**TP 1 Traduction Automatique**

LIU Pan, HASSANI Leila

Université : Inalco

Contact : liupan.lucine@gmail.com, leilahassani@hotmail.fr

Dans le fichier zip « TP1\_Pan\_LIU\_Leila\_HASSANI », vous trouverez tous nos scripts et rapport dans le dossier « scripts\_rapport », le dossier « TP\_1 » contient les fichiers dont nous avons besoin pour exécuter les scripts.

图形用户界面, 应用程序

描述已自动生成图片包含 形状

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

1. **Installation et évaluation de l’outil de désambiguïsation morpho- syntaxique de l’université de Stanford**

**1. Installation**

Du fait que notre ordinateur est équipé de MacOs, nous pouvons sauter l’étape de télécharger une machine virtuelle Linux. Après avoir téléchargé et compressé le *Stanford Log-linear Part-Of-Speech Tagger*, nous avons utilisé la ligne de commande fournie (*./stanford-postagger.sh models/english-left3words-distsim.tagger Sample\_eng.txt > Sample\_eng.txt.pos*) pour analyser le texte du fichier « simple-input.txt) et nous avons obtenu le résultat attendu.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**2. Évaluation**

1. 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

文本

描述已自动生成

1. 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.

Programme Python :文本

描述已自动生成

conversion.py

Resultat :

文本

描述已自动生成

1. 文本

   中度可信度描述已自动生成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
2. 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.

文本

描述已自动生成Programme Python :

图形用户界面, 文本

描述已自动生成

pennTreeBank2etiquettesUniverselles.py

Résultat :

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. 图形用户界面, 文本, 应用程序, 电子邮件

   描述已自动生成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

**Résultat :**

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

Plus fines les parties du discours sont catégorisées, on obtiendrait un taux de précision plus bas. Quand on remplace les « micro-étiquettes » par des macro-étiquettes, on obtient un taux de précision plus haut. Étant donné que Penn TreeBank possède des parties du discours plus fines (ex : JJ JJR JJS correspondent tous les trois à ADJ dans les étiquettes universelles), on a un taux de précision plus bas en utilisant le pos-tagger du Stanford Core NLP. Quand on utilise des normes différentes pour évaluer nos résultats, on obtient probablement un taux de précision différent.

1. **Installation et utilisation de l’outil de reconnaissance d’entités nommées de l’université de Stanford**

**1. Installation**

Après avoir installé et décompressé le fichier téléchargé, nous avons analysé le texte du fichier (simple\_NER\_en.txt), voici le résultat obtenu :

**图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成**

**2. Extraction d’entités nommées**

En saisissant la ligne de commande suivante, nous avons obtenu le fichier wsj\_0010\_sample.txt.ner.stanford pour traiter ultérieurement :

*java -mx600m -cp stanford-ner.jar:lib/\* edu.stanford.nlp.ie.crf.CRFClassifier -loadClassifier  
classifiers/english.all.3class.distsim.crf.ser.gz -textFile wsj\_0010\_sample.txt > wsj\_0010\_sample.txt.ner.stanford*

文本

描述已自动生成Programme Python :

Résultat :文本

低可信度描述已自动生成