Université de Bouira		Faculté de sciences
Série TP n°8	Module	TAL
	Filière	**Master ISIL** — **1ère Année**

Qu'est-ce que c'est NLTK ?

Natural Language Toolkit (NLTK)[1] est une boîte-à-outil permettant la création de programmes pour l'analyse de texte. Cet ensemble a été créé à l'origine par Steven Bird et Edward Loper, en relation avec des cours de linguistique informatique à l'Université de Pennsylvanie en 2001. Il existe un manuel d'apprentissage pour cet ensemble titré Natural Language Processing with Python [2](en anglais).

Installation

Linux	Windows

`sudo pip install nltk`
 	
ou bien pour le python3
 	
`sudo pip3  install nltk`
	C:\python2.7\Scripts\pip.exe install nltk
	ou bien installer le fichier attaché

Travailler avec NLTK

La première chose à faire pour utiliser NLTK est de télécharger ce qui se nomme le *NLTK corpora*. On va télécharger quelques corpus.

Dans votre éditeur Python IDLE, écrivez ceci :

```
import nltk
nltk.download()
```

Dans ce cas précis, une interface graphique s'affiche, vous permettant de définir la destination des fichiers et de sélectionner ce dont vous avez besoin:

Sélectionner d'installer les fichiers suivants :

- models : punkt, averaged_perceptron_tagger
- corpora:stopwords

Hors ligne :

copier le repertoire nltk_data sur le c:\ sous windows ou bien sur le *home*/votre_rep/

1 http://www.nltk.org/
2 http://www.nltk.org/book/

Lister les mots vides :

```
from nltk.corpus import stopwords
print('mots vides en  anglais')
print(stopwords.words("english"))
print('mots vides en  français')
print(stopwords.words("french"))
print('mots vides en  arabe')
print(u", ".join(stopwords.words("arabic")))
```

Eliminer les mots vides

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
text = 'In this tutorial, I\'m learning NLTK. It is an interesting platform.'
stop_words = set(stopwords.words('english'))
words = word_tokenize(text)
new_sentence = []
for word in words:
    if word not in stop_words:
    new_sentence.append(word)
print(new_sentence)
```

Disons que dans le fichier texte suivant (fichier attaché). Nous désirerions rechercher (fouiner) le mot ‘language’. Nous pourrions utiliser la librairie NLTK comme suit :

```
file2 = codecs.open('NLTK.txt', 'r', encoding='utf-8')
read_file = file2.read()
text = nltk.Text(nltk.word_tokenize(read_file))
```

Tokenizing sentences

```
from nltk.tokenize import sent_tokenize, word_tokenize
data = "All work and no play makes jack dull boy. All work and no play makes jack
a dull boy."
print(sent_tokenize(data))
```

Stemming

```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
words = ["game","gaming","gamed","games"]
ps = PorterStemmer()
for word in words:
    print(ps.stem(word))
```

Analyse des sentiments

```
import nltk.classify.util

from nltk.classify import NaiveBayesClassifier
from nltk.corpus import names

def word_feats(words):
    return dict([(word, True) for word in words])

positive_vocab = [ 'awesome', 'outstanding', 'fantastic', 'terrific', 'good',
'nice', 'great', ':)' ]
negative_vocab = [ 'bad', 'terrible','useless', 'hate', ':(' ]
neutral_vocab = [
'movie','the','sound','was','is','actors','did','know','words','not' ]

positive_features = [(word_feats(pos), 'pos') for pos in positive_vocab]
negative_features = [(word_feats(neg), 'neg') for neg in negative_vocab]
neutral_features = [(word_feats(neu), 'neu') for neu in neutral_vocab]

train_set = negative_features + positive_features + neutral_features

classifier = NaiveBayesClassifier.train(train_set)

# Predict
neg = 0
pos = 0
sentence = "Awesome movie, I liked it"
sentence = sentence.lower()
words = sentence.split(' ')
for word in words:
    classResult = classifier.classify( word_feats(word))
    if classResult == 'neg':
        neg = neg + 1
    if classResult == 'pos':
        pos = pos + 1

print('Positive: ' + str(float(pos)/len(words)))
print('Negative: ' + str(float(neg)/len(words)))
```

Travail à domicile (facultatif)

1- Choisir un corpus parmi les corpora de NLTK.
2- Calculer les fréquences des mots, et afficher les mots les plus fréquents.
3- Créer un index des mots basé sur leurs stems.
4- Créer un moteur de recherche basé sur l'index précedent.

Définitions

Corpus, pluriel : corpora ; Une collection de données linguistiques, parfois une compilation de textes écrits, ou de transcriptions d'enregistrement de discours. La raison principale d'un corpus est de vérifier une hypothèse sur le langage - par exemple : déterminer comment l'utilisation d'un son particulier, d'un mot ou d'une construction syntaxique varie. Les corpus linguistiques agissent avec les lois et les pratiques d'utilisation de corpora dans l'étude du langage. Un corpus informatique contient un ensemble vaste de textes traduisibles en langage-machine.

Universal Part-of-Speech Tagset

Tag	Meaning	English Examples
ADJ	adjective	new, good, high, special, big, local
ADP	adposition	on, of, at, with, by, into, under
ADV	adverb	really, already, still, early, now
CONJ	conjunction	and, or, but, if, while, although
DET	determiner, article	the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !
X	other	ersatz, esprit, dunno, gr8, univeristy

Autres bibliothèques :

1. **Apache OpenNLP** (java)
2. Spacy (python)
3. **Stanford's Core NLP Suite**,
4. **Apache Lucene and Solr**
5. **GATE** and **Apache UIMA**

Références :

https://code.tutsplus.com/fr/tutorials/introducing-the-natural-language-toolkit-nltk--cms-28620

https://pythonspot.com/category/nltk/