Medical Informatics: Lab 2

Working with Ontologies and Semantic Web Data

In this lab you will learn how to use different tools for specifying and querying semantic web data, as well as building and navigating ontologies. To make the most of the lab, please download the zip file Lab2_files.zip (which contains the following files: i) example-00.rq, ii) example-01.rq, iii) example-02.rq, iv) knows.ttl and v) cafes.ttl).

Part 1: SPARQL Querying with Apache Jena ARQ

Downloading the tool and setting up the environment

1. To download Apache Jena ARQ:
    a. Go to https://archive.apache.org/dist/jena/binaries/ and download the 3.12.0 binary distribution of Apache Jena (apache-jena-3.12.0.zip for Windows or apache-jena-3.12.0.tar.gz for Mac OS).
    b. Unzip the file downloaded in a folder of your choice.
        i. In my case, I have extracted the contents of this compressed folder in Documents/tools/jena3, while my lab files are located in Documents/Lab2. I recommend that you do the same.
2. To set up the environment, type the following in the command prompt/terminal.
    a. On Linux / Mac:
    ```
    export JENA_HOME=the_directory_where_you_downloaded_Jena
    export PATH=$PATH:$JENA_HOME/bin
    ```

    b. On Windows:
    ```
    SET JENA_HOME=the_directory_where_you_downloaded_Jena
    SET PATH=%PATH%;%JENA_HOME%\bat
    ```

3. To test that the environment has been set up correctly, type the following in the command prompt/terminal. If this command returns an error then JENA_HOME is not pointed to a valid directory.
    a. On Linux / Mac:
    ```
    cd $JENA_HOME
    ```

    b. On Windows:
    ```
    cd %JENA_HOME%
    ```

4. You should now be able to run the Apache Jena tools from the command line. For example, the first command below prints the versions of Jena and ARQ used in your distribution, while the second command prints information on how to use the ARQ tool.

    a. On Linux / Mac:

```
sparql --version
arq --help
```

    b. On Windows:

```
sparql.bat --version
arq.bat --help
```

5. From the command line, navigate to the folder where you have placed your files for Lab 2.[1]

    a. In my case, I am currently in Documents/tools/jena3 and I have placed my lab files in Documents/Lab2. To navigate to that folder I type:

        i. On Linux / Mac:

```
cd
cd Documents/Lab2
```

        ii. On Windows:

```
cd ..
cd ..
cd Lab2
```

6. The general form of a command for running a SPARQL query is

```
arq --query=<yourQueryFile> --data=<yourDataFile>
 or
arq --query=<yourQueryFile>
 or
arq --query <yourQueryFile>
```

---

[1] For further information on how to use the command prompt in Windows and the terminal in Linux/Mac, please see the following references: http://www.cs.columbia.edu/~sedwards/classes/2017/1102-spring/Command%20Prompt%20Cheatsheet.pdf and http://abacus.gene.ucl.ac.uk/software/CommandLine.MACosx.pdf

Running example-00.rq over data published on the web

1. Open the file example-00.rq in a text editor (e.g. Notepad, TextEdit or Sublime) and inspect its content. Note that the FROM clause specifies that the query should be run against the RDF data to be found at
http://www.homepages.ed.ac.uk/amanatak/medinf/book.ttl

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?title ?creator
FROM <http://www.homepages.ed.ac.uk/amanatak/medinf/book.ttl>
WHERE {
        dbpedia:Logicomix dc:title ?title .
        dbpedia:Logicomix dc:creator ?creator .
        }
```

2. Access the dataset at http://www.homepages.ed.ac.uk/amanatak/medinf/book.ttl and inspect its content.
3. To run the query, type the following in the command prompt/terminal:
   `arq --query=example-00.rq`

Running example-01.rq over a single local data file

1. Open the file example-01.rq in a text editor (e.g. Notepad, TextEdit or Sublime) and inspect its content.

```
PREFIX : <http://inf.ed.ac.uk/ont#>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?title
WHERE
    {
    ?cafe dbp:locatedIn :central .
    ?cafe dc:title ?title .
    }
```

2. Open the dataset cafes.ttl in a text editor (e.g. Notepad, TextEdit or Sublime) and inspect its content.
3. To run the query example-01.rq over the data in cafes.ttl, type the following in the command prompt/terminal:
   `arq --query=example-01.rq --data=cafes.ttl`
4. Try modifying the cafes.ttl file to add another café located centrally, and rerun the above command.

Running example-02.rq over multiple local data files

1. Open the file example-02.rq in a text editor (e.g. Notepad, TextEdit or Sublime) and inspect its content.

```
PREFIX : <http://inf.ed.ac.uk/ont#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?cafe ?x
WHERE
    {
      ?cafe :lovedBy ?x .
      {
       ?x foaf:knows ?y .
      }
      UNION
      {
       ?y foaf:knows ?x .
      }
    }
```

2. Open the datasets cafes.ttl and knows.ttl in a text editor (e.g. Notepad, TextEdit or Sublime) and inspect their content.
3. To run the query example-02.rq over the data in knows.ttl and cafes.ttl, type the following in the command prompt/terminal:
   ```
   arq --query=example-02.rq --data=knows.ttl --data=cafes.ttl
   ```

Part 2: SPARQL Querying online

There are a few general purpose SPARQL engines available online, including:

- http://tw.rpi.edu/endpoint/sparql.html
- http://www.sparql.org/sparql.html

You can run query example-00.rq discussed in Part 1 with the use of these SPARQL engines, where you provide the URI of the dataset in the target graph URI field or in the FROM clause within the query.

Note, however, that these general-purpose SPARQL engines are not always stable and they often tend to crash. So, in general, it is best to rely on Apache Jena ARQ for SPARQL querying.

Part 3: Querying triple stores with SPARQL endpoints

Several triple stores have SPARQL endpoints so you can query them. This includes the following:

- DBpedia: https://dbpedia.org/sparql
- Wikidata: https://query.wikidata.org/
- Scottish Government Statistics: http://statistics.gov.scot/sparql

In this lab, we'll try the following query over DBpedia, so as to find people that have the same nationality/nationalities as Albert Einstein.

```
SELECT DISTINCT ?person ?nationality

WHERE {

<http://dbpedia.org/resource/Albert_Einstein>
  <http://dbpedia.org/ontology/citizenship> ?nationality .

?person <http://dbpedia.org/ontology/citizenship> ?nationality .

} LIMIT 1000
```

Part 4: Ontology authoring and editing with Web Protégé

Protégé[2] and its web-based equivalent Web Protégé are among the most popular tools for developing ontologies. In this lab we'll be using Web Protégé, but it is worth keeping in mind that Protégé has more functionalities.

In this lab we'll be creating a simple ontology that describes people. It consists of:

- Classes: Person, Man, Woman, Gender
- Properties: hasGender, hasBirthYear
- Instances: male, female

1. Go to: https://webprotege.stanford.edu/ and sign up for an account.
2. Click on Create Project and name your project "People".
3. Click on "People" on the list of projects to load your new project.
4. To create a new class, click on its superclass on the left panel (e.g. owl:Thing) and then click on Create.
    a. In this lab we'll create the class Person, which is a subclass of owl:Thing. Within annotation we will add an skos:comment with content "This class specifies people".
    b. We will also create classes Man and Woman as subclasses of Person.
    c. We will also create class Gender as subclass of owl:Thing.
5. To create individuals (i.e. instances of a class), click on Individuals in the menu, choose the class of interest and click on Create within "Individuals by Class".
    a. In this lab, we'll create two instances of the Gender class: Male and Female.
6. To specify properties, click on Properties on the menu, choose owl:topObjectProperty for object properties or owl:topDataProperty for data properties and click Create.
    a. In this lab, we'll create an object property called "hasGender", which has domain Person and range Gender.
    b. We will also create a data property called "hasBirthYear" with domain Person and range xsd:integer. We'll tick the functional checkbox, as this is a single valued property.
7. To define the properties of a particular class, go to Classes on the menu, click on the class of interest and fill in the corresponding information within the Relationships section.
    a. In this lab, we will set Man to have gender male, and Woman to have gender female.

---

[2] https://protege.stanford.edu/