Cours TAL – Labo 2 : Mise en œuvre et évaluation de « POS taggers » pour le français

Distribué le mardi 5 mars 2019, pour le lundi 18 mars 2019

Objectif

Appliquer des étiqueteurs morphosyntaxiques (POS taggers) disponibles dans NLTK et dans les outils Stanford NLP à des textes français, puis quantifier leurs performances.

Informations

- Travail à effectuer en binôme.
- Le notebook Jupyter avec les expériences, les résultats obtenus, et leur analyse est à envoyer avant le **lundi 18 mars à 23h59** comme un fichier ZIP (IPYNB et PDF) par email.

Instructions

1. **Télécharger des données en français annotées avec les POS tags : trois fichiers de données** UD[1] (mot de passe = reference) https://drive.switch.ch/index.php/s/0wUwDoCmaVPU4Gw qui sont une adaptation des données d'origine avec la suppression de quelques lignes. Quel est le format de ces 3 fichiers ? Dans quelles colonnes se trouvent les mots et leur POS tags ? Pouvez-vous trouver sur le Web la liste des POS tags du projet Universal Dependencies ?

2. **Évaluer le *Stanford POS tagger* pour le français avec les modèles fournis**

    a. Le labo de TAL de Stanford fournit un étiqueteur morpho-syntaxique (POS tagger) qui utilise l'apprentissage automatique : https://nlp.stanford.edu/software/tagger.html. Ce tagger est en Java, et peut être exécuté en ligne de commande en utilisant les instructions fournies à l'URL précédente. Notez qu'il est aussi possible de l'appeler depuis NLTK (un notebook Jupyter) grâce au module nltk.tag.stanford (voir code source à https://www.nltk.org/_modules/nltk/tag/stanford.html) mais la gestion des différents chemins (de Java, des classes Stanford, des modèles) est un peu pénible.

    b. Téléchargez les modèles pour le français (*full Stanford Tagger version 3.9.2*) et testez-les sur les fichiers « dev » et « test », en utilisant le modèle *french-ud.tagger*. Suivez les indications sous « *javadoc for MaxentTagger* » à l'URL précédente. Quels sont les scores obtenus ?

[1] Source originale : https://github.com/UniversalDependencies/UD_French-GSD

3. **Entraîner le Stanford POS tagger sur les données UD en français, et comparer le modèle obtenu avec les modèles français de la partie 1A (dans GATE).** On suivra la documentation du Stanford POS tagger pour l'entraîner avec le fichier *fr-ud-train.conllu2* et le tester avec *fr-ud-test.conllu2*. Cette documentation se trouve à : https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/MaxentTagger.html – considérer la section « *Training from the command line* ».
    a. La configuration du système pour l'entraînement est donnée dans un fichier (texte) qui peut être produit soit en suivant la documentation (option –genprops pour obtenir un template, qui doit être modifé), soit en s'inspirant du fichier *french-ud.tagger.props* qui accompagne le fichier modèle *french-ud.tagger* que vous avez utilisé ci-dessus. On peut aussi mélanger les approches. Le but est d'aboutir à un nouveau modèle *myFrench-UD.tagger.props* qui offre un bon entraînement.
    b. Lancer l'entraînement sur le fichier *fr-ud-train.conllu2.* S'il est trop grand, s'entraîner seulement sur le fichier *fr-ud-dev.conllu2.* Pendant l'entraînement (> 10 minutes, 500 itérations), on peut regarder la suite du travail.
    c. Quel modèle est meilleur, le vôtre ou celui téléchargé en 2 ? On peut l'évaluer selon https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/MaxentTagger.html (*tagging and testing from the command line*). On peut p.ex. modifier le fichier *myFrench-UD.tagger.props* en indiquant une *testFile* (choisir l'option verboseResults = false).

4. **Entraîner un POS tagger dans NLTK pour qu'il fonctionne sur le français.**
    a. Importez les données en français dans NLTK (comme dans le Labo 1). Pour observer quelle forme ont les textes tagués dans NLTK, considérez le Chapitre 5, sections 1, 2.1 et 2.2 : http://www.nltk.org/book/ch05.html. L'importation sera facilitée par le module nltk.corpus.reader.conll (https://www.nltk.org/api/nltk.corpus.reader.html).
    b. Dans le module de NLTK avec des taggers, http://www.nltk.org/api/nltk.tag.html, considérez le module *nltk.tag.perceptron*, pour lequel on explique de façon précise l'entraînement (voir « *train the model* ») et le test.
    c. Entraînez ce module sur les données *train* puis testez-le sur les données *test* grâce à la méthode *evaluate*. Comment se compare-t-il avec les deux modèles du POS tagger MaxEnt de Stanford ?