TD 2 - Annotation sémantique (Named-Entity Linking) et enrichissement du texte

Introduction

L'objectif principal de ce TD est de développer un système automatique qui prend un texte en entrée, reconnaît les entités qui y sont mentionnées et détermine la ressource de DBpedia la plus appropriée pour chaque entité reconnue en fonction du contexte. Ensuite, en interrogeant chaque ressource, le système est capable de récupérer des informations pour finalement enrichir/augmenter le texte original.

Prenons par exemple le texte suivant :

> *With an equilibrium temperature of about 4,050 kelvin, the exoplanet KELT-9b (also known as HD 195689b) is an archetype of the class of ultrahot Jupiters that straddle the transition between stars and gas-giant exoplanets and are therefore useful for studying atmospheric chemistry*

Vous connaissiez déjà la planète KELT-9b ? Vous souhaitez en savoir plus ? Pour cela, le système doit d'abord trouver les mentions correspondant aux ressources de DBpedia.

> *With an* ***<entity URI="http://dbpedia.org/resource/Planetary_equilibrium_temperature">****equilibrium temperature****</entity>*** *of about 4,050* ***<entity URI="http://dbpedia.org/resource/Kelvin">****kelvin****</entity>****, the* ***<entity URI="http://dbpedia.org/resource/Exoplanet">****exoplanet****</entity>*** ***<entity URI="http://dbpedia.org/resource/KELT-9b">****KELT-9b****</entity>*** *(also known as* ***<entity URI="http://dbpedia.org/resource/Henry_Draper_Catalogue">****HD****</entity>*** *195689b) is an* ***<entity URI="http://dbpedia.org/resource/Archetype">****archetype****</entity>*** *of the class of ultrahot* ***<entity URI="http://dbpedia.org/resource/Jupiter_mass">****Jupiters****</entity>*** *that straddle the transition between stars and* ***<entity URI="http://dbpedia.org/resource/Gas_giant">****gas-giant****</entity>*** ***<entity URI="http://dbpedia.org/resource/Exoplanet">****exoplanets****</entity>*** *and are therefore useful for studying* ***<entity URI="http://dbpedia.org/resource/Atmospheric_chemistry">****atmospheric chemistry****</entity>***

Ensuite, à partir de chaque ressource identifiée, le système doit récupérer des informations supplémentaires à travers une requête SPARQL. Par exemple à partir de la ressource **`http://dbpedia.org/resource/Jupiter_mass`**, nous pouvons interroger la propriété **`rdfs:comment`** et récupérer l'information :

> *Jupiter mass (MJ or MJup) is the unit of mass equal to the total mass of the planet Jupiter (1.898×1027 kg, 317.83 Earth mass; one Earth mass equals 0.00315 Jupiter masses). Jupiter mass is used to describe masses of the gas giants, such as the outer planets and extrasolar planets. It is also used in describing brown dwarfs. In the Solar System, the masses of the outer planets can be listed in Jupiter mass. The other gas giants are far less massive than Jupiter. * Jupiter – 1.000 * Saturn – 0.299 * Uranus – 0.046 * Neptune – 0.054 One Jupiter mass can be converted to related units*

Méthodologie

Installation de DBpedia Spotlight.
Pour des annotations plus rapides, il est recommandé d'installer et d'exécuter ce système en local configuré pour la langue anglaise (en). Le moyen le plus simple est de passer par Docker. Pour ce faire, suivez les instructions sur ce lien : https://hub.docker.com/r/dbpedia/dbpedia-spotlight

Si vous ne pouvez pas installer DBpedia Spotlight localement, vous pouvez également utiliser une version de démonstration disponible en ligne. Cette option **est moins recommandée**, car les réponses sont généralement plus lentes ou le service n'est pas disponible.

Annoter des entités dans un texte
Utilisez la bibliothèque Python *pyspotlight* (https://github.com/aolieman/pyspotlight) pour invoquer la méthode d'annotation de DBpedia Spotlight, en passant comme paramètres:

- l'url DBpedia Spotlight: si installé localement, utilisez http://localhost:2222/rest/annotate, si vous voulez utiliser la version de démo (https://demo.dbpedia-spotlight.org/ https://www.dbpedia-spotlight.org/api)
- le texte à annoter
- les paramètres *confidence* et *support*[1]

```
>>> import spotlight

>>> annotations = spotlight.annotate('https://api.dbpedia-spotlight.org/en/rest/annotate',

...                                  'Votre texte ici',

...                                  confidence=0.4, support=20)
```

Vérifier les résultats et ajuster les paramètres
Vérifiez les résultats (les entités DBpedia, qui se trouvent dans le JSON résultant) et ajustez les paramètres jusqu'à ce que les meilleurs résultats possibles soient obtenus.

Rechercher des informations supplémentaires pour chaque résultat trouvé
Créez une fonction qui lit le JSON renvoyé et extrait chaque URI qu'il contient.
Ensuite, la fonction doit, en utilisant la bibliothèque SPARQLWrapper (https://rdflib.dev/sparqlwrapper/), interroger DBpedia pour obtenir des informations supplémentaires sur chaque ressource (voir l'exemple « SELECT query » dans la documentation de la bibliothèque). Commencez par utiliser la requête suivante pour obtenir la propriété `rtfs:comment` en anglais.

---

[1] **Support** : il s'agit du paramètre « Resource Prominence », qui vous aide à ignorer les ressources peu importantes ou peu informatives. Lorsque vous lui attribuez une valeur X, cela signifie que les ressources dont le nombre de "in-links" Wikipedia est inférieur à X seront ignorées et ne vous seront pas renvoyées. **Confidence** : c'est le paramètre « Disambiguation Confidence », c'est un seuil qui prend une valeur entre 0 et 1. Lorsque vous lui donnez une valeur élevée, vous obtenez des annotations meilleures et plus fiables mais vous risquez de perdre certaines annotations correctes.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?comment
WHERE {
    <URI_de_votre_ressource_ICI> rdfs:comment ?comment .
    FILTER(langMatches(lang(?comment), "EN"))
}
```

Récupérer d'autres informations et dans d'autres langues

En modifiant la requête SPARQL fournie à l'étape précédente, essayez de récupérer des informations pertinentes à partir d'autres propriétés de chaque ressource DBpedia (comme *abstract* ou `dbo:populationTotal` par exemple). Vous pouvez également récupérer chaque information dans d'autres langues.

Cas d'Utilisation

Dans cette partie du TD, vous utiliserez le dataset Zeroshot/Twitter-Financial-News-Topic (https://huggingface.co/datasets/zeroshot/twitter-financial-news-topic?row=0), qui contient des textes de tweets. L'objectif est d'appliquer DBpedia Spotlight pour identifier les entités dans chaque tweet, puis de récupérer des informations sur chaque entité pour enrichir le texte du tweet.

1. Sélectionnez un ensemble de tweets du dataset. Notez que le *label* associé à chaque tweet ne sera pas utilisé dans cet exercice.

2. Utilisez DBpedia Spotlight pour annoter les entités dans chaque tweet. Comme précédemment, utilisez la bibliothèque Python pyspotlight pour cela.

3. Après avoir identifié les entités, utilisez SPARQLWrapper pour interroger DBpedia et récupérer des informations supplémentaires sur chaque entité identifiée. Vous pouvez utiliser par exemple des propriétés telles que dbo:abstract pour obtenir un résumé de l'entité ou dbo:wikiPageExternalLink pour un lien externe associé.

4. Enrichissez le texte du tweet avec les informations récupérées. Par exemple, un tweet tel que
"Deutsche Bank downgrades General Motors, cites looming pricing risks to automakers https://t.co/40yhalOoOU"

pourrait être enrichi pour inclure un résumé sur la Deutsche Bank et General Motors, basé sur les informations récupérées de DBpedia.

Exemple de Résultat Attendu:
Un tweet enrichi pourrait ressembler à ceci :
"Deutsche Bank [Résumé de DBpedia sur la Deutsche Bank] downgrades General Motors [Résumé de DBpedia sur General Motors], cites looming pricing risks to automakers https://t.co/40yhalOoOU"

5. Pour chaque tweet, créez une visualisation montrant le tweet original, les entités identifiées, et le tweet enrichi avec les informations supplémentaires.

6. Discutez des avantages et des limites de cette approche d'enrichissement de texte, notamment en ce qui concerne la pertinence et la précision des informations extraites.