

# CECS 571 Fundamentals of Semantic Web Technologies

Spring 2020

## Project 3: Semantic Data Retrieval

### Overview

Work in teams to build a search system using SPARQL to retrieve results from the semantic datasets (in RDF or OWL format) you have generated in project 2. The search system may support a limited set of queries (given the scope of your datasets), although it should ideally demonstrate its ability to answer complex questions using semantics. Your system should query RDF/OWL data using SPARQL, and display results in HTML. Your search system may look similar to the one found at <http://dbpedia.org/sparql>

You are encouraged to form new teams of 5 with different members from project 1 and 2, although you can also choose to remain in the same team.

### Deadline

Monday, May 4, 2020, 5pm local time

### An Example

An example of open semantic data is DBpedia's Wikitionary (see <https://wiki.dbpedia.org/wiktionary-rdf-extraction>), which provides a free dictionary for several languages. For example, the English word *dog* has the following data from Wikitionary:

wr:dog	doap:creator	< <a href="http://en.wiktionary.org/w/ind[...]=dog&amp;action=history">http://en.wiktionary.org/w/ind[...]=dog&amp;action=history</a> >
wr:dog	lemon:sense	wr:dog-English-Noun-1
wr:dog	rdfs:label	dog@en
wr:dog	rdfs:seeAlso	< <a href="http://en.wiktionary.org/wiki/dog">http://en.wiktionary.org/wiki/dog</a> >
wr:dog-English-Noun	wt:hasSense	wr:dog-English-Noun-1
wr:dog-English-Noun-1	dc:language	wt:English
wr:dog-English-Noun-1	wt:hasPoS	wt:Noun
wr:dog-English-Noun-1	wt:hasExample	'The `dog` barked all night long.'@en
wr:dog-English-Noun-1	wt:hasMeaning	'An animal, member of the genus `Canis` (probably descended from the common wolf) that has been domesticated for thousands of years; occurs in many breeds. Scientific name: `Canis lupus familiaris`.'@en

A search system querying the above RDF can display, for instance, a definition for the word *dog* and an example usage of this word.

### Requirements

Your code must be hosted on GitHub and is self-contained, i.e. once downloaded, anyone can open and run your program without having to configure any other dependencies. The source code should run without error.

### Submission

Submit two items electronically to the designated DropBox folder on BeachBoard:

- (1) Presentation slides in .pdf or .ppt summarizing how you approached this project. Presentations should be prepared for a 12-minute talk and 3-minute Q&A. In particular, your talk should include the following:
  - a. A reminder of the datasets your system intends to work with
  - b. The kind of questions your search system aims to answer
  - c. A live/recorded demo of your search system
  - d. Technical challenges encountered and how you overcome these issues
- (2) A link to your project's GitHub page in the comment field as you submit your presentation slides.

### **Grading Guidelines**

This assignment is worth 20% of the final grade. All members of the same team will receive the same points, which are subject to peer review (see below). The code/search system and the presentation are marked out of 10% each. You will be graded on the extent to which the required deliverables discussed above have been successfully met.

In particular, the search system will be evaluated on its overall functionality, difficulty of the queries & datasets, and appropriate use of SPARQL. Furthermore, your code will be evaluated against conventions, complexity, reuse and extensibility. Finally, the presentation will be judged on the quality of the discussions, Q&As, time keeping, and the overall clarity.

**Peer review:** contributions in a team-based assignment should be understood as the individual input that is valued by your peers and is advancing the collective team outcome positively. Thus, your final grades may be adjusted to reflect the evaluations rated by your peers. Please be reminded that the instructor cannot give credit to any individual who makes little or no contribution to group-based assignments. Further peer review submission instructions will be provided to the class after the due-date of this assignment.

### **Tips on Earning Points**

Consider the following questions when completing this project:

- Do you have at least two good examples of complex questions that can be answered by your search system?
- Can you demonstrate at least two SPARQL queries that your search system supports?
- Does your system run without error and is self-contained?
- Have you exercised good software management practices such as adding informative comments etc. in your code?
- Can someone not involved in the design or development of your project gain a basic understanding of the various aspects of the semantic queries after your presentation?