Semantic Web

Assignment 7

Johannes Härtel Iryna Dubrovska

Institute of Web Science and Technologies
Department of Computer Science
University of Koblenz-Landau

Some of the tasks may require you to do additional research extending the lecture. Please keep the citation rules in mind.

For all the assignment questions that require you to write a code, make sure to include the code in the answer sheet, along with a separate python file. Where screen shots are required, please add them in the answers directly and not as separate files.

Please submit your XML solutions in separate files.

Team Name: XXXX
Team Members: XXXX

1 Ontology Matching

Consider two ontologies O1 and O2 as depicted below:

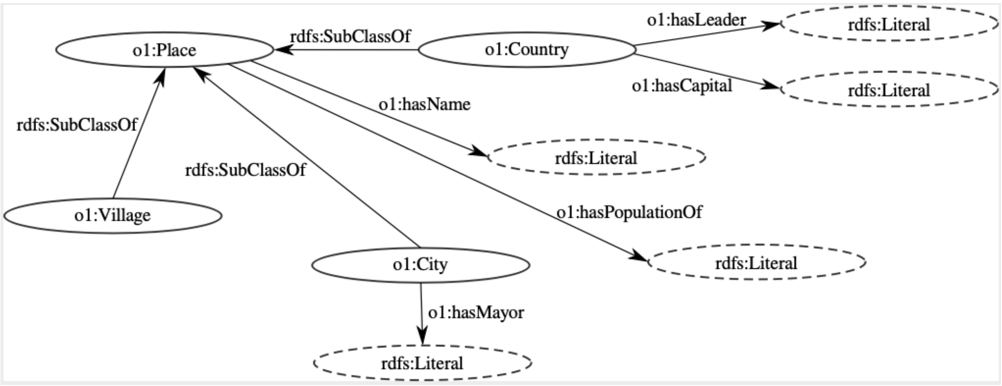

Figure 1: Ontology O1

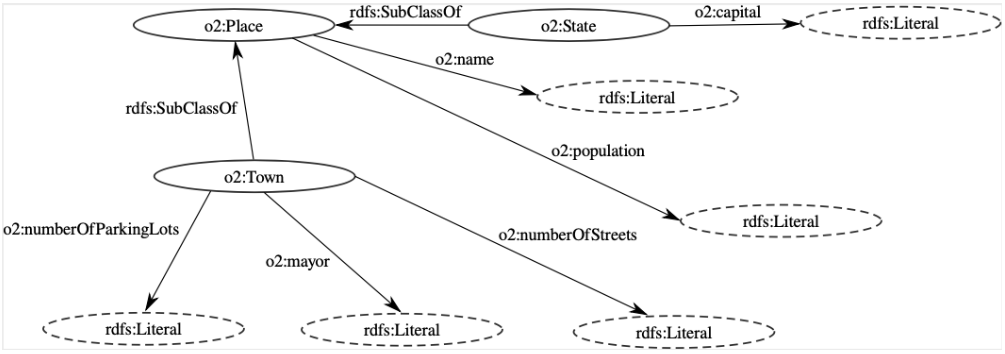

Figure 2: Ontology O2

In this task you shall identify similarities between ontologies O1 and O2. In your analysis, disregard the namespace prefixes. **Give detailed explanations** when answering the following questions:

1.1 What are the Levenshtein distances between all possible pairs of `o1:hasLeader`, `o1:hasName`, and `o1:hasCapital`?

1.2 Calculate the suffix similarity for both `o2:mayor` and `o2:population` to elements in O1. Indicate the pair(s) with the confidence value higher than zero.

1.3 Using graph-based techniques, identify two pairs of entities between non-leaf elements of O1 and O2 that are similar. For each pair $(e1, e2)$ that you provide, explain why they are similar.

2 Alignment

Specify an alignment of the following instances to ones found in the Linked Open Data cloud (e.g. dbpedia.org is a good starting point for this kind of search, as well as wikipedia.org). The alignment should be done, for example, for `http://example.org/places/Berlin` and `http://dbpedia.org/page/Berlin`.
Provide 2 equality-correspondences per instance (a total of 10). You must provide the level of confidence for each correspondence. For this, you can use the following normalized version of the Levenshtein distance. Given two strings $s1$ and $s2$, let N be the size of the longest string, and L be the Levenshtein distance between $s1$ and $s2$. The normalized Levenshtein distance is $(N - L)/N$. You can use the Levenshtein calculator at `http://www.unit-conversion.info/texttools/levenshtein-distance/` to assist with the task.

- http://example.org/places/Berlin
- http://example.org/institutions/Reichstag
- http://example.org/people/JimRakete
- http://example.org/artwork/LaTrahisonDesImages
- http://example.org/places/Myanmar

Important Notes

Submission

- Solutions have to be submitted to the OLAT repository Submission in the respected folder.
- The name of the group and the names of all participating students with matriculation numbers must be listed on each submission.
- Solution format: all solutions as *one* PDF document. Programming code has to be submitted as Python code to the OLAT repository. Upload *all* `.py` files of your program! Use `UTF-8` as the file encoding. *Other encodings will not be taken into account!*
- Check that your code compiles without errors.
- Make sure your code is formatted to be easy to read.
  - Make sure you code has consistent indentation.
  - Make sure you comment and document your code adequately in English.
  - Choose consistent and intuitive names for your identifiers.
- Do *not* use any accents, spaces or special characters in your filenames.

Acknowledgment

This pdfLaTeX template was adapted by Jun Sun and Iryna Dubrovska based on the LuaLaTeX version by Lukas Schmelzeisen.

LaTeX

Use `pdflatex assignment_X.tex` to build your PDF.