Finding Unexplainable Triples in an RDF Graph

Jedrzej Potoniec[1][0000−0002−6115−6485]

Faculty of Computing, Poznan University of Technology,
ul. Piotrowo 3, 60-965 Poznan, Poland,
Jedrzej.Potoniec@cs.put.poznan.pl

Abstract. We consider how to select a subgraph of an RDF graph in an ontology learning problem in order to avoid learning redundant axioms. We propose to address this by selecting RDF triples that can not be inferred using a reasoner and we present an algorithm to find them.

Keywords: ontology learning, explanation, RDF, OWL 2 RL

1 Introduction

Consider the following ontology learning problem: given an RDF graph and an ontology describing it, extend the ontology with new axioms, that inductively follow from the data. One of the possible pitfalls of an algorithm solving this problem is generating variants of an axiom already present in the ontology. If such a variant does not deductively follow the ontology, then the user must deal with the resulting redundancy. We postulate that the ontology in the ontology learning problem is a hypothesis, as understood in inductive reasoning, and so we expect it to explain some parts of the graph. It follows that to extend the ontology, one must concentrate on the triples that are not explained. We thus consider the following problem: how to select a subset of triples from the graph that are not explained by the ontology and thus provide new knowledge, which can be generalized and represented as new axioms in the ontology.

The contributions of the paper are as follows: (i) we introduce the notions of unexplained and unexplainable triples; (ii) we propose two algorithms to identify unexplainable triples. We use the following notation conventions: RDF triples are presented in Turtle syntax [2], while OWL axioms are expressed in Manchester syntax [4]. We use $\models$ to denote deductive inference, i.e. $\mathcal{O} \models \{t\}$ means that a triple t deductively follows from an ontology $\mathcal{O}$.

2 Unexplained and unexplainable triples

Consider an RDF graph consisting of the following two triples (expressed in Turtle sytnax): {`:rex a :Dog, :Animal.`} and an ontology consisting of a single axiom {`:Dog` SubClassOf: `:Animal`}. We observe that the triple `:rex a :Animal.` in the sample graph is explained by the ontology, i.e., even if removed from the graph it can be restored using deductive inference. Conversely, the other triple could not be restored if removed, and thus represents new knowledge.

Definition 1. *Given an RDF graph $\mathcal{G}$ and an ontology $\mathcal{O}$, an* unexplained part *$\mathcal{G}_{\overline{\varepsilon}}$ is a subgraph of $\mathcal{G}$ such that: (i) it is sufficient to restore the rest of the graph: $\mathcal{O} \cup \mathcal{G}_{\overline{\varepsilon}} \models \mathcal{G} \backslash \mathcal{G}_{\overline{\varepsilon}}$; (ii) no triple from it can be restored if removed: $\forall t \in \mathcal{G}_{\overline{\varepsilon}}\colon \mathcal{O} \cup \mathcal{G} \backslash \mathcal{G}_{\overline{\varepsilon}} \not\models \{t\}$; (iii) it is subset-minimal, i.e., none of its proper subset has both properties. We call the remaining part of the graph an* explained part *and denote by $\mathcal{G}_{\varepsilon} = \mathcal{G} \backslash \mathcal{G}_{\overline{\varepsilon}}$.*

It is easy to observe that there may be multiple unexplained parts in a single graph. Consider $\mathcal{G} = \{$`:rex a :Dog, :MansBestFriend.`$\}$ and $\mathcal{O} = \{$`:Dog` EQUIVALENTTO: `:MansBestFriend`$\}$. There exists two different unexplained parts: $\mathcal{G}^{(1)}_{\overline{\varepsilon}} = \{$`:rex a :Dog.`$\}$ and $\mathcal{G}^{(2)}_{\overline{\varepsilon}} = \{$`:rex a :MansBestFriend.`$\}$. Due to this, for the ontology learning problem, the usability of a single unexplained part is of limited use. Instead, we propose to consider a set of unexplainable triples, as defined below.

Definition 2. *Given an ontology $\mathcal{O}$ and an RDF graph $\mathcal{G}$, the set of* unexplainable triples *is the intersection of all possible sets of unexplained triples: $\bigcap_i \mathcal{G}^{(i)}_{\overline{\varepsilon}}$, where i iterates over all possible unexplained parts of the graph.*

These triples are the most interesting triples for learning new axioms, as they necessarily contain new knowledge, which is not explained by the ontology.

Theorem 1. *A triple t is unexplainable if, and only if, once removed from a graph it can not be restored using deductive inference.*

$$\forall t \in \mathcal{G}\colon \left(\mathcal{O} \cup \mathcal{G} \backslash \{t\} \not\models \{t\} \iff t \in \bigcap_i \mathcal{G}^{(i)}_{\overline{\varepsilon}} \right)$$

Proof. Assume there exists a triple t such that $\mathcal{O} \cup \mathcal{G} \backslash \{t\} \not\models \{t\}$, but $t \notin \bigcap_i \mathcal{G}^{(i)}_{\overline{\varepsilon}}$. It follows that there exists j such that $t \notin \mathcal{G}^{(j)}_{\overline{\varepsilon}}$ and from Definition 1 we get $\mathcal{O} \cup \mathcal{G}^{(j)}_{\overline{\varepsilon}} \models \{t\}$. As $\mathcal{G}^{(j)}_{\overline{\varepsilon}} \subseteq \mathcal{G} \backslash \{t\}$, from the monotonicity of reasoning, we conclude that $\mathcal{O} \cup \mathcal{G} \backslash \{t\} \models \{t\}$, contradicting the assumption.

Now assume that there exists a triple t such that $t \in \bigcap_i \mathcal{G}^{(i)}_{\overline{\varepsilon}}$, but $\mathcal{O} \cup \mathcal{G} \backslash \{t\} \models \{t\}$. From Definition 1 it follows that there exists j such that $t \in \mathcal{G}^{(j)}_{\varepsilon}$ and $t \notin \mathcal{G}^{(j)}_{\overline{\varepsilon}}$, but this contradicts the assumption.

Using Theorem 1, we can construct a naïve algorithm for computing the set of unexplainable triples by iterating over the graph, and for each triple checking whether the left-hand side of the theorem holds. While correct, such an algorithm is impractical due to its complexity and so we consider a special case of OWL 2 RL to construct a more practical algorithm.

3 Unexplainable triples in OWL 2 RL

Using De Morgan's laws it follows from Theorem 1 that a triple can be restored iff it belongs to at least one set of explained triples: $\forall t \in \mathcal{G}\colon (\mathcal{O} \cup \mathcal{G} \backslash \{t\} \models \{t\}$

$\iff t \in \bigcup_i \mathcal{G}_\varepsilon^{(i)}$) We can use this to construct an algorithm suitable for OWL 2 RL [6]. Assume that $\mathcal{O}$ is a consistent OWL 2 RL ontology and $\mathcal{G}$ is an RDF graph closed w.r.t. logical conclusions following from the ontology and the graph, i.e., there is no such triple t that $t \notin \mathcal{G}$, but $\mathcal{O} \cup \mathcal{G} \models \{t\}$. As deductive inference in OWL 2 RL can be realized using a set of rules, it follows that a triple t can be restored if, and only if, there exists a rule $P \to C$ such that for some assignment σ of RDF nodes to the variables of the rule: (i) all its premises and conclusions are present in the graph: $\sigma(P) \subseteq \mathcal{G}, \sigma(C) \subseteq \mathcal{G}$; (ii) t is in the conclusions: $t \in \sigma(C)$; (iii) t is not in the premises: $t \notin \sigma(P)$. By rewritting all the rules as SPARQL SELECT queries and answering them over the graph, we obtain all the triples that can be restored, and by subtracting them from the graph, we arrive at the set of unexplainable triples.

Consider $\mathcal{G} = \{$`:rex a :Dog.`$\}$ and $\mathcal{O} = \{$`:Dog` SubClassOf: `:Animal`$\}$, and let $\mathcal{G}' = \{t \colon \mathcal{O} \cup \mathcal{G} \models \{t\}\}$ be the graph closed w.r.t. the ontology. Consider the rule *cax-sco*[1]: *If* `T(?c1, rdfs:subClassOf, ?c2)` ***and*** `T(?x, rdf:type, ?c1)` *then* `T(?x, rdf:type, ?c2)`. Each of the literals in the rule corresponds to a SPARQL triple pattern, so the corresponding query can be written as follows:

```
SELECT (?x AS ?subject) (rdf:type AS ?predicate) (?c2 AS ?object)
WHERE {?c1 rdfs:subClassOf ?c2. ?x a ?c1, ?c2. FILTER(?c1!=?c2)}
```

The answer to the query w.r.t. $\mathcal{G}'$ contains the triple `:rex a :Animal.` Should there be no filter clause in the query, it would also contain the triple `:rex a :Dog.`, making the triple self-explanatory.

Some additional consideration must be given to the rules that require checking a variable number of premises, i.e., *prp-spo2*, *prp-key*, *cls-int1*. For them, we must make an assumption about the maximal length of the premises, e.g. by analyzing the axioms in the ontology and generating an appropriate queries in the run-time. Algorithm 1 presents a complete algorithm for computing the set of unexplainable triples. A proof of concept implementation is available at `https://github.com/jpotoniec/UnexplainedTriples`.

4 Related Work

The considered problem is rooted in the research on ontology learning. Multiple setups of the problem were considered, e.g. Lehmann et al. proposed a supervised learning framework DL-Learner and adapted it for ontology engineering [7]; Potoniec et al. developed Swift Linked Data Miner (SLDM) suitable for mining OWL 2 EL class hierarchy from a SPARQL endpoint [8].

The problem at hand is also closely related to the problem of providing a justification for an entailment, i.e. a minimal subset of axioms for the entailment to hold, as considered by Horridge et al. [5]. In this work we are interested rather in detecting the parts of a graph that lack any justification. The problem can be also seen as a ontology modularization/segmentation problem, discussed e.g. by Seidenberg and Rector [9] or d'Aquin et al. [3]. It can also be seen as an inverse of the ontology completion problem, as defined by Baader et al. [1].

[1] The rules and their names from [6], Section 4.3

Algorithm 1: Computing the set of unexplained triples given a OWL 2 RL ontology $\mathcal{O}$ and an RDF graph $\mathcal{G}$ closed w.r.t. the ontology. $Q_{\mathcal{O}}$ is the set of all SPARQL SELECT queries corresponding to the inference rules of OWL 2 RL w.r.t. the ontology $\mathcal{O}$ and $ANS(q, \mathcal{G})$ is the set of answers to the query q w.r.t. the graph $\mathcal{G}$.

$T \leftarrow \emptyset$
forall $q \in Q_{\mathcal{O}}$ **do**
  $T \leftarrow T \cup ANS(q; \mathcal{G})$
end
return $\mathcal{G} \backslash T$

5 Conclusions

In this paper we introduced the notion of explained and unexplained triples, as a problem arising from selecting an appropriate subgraph of an RDF graph for ontology learning. We then used it to construct the set of unexplained triples for a given graph and demonstrated how such a set can be computed. Finally, we provide a proof of concept implementation of the presented algorithm. In the future, we plan to integrate the results with SLDM to measure their impact on the actual learning problem.

Acknowledgement. We acknowledge the support from the grant 09/91/DSPB/0627.

References

1. Baader, F., et al.: Completing description logic knowledge bases using formal concept analysis. In: Veloso, M.M. (ed.) IJCAI 2007, Proc. of the 20th Int. Joint Conf. on AI. pp. 230–235 (2007), http://ijcai.org/Proceedings/07/Papers/035.pdf
2. Carothers, G., Prud'hommeaux, E.: RDF 1.1 turtle. W3C recommendation, W3C (Feb 2014), http://www.w3.org/TR/2014/REC-turtle-20140225/
3. d'Aquin, M., et al.: Modularization: a key for the dynamic selection of relevant knowledge components. In: Haase, P., et al. (eds.) Proc. of WoMO'06. vol. 232. CEUR-WS.org (2006), http://ceur-ws.org/Vol-232/paper2.pdf
4. Horridge, M., Patel-Schneider, P.: OWL 2 web ontology language manchester syntax (second edition). W3C note, W3C (Dec 2012), http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/
5. Horridge, M., et al.: Toward cognitive support for OWL justifications. Knowl.-Based Syst. 53, 66–79 (2013), https://doi.org/10.1016/j.knosys.2013.08.021
6. Horrocks, I., et al.: OWL 2 web ontology language profiles (second edition). W3C recommendation, W3C (Dec 2012), http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/
7. Lehmann, J., et al.: Class expression learning for ontology engineering. J. Web Sem. 9(1), 71–81 (2011), https://doi.org/10.1016/j.websem.2011.01.001
8. Potoniec, J., et al.: Swift Linked Data Miner: Mining OWL 2 EL class expressions directly from online RDF datasets. J. Web Sem. 46, 31–50 (2017), https://doi.org/10.1016/j.websem.2017.08.001
9. Seidenberg, J., Rector, A.L.: Web ontology segmentation: analysis, classification and use. In: Carr, L., et al. (eds.) Proc. WWW 2006. pp. 13–22. ACM (2006), http://doi.acm.org/10.1145/1135777.1135785