Analyse sémantique automatique du langage naturel Marie Candito
M2 Linguistique Informatique – Univ. Paris Diderot

Evaluation de vecteurs de mots par une tâche de substitution lexicale

A rendre par mail pour le ??: un zip, contenant votre programme votrenomenminuscules**.py2** ou votrenomenminuscule**.py3**
et votre rapport ds votrenomenminuscule.pdf
Possibilité de travailler en binôme.

1 La tâche de substitution lexicale SemDis 2014

Etudiez la définition de la tâche

https://www.irit.fr/semdis2014/fr/task1.html

ainsi que le jeu de données de test fourni, les réponses gold, le format attendu des réponses, les métriques d'évaluation et le script d'évaluation.

2 Expérimentations avec vecteurs de mots

Implémentez la méthode ci-dessous, utilisez le module d'évaluation fourni par les utilisateurs, et testez différentes configurations d'hyperparamètres.

2.1 Génération des candidats substituts :

On propose d'utiliser comme candidats substitut pour un mot cible ses n plus proches voisins dans un espace vectoriel de mots, de même catégorie morpho-syntaxique que le mot cible. On testera deux ressources :

- FREDIST : (Henestroza Anguiano & Denis, 2011) : les plus proches voisins sont déjà calculés, téléchargeable ici : https://gforge.inria.fr/projects/fredist/ .
- FRWAK-SKIPGRAM : Des vecteurs obtenus avec word2vec, par J.P. Fauconnier, et disponibles ici : http://fauconnier.github.io/#data, plus précisément nous utiliserons ceux calculés avec skip-gram negative sampling, sur le corpus FRWAK, en version lemmatisés, avec catégories morphosyntaxiques, 700 dim, cutoff de 50 :
  http://embeddings.org/frWac_postag_no_phrase_700_skip_cut50.bin
  (utilisez le package de manipulation de .bin word2vec, cf.
  http://fauconnier.github.io/#data)

2.2 Score pour les candidats substituts :

Vous utiliserez la similarité entre le substitut et un Continuous Bag of Words, i.e. le vecteur obtenu en sommant les vecteurs des mots de contexte. Plus précisément, pour un couple (c=mot cible à substituer, s=candidat substitut) :
Soit CXT l'ens. des mots pleins dans le contexte de c (limitable à 2F mots pleins, F étant un hyperparamètre), soit Z le vecteur obtenu en faisant une somme des vecteurs des mots de CXT, on utilise simplement score(vecteur(s), Z). Une variante est d'inclure ou pas dans le CXT le mot cible lui-même, cf. le candidat substitut peut être un voisin plus ou moins proche de c.

Les hyperparamètres à tester sont donc :

- L'espace vectoriel à utiliser : soit FREDIST soit FRWAK-SKIPGRAM (NB : fredist se présente comme un thésaurus précalculé à partir de vecteurs distributionnels creux. On considèrera que les XXX plus proches voisins de chaque mot sont les seules dimensions non nulles de ces vecteurs)
- CIBLE_INCLUSE : un booléen pour inclure ou pas au contexte le mot cible lui-même
- La taille F de la fenêtre de contexte (on considèrera F mots pleins à gauche + F mots pleins à droite)
  - la taille 0 n'a pas de sens si la cible n'est pas incluse, et correspond à un choix non contextuel
  - prévoir une valeur « infinie » : tous les mots pleins de la phrase

Vous aurez besoin de tagger/lemmatiser les phrases de contexte du jeu SemDis (en utilisant par exemple le tagger MElt Denis et Sagot, 2009 :
https://gforge.inria.fr/frs/?group_id=481&release_id=9930#melt-_2.0b12-title-content)

Vous ferez un petit rapport réexpliquant la tâche, la méthode, et commentant vos résultats.
Votre programme doit contenir une aide en ligne (l'option –h doit indiquer comment utiliser le programme).
Toute idée d'amélioration est la bienvenue.