Scalable Authoritative OWL Reasoning on a Billion Triples

Aidan Hogan, Andreas Harth, and Axel Polleres

Digital Enterprise Research Institute, National University of Ireland, Galway

Abstract. In this paper we present a scalable algorithm for performing a subset of OWL reasoning over web data using a rule-based approach to forward-chaining; in particular, we identify the problem of ontology hijacking: new ontologies published on the Web re-defining the semantics of existing concepts resident in other ontologies. Our solution introduces consideration of authoritative sources. We present the results of applying our methods on a re-crawl of the billion triple challenge dataset.

1 Introduction

Reasoning over aggregated Web data is useful, for example: to infer new assertions using terminological knowledge from ontologies and therefore provide a more complete data set; to unite fractured knowledge about individuals collected from disparate sources; and to execute mappings between domain descriptions and therefore provide translations from one conceptual model to another. Our work on reasoning is motivated by the requirements of the Semantic Web Search Engine (SWSE) project[1], within which we strive to offer search, querying and browsing over the Semantic Web.

Reasoning on Web data poses a number of requirements:

- the system has to perform on web-scale, with implications on the completeness of the reasoning procedure, algorithms and optimisations
- the method has to perform on collaboratively created knowledge bases, which has implications on trust and the privileges of data publishers
- the web search scenario requires sub-second response times, which has implications on the reasoning and query processing strategy

In [5], we presented SAOR – Scalable Authoritative OWL Reasoner – which focuses on performing best-effort RDFS and OWL reasoning on Web data. SAOR is designed to accept as input a web knowledge-base in the form of a body of statements as produced by a web-crawl and to output by forward-chaining a knowledge-base enhanced by a given fragment of OWL reasoning. In [5], we presented some initial evaluation of rules which exhibited linear scale. For the Billion Triple Challenge we now apply this subset of OWL and conduct forward-chaining materialiation and present our results.

[1] http://swse.deri.org/

2 SAOR: Inferencing for the Web

Complete inference at the instance level is neither feasible nor desirable: firstly, for the computational infeasibility of complete OWL Full reasoning, and secondly, since we do not deem the explosive nature of contradiction in classical logics desirable in a Web reasoning scenario. Thus, rather than striving for complete inference, we adopt a "best effort" reasoning strategy, optimising inference based on the following principles:

1. We assume a separation of T-Box and A-Box.
2. We trade completeness for implementational feasibility following a rule-based, finite, forward-chaining approach to OWL inference.
3. We trade completeness for producing a much smaller subset of inferred statements; i.e, we deliberately ignore (i) the explosive behaviour of classical inconsistency, (ii) arguably "void" statements in terms of non-standard use of the RDF(S) and OWL vocabularies, (iii) non-authoritative T-Box statements.

2.1 Separating A-Box from T-Box

In SAOR, we strictly separate terminological knowledge from assertional data according to their use of the RDF(S) and OWL vocabulary; we call these the "A-Box" and the "T-Box" respectively (loosely borrowing Description Logics terminology).

Table 2 provides a list of graph patterns (inspired by N3's[1, 2] syntax) in RDF graphs we consider to be part of the T-Box wherein T-Box patterns are idenitifed by underlining. Note that when retrieving graphs from the Web, the instances of these patterns are all of the T-Box statements we consider in our reasoning process: triples that do not match one of these patterns are not considered being part of the T-Box, but are treated purely as assertional "data" triples.

The materialisation of axiomatic statements and completing the entire T-Box may create a bulk of statements with little practical utility. In fact, we deliberately accept the omission of T-Box inference rather as an optimisation: we focus on answering queries over A-Box data rather than, e.g., inferring all members of `:Class`.

SAOR does not support meta-modelling [6], except by conceptually separating the instance- class- or property-meanings of a resource: by separating the T-Box and A-Box segment of the knowledge base, we do not support possible entailments from the simultaneous description of both a class and an instance. Particularly, we treat URIs in the context they appear, in the spirit of "punning";[2] e.g., we do not carry over `:sameAs` inferences to the T-Box. This is in-line with first-order-logic point of view, where equalities do not affect predicates.

We filter out further triples when extracting the T-Box; namely, we ignore non-standard use of RDF in our reasoning efforts. Non-standard use of RDF

[2] `http://www.w3.org/2007/OWL/wiki/Punning`

briefly equates to the use of properties and classes which make up the RDF(S) vocabulary in locations where they have not been been intended, cf. [3, 7].

Our reasoning shall be tailored towards A-Box inferences. Along these lines, certain "inflationary" A-Box statements involving the RDFS vocabulary are not of interest for our inferences. In the quest for scalability, we are not interested in inferring/explicitly storing "quasi-axiomatic" RDF statements such as r `a rdfs:Resource.`, p `a rdf:Property`, etc. which essentially hold for any URI r, class c and property p mentioned in a graph, but are not appearing in the prerequisites of any of our inference rules.

2.2 Rule-based OWL Reasoning

Reasoning in SAOR is inspired by previous approaches, particularly the pD* fragment defined by ter Horst [8], to cover large parts of OWL by positive inference rules which can be implemented in a forward-chaining engine. In this paper, we only present and execute the rules which we found to be linearly scalable in [5].

Although certain triples matched in the inference rule bodies come from the T-Box and others come from the A-Box, inferences are reflected in the A-Box only. Thus, on exhaustive application of the rules, the T-Box remains unchanged.

2.3 Authoritatively Reasoning against Ontology Hijacking

SAOR is also designed to counter-act a behaviour we discovered from initial evaluations which we term ontology hijacking. We counter such non-authoritative extensions of ontologies by ignoring possibly problematic statements during the T-Box generation.

Before formally defining ontology hijacking, let us give some preliminary definitions:

Definition 1 (Authoritative Source).
A graph $s \in \mathcal{KB}$ speaks authoritatively about a concept $c \in \mathcal{C}_{\mathcal{KB}} \cup \mathcal{P}_{\mathcal{KB}}$ if c appears in a triple t of s and one of the following holds true:

1. *c is not identified by a URI (i.e., identified by a blank node)*
2. *s is retrievable from a URI which coincides with (or redirects to) the namespace[4] of the URI identifying c.*

Firstly, all sources are authoritative for anonymous classes or properties defined in that source. The second condition is designed to support best practices as currently adopted by web ontology publishers[5].

[4] Here, slightly abusing XML terminology by "namespace" of a URI we mean the prefix of the URI obtained from stripping off the final NCname

[5] See Appendix A&B of http://www.w3.org/TR/swbp-vocab-pub/

#	DL Syntax	Rule
		$\mathcal{G}0$ **: NO A-BOX PATTERNS IN ANTECEDENT**
00	$\{o_i....o_n\}$	**?C** :oneOf (?o$_1$ ... ?o$_n$) . ⇒ ?o$_1$ ... ?o$_n$ a ?C .
		$\mathcal{G}1$ **: ONE A-BOX PATTERN IN ANTECEDENT**
01	$C \sqsubseteq D$	**?C** rdfs:subClassOf ?D . ?s a ?C . ⇒ ?s a ?D .
02_b	$C \equiv D$	**?C** :equivalentClass ?D . ?s a ?C . ⇒ ?s a ?D .
02_b		?C :equivalentClass **?D** . ?s a ?D . ⇒ ?s a ?C .
03	$P \sqsubseteq Q$	**?P** rdfs:subPropertyOf ?Q . ?s ?P ?o . ⇒ ?s ?Q ?o .
04_a	$P \equiv Q$	**?P** :equivalentProperty ?Q . ?s ?P ?o . ⇒ ?s ?Q ?o .
04_b		?P :equivalentProperty **?Q** . ?s ?Q ?o . ⇒ ?s ?P ?o .
05_a	$P \equiv P_0^-$	**?P** :inverseOf ?Q . ?s ?P ?o . ⇒ ?o ?Q ?s .
05_b		?P :inverseOf **?Q** . ?s ?Q ?o . ⇒ ?o ?P ?s .
06	$\top \sqsubseteq \forall P^-.C$	**?P** rdfs:domain ?C . ?s ?P ?o . ⇒ ?s a ?C .
07	$\top \sqsubseteq \forall P.C$	**?P** rdfs:range ?C . ?s ?P ?o . ⇒ ?o a ?C .
08	$P \equiv P^-$	**?P** a :SymmetricProperty . ?s ?P ?o . ⇒ ?o ?P ?s .
09_a	$\exists P.x$	**?C** :hasValue ?x; :onProperty **?P** . ?y ?P ?x . ⇒ ?y a ?C .
09_b		**?C** :hasValue ?x; :onProperty ?P . ?y a ?C . ⇒ ?y ?P ?x .
10	$C_1 \sqcup ... \sqcup C_n$	**?C** :unionOf (?C$_1$...**?C$_i$**...?C$_n$) . ?x a ?C$_i{}^3$. ⇒ ?x a ?C .
11	$(\geq 1P)$	**?C** :minCardinality 1; :onProperty **?P** . ?x ?P ?y . ⇒ ?x a ?C .
12	$C_1 \sqcap ... \sqcap C_n$	**?C** :intersectionOf (?C$_1$ ... ?C$_n$) . ?y a ?C . ⇒ ?y a ?C$_1$, ..., ?C$_n$.

Table 1: Supported rules with N3-style syntax used for triple patterns. Patterns found in the T-Box are underlined whereas A-Box statements are not; further, rules are grouped according to T-Box/A-Box segments of antecedents. The source of a matching T-Box pattern must speak authoritatively for boldface variable bindings for the rule to fire. Where italicised elements exist, only one of the bindings must be authoritative.

Definition 2 (Ontology Hijacking). *Let $s \in \mathcal{KB} = (\mathcal{T}, \mathcal{A})$ and $\mathcal{KB}' = (\mathcal{T}', \mathcal{A}') = \mathcal{KB} \setminus \{s\}$ be the knowledge base constructed from all graphs in $\mathcal{KB}$ except s. By* Ontology Hijacking *we now mean that a source s speaks non-authoritatively about a concept $c \in \mathcal{C}_{\mathcal{KB}'} \cup \mathcal{P}_{\mathcal{KB}'}$ (i.e., where c appears in $\mathcal{T}'$), in such a way that $Cl_{\mathcal{T}}(\mathcal{A}') \neq Cl_{\mathcal{T}'}(\mathcal{A}')$.*

Ontology hijacking is the re-definition or extension of a definition of a legacy concept (class or property) in a non-authoritative source such that performing reasoning on legacy A-Box data results in a change in inferencing. One particular method of ontology hijacking is defining new super-concepts of legacy concepts. As a concrete example, if one were to publish today a property in an ontology (in a non-authoritative location for FOAF), `my:name`, within which the following was stated: `foaf:name rdfs:subClassOf my:name .`, that person would be hijacking the `foaf:name` property and effecting the translation of all `foaf:name` statements in the web knowledge base into `my:name` statements as well.

#	DL Syntax	Rule	# Inferred
		$\mathcal{G}0$ **: NO A-BOX PATTERNS IN ANTECEDENT**	
00	$\{o_i ... o_n\}$	**?C** :oneOf (?o$_1$ ... ?o$_n$) . ⇒ ?o$_1$ ... ?o$_n$ a ?C .	35,161
		$\mathcal{G}1$ **: ONE A-BOX PATTERN IN ANTECEDENT**	
01	$C \sqsubseteq D$	**?C** rdfs:subClassOf ?D . ?s a ?C . ⇒ ?s a ?D .	1,124,758,631
02$_a$	$C \equiv D$	**?C** :equivalentClass ?D . ?s a ?C . ⇒ ?s a ?D .	8,137,162
02$_b$		?C :equivalentClass **?D** . ?s a ?D . ⇒ ?s a ?C .	90,372
03	$P \sqsubseteq Q$	**?P** rdfs:subPropertyOf ?Q . ?s ?P ?o . ⇒ ?s ?Q ?o .	156,462,399
04$_a$	$P \equiv Q$	**?P** :equivalentProperty ?Q . ?s ?P ?o . ⇒ ?s ?Q ?o .	5,667,464
04$_b$		?P :equivalentProperty **?Q** . ?s ?Q ?o . ⇒ ?s ?P ?o .	6,642
05$_a$	$P \equiv P_0^-$	**?P** :inverseOf ?Q . ?s ?P ?o . ⇒ ?o ?Q ?s .	230,945,040
05$_b$		?P :inverseOf **?Q** . ?s ?Q ?o . ⇒ ?o ?P ?s .	230,941,648
06	$\top \sqsubseteq \forall P^-.C$	**?P** rdfs:domain ?C . ?s ?P ?o . ⇒ ?s a ?C .	588,530,865
07	$\top \sqsubseteq \forall P.C$	**?P** rdfs:range ?C . ?s ?P ?o . ⇒ ?o a ?C .	528,995,909
08	$P \equiv P^-$	**?P** a :SymmetricProperty . ?s ?P ?o . ⇒ ?o ?P ?s .	560,460
09$_a$	$\exists P.x$	**?C** :hasValue ?x; :onProperty **?P** . ?y ?P ?x . ⇒ ?y a ?C .	98,601
09$_b$		**?C** :hasValue ?x; :onProperty ?P . ?y a ?C . ⇒ ?y ?P ?x .	104,780
10	$C_1 \sqcup ... \sqcup C_n$	**?C** :unionOf (?C$_1$...**?C**$_i$...?C$_n$) . ?x a ?C$_i$. ⇒ ?x a ?C .	81,736,234
11	$(\geq 1P)$	**?C** :minCardinality 1; :onProperty **?P** . ?x ?P ?y . ⇒ ?x a ?C .	65,283,322
12$_a$	$C_1 \sqcap ... \sqcap C_n$	**?C** :intersectionOf (?C$_1$ ... ?C$_n$) . ?y a ?C . ⇒ ?y a ?C$_1$, ..., ?C$_n$.	115,383
12$_b$	$C_1 \sqcap ... \sqcap C_n$	**?C** :intersectionOf (**?C$_1$**) . ?y a ?C$_1$. ⇒ ?y a ?C .	42

Table 2: SAOR Scan-Rules: (T-Box patterns, **Authoritative**).

Ontology hijacking is problematic in that it vastly increases the amount of statements that are materialised and can potentially harm inferencing on data contributed by other parties.

Following from the definition of ontology hijacking, which we wish to counter in SAOR, we place authoritative restrictions on T-Box patterns in the antecedents of our supported rules. For each rule, at least one of the concepts matched by the A-Box segment of the antecedent must be authoritatively spoken for in the T-Box segment. This ensures that a rule cannot fire using T-Box axiom(s) which speak entirely non-authoritatively for the given A-Box membership assertions.

Table 2 identifies the authoritative restrictions wherein the underlined T-Box pattern is matched by a set of triples from a source *s* iff **both** of the following hold true:

- *s* speaks authoritatively for all concepts matching a boldface variable in Table 2.
- *s* speaks authoritatively for at least one concept matching an italicised variable in Table 2.

For further discussion, we again refer the reader to [5].

3 Reasoning Algorithm

In the following we first present observations on web data that influenced the design of the algorithm, then give an overview of the algorithm, and next discuss details of how we handle T-Box information and perform statement-wise reasoning.

The design of our algorithm is motivated by observations on our Web dataset:

1. Reasoning accesses a large slice of data in the index: around 45% of statements produced uniquely inferred statements.
2. Relative to A-Box (instance) data, the volume of T-Box (structural) data on the Web is small: only around 0.7% of statements were classifiable as T-Box statements.
3. T-Box data is the most frequently accessed segment of data for reasoning:.

Following from the first observation, we employ a file-scan approach which is more efficient in this scenario than query processing lookups. Thus, we avoid the overhead of indexing the data and running full query processing; also we avoid probing the same statements repeatedly for different rules at the low cost of scanning a given percentage of statements not useful for reasoning.

Following from the second and third observations, we optimise by placing T-Box data in a separate data structure accessible by the reasoning engine. Currently, we hold the entire T-Box data in-memory, but the algorithm can be generalised to provide for an on-disk structure or a distributed in-memory structure as needs require.

3.1 Algorithm Overview

The algorithm involves two scans over the data as follows:

1. SCAN 1: separate T-Box information and build in-memory representation
2. PRE SCAN 2: execute A-Box rules with only T-Box patterns in the antecedent ($\mathcal{G}0$)
3. SCAN 2: perform reasoning in a statement-wise manner:
   - Execute rules in $\mathcal{G}1$: join A-Box pattern with in-memory T-Box; recursively execute steps over inferred statements; write inferred statements to intermediate inferred statement output file.

3.2 Handling Structural Data

In the following, we describe how to separate the T-Box data and how to create the data structures for representing the T-Box.

T-Box data from RDFS and OWL specifications can be acquired either from conventional crawling techniques, or by accessing the locations pointed to by the dereferenced URIs of classes and properties in the instance data. We assume for brevity that all the pertinent structural data has already been collected and exists in the input data. if T-Box data is sourced via different means we can build an in-memory representation directly, without requiring the first scan of the entire input data.

Firstly, we separate all possible T-Box statements from the main bulk of input data. We next apply authoritative analysis to the T-Box data and load the results into our in-memory representation. For the in-memory T-Box we employ two separate hashtables, one for classes and another for properties, with the concept identifiers as key and a Java representation of the class or property as value. The property and class objects are designed to contain all of the information required for reasoning on a membership assertion of that property or class: that is, concepts used in the antecedent of a rule are linked to the concepts appearing in the consequent of that rule with the link labelled according to that rule. During reasoning, the property/class identifier used in the membership assertion is sent to the corresponding hashtable and the returned value used for reasoning on that assertion. The in-memory T-Box will remain completely unchanged throughout the whole reasoning process.

3.3 Initial Input Scan

Having loaded the structural data, the SAOR engine is now prepared for reasoning by statement-wise scan of the assertional data.

Directly from the T-Box we can execute Rule 00 which does not require any A-Box data to compute.

To compute the rules in $\mathcal{G}1$, there are two distinct types of statements which require different handling, namely `rdf:type` statements and general non-`rdf:type` statements, reflected by the separation in the T-Box of class and property hashtables. The `rdf:type` statements are subject to class-based entailment reasoning (Rules 1-3 & 10,12), and require joins with class descriptions in the T-Box. The non-`rdf:type` statements are subject to property-based entailments (Rules 4-9,11) and thus requires joins with T-Box property descriptions.

We assume disjointness between the statement categories: we do not allow any external extension of the core `rdf:type` semantics (non-standard use / non-authoritative extension).

The reasoning scan process can be described as recursive depth-first reasoning whereby each unique statement produced is input immediately for reasoning. Statements produced thus far for the original input statement are kept in a set to provide uniqueness testing and avoid cycles; a uniquing function is maintained for a commmon subject group in the data, ensuring that statements are only produced once for that statement group. Once all of the statements produced by a rule have been themselves recursively analysed, the reasoner moves on to analysing the proceeding rule. The process continues until the input data has been exhausted.

4 Evaluation and Discussion

Briefly, in [5], we found that authoritative analysis significantly reduces the number of materialised statements – we took `foaf:Person` as an example and showed that authoritative reasoning produced 64.8x less statements than non-authoritative.

We conducted reasoning on a recrawled dataset provided for the Billion Triple Challenge. We used MultiCrawler [4] to download the data, using a seed set of the de-referenced URIs from the original dataset and the original source URLS. The dataset is derived from 6.5M sources and contains 1.1 billion statements.

Separating, authoritatively analysing and loading the T-Box took 6.47 hours. Scan reasoning took 9.82 hours and produced 1.925 billion newly inferred statements. This dataset is available upon request to the authors. Performance is illustrated in Figure 1 and indicates near-linear scale. Slow-down in input rate is shown to correlate with increased output rate, and vice-versa: we can conclude that system performance is most greatly influenced by read/write disk operations for input/output statements.

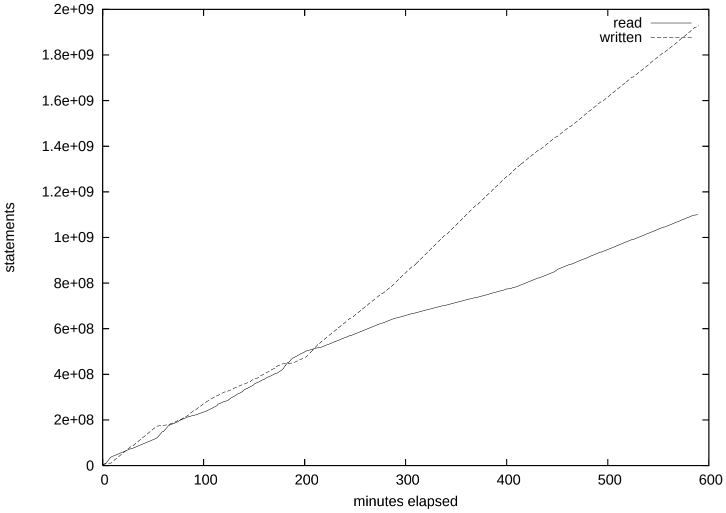

Fig. 1: Performance of the inferencing algorithm.

5 Conclusion

We have presented SAOR: a reasoning methodology for performing reasoning over Web data based on loading the T-Box in-memory and scanning the rest of the data to perform inferencing. To keep the resulting knowledge base manageable, both in size and quality, we made the following modifications to traditional reasoning procedures:

- allow only standard use of RDF and disallow meta-modelling
- allow extension of classes and properties only from authoritative sources (no ontology hijacking)

References

1. T. Berners-Lee and D. Connolly. Notation3 (N3): A readable RDF syntax, Jan. 2008. W3C Team Submission, available at `http://www.w3.org/TeamSubmission/n3/`.
2. T. Berners-Lee, D. Connolly, L. Kagal, Y. Scharf, and J. Hendler. N3Logic: A logical framework for the World Wide Web. *Theory and Practice of Logic Programming*, 8:249–269, 2008.
3. J. d. Bruijn and S. Heymans. Logical foundations of (e)RDF(S): Complexity and reasoning. In *6th International Semantic Web Conference*, number 4825 in LNCS, pages 86–99, Busan, Korea, Nov 2007.
4. A. Harth, J. Umbrich, and S. Decker. Multicrawler: A pipelined architecture for crawling and indexing semantic web data. In *5th International Semantic Web Conference*, pages 258–271, 2006.
5. A. Hogan, A. Harth, and A. Polleres. SAOR: Authoritative Reasoning for the Web. In *Proceedings of the 3rd Asian Semantic Web Conference (ASWC 2008)*, Bankok, Thailand, Dec. 2008. To appear, available at `http://sw.deri.org/~aidanh/docs/aswc08.pdf`.
6. B. Motik. On the properties of metamodeling in OWL. *J. Log. Comput.*, 17(4):617–637, 2007.
7. S. Muñoz, J. Pérez, and C. Gutiérrez. Minimal deductive systems for rdf. In *ESWC*, pages 53–67, 2007.
8. H. J. ter Horst. Completeness, decidability and complexity of entailment for rdf schema ans a semantic extension involving the owl vocabulary. *Journal of Web Semantics*, 3:79–115, 2005.