POLYTECH
PARIS-SUD

Et5-Info

Module : Traitement Automatique des Langues

TD-TP : Analyse linguistique avec le framework NLTK

Contexte :

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

1. **Découpage (Tokenization):** 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 (Morphological analysis):** 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 (Part-Of-Speech tagging):** 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 (Syntactic analysis ou Parsing):** 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 (Named Entity recognition):** 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 NLTK (une boite à outils pour le traitement automatique de la langue utilisant des approches hybrides).

I. Installation de la plateforme d'analyse linguistique NLTK

Avant de démarrer l'installation de la plateforme NLTK sur un poste équipé d'une distribution Linux (de préférence Ubuntu 20.04 LTS), il faudrait vérifier la version installée de Python:

```
python –version
```

Si `Python 2.7.12`, utiliser `pip`

Si `Python3.5` (ou supérieur), utiliser `pip3`

Installation : http://www.nltk.org/install.html

1. Installation de NLTK

   ```
   pip install --user -U nltk
   ```

2. Installation de Numpy

   ```
   pip install --user -U numpy
   ```

3. Test de NLTK en interactif

   ```
   python
   >>> import nltk
   >>> nltk.download('punkt')
   >>> from nltk.tokenize import word_tokenize
   >>> text = "It's works!"
   >>> print(word_tokenize(text))
   ```

Résultat:

```
['It', '’', 's', 'works', '!']
```

1. Evaluation de l'analyse morpho-syntaxique de la plateforme NLTK

1. Ecrire un programme Python utilisant le package `pos_tag` pour désambiguïser morpho-syntaxiquement le texte du fichier `wsj_0010_sample.txt`. Le résultat de ce module sera mis dans le fichier `wsj_0010_sample.txt.pos.nltk`.

   Note :

   - Un exemple d'utilisation du package `pos_tag` se trouve sur le lien https://www.guru99.com/pos-tagging-chunking-nltk.html.
   - L'analyseur morpho-syntaxique de la plateforme NLTK utilise les étiquettes du Penn TreeBank.

   Format du résultat de l'analyse morpho-syntaxique:

   Le fichier `wsj_0010_sample.txt.pos.nltk` doit avoir le format suivant :

   Token \t Tag (\t correspond à tabulation)

Exemple :

```
When     WRB
it PRP
's VBZ
time     NN
for      IN
their    PRP$
biannual JJ
powwow   NN
,  ,
…
```

2. Ecrire un programme Python permettant de convertir les étiquettes Penn TreeBank du fichier `wsj_0010_sample.txt.pos.nltk` en étiquettes universelles en utilisant la table de correspondance POSTags_PTB_Universal_Linux.txt. Le fichier résultat sera nommé `wsj_0010_sample.txt.pos.univ`.
3. Ecrire un programme Python permettant de construire à partir du fichier annoté au format CONLL `wsj_0010_sample.txt.conll` un fichier contenant uniquement les phrases à annotées. Le fichier résultat sera nommé `wsj_0010_sample.txt`.

Format du fichier contenant uniquement les phrases à annotées:

Le fichier `wsj_0010_sample.txt` doit avoir le format suivant :

Phrase 1

Phrase 2

…

Exemple :

```
When it's time for their biannual powwow, the nation's
manufacturing titans typically jet off to the sunny confines of
resort towns like Boca Raton and Hot Springs.
Not this year.
…
```

4. Ecrire un programme Python permettant de construire à partir du fichier annoté au format CONLL `wsj_0010_sample.txt.conll` un fichier contenant uniquement les annotations en POS tags. Le fichier résultat sera nommé `wsj_0010_sample.txt.pos`.

Format du fichier contenant uniquement les annotations en POS tags:

Le fichier `wsj_0010_sample.txt.pos` doit avoir le format suivant :

Token \t Tag (\t correspond à tabulation)

Exemple :

```
When     WRB
it PRP
's VBZ
time     NN
 …
```

2. Utilisation de la plateforme NLTK pour l'analyse syntaxique

1. Ecrire un programme Python utilisant le package `parse` pour extraire les mots composés (chunks) ayant la structure syntaxique Déterminant-Adjectif-Nom (`grammar = "Compound: {<DT>?<JJ>*<NN>}"`) présents dans le texte du fichier `wsj_0010_sample.txt`. Le résultat de ce module sera mis dans le fichier `wsj_0010_sample.txt.chk.nltk`.

   Note :
   Un exemple d'utilisation du package `parse` se trouve sur le lien https://www.guru99.com/pos-tagging-chunking-nltk.html.

2. Généraliser le programme Python précédent pour extraire les mots composés (chunks) compatibles avec les structures syntaxiques ci-dessous :
   Adjectif-Nom
   Nom-Nom
   Adjectif-Nom-Nom
   Adjectif-Adjectif-Nom

   Note : Il est recommandé d'utiliser un fichier déclaratif contenant ces structures syntaxiques.

Format du résultat de l'analyse syntaxique:

Le fichier `wsj_0010_sample.txt.chk.nltk` doit avoir le format suivant :

Pattern:
Mot composé

Exemple :

```
Adjectif-Nom:
it's time
nation's manufacturing
corporate decision
good place

Nom-Nom:
fall board
…
```

3. Utilisation de la plateforme NLTK pour l'extraction d'entités nommées

1. Ecrire un programme Python utilisant le package `ne_chunk` pour extraire les entités nommées présentes dans le texte du fichier `wsj_0010_sample.txt`. Le résultat de ce module sera mis dans le fichier `wsj_0010_sample.txt.ne.nltk`.

   Note : Un exemple d'utilisation du package `ne_chunk` se trouve sur le lien https://pythonprogramming.net/named-entity-recognition-nltk-tutorial/.

Format du résultat de l'extraction d'entités nommées:

Le fichier `wsj_0010_sample.txt.ne.nltk` doit avoir le format suivant :

Token \t Tag (\t correspond à tabulation)

Exemple :

```
Boca Raton      PERSON
Hot Springs     PERSON
National Association ORGANIZATION
Hoosier  ORGANIZATION
Indianapolis    GPE
Rust Belt       ORGANIZATION
```

2. Ecrire un programme Python permettant de convertir les étiquettes NLTK du fichier `wsj_0010_sample.txt.ne.nltk` en étiquettes universelles en utilisant la table de correspondance NERTags_NLTK_Universal_Linux.txt. Le fichier résultat sera nommé `wsj_0010_sample.txt.ne.univ`.
3. A partir du résultat de l'outil de reconnaissance des entités nommées `wsj_0010_sample.txt.ne.univ`, écrire un programme Python permettant de représenter les entités nommées sous un format tabulé XLS ou CSV (4 colonnes). Le fichier résultat sera nommé `wsj_0010_sample.txt.ne.xls`.

Format du résultat de l'extraction d'entités nommées au format tabulé:

Le fichier `wsj_0010_sample.txt.ne.xls` doit avoir le format suivant :

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

Exemple :

Entité nommée	Type	Nombre d'occurrences	Proportion dans le texte (%)
Boca Raton	PERSON	1	
Hot Springs	PERSON	1	
National Association	ORGANIZATION	1	
…			

Note:
Faire les expérimentations sur le fichier `formal-tst.NE.key.04oct95_small.txt`