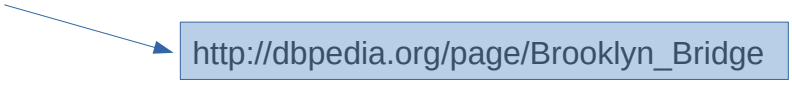
TP (3)-4-(5): Conception et implémentation d'un système de questions réponses en langue naturelle sur des données structurées

Continuation du TP3.

Sous-tache 1c. Identifier les entités de la base mentionnés dans la question

Pour faire le matching entre les entités nommées que vous avez identifié dans votre question et les entités dans la base DBpedia, servez-vous du **DBpedia Lookup Service** - Find DBpedia URIs for keywords (https://wiki.dbpedia.org/lookup). Ce service de recherche DBpedia peut être utilisé pour rechercher des URI DBpedia à l'aide de mots-clés associés. Associé signifie que soit le libellé d'une ressource correspond, soit un texte d'ancrage fréquemment utilisé dans Wikipédia pour faire référence à une ressource spécifique correspond (par exemple, la ressource `http://dbpedia.org/resource/United_States` peut être recherchée par la chaîne "USA").

`Which river does the Brooklyn Bridge cross ?`



A ce stade, normalement pour chaque question vous avez du trouver l'objet sur lequel la question porte [Brooklyn_Bridge], et le type de réponse attendue [River]. **Attention !** La réponse attendue est la **variable inconnue** dans votre SPARQL query (mais c'est bien de connaitre son type pour pouvoir trier les réponses reçues par le DBpedia SPARQL endpoint).

Sous-tache 1d. Identifier les relations de la base mentionnés dans la question

La variable manquante pour pouvoir écrire votre SPARQL query est la relation qui relie l'objet sur lequel la question porte à la réponse attendue (dans la question-exemple, la relation qu'on cible est la relation `[crosses]`, qui relie le Brooklyn Bridge a la réponse attendue par le système, c'est a dire `East River`).

Ici (http://mappings.dbpedia.org/server/ontology/classes/) vous pouvez trouver l'ontologie de DBpedia, qui contiens toutes les relations de la base.

Comme elles sont vraiment très nombreuses (presque 2800 *properties*), vous pouvez télécharger du Moodle un échantillon de ces relations **relations.txt** (il s'agit de toutes les relations qui vous permettront de répondre aux questions de votre jeu de questions, plus d'autres relations).

Vous devez maintenant trouver une bonne stratégie pour trouver parmi les tokens de votre question en entrée, ceux qui matchent avec une des propriétés de la liste.
Pour ce faire, vous pouvez tester plusieurs stratégies :
- exact match
- String edit distance (Levenshtein) (http://www.nltk.org/howto/metrics.html)

- WordNet similarities (https://www.nltk.org/howto/wordnet.html)

Cette tâche pose le problème des variations lexicales entre les labels associés aux entités et relations de la base et les termes employés par l'utilisateur, puisque celui-ci n'est pas guidé par la connaissance du schéma de la base. Se pose aussi le problème de résolution d'ambiguïtés sémantiques, car un même terme peut faire référence à différents objets ou prédicats. Par exemple, le verbe *married to* peut faire référence aux relations *dbo:spouse*, *dbo:partner*, *dbp:wife*, *dbp:husband*, *dbp:union*, *dbp:relationship*.

Tache 2. Création de la requête SPARQL

Normalement vous disposez des trois éléments qui vous permettront de poser la requête SPARQL au DBpedia SPARQL endpoint/

```
<query>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX res: <http://dbpedia.org/resource/>
SELECT DISTINCT ?uri WHERE {
        res:Brooklyn_Bridge dbo:crosses ?uri .
        }
</query>
```

et de recevoir une réponse :

```
<answer>
<uri>http://dbpedia.org/resource/East_River</uri>
</answer>
```

Tache 3. Évaluation (à suivre...)