Université de Bouira
Département d'Informatique
Module : Traitement Automatique des Langues
Niveau M1-ISIL

Série TP 3

Analyse lexicale – PreProcessing. Stemming et lemmatisation avec NLTK.

1 – Qu'est-ce que le package Python nltk ?

Natural Language Tool Kit (NLTK) est une bibliothèque Python permettant de créer des programmes fonctionnant avec le langage naturel. Il fournit une interface conviviale aux ensembles de données contenant plus de 50 corpus et ressources lexicales telles que WordNet. La bibliothèque peut effectuer différentes opérations telles que la tokenization, le stemming, la lemmatisation, la classification, le parsing, tagging, etc.

NLTK peut être utilisé par les étudiants, les chercheurs et les industriels. C'est une bibliothèque Open Source et gratuite. Il est disponible pour Windows, Mac OS et Linux.

- Pour l'installer en utilisant PIP depuis cmd.exe : >> `pip install nltk`

Pratique : exécuter le code suivant sous Python 3

```
# fixer le codage des caractères dans l'entête de script
# -*- coding: utf-8 -*-

# ------------Importer les bibliothèques nécessaires------------

import nltk

# importing tokenization
from nltk.tokenize import word_tokenize
from nltk.tokenize import RegexpTokenizer

# importing the stopwords
from nltk.corpus import stopwords

# importing Porter and Lancaster stemmers from nltk
from nltk.stem import PorterStemmer, LancasterStemmer

# importing WordNetLemmatizer
from nltk.stem import WordNetLemmatizer

# ----------------- Stemming ----------------------------------

# LancasterStemmer est simple, mais pratique un stemming lourd du aux
itérations et donc un sur-stemming peut se produire.
# Le sur-stemming fait que les racines obtenues ne sont pas
linguistiques, ou elles peuvent n'avoir aucune signification.
# LancasterStemmer produit une racine encore plus courte que porter
en raison des itérations et du sur-stemming.
```

```
# Créer des objets des classes PorterStemmer and LancasterStemmer
porter = PorterStemmer()
lancaster = LancasterStemmer()

# - Stemming a word - PorterStemmer
print("Porter Stemmer")
print(porter.stem("cats"))
print(porter.stem("trouble"))
print(porter.stem("troubling"))
print(porter.stem("troubled"))

# - Stemming a word - LancasterStemmer
print("Lancaster Stemmer")
print(lancaster.stem("cats"))
print(lancaster.stem("trouble"))
print(lancaster.stem("troubling"))
print(lancaster.stem("troubled"))

# --------------------------

# - Stemming une liste de mots
word_list = ["friend", "friendship", "friends",
"friendships","stabil","destabilize","misunderstanding","railroad","m
oonlight","football"]

print("{0:20}{1:20}{2}".format("Word","Porter Stemmer",
                                                "lancaster Stemmer"))

for word in word_list:
    print("{0:20}{1:20}{2}".format(word, porter.stem(word),
                                          lancaster.stem(word)))

# --------------------------

# - Stemming une phrase sans tokenization
sentence = "Pythoners are very intelligent, and work very pythonly
and now they are pythoning their way to success."

porter.stem(sentence)

# --------------------------

# - Stemming une phrase avec tokenization (les ponctuations sont
gardées avec word_tokenizer)

nltk.download('punkt')

sentence = "Pythoners are very intelligent, and work very pythonly
and now they are pythoning their way to success."

def stemSentence(sentence):
     # Tokenization
```

```
    token_words = word_tokenize(sentence)
    print('Tokens:', token_words)
    # Stemming
    stem_sentence = []
    for word in token_words:
        stem_sentence.append(porter.stem(word))
    return " ".join(stem_sentence)

stems = stemSentence(sentence)
print('Stems: ', stems)

# ---------------------------

# - Stemming une phrase avec tokenization, et en supprimons stopwords

# downloading stopwords from nltk
nltk.download('stopwords')

# affectation des stopwords anglais à la liste sw
sw = stopwords.words('english')

sentence = "Pythoners are very intelligent, and work very pythonly
and now they are pythoning their way to success."

def cleanStemSentence(sentence):
    token_words = word_tokenize(sentence)
    # éliminer les stopwords de la liste des tokens obtenus
    clean_tokens = [token for token in token_words if token not in
sw]
    print('Tokens:', clean_tokens)
    stem_sentence = []
    for word in clean_tokens:
        stem_sentence.append(porter.stem(word))
    return " ".join(stem_sentence)

stems = cleanStemSentence(sentence)
print('Stems: ', stems)

# ---------------------------

# - Stemming une phrase avec tokenization, en supprimons la
ponctuation avec RegexpTokenizer

sentence = "Pythoners are very intelligent and work very pythonly and
now they are pythoning their way to success."

def stemSentence(sentence):
     # Tokenization
    tokenizer = RegexpTokenizer(r'\w+')
    token_words = tokenizer.tokenize(sentence)
    print('Tokens:', token_words)
    # Stemming
```

```
    stem_sentence = []
    for word in token_words:
        stem_sentence.append(porter.stem(word))
    return " ".join(stem_sentence)

stems = stemSentence(sentence)
print('Stems: ', stems)

# --------------------------

# - Stemming a document

with open("my_text.txt") as file:
     my_lines_list = file.readlines()

def stemSentence(sentence):
    token_words = word_tokenize(sentence)
    stem_sentence=[]
    for word in token_words:
        stem_sentence.append(porter.stem(word))
    return " ".join(stem_sentence)

# Stemming la prmière ligne du document
stems = stemSentence(my_lines_list[0])
print('Sentence: ', my_lines_list[0])
print('Stemmed sentence: ', stems)

# Stemming toutes les lignes du document
for line in my_lines_list:
    stem_sentence = stemSentence(line)
    print('Sentence: ', line)
    print('Stemmed sentence: ', stem_sentence)


# -------------------------- Lemmatization --------------------

# Lemmatization WordNet Lemmatizer, avec et sans context-POS

# download wordnet
nltk.download('wordnet')

# Instantiating the lemmaztizer object
lemmatizer = WordNetLemmatizer()

# --------------------------

# - Lemmatiser un mot sans contexte
print(lemmatizer.lemmatize("bats"))
print(lemmatizer.lemmatize("feet"))
print(lemmatizer.lemmatize("are"))

# - Lemmatiser un mot avec contexte : parts-of-speech parameter
```

```
print(lemmatizer.lemmatize("are", pos='v'))
print(lemmatizer.lemmatize("swimming", pos='v'))
print(lemmatizer.lemmatize("swimming", pos='n'))
print(lemmatizer.lemmatize("stripes", pos='v'))
print(lemmatizer.lemmatize("stripes", pos='n'))

# ----------------------------

# - Lemmatize a sentence
sentence = "He was running and eating at same time. He has bad habit
of swimming after playing long hours in the Sun."

# tokenize the sentence into a list of words
tokenizer = RegexpTokenizer(r'\w+')
sentence_words = tokenizer.tokenize(sentence)

# sans contexte
print("{0:20}{1:20}".format("Word", "Lemma"))
for word in sentence_words:
    print ("{0:20}{1:20}".format(word, lemmatizer.lemmatize(word)))
```