

CS 348 Project 4

Semantic Web

Fall 2019

This project is due on 12/04/2019. Note: There will be a 10% penalty for each late calendar day. After five calendar days, the Project will not be accepted.

1 Introduction

The Semantic Web is an extension of the World Wide Web through standards by the World Wide Web Consortium (W3C). It applies a framework that includes a **data-centric publishing language** such as RDF, OWL or XML that allows meaning and structure (through new data and metadata) to be added to content in a way that is machine readable.

The machine-readable descriptions enable content managers to add meaning to the content, i.e., to describe the structure of the knowledge we have about that content. In this way, a machine can process knowledge itself, instead of text, using processes similar to human deductive reasoning and inference, thereby obtaining more meaningful results and helping computers to perform automated information gathering and research.

In this project, you will be introduced to an **RDF query language**, SPARQL (pronounced "sparkle"), and you will learn how to issue SPARQL queries over an endpoint that serves the DBpedia RDF dataset. An endpoint is essentially a server that exposes a dataset through a public interface/website that can be queried using the SPARQL query language. The endpoint we will be using is for the DBpedia dataset, a rich RDF dataset that represents Wikipedia as RDF entries.

There are many web-based interfaces and cross-platform applications for you to execute SPARQL queries. A good example is [YASGUI](#), which provides syntax checking and highlighting and allows you to access any SPARQL endpoints.

2 Example

Please examine the webpage http://dbpedia.org/page/Purdue_University and get an idea about what a DBpedia entry looks like. Try to run followig query, which retrieves the name, founding date, mascot, number of student, president and state of Purdue Univerity.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX : <http://dbpedia.org/resource/>
SELECT ?name ?foundDate ?mascot ?numOfStudents ?president ?state
WHERE {
    :Purdue_University rdfs:label ?name .
    :Purdue_University dbo:foundingDate ?foundDate .
    :Purdue_University dbo:mascot ?mascot .
    :Purdue_University dbo:numberOfStudents ?numOfStudents .
    :Purdue_University dbo:president ?president .
    :Purdue_University dbo:state ?state .
    FILTER(lang(?name) = 'en')
}
```

A SPARQL query consists a set of triple patterns that correspond to information we are interested about. A triple pattern consists of three main parts, a subject, predicate, and an object. Each part can be either bound (has a value such as :Purdue_University) or unbound (a variable such as ?name). The FILTER operation is also used to present only strings with the English tag (i.e. with the @en tag).

3 Queries

Write queries for the following questions. Your answer should include the prefix for each questions.

3.1

Find city, number of undergraduate students, usnews ranking, and homepage of Purdue University.

Resources:

<http://dbpedia.org/ontology/city>

<http://dbpedia.org/ontology/numberOfUndergraduateStudents>

<http://dbpedia.org/property/usnwrNu>

<http://xmlns.com/foaf/0.1/>

3.2

Find the English name ('en') and number of students of all resources with type University that have student number greater than 35000. Entries should be sorted in descending order by student number and limit your result to 15 entries.

Tips:

1. To find resources with type University, use `rdf:type dbo:University`
2. Syntax for ORDER BY clause is `ORDER BY [DESC] (?name)`

3.3

Find the name and founding date of universities that are located in Indiana state. Sort the universities from oldest to youngest.

Resources:

<http://dbpedia.org/ontology/state>

<http://dbpedia.org/ontology/foundingDate>

3.4

The result from previous question may not contain all the universities in Indiana since some of the entries do not have the property of founding date. Write a query that lists all the universities in Indiana state no matter if they have a founding date. Order the universities from newest to oldest.

3.5

Find the affiliations of Purdue whose name contain "Association".

Resources:
<http://dbpedia.org/ontology/affiliation>

3.6

For all the affiliations of Purdue, find there founding dates and name of leaders if exit.

Resources:
<http://dbpedia.org/ontology/formationYear>
<http://dbpedia.org/property/leaderName>

Tips:
Consider the subquery syntax:

```
...  
WHERE {  
    ...  
    {  
        SELECT [ Vars ] WHERE {  
            [ SubSelect ]  
        }  
    }  
}
```

3.7

Find at least 4 properties from DBpedia entries you like. Use string filter and integer filter in your query.