

Web Sémantique et Social : Interrogation de données RDF

1) SPARQL : Interrogation et méta-interrogation

Créer un modèle Jena (voir TP précédent) contenant les triplets suivant.

```
@prefix movies: <http://www.lirmm.fr/ulliana/movies#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dbp: <http://dbpedia.org/> .

movies:directedBy rdfs:domain movies:Movie .
movies:title rdfs:domain movies:Movie .
movies:directedBy rdfs:range movies:Director .
movies:playsIn rdfs:domain movies:Actor .
movies:playsIn rdfs:range movies:Movie .
movies:Actor rdfs:subClassOf movies:Artist .
movies:Director rdfs:subClassOf movies:Artist .
movies:title rdf:type owl:DataTypeProperty .
movies:title rdfs:domain movies:Movie .
movies:m2 movies:title "Vertigo" .
movies:m1 rdf:type movies:Movie .
movies:m3 movies:directedBy dbp:Alfred_Hitchcock .
movies:a1 movies:playsIn movies:m4 .
movies:Director rdfs:subClassOf movies:Artist .
```

1. Séparer les triplets contenant des connaissances ontologiques des triples représentant des données.

Évaluer les requêtes suivantes avec Jena. Une classe d'exemple est fournie à l'adresse <https://gitlab.etu.umontpellier.fr/p00000013857/jena-models>.

2. Donner la requête SPARQL qui sélectionne tous les (identifiants des) films
3. Donner la requête SPARQL qui sélectionne tous les (identifiants des) films, cette fois ci en prenant en compte la sémantique de `rdfs:subClassOf`
4. Donner la requête SPARQL qui sélectionne toutes les sous-classes de la classe artiste
5. Donner la requête SPARQL qui sélectionne tous les artistes, cette fois ci en prenant en compte la sémantique de `rdfs:subClassOf`
6. Pour chaque requête, dire s'il s'agit d'une interrogation ou d'une méta-interrogation.

2) L'exploration d'un endpoint SPARQL : le cas de DBPedia.

Dans le Web Sémantique, on est souvent confrontés au problème d'extraire un résumé de la structure des données RDF accessibles via les endpoints SPARQL. Une solution pragmatique consiste à exécuter une série de requêtes simples et génériques permettant de comprendre la structure des classes ainsi que l'utilisation des propriétés dans les données.

L'objectif de cet exercice est de récupérer une description de la structure d'un fragment de DBPedia.

Connectez vous à l'endpoint SPARQL de DBPedia (<http://dbpedia.org/sparql> ou <http://dbpedia.org/snorql/>) et testez-en le bon fonctionnement avec la requête suivante.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?varClass
WHERE { ?varClass rdf:type owl:Class
}
LIMIT 100
```

Voici ensuite une trace à suivre pour l'exploration de DBPedia.

1. Choisissez une classes de l'ontologie qui commence par la première lettre de votre nom, et donnez la liste de ses sous-classes.

Pour tester si une classe commence par exemple avec la lettre "G" vous pouvez ajouter

`FILTER regex(?varClass, "http://dbpedia.org/ontology/G")` dans la clause WHERE de la requête

2. Donner la liste des propriétés employées pour décrire les instances de la classe choisie.
3. Vérifier si ces propriétés ont des sous-propriétés.
4. (Optionnel) Trier les sous-classes selon le nombre d'instances.
5. (Optionnel) Trouver les couples de propriétés les plus utilisées conjointement.

Dans le rendu, reporter les requêtes correspondantes aux interrogations, ainsi que les 10 premières résultats obtenus.

3) Interrogation du réseau social de l'UE

Récupérer les données RDF (https://notes.inria.fr/LnHBdHhUQlG74P_65iAcjQ) et les enregistrer dans un fichier. Vous devrez utiliser Jena pour l'interrogation (QueryLocalRDF.java).

Exécuter cette requête de test; attention aux occurrences du caractère " ~ ".

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX tsd: <http://www.umontpellier.fr/traitementsemantiquedesdonnees#> .
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT *
WHERE { ?s ?p ?o }
ORDER BY ?s ?p ?o
```

Considérez les graphes G_{knows} , $G_{\text{personalFriend}}$, et $G_{\text{socialFriend}}$, induits par les propriétés `foaf:knows`, `ws:personalFriend`, et `ws:socialFriend`, respectivement. On assume qu'un noeud n (ou bien m) appartient au graphe induit par la propriété p lorsque G_p contient le triplet $\langle n, p, m \rangle$. Enfin, n'oubliez surtout pas que `foaf:knows` est une propriété symétrique, il vous faudra prendre en compte cela soit au niveau du modèle Jena soit au niveau des interrogations.

Donner des requêtes SPARQL permettant de répondre aux questions suivantes.

1. Est il vrai que $G_{\text{personalFriend}} \subseteq G_{\text{socialFriend}}$?
2. Est il vrai que $G_{\text{socialFriend}} \subseteq G_{\text{personalFriend}}$?
3. Est il vrai que $G_{\text{personalFriend}}, G_{\text{socialFriend}} \subseteq G_{\text{knows}}$?

Répondez aux questions suivantes pour chaque graphe ($G_{\text{personalFriend}}$, et $G_{\text{socialFriend}}$).

4. Combien de noeuds et d'arêtes y a-t-il dans le graphe ?
5. S'agit-il d'un graphe fortement connexe ?
6. S'agit-il d'un graphe connexe ?
7. Donner la liste de vos voisins dans le graphe.
8. Calculez la degré de chaque noeud (nombre de voisins)
9. Combien de triangles¹ y-a-t il dans le graphe ?
10. Combien de triangles peuvent encore se former dans le graphe ?
11. Donner la liste de toutes les cliques² de cardinalité 4 dans le graphe.
12. Donner la taille de chaque composant connexe du graphe

1. un triangle, est un ensemble de trois ressources A, B, C telles que la propriété P vaut entre A et B , entre B et C , et entre C et A . Il s'agit d'une topologie classique dans graphe social, par exemple +90% des recommandations d'amitié dans Facebook suivent cette topologie.

2. on verra le graphe comme non-orienté pour cette question. C'est à dire, si un arc étiqueté avec P existe de A vers B , alors nous supposons implicitement que un arc étiqueté avec P existe de B vers A .