TD4 : La désambiguïsation lexicale (wordnet)

Et pondération des mots

LO17

Fatma Chamekh

2021-2022

Objectifs :

L'objectif de ce TD est d'explorer la désambiguïsation lexicale en utilisant wordnet/ NLTK et la pondération des mots via tdf-IDF. Il comporte deux parties :

- Partie 1 : découverte, manipulation de la base Wordnet;
- Partie 2 : se familiariser avec un enchainement de traitement des différentes tâches NLP vu en cours. Continuez le travail sur la base Reuters (TD2) en s'inspirant des exercices du TD.

Partie 1 :

Exercice 1 :

Pour se familiariser avec Wordnet/NLTK, vous pouvez vous baser sur la documentation suivante : https://www.nltk.org/_modules/nltk/corpus/reader/wordnet.html.

1. Ecrire une fonction python pour obtenir l'ensemble des synsets du nom resistance.
2. Ecrire une fonction qui prend un synset en paramètre et qui renvoie la liste des mots contenus dans le synset.
3. Ecrire une fonction qui prend un synset en paramètre et qui renvoie la liste de ses hyperonymes (les mots contenus dans les synsets hyperonymes).
4. Ecrire une fonction qui prend un synset en paramètre et renvoie la liste des hyponymes de ses hyperonymes.
5. Ecrire une fonction qui prend un synset en paramètre et renvoie la liste des hyponymes des hyponymes de ses hyperonymes.
6. Ecrire une fonction voisins qui prend un synset en paramètre et renvoie l'union de l'ensemble de ses hyperonymes, de l'ensemble des hyponymes de ses hyperonymes et l'ensemble des hyponymes des hyponymes de ses hyponymes.

Exercice 2 :

NB : l'ensemble des documents dont vous avez besoin pour réaliser les exercices sont disponible dans le dossier exercicewordnet disponible sur wordnet.

Le nom resistance a 10 sens dans Wordnet (le nombre n'est pas fixe car il ya des mises à jour régulières). Le but de l'exercice est d'assigner automatiquement le bon sens de resistance à toutes les occurrences dans le texte *wsd-data1.txt*. Chaque ligne du texte contient un paragraphe comprenant au moins une occurrence de resistance. Chacune de ces lignes est suivie d'un séparateur de paragraphes sur une autre ligne.

Pour chacune des occurrences de resistance, il s'agira d'extraire la liste des noms voisins dans le texte et d'évaluer le recouvrement avec la liste des mots "voisins" dans Wordnet pour chacun des sens (synsets) possibles.

Le sens sélectionné sera celui qui a le meilleur recouvrement. Pour cela, vous vous aiderez de la classe wsdData dans wsd.py (cf. exemple d'utilisation td10.py). La classe wsdData permet pour chacun des paragraphes d'un texte d'extraire la liste des noms (méthode getInstances) et le textVector associé (méthode getInstanceVectors).

Attention: pensez à mettre dans votre répertoire de travail les scripts tagging.py et textSpaceVector.py.

Nous supposons que les voisins d'un sens d'un mot dans Wordnet est la liste renvoyée par la fonction voisins implémentée ci-dessus.

1. Ecrire un script python qui permet d'assigner à chaque occurrence de resistance dans le texte un sens (un synset).
2. Evaluer les résultats.
3. N'y a-t-il pas moyen d'améliorer ces résultats ? proposez vos solutions.

Partie 2 :

Exercice1 :

Dans le document textminingTP (J'ai proposé ce TP quand j'étais à Telecom Paris pour des étudiants de AgroParistech qui veulent s'orientés vers la data science. Les collègues ont conservé ce TP pour les futurs étudiants), je présente un enchainement pour analyser un texte et découvrir les mots les plus pertinents. Veuillez tester cet enchainement.

Exercice2 :

En se basant sur les exercices de ce Td (voir les TD précédents). Proposez les mots les plus pertinents pour le corpus Reuters (voit TD2).

Attention : L'enchainement proposé dans l'exercice précèdent n'est pas universel/optimal. Vous pouvez vous inspirer pour réaliser le vôtre.