COMP 6721 Applied Artificial Intelligence (Fall 2023)

Worksheet #7: Knowledge Graphs & Intelligent Agents, Part I

Your first Knowledge Graph. We start by modeling some university-related knowledge in form of a *graph*: (1) Joe studies at Concordia University; (2) Joe knows Jane; (3) Jane studies at McGill University. Draw the graph:

Graph Updates. Let's add some additional knowledge: (4) Joe's email address is `joe@example.com`; (5) Concordia is located in Montreal; (6) McGill is located in Montreal. Add these to the graph above.

Triples. Graphs can be represented as *triples* (and vice versa), consisting of

`<subject> <predicate> <object>`

Write the triples corresponding to the first three statements above:

1. ....
2. ....
3. ....

More Triples. Here's another triple: (*Joe, is a, Person*). Add it to the graph you drew in the first task.

Wikidata. So far, we defined everything in terms of natural language. That's not very useful for a knowledge base to be used in an intelligent system. Rather than writing *"Concordia"*, we will use a URI that points to a machine-readable description in the RDF *(Resource Description Framework)* format. Using your phone or laptop, look up (our) Concordia University in the open knowledge base *Wikidata* (https://wikidata.org):[1]

- URI: ....

Using URIs. Ok, now let's go back to the graph from the first task. Replace the string "Concordia" with the URI you obtained in the previous step in your graph above. *Note:* To obtain a complete RDF graph, you'd have to continue replacing all subjects, predicates and objects (except literals) with URIs.

More URIs. What about our Joe and Jane? Unlike some famous persons, we will most likely not find them in Wikidata (or any other public knowledge graph, like DBpedia). However, an organization might have them in their own graph (for example, in Facebook, LinkedIn, or Concordia's student database). For this example, we simply name them as `http://example.org/joe#me` (likewise for Jane). How do we model the predicate that Joe *knows* Jane in a machine-readable way? We will discuss the details in the next lecture; for now, simply use the URI `http://xmlns.com/foaf/0.1/knows`. Re-write the triple (2) from the first task using three URIs:

- ....

Observe how an intelligent agent can now autonomously learn more about each part of this triple simply by resolving the URI and reading the RDF data retrieved through it.

Namespaces. Writing these full URIs (technically IRIs) is tiring (and uses up storage space). Using the following prefixes:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX user:  <http://example.org/>
```

re-write the triple from the previous task, this time with namespaces:

- ....

Enriching Graphs. Wikidata contains (among other information) the information which *city* Concordia is located in. Find the *property* information and write it in form of a triple, using the defined prefixes:

```
PREFIX wdt:  <http://www.wikidata.org/prop/direct/>
PREFIX wd:  <http://www.wikidata.org/entity/>
```

- ....

Now add this triple to the graph from the first task.

AI Agent. Given the knowledge graph you constructed above, can an intelligent agent answer the question *"Which city is Joe studying in?"*? If yes, how (explain, step-by-step, using your graph)? If no, why not (what is missing)?

[1] What you see in your browser is actually a human-readable web page obtained through a 303 redirect (from `/entity/...` to `/wiki/...`), not the raw RDF data that would be consumed by a program (`/entity/...`). The returned file format is determined by the extension, e.g., https://www.wikidata.org/wiki/Special:EntityData/Q326342.rdf for raw RDF data.