Homework 3: Representing knowledge and linking entities
INF 558 BUILDING KNOWLEDGE GRAPH

DUE DATE: Monday, 09/17/2017 @ 11:59pm on Blackboard

Ground Rules

This homework must be done individually. You can ask others for help with the tools, but the submitted homework needs to be your own work.

Summary

In this homework, you will link artists in SAAM museum to the Getty Union List of Artist Names (ULAN), and represent your extracted data in previous homework using RDF.

Task 1 (6 pts)

In this task, you are given a dataset of artists from ULAN (ulan_artists.json) and a list of artist URLs in SAAM (saam_artists.json). Your goal is to match records from these 2 datasets using entity linking methods. This means you need to figure out which pairs of artists in the two datasets are referring to the same artists.

1.1 String similarity (3 pts)

ULAN dataset contains 2 fields: artist name and birth date. Analyze the given data and choose string similarities that you think are appropriate for each field. Explain your choices in the report. Write a program that compute the field similarities between records from 2 datasets.

Note that you can customize string similarity methods or change field values if necessary. For example, you choose Levenshtein for artist last name, which you derive from artist name.

1.2 Entity linking (3 pts)

Design a scoring function to combine your field similarities. Explain your choices of weights in the scoring function in the report. Write a program that predict corresponding ULAN artists of SAAM artists using your scoring function.

Apply your program on the development set (saam_ulan.dev.json) and report its accuracy, which is number of correctly linked artists divided by number of artists. An artist is correctly linked if it links to its corresponding ULAN artist, or NULL if there is no corresponding one.

$$accuracy = \frac{|correctly\ linked\ artists|}{|artists|}$$

Apply your program on the test set (saam.test.json), and export your prediction to an output file (saam_ulan.test.json) with the following format (JSON):

```
{"saam_artist": <artist_url>, "ulan_artist": <ulan_url>}
```

For example:

```
[
  {
    "saam_artist": " https://americanart.si.edu/artist/john-frederick-kensett-2599",
    "ulan_artist": "http://vocab.getty.edu/ulan/500030726"
  },
  {
    "saam_artist": "https://americanart.si.edu/artist/elbridge-kingsley-2530",
    "ulan_artist": "http://vocab.getty.edu/ulan/500030726"
  }
]
```

The ground-truth of this task will not be released. You will be graded based on your performance.

Task 2 (4 pts)

In this task, you will represent your artist and artwork data using RDF. The ontology you will use is schema.org, which can be downloaded from: http://schema.org/docs/developers.html. As this ontology may not include all necessary classes and properties to model your data, you will need to extend the ontology to do the job.

2.1 Define ontologies (2 pts)

Define classes and properties you needed if there is no suitable property in schema.org. For example, you create property <https://schema.org/extent> to represent dimension of an artwork, but you do not need to create <https://schema.org/dateCreated> to represent artwork creation date since it is defined in the ontology.

Requirements: you must create Artwork class as subclass of CreativeWork to model SAAM artworks. Save your ontology extension to extended_schema.ttl (turtle format).

2.2 Convert artist and artwork data to RDF (2 pts)

Write a program that convert your extracted artist and artwork data in homework 2 to RDF triples (turtle format) using schema.org and the extension you defined in task 2.1.

You can validate your ontology and RDF triples using http://rdf-translator.appspot.com/.

Submission Instructions

You must submit the following files/folders in a single .zip archive named Firstname_Lastname_hw3.zip and submit it via Blackboard:

- **Firstname_Lastname_hw3_report.pdf**: A pdf file containing your answers to the Task 1.
- **saam_ulan.test.json:** contains your linkage result in Task 1.2.

- **artist.ttl, artwork.ttl**: RDF triples you created in Task 2.2.
- **extended_schema.ttl**: The ontology extension you defined in Task 2.1
- **source**: This folder includes all the code you wrote to accomplish Task 1 and 2.