One Simple Ontology for Linked Data Sets

Lihua Zhao and Ryutaro Ichise

Principles of Informatics Research Division,
National Institute of Informatics, Tokyo, Japan,
{lihua, ichise}@nii.ac.jp

Abstract. The Linking Open Data (LOD) cloud includes over 26 billion RDF triples from various domains. In order to access linked data sets, Semantic Web users have to understand the ontology schema of the data sets. However, understanding all the ontologies used in the LOD cloud is not feasible and is time-consuming. A simple and easily understandable ontology that integrates ontology schema from different data sets is a solution to this problem. This paper proposes an automatic ontology learning method that integrates ontologies from different linked data sets, and presents case studies to show the advantages of our approach.

Keywords: linked data, ontology learning, ontology integration

1 Introduction

The Linking Open Data (LOD) cloud contains 203 data sets with over 26 billion RDF triples [3]. An RDF triple is in the subject-predicate-object form, where Uniform Resource Identifiers (URIs) are used to represent the subject or object [3]. Semantic web developers can query linked data with SPARQL, which is an RDF query language. However, Semantic Web developers have to be familiar with the ontology schema of the data sets. Learning all the ontology schema, which may contain thousands of distinct ontology predicates, is time-consuming and cumbersome. Querying based on one simple ontology that integrates various ontologies can simplify SPARQL queries and help developers of Semantic Web applications to easily understand ontology schema. In order to automatically integrate ontologies, we proposed an ontology learning method that includes ontology manipulations such as ontology term extraction, ontology matching, and ontology integration. The ontology learning method can automate or semi-automate the ontology construction process from structured, semi-structured, or unstructured data [1].

In this paper, we present an automatic ontology learning method that can be applied to linked data sets from diverse domains. An automatically constructed ontology is called a Mid-Ontology, which integrates related ontology predicates from diverse data sets. The Mid-Ontology simplifies SPARQL queries and effectively retrieves information in case studies. Furthermore, by using the Mid-Ontology, we can find missing links by querying with integrated ontology predicates.

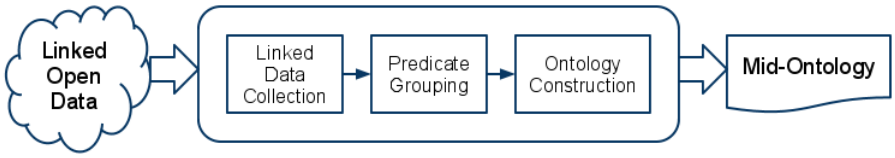

Fig. 1: Architecture of our ontology learning approach.

2 Ontology Learning Approach

In this section, we describe our ontology learning approach, which consists of three phases: linked data collection, predicate grouping, and ontology construction (Fig. 1).

Linked Data Collection
: Since our aim is to construct a simple integrated ontology from linked data sets, we are interested in instances linked with owl:sameAs. We select a core data set that has links to other data sets, and then retrieve instances that have links from or to this core data set. An instance is deleted, if all the triples of the instance are SameAs links, broken links, or types, which may add noise to the collected data. In the final step, we collect all the triples from retrieved linked instances, except triples with the owl:sameAs predicate, because we have already collected all the contents of the linked instances.

Predicate Grouping
: Grouping related predicates from different ontology schema is critical for Mid-Ontology construction, because there exist many similar ontology predicates representing the same thing. The predicate grouping phase involves three main steps: creating initial predicate groups, pruning groups, and refining groups.

  The first step involves creating initial groups of predicates that share the same information or the same object by implementing the exact string matching method on predicates and objects. The second step involves pruning initial groups by knowledge-based similarity matching and string-based similarity matching, which are commonly used to match ontologies at the concept level [2]. Knowledge-based similarity measures [5] are based on WordNet, which is a large lexical database of English, and string-based similarity measures are prefix, suffix, Levenshtein distance, and n-gram, as introduced in [4]. The final group refining step involves splitting predicates according to the relations of rdfs:domain or rdfs:range, because even though the values or terms of predicates are similar, the predicates may belong to different domains or ranges.

Ontology Construction
: A Mid-Ontology is automatically created with selected terms, refined predicate groups, and the mo-prop:hasMembers predicate, which connects our Mid-Ontology classes with groups of predicates. An ontology class is designed

Table 1: Predicates grouped in mo-onto:population.

```
<rdf:Description rdf:about="mid-onto:population">
<mo-prop:hasMembers rdf:resource="http://dbpedia.org/property/population"/>
<mo-prop:hasMembers rdf:resource="http://dbpedia.org/property/popLatest"/>
<mo-prop:hasMembers rdf:resource="http://dbpedia.org/property/populationTotal"/>
<mo-prop:hasMembers rdf:resource="http://dbpedia.org/ontology/populationTotal"/>
<mo-prop:hasMembers rdf:resource="http://dbpedia.org/property/einwohner"/>
<mo-prop:hasMembers rdf:resource="http://www.geonames.org/ontology#population"/>
</rdf:Description>
```

Table 2: SPARQL Example: Find places with a population of more than 10 million.

```
SELECT DISTINCT ?places
WHERE{  mid-onto:population mo-prop:hasMembers ?prop.
        ?places ?prop ?population.   FILTER (xsd:integer(?population) > 10000000). }
```

as mo-onto:$Term$, where "$Term$" is automatically selected by choosing the longest term among the most frequent pre-processed terms from each group of predicates. The pre-process includes tokenization, removal of stop words, and stemming, which are commonly applied in natural language processing (NLP).

3 Case Study

In this section, we show the advantages of our ontology learning approach by presenting case studies conducted on DBpedia, Geonames, and NYTimes data sets, which are from the cross-domain, geographical domain, and media domain, respectively. Since DBpedia is the main cross-domain data set and has links with both Geonames and NYTimes, we select DBpedia as the core data set and implement our ontology learning approach.

As a result, the automatically constructed Mid-ontology contains 28 groups of predicates from these three data sets. Table 1 shows one of the 28 groups in the constructed Mid-Ontology, which integrates predicates that indicate the population from DBpedia and Geonames. This group does not contain any NYTimes predicate, because there is no predicate that indicates the population in NYTimes data. We observed that the instances of NYTimes are linked with instances of other data sets according to the labels of the news heading, which is represented by "http://www.w3.org/2004/02/skos/core#prefLabel". This predicate is included in the group of mo-onto:name, which contains predicates related to names of places, persons, or events.

One advantage of our approach is that we can retrieve related information with the automatically integrated Mid-Ontology. Table 2 shows a SPARQL example in which this mo-onto:population is used to find places that have a population of more than 10 million. This simple SPARQL query automatically queries

with all the predicates listed under mo-onto:population as illustrated by Table 1. We can find 517 places with mo-onto:population, while with the single predicate listed in Table 1, we can find 177, 1, 107, 129, 1, and 244 places. The results queried with mo-onto:population are a combination of the results retrieved with each predicate in that group. Furthermore, it is difficult to manually find all the six predicates that indicate the population in different data sets. As this example shows, our approach simplifies SPARQL queries and returns all the possible results without user interaction. In contrast, querying in which each single predicate has to be found manually is time-consuming.

Another advantage of our approach is that potential missing links with the Mid-Ontology that integrates related ontology predicates can be found. Since our ontology learning approach is processed on a data set extracted with owl:sameAs, we can expect to find missing links that should be linked with owl:sameAs. For instance, if we search for places that have a population of "119549", we can get "http://dbpedia.org/resource/Cyclades" and "http://sws.geonames.org/259819/" which correspond to the same place: "Cyclades". However, there is no owl:sameAs link between these two URIs. Therefore, we can find missing links with our Mid-Ontology if there exist predicates from different domains grouped under the same Mid-Ontology class, such as mo-onto:postalcode and mo-onto:name.

4 Conclusion

In this paper, we proposed an ontology learning approach with linked open data. Our approach can help Semantic Web application developers to build an ontology with linked data sets without learning all the ontology schema. The main procedures of our approach are linked data collection, ontology predicate grouping, and ontology construction. Our approach can automatically extract the most related predicates between linked data sets to construct a Mid-Ontology. Case studies show that with the automatically created Mid-Ontology, we can effectively retrieve potential related information in a simple SPARQL query and can find missing links if predicates from different data sets are integrated in the constructed ontology.

References

1. Drumond, L., Girardi, R.: A survey of ontology learning procedures. In: Proceedings of the 3rd Workshop on Ontologies and their Applications. CEUR Workshop Proceedings, vol. 427 (2008)
2. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Heidelberg (2007)
3. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool (2011)
4. Ichise, R.: An analysis of multiple similarity measures for ontology mapping problem. International Journal of Semantic Computing 4(1), 103–122 (2010)
5. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet::similarity: Measuring the relatedness of concepts. In: Proceedings of the Nineteenth National Conference on Artificial Intelligence. pp. 1024–1025 (2004)