COMP 474/6741 Intelligent Systems (Winter 2022)

Worksheet #4: Knowledge Base Design & Applications

Task 1. Quick refresher: How do you select *all* triples in a graph using SPARQL?

```
SELECT    .     .     .
WHERE {

          .     .     .

}
```

Task 2. Now for something slightly different: Can you write a SPARQL query that selects all the properties that were declared in a graph?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?property
WHERE {

          .     .     .

}
```

This is an example for a query that's useful during knowledge base development, to show or test the metadata of a graph.

⌨ **Task 3.** Find the URI for *Meat Loaf* (the musician, not the dish) in both DBpedia and Wikidata. What's a major difference between the two graphs? And what is the technical reason for it?

....

⌨ **Task 4.** Wikidata also has a public SPARQL query interface, located at https://query.wikidata.org/. You previously found the URI for Concordia in Wikidata. Now, try to write a SPARQL query that returns the *city* (URI, name) for Concordia from Wikidata:

```
SELECT ?city ?cityname
WHERE {

          .     .     .

}
```

Task 5. Create a *competency question* and a corresponding SPARQL query for our FOCU university example:

....

Testing query:

```
SELECT  .      .      .
WHERE {


          .      .      .


}
```

⌨ **Task 6.** An early, well-known commercial service for semantic annotation of textual (mostly news) documents was Thompson Reuter's *OpenCalais*, which has since been spun out and re-branded as *Refinitif Intelligent Tagging*: Try out the online demo at https://permid.org/onecalaisViewer on a document, for example the first part of the Wikipedia article on Concordia. Look at the entities that were detected and go to the "RDF view": what ID is given to Concordia in this knowledge graph?

....

Hint: There is another tool at the top of the page, *Entity Search*, where you can cross-check your entities.

⌨ **Task 7.** Go to the DBpedia *Spotlight* online demo at https://www.dbpedia-spotlight.org/demo/. Try analyzing a test document with some ambiguities, e.g, *"Paris Hilton went to the Hilton in Paris."* Inspect the entities that were linked to DBpedia. Are they correct?

⌨ **Task 8.** Using *Google's Structured Data Testing Tools*,[1] examine your favorite movie on IMDB.[2]

1. Find some vocabulary terms that are used to model the movie information:

   ....

2. Find the corresponding vocabulary definition URL online:

   ....

⌨ **Task 9.** Find an article online, let's say from the always trustworthy *Mtl Blog*.[3] Look at the HTML source in your browser and find the `META` tags. Identify entries used by Facebook's *Open Graph Protocol*.

....

Now try running the same article through W3C's *RDFa 1.1 Distiller and Parser* at https://www.w3.org/2012/pyRdfa/. Choose Turtle format and compare the triples linking the article using Facebook's OGP.

[1] https://search.google.com/structured-data/testing-tool

[2] https://www.imdb.com

[3] https://www.mtlblog.com