POLYTECH PARIS-SUD

Et5-Info

Module : Traitement Automatique des Langues - Traduction Automatique

TP 1 - Analyse linguistique avec la plateforme Stanford CoreNLP

Contexte :

La traduction automatique à base de corpus des textes des langues morphologiquement riches nécessite un pré-traitement impliquant au minimum une analyse morpho-syntaxqiue des données d'apprentissage. L'objectif de ce TP est l'installation et l'expérimentation de la plateforme d'analyse linguistique Stanford Core NLP nécessaire à la préparation des corpus d'apprentissage des modèles de traduction statistique Moses et neuronal OpenNMT.

Une plateforme d'analyse linguistique standard se compose des modules suivants :

1. **Tokenisation :** Ce module consiste à découper les chaînes de caractères du texte en mots, en prenant en compte le contexte ainsi que les règles de découpage. Ce module utilise généralement des règles de segmentation ainsi que des automates d'états finis.
2. **Analyse morphologique :** Ce module a pour but de vérifier si le mot (token) appartient à la langue et d'associer à chaque mot des propriétés syntaxiques qui vont servir dans la suite des traitements. Ces propriétés syntaxiques sont décrites en classes appelées catégories grammaticales. La consultation de dictionnaires de formes ou de lemmes permet de récupérer les propriétés syntaxiques concernant les mots à reconnaître.
3. **Analyse morpho-syntaxqiue :** Après l'analyse morphologique, une partie des mots restent ambigus d'un point de vue grammatical. L'analyse morphosyntaxique réduit le nombre des ambiguïtés en utilisant soit des règles ou des matrices de désambiguïsation. Les règles sont généralement construites manuellement et les matrices de bi-grams et tri-grams sont obtenues à partir d'un corpus étiqueté et désambiguïsé manuellement.
4. **Analyse syntaxique :** Ce module consiste à identifier les principaux constituants de la phrase et les relations qu'ils entretiennent entre eux. Le résultat de l'analyse syntaxique peut être une ou plusieurs structures syntaxiques représentant la phrase en entrées. Ces structures dépendent du formalisme de représentation utilisé : un arbre syntagmatique, un arbre de dépendance ou une structure de traits. L'analyse en dépendance syntaxique consiste à créer un arbre de relations entre les mots de la phrase. Le module d'analyse syntaxique utilise des règles pour l'identification des relations de dépendance ou des corpus annotés en étiquettes morpho-syntaxiques et en relations de dépendance.
5. **Reconnaissance d'entités nommées :** Ce module consiste à identifier les dates, lieux, heures, expressions numériques, produits, évènements, organisations, présentes sur un ou plusieurs tokens, et à les remplacer par un seul token.

Travail demandé

Vous allez installer et expérimenter la plateforme d'analyse linguistique Stanford Core NLP (une boite à outils linguistiques utilisant l'apprentissage statistique à partir de corpus annotés).

I. Installation et évaluation de l'outil de désambiguïsation morpho-syntaxique de l'université de Stanford

Informations sur l'outil : https://nlp.stanford.edu/software/tagger.shtml

1. Installation

a. Installer Java sur Ubuntu 16.04 (Les outils TAL de Stanford nécessite Java 1.8+) :
   i. Mettre à jour les paquets courants : `apt-get update`
   ii. Installer la dernière version du Java Runtime Environment (JRE): `apt-get install default-jre`
   iii. Installer Java Development Kit (JDK): `apt-get install default-jdk`

   Note : Vérifier la version de Java à l'aide de : `java –version, javac –v`

b. Télécharger la basic English Stanford Tagger version 3.9.2 (stanford-postagger-2018-10-16.zip)

c. Décompresser le fichier téléchargé: `unzip stanford-postagger-2018-10-16.zip`

d. Analyser le texte du fichier « simple-input.txt) :

```
cd stanford-postagger-2018-10-16
```

```
./stanford-postagger.sh models/english-left3words-distsim.tagger sample-input.txt > sample-input.txt.pos
```

Exemple :

Le fichier "Sample_eng.txt" contient la phrase "John ate dicious pizza with friends."

```
./stanford-postagger.sh models/english-left3words-distsim.tagger Sample_eng.txt > Sample_eng.txt.pos
```

Stanford POS tagger output (Fichier Sample_eng.txt.pos):

John_NNP ate_VBD dicious_JJ pizza_NN with_IN friends_NNS ._.

2. Evaluation

a. Lancer le POS tagger sur le fichier « wsj_0010_sample.txt » : `./stanford-postagger.sh models/english-left3words-distsim.tagger wsj_0010_sample.txt > wsj_0010_sample.txt.pos.stanford`

b. Ecrire un programme Python qui permet de convertir le fichier de référence `wsj_0010_sample.pos.ref` au format de la sortie du Stanford POS tagger. Mettre le résultat de cette conversion dans le fichier `wsj_0010_sample.pos.stanford.ref`.

c. Calculer la précision de ce POS tagger en utilisant les étiquettes PTB : `python evaluate.py wsj_0010_sample.txt.pos.stanford wsj_0010_sample.pos.stanford.ref`

d. Remplacer à l'aide d'un programme Python les étiquettes Penn TreeBank des fichiers `wsj_0010_sample.txt.pos.stanford` et `wsj_0010_sample.pos.stanford.ref` par les étiquettes universelles en utilisant la table de correspondance « POSTags_PTB_Universal.txt ». Mettre le résultat de

cette conversion dans les fichiers : `wsj_0010_sample.txt.pos.univ.stanford` et `wsj_0010_sample.txt.pos.univ.ref.`

e. Calculer la précision de ce POS tagger en utilisant les étiquettes universelles : `python evaluate.py wsj_0010_sample.txt.pos.univ.stanford wsj_0010_sample.txt.pos.univ.ref`

f. Quelles conclusions peut-on avoir à partir de ces deux évaluations ?

Notes :

1 Exemple de conversion de format (Exercice b) :

Entrée:

Format: Token Tabulation Etiquette morpho-syntaxique

Exemple :

When WRB

it PRP

's VBZ

time NN

for IN

their PRP$

biannual JJ

powwow NN

Sortie :

Format: Token_Etiquette morpho-syntaxique

Exemple :

When_WRB it_PRP 's_VBZ time_NN for_IN their_PRP$ biannual_JJ powwow_NN

Notons que les phrases dans le fichier wsj_0010_sample.pos.ref sont séparées par une ligne vide.

2 Exemple de conversion d'étiquettes morpho-syntaxiques (Exercice d) :

Entrée :

Exemple :

When_WRB it_PRP 's_VBZ time_NN for_IN their_PRP$ biannual_JJ powwow_NN

Sortie :

Exemple :

When_ADV it_PRON 's_VERB time_NOUN for_ADP their_PRON biannual_ADJ powwow_NOUN

Dans cet exemple, les étiquettes morpho-syntaxiques converties (en utilisant la table POSTags_PTB_Universal.txt) sont les suivantes :

WRB => ADV

PRP => PRON

VBZ => VERB

NN => NOUN

PRP$ => PRON

JJ => ADJ

II. Installation et utilisation de l'outil de reconnaissance d'entités nommées de l'université de Stanford

Informations sur l'outil : https://nlp.stanford.edu/software/CRF-NER.html

1. Installation

g. Installer Java sur Ubuntu 16.04 (Les outils TAL de Stanford nécessite Java 1.8+) :
   i. Mettre à jour les paquets courants : `apt-get update`
   ii. Installer la dernière version du Java Runtime Environment (JRE): `apt-get install default-jre`
   iii. Installer Java Development Kit (JDK): `apt-get install default-jdk`

   Note : Vérifier la version de Java à l'aide de : `java –version, javac –v`

h. Télécharger la Stanford Named Entity Recognizer version 3.9.2 (stanford-ner-2018-10-16.zip)

i. Décompresser le fichier téléchargé: `unzip stanford-ner-2018-10-16.zip`

j. Analyser le texte du fichier « simple-input.txt) :

```
cd stanford-ner-2018-10-16
```

```
java -mx600m -cp stanford-ner.jar:lib/*
edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier
classifiers/english.all.3class.distsim.crf.ser.gz -textFile
Sample_NER_en.txt > Sample_NER_en.txt.output
```

Exemple :

Le fichier " Sample_NER_en.txt" contient la phrase "John ate dicious pizza with friends."

```
java -mx600m -cp stanford-ner.jar:lib/*
edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier
classifiers/english.all.3class.distsim.crf.ser.gz -textFile
Sample_NER_en.txt > Sample_NER_en.txt.output
```

Stanford NE recognizer output (Fichier Sample_NER_en.txt.output):

John/PERSON ate/O dicious/O pizza/O with/O friends/O ./O

2. Extraction d'entités nommées

A partir du résultat de l'outil de reconnaissance des entités nommées « wsj_0010_sample.txt.ner.stanford », écrire un programme Python permettant de représenter les entités nommées sous le format suivant :

Entité nommée Type Nombre d'occurrences Proportion dans le texte (%)

Exemple :

Entité nommée	**Type**	**Nombre d'occurrences**	**Proportion dans le texte (%)**
John	PERSON	1	1 (1/1)