CITY
UNIVERSITY OF LONDON
EST 1894

INM713 Semantic Web Technologies and Knowledge Graphs

Laboratory 8: Ontology Alignment

Ernesto Jiménez-Ruiz

Academic course: 2020-2021
Updated: March 15, 2021

Contents

1 Git Repositories

Support codes for the laboratory sessions are available in *GitHub*. There are two repositories, one in Python and another in Java:

`https://github.com/city-knowledge-graphs`

2 Support Codes and Datasets

The GitHub repositories include support codes for this lab session including the ontologies (owl format) and reference alignments (ttl format). The reference alignment files can also serve as an example about how your computed mapping should look like.

In **Python** we will rely on the Owlready API to load and process the ontologies. In **Java** we will continue with the Jena API. The scripts `AccessEntityLabels.py` and `AccessEntityLabels.java` load an ontology, iterate over the classes and access the lexical information of the classes (*i.e.*, name/labels available as part of the URI or via the `rdfs:label` annotation).

In `lab6` we also saw how to compute lexical similarities among strings, and in `lab2` how to create RDF triples.

The scripts `CompareWithReference.py` and `CompareWithReference.java` compute Precision and Recall given as input the system computed mappings and the reference mappings for the corresponding matching task.[1]

3 Ontology Alignment Evaluation

The Ontology Alignment Evaluation Initiative[2] (OAEI) is an international effort for the systematic evaluation of ontology alignment systems that has been running since 2004. In this lab session we are going to use two datasets from the OAEI.

Conference. This dataset aligns 7 ontologies from the same domain (conference organization): Cmt, ConfTool, Edas, Ekaw, Iasted, Sigkdd, Sofsem.

Anatomy. This evaluation dataset consists of finding an alignment between the Adult Mouse Anatomy (mouse) and a part of the NCI Thesaurus describing the human anatomy (human).

Task 2.1 Design and implement a (simple) lexical matcher that loads two ontologies and finds equivalence correspondences among their entities. Save the alignments as RDF triples in turtle format. For example:

```
cmt:Conference owl:equivalentClass confOf:Conference .
cmt:title owl:equivalentProperty confOf:hasTitle .
```

***Tip:** align classes against classes and properties against properties.*

[1] `https://en.wikipedia.org/wiki/Precision_and_recall`

[2] OAEI: `http://oaei.ontologymatching.org/`

Task 2.2 Apply your algorithm over the following pairs of ontologies from the OAEI's conference track:

- cmt.owl - ekaw.owl
- cmt.owl - confOf.owl
- confOf.owl - ekaw.owl

Give suitable names to the mapping files (*e.g.*, `ernesto-cmt-ekaw.ttl`). Use the provided script and reference alignments to compute the precision and recall of the mappings computed by your system.

Task 2.3 Apply your algorithm over the OAEI's anatomy track and get the precision and recall of your mappings. Note that, unlike in the conference track, the ontologies in the anatomy track are relatively large and contain $2,744$ classes (mouse.owl) and $3,304$ classes (human.owl). If your algorithm computes a pair-wise comparison, this will imply $> 9,000,000$ lexical comparisons.

Task 2.4 (Optional) Submit your mappings in moodle so that I can rank your contributions. You can use a "system name" instead of your name.

Task 2.5 (Optional) Create an algorithm that *(i)* is scalable for large ontologies (*e.g.*, uses a lexical index) and *(ii)* involves not only the labels of the entities but also the structure of the ontology.[3]

[3] LogMap (`https://github.com/ernestojimenezruiz/logmap-matcher`) is a state-of-the-art system that can match the ontologies in the anatomy track in 7s while producing competitive results. Its simple lexical variant LogMapLt takes only 2s. Results: `http://oaei.ontologymatching.org/2020/results/anatomy/`