A method for the synthesis of an ontology from predefined data

Renato Fabbri[1, a)] and Osvaldo N. Oliveira Jr.[b)]

São Carlos Institute of Physics, University of São Paulo (IFSC/USP)

(Dated: 22 June 2016)

OWL Ontologies are critical tools to describe taxonomies and the structure of knowledge. Most ontologies are created by domain experts even though the data they arrange is often given by a software system. This work presents a method for obtaining an ontology from predefined data. The resulting structure has the simplest form with an accurate support for the undelying information.

Keywords: linked data, semantic web, big data, artificial intelligence

I. INTRODUCTION

Ontologies and data-driven ontologies.

A. Related work

Ontology creation methods and data-driven methods.

II. METHOD

The method consists of probing the ontological structure in data with SPARQL queries and post-processing which can be divided in the following steps:

1. Obtaining all distinct classes with the query:
   `SELECT DISTINCT ?class WHERE { ?s a ?class }`

2. Obtaining all distinct properties with the query:
   `SELECT DISTINCT ?p WHERE { ?s ?p ?o }`

3. For each class, get distinct subject classes and predicates where the the object is an instance of the class:
   `SELECT DISTINCT ?p ?cs WHERE { ?i a <class_uri> . ?s ?p ?i . ?s a ?cs . }`

4. For each class, get distinct predicates and object classes or datatypes where the subject is an instance of such class:
   `SELECT DISTINCT ?p ?co (datatype(?o) as ?do) WHERE { ?i a <class_uri> . ?i ?p ?o . OPTIONAL { ?o a ?co . } }`

5. For each property, check if it is functional, i.e. if it occurs only once with each subject:
   `SELECT DISTINCT (COUNT(?o) as ?co) WHERE { ?s <property_uri> ?o } GROUP BY ?s`

6. For each property, find the incident range and domain with the queries:
   `SELECT DISTINCT ?co (datatype(?o) as ?do) WHERE { ?s <property_uri> ?o . OPTIONAL { ?o a ?co . } }` `SELECT DISTINCT ?cs WHERE { ?s <property_uri> ?o . ?s a ?cs . }`

7. For each instance of each class, get all distinct predicates. For each predicate, check if all instances of the class hold such relationship (existential restriction):
   `SELECT DISTINCT ?p WHERE { ?s a <class_uri>. ?s ?p ?o . }`
   `SELECT DISTINCT ?s WHERE { ?s a <class_uri> }`
   `SELECT DISTINCT ?s ?co (datatype(?o) as ?do) WHERE {?s a <class_uri>. ?s <property_uri> ?o . OPTIONAL {?o a ?co . }}`

8. and if all instances that hold such relationship are instances of the class (universal restriction):
   `SELECT DISTINCT ?s WHERE { ?s <property_uri> ?o . }`

9. Draw each class, each property and the overall figure.

10. Make `rdfs:subClassOf` and `rdfs:subPropertyOf` statements to better organize knowledge and link to third party ontologies and data.

11.

III. RESULTS AND DISCUSSION

Figure

IV. CONCLUSIONS

ACKNOWLEDGMENTS

Renato Fabbri is grateful to CNPq (process: 140860/2013-4, project 870336/1997-5), United Nations Development Program (PNUD/ONU, contract: 2013/000566; project BRA/12/018) and the Postgraduate Committee of the IFSC/USP.

Appendix A: Foo

Bar

[a)] http://ifsc.usp.br/~fabbri/; Electronic mail: fabbri@usp.br
[b)] www.polimeros.ifsc.usp.br/professors/professor.php?id=4; Electronic mail: chu@ifsc.usp.br; Also at IFSC-USP