Mustafa Bedi Eğilmez
ec14321 - 140637869

ECS735P - The Semantic Web Coursework

1.Introduction

According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". In this coursework, we have required to create an ontology and populate that ontology with external semantic data repositories. This local ontology should answer that one of the repository cannot answer itself alone.

Project Gutenberg (PG) is an open resource for ebooks. However, it is not possible in PG to read the abstracts of the books available. DBPedia (DBP) is crowd sourced structured knowledge base for Wikipedia. I have used those websites' endpoints (PG-endpoint DBP-endpoint) to populate my ontology.

2.Tasks

2.1. OWL ontology

I have used Protege to generate my ontology, namely `gutenbuch.owl`. Protege is a quite powerful tool to generate ontologies because of it's interface. I made a list of the entities (classes and objects) that I am going to generate beforehand. I have added those and defined URIs. Then I have defined object properties and their URIs as well. After his I worked on the SPARQL query on Python, using rdflib module.

2.2.Basic Task

I have selected Franz Kafka and Mark Twain as my samples. I intentionally choose a small dataset, to make it less complicated. I have gathered their works, those available on DBP endpoint(ie book name, publication date, language, page number, abstracts, genre etc.) I mapped these to my existing entities, that I have already created within my ontology.

2.3.Bonus Task

For the bonus task, I have added writers' information available on PG endpoint. From this endpoint I get book title, author name, date added to PG. Limitation here is the semantic entities provided by PG are limited.

2.4.SPARQL Queries

I build my two queries, where user can easily modify and create his/her own query. The queries do two things. First query gathers data collected from DBP. It lists author name, page number, and name of the work. Second query connects two datasets, namely PG and DBP. It gets summary from DBP and author and its related work. After running this query, it is clearly indicated that which data is gathered from which data source.

3.Conclusion & Challenges

Connecting two different data sources helped us to link two different datasets and answered a need that cannot be obtained from both data resources. From PG users can

download complete book, from DBP they can read the summaries of the books before downloading it.

First challenge I have come across is understanding the entities and objects in given datasets. I have made lots of online research and tested some simple queries in order to comprehend the structure of the endpoints.

Another challenge is the datasets are not structured inter exchangeable. For example, famous work of Franz Kafka, *The Metamorphosis* is can be reached at http://dbpedia.org/resource/The_Metamorphosis in DBP where in PG this is http://wifo5-04.informatik.uni-mannheim.de/gutendata/resource/etext5200 . More uniformed datasets will make the data linkage more easy.

4.Running Code Snippets (assuming python is installed to computer)

1. Open a command line tool of your preference (Terminal, PowerShell etc.) and go to directory where you extracted the archive file.
2. Type `>> python dbpedia_hook.py` this will load the data from the both data soruces. You do not have to run other hook. This script calls the other script, follow the responses from the command line. This will result tho more .owl ontologies, similar to the course work sample provided by the lecturer.
3. To run queries, type following to the command line. Output will be visible on command line:

   ```
   >> python dbpedia_query.py
   … you should see results here
   >>python gb_dbpedia_query.py
   ```
4. You can modify `dbpedia_query.py` and `gb_dbpedia_query.py` scripts to make your own queries.