset, (ii) optimisations based on seperating out terminological data; and (iii) the inclusion of authoritative reasoning, we can perform a cautious, distributed form of materialisation which has good competency with respect to popular Linked Data vocabularies, and which roughly doubles input data size (thus not overly-burdening consumers).

3 Annotated Reasoning and Inconsistency Repair

Our next use-case was detecting and repairing inconsistency in the closed corpus, where OWL 2 RL/RDF contains various constraint rules. First, using PageRank scores of documents in our corpus, we rank input triples as the sum of the ranks of documents in which they appear. We then propose a formal annotation framework which propagates ranks to inferred triples based on the ranks of the triples and rules involved in their proof. Enabling scale, we propose a straightforward aggregation: the rank of an inference is given as the minimum rank of the triples involved in its proof. This simple aggregation avoids introducing new annotation terms, thus ensuring *decidability*.

Thereafter, we use the constraint rules to extract inconsistencies—sets of triples which represent a contradiction—from the merge of the ranked input and inferred data. We then propose a straightforward, scalable diagnosis method—deriving a "parsimonious" set of triples which, when removed, will restore consistency—which (i) iterates over inconsistent sets in decreasing order of minimum ranked triple; (ii) collects minimum ranked triples from each inconsistency which has not already been diagnosed; (iii) determines triples which *require* triples in the diagnosis to be inferred, appending them to the diagnosis. Triples in the diagnosis are then removed to repair the corpus.

We again apply our methods in a distributed setting. Applying annotated reasoning over nine machines took 14.6 h: $\sim$4.4$\times$ longer than the baseline where more duplicates are inferred, and non-optimal triples must be removed in a batch-sort post-processing step. Detecting and extracting all inconsistencies took 2.9 h, finding 301 k unique inconsistencies: 97.7% of these were invalid datatype literals, and the remaining 2.4% were disjoint classes, mostly from FOAF. Generating and applying the repair took 2.82 h, removing 418 k triples (0.02% of the closed corpus).

In terms of lessons learnt, our main observation is that—with the trivial exception of ill-typed literals—there is not much formal inconsistency in Linked Data, primarily due to a lack of axiomatisation of constraints on a vocabulary level, in turn possibly also due to the open-world nature of OWL. This makes it difficult to detect (and thus repair) noise and modelling errors in the merged corpus (cf. [12]).

4 Handling Coreference Identifiers

We have thus far excluded equality reasoning involving `owl:sameAs` since the standard rules are quadratic. The last use-case we investigated was handling coreferent assertional identifiers. We investigated two approaches: (i) a baseline approach using explicit `owl:sameAs` relations; (ii) an extended approach which also considers inferable `owl:sameAs` relations. Instead of applying quadratic *replacement*, we *canonicalise* (aka.

consolidate) the data by choosing one canonical identifier from each coreferent set and rewriting the data according to the chosen identifiers. (We preserve all non-canonical URIs in the output by means of an `owl:sameAs` link.)

For the baseline approach, we extract the explicit `owl:sameAs` data from the corpus in parallel, load them into memory, and replicate them across all machines; thereafter, we apply canonicalisation of the corpus by means of a single scan. Applying this method over the input corpus took 1.05 h and extracted 11.93 m `owl:sameAs` quadruples (only 3.8 m unique triples), forming 2.16 m coreference sets mentioning 5.75 m terms (6.24% of all URIs)—an average of 2.65 elements per set. Of the 5.75 m terms, only 4,156 were blank-nodes. The largest set contained 8,481 terms, but was (manually) deemed to be incorrect due to over-use of `owl:sameAs` for linking drug-related entities in the DailyMed and LinkedCT exporters; however, we sampled 100 sets and manually verified that all were correct (see [9] for more details).

We then extended the approach to consider `owl:sameAs` inferable through inverse-functional and functional properties, and cardinalities; note that we found no inferences through the latter, and that we had to "blacklist" various void values for inverse-functional properties found in the data [9]. Using this approach, the `owl:sameAs` data could no longer fit in memory, where we instead used on-disk batch processing techniques (e.g., sort-merge-joins). The extended approach took 12.34 h on nine machines, and found 2.82 m equivalence classes (an increase of 1.31× the baseline) mentioning a total of 14.86 m terms (an increase of 2.58× from baseline; 5.77% of all URIs and blank-nodes), of which 9.03 million were blank-nodes (an increase of 2173× from baseline; 5.46% of all blank-nodes) and 5.83 million were URIs (an increase of 1.014× baseline; 6.33% of all URIs).

Finally, we also investigated some initial probabilistic approaches to find new coreference, as well as methods for detecting and repairing incorrect coreferences. These approaches had mixed results for our data; see [9] for details.

In terms of lessons learnt, compared to considering only explicit `owl:sameAs` relations in Linked Data, vastly more coreferent blank-nodes but very few novel coreferent URIs are found through inverse-functional and functional properties. Many of the additional inferences come from FOAF data exported by blogging platforms like `hi5.com` which identify users with blank-nodes and legacy inverse-functional property values (not URIs). Cardinality-based reasoning found no new coreference. We also found many examples of erroneous linking, esp. for inferred `owl:sameAs`, where the semantics of (inverse-)functional properties are not respected on the Web; explicit `owl:sameAs` relations were of higher quality, but also not entirely accurate (cf. [7]).

5 Conclusion

We argue that many consumers of Linked Data (will) benefit from lightweight reasoning, where we presented and investigated three particular use-cases involving OWL 2 RL/RDF. Although issues relating to *scale* are now being tackled in the literature [5, 10, 17, 16, 15, 2, 13], few approaches discuss *robustness* or applicability for open Linked Data [10, 5]. In this thesis, we designed and investigated a variety of reasoning methods for a Linked Data corpus of 1.1 g quadruples crawled from 4 m Web documents.

The high-level lessons learnt are: (i) various trade-offs are necessary to enable (incomplete) reasoning over such data, but the resulting profile still has good competency wrt. popular vocabularies; (ii) separating terminological data enables efficient distribution and further optimisation; (iii) cautious consideration of the source of data (authoritative reasoning) is needed to ensure reasonable materialisation sizes; (iv) little of the noise inherent in Linked Data is symptomised as formal inconsistency; (v) explicit `owl:sameAs` *relations give 99% of coreferent URIs possible through OWL 2 RL/RDF rules, whereas inferred* `owl:sameAs` *relations mainly identify coreferent blank-nodes.*

Acknowledgements: I would like to thank to my supervisor Axel Polleres and my examiners Stefan Decker and James Hendler for their support. This work has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Líon-2) and by an IRCSET scholarship.

References

1. S. Auer and J. Lehmann. Creating knowledge out of interlinked data. *Semantic Web*, 1(1-2):97–104, 2010.
2. B. Bishop, A. Kiryakov, D. Ognyanoff, I. Peikov, Z. Tashev, and R. Velkov. OWLIM: A family of scalable semantic repositories. *Sem. Web J. (to appear)*, 2011.
3. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
4. G. Cheng, W. Ge, H. Wu, and Y. Qu. Searching Semantic Web Objects Based on Class Hierarchies. In *Proceedings of Linked Data on the Web Workshop*, 2008.
5. R. Delbru, A. Polleres, G. Tummarello, and S. Decker. Context Dependent Reasoning for Semantic Documents in Sindice. In *Proc. of 4th SSWS Workshop*, 2008.
6. B. C. Grau, B. Motik, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language: Profiles. W3C Recommendation, Oct. 2009. http://www.w3.org/TR/owl2-profiles/.
7. H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson. When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In *ISWC*, 2010.
8. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space (1st Edition)*. Morgan & Claypool, 2011.
9. A. Hogan. *Exploiting RDFS and OWL for Integrating Heterogeneous, Large-Scale, Linked Data Corpora*. PhD thesis, Digital Enterprise Research Institute, National University of Ireland, Galway, 2011. Available from `http://aidanhogan.com/docs/thesis/`.
10. A. Hogan, A. Harth, and A. Polleres. Scalable Authoritative OWL Reasoning for the Web. *Int. J. Semantic Web Inf. Syst.*, 5(2), 2009.
11. A. Hogan, J. Z. Pan, A. Polleres, and S. Decker. SAOR: Template Rule Optimisations for Distributed Reasoning over 1 Billion Linked Data Triples. In *ISWC*, 2010.
12. P. Jain, P. Hitzler, P. Z. Yeh, K. Verma, and A. P. Sheth. Linked Data is Merely More Data. In *AAAI Spring Symposium "Linked Data Meets Artificial Intelligence"*, Mar. 2010.
13. V. Kolovski, Z. Wu, and G. Eadon. Optimizing Enterprise-scale OWL 2 RL Reasoning in a Relational Database System. In *ISWC*, 2010.
14. G. Meditskos and N. Bassiliades. DLEJena: A practical forward-chaining OWL 2 RL reasoner combining Jena and Pellet. *J. Web Sem.*, 8(1):89–94, 2010.
15. J. Urbani, S. Kotoulas, J. Maassen, F. van Harmelen, and H. E. Bal. OWL Reasoning with WebPIE: Calculating the Closure of 100 Billion Triples. In *ESWC (1)*, pages 213–227, 2010.
16. J. Urbani, S. Kotoulas, E. Oren, and F. van Harmelen. Scalable Distributed Reasoning Using MapReduce. In *ISWC*, pages 634–649, 2009.
17. J. Weaver and J. A. Hendler. Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples. In *ISWC*, pages 682–697, 2009.