Web Data Mining and Semantics Project

Note: A Group of **at most 3 students** can work in this project

Part I: Modeling the ontology (5 points)

In this project, we aim to create an ontology, using the Protégé editor, which models movies. A movie has one or several directors, writers and actors. It also has a title, one or several genres, a year, a country and a language. To define the Genre of a movie, possible choices are: Thriller, Crime, Action, Drama or Comedy. Actors, directors and writers are persons. Persons have a gender (male or female), a name, an age and a nationality.

Indications :

1. The range of the *hasActor, hasDirector and hasWriter* properties is Person
2. Define classes that are disjoint, restrictions and conditions on classes if necessary. Example of a restriction: an actor has the restriction: *isActor* of a Movie.
3. Define the types of the properties (transitive, symmetric, inverseOf, etc.) if necessary.
4. While defining a property, define its inverse (*hasActor* and isActorOf) if necessary.
5. While defining a property, define its domain and range.
6. Define **any other concepts or properties** if it is needed
7. Check the consistency of your ontology with PELLET
8. Add at least two defined classes (for example, with universal and/or existential restrictions)

Optional :

1. You may use existing classes and properties in Dbpedia (or any remote knowledge base) and links them to you own classes using **equivalentClass, equivalentProperty** properties

Part II: Populating the ontology (5 points)

Create some individuals to the Movie class such as:

- Pulp Fiction, Genre: Crime Thriller, 1994, USA, English.
- Kill Bill (volume 1), Genre: Action Crime Thriller, 2003, USA, English.

Create some individuals to different classes such as:

- Quentin Tarantino, American, 53 years old, writer and director of Pulp Fiction and Kill Bill (volume1). He also played a role in that movie.
- John Travolta, American, 59 years old, actor in Pulp Fiction.
- Uma Thurman, 43 years old, actress in Pulp Fiction. She also participated as a writer in Kill Bill (volume1).
- ……..

Part III: Querying the ontology (4 points)

Write SPARQL queries to response to the following:

1. List the instances of the class Actor
2. List the name of all Thriller movies. For each one, display its director.
3. List the name of all Crime Thriller movies.
4. List the name of Actors older than 51 years.

Propose 5 SPARQL queries:

1. A query that contains at least 2 Optional Graph Patterns
2. A query that contains at least 2 alternatives and conjunctions
3. A query that contains a CONSTRUCT query form
4. A query that contains an ASK query form
5. A query that contains a DESCRIBE query form

Optional :

1. Use Dbpedia (or any remote knowledge base) to extract some individuals and add them to your Knowledge base using SPARQL queries

Part IV: Manipulating the ontology using Jena (10 points)

Using an IDE (for example eclipse), create a new Java project and configure its build path to add the Jena libraries. In that project, create a new folder called data and copy your OWL file from part II inside it.

1. Create a java program (Jena1.java) that loads the ontology and displays all the Persons (**without** using queries, **without** inference).
2. Create a java program (Jena2.java) that loads the ontology and displays all the Persons (**using** a query, **without** inference). Create the used query in text file under the data folder.
3. Create a java program (Jena3.java) that loads the ontology and displays all the Actors (**without** using queries, **using** inference). To load the inferred model, use the `JenaEngine.readInferencedModelFromRuleFile` method and use owl rules.
4. Create a java program (Jena4.java) that:
    a. Reads a name of a movie
    b. If it doesn't exist displays an error message
    c. Else, display its year, country, genres and actors
5. Create a java program (Jena5.java) that displays all persons that are actors and directors. Do this using a rule that defines a new class ActorDirector. The rule file must be saved in the data folder.
6. Specify three different rules and implement them in a java program (Jena6.java). These rules should complement the ontology (cannot be expressed using only OWL).

Part V: Java application (6 points)

- Create a GUI (or console) java application that returns you a list of films based on the included/excluded actors, directors and genres.

Optional :

- Use Dbpedia (or any remote knowledge base) to extract some individuals and add them to your Knowledge base using Sparql queries

Work to send:

Create an archive **name1-name2-name3.zip** with:

1. The ***.owl *.rdfs** file generated by Protégé (part I and II)
2. A (***.txt or *.doc)** file containing the SPARQL queries (part III)
3. The eclipse **src** and **data** folders of (part IV and V)
4. The presentation file ***.ppt**

Send the archive by e-mail, the subject of the e-mail **must** be:

[**KM-Project]name1-name2-name3**, to teaching.gaaloul@gmail.com