Natural Language processing

TP 2 – Bag of words (BOW)

L'objectif de ce TP est la construction d'un BOW pour la détection de SPAM sur une base des données des mails. La pratique consiste d'abord à vectoriser les observations (word embeddings), puis à implémenter un modèle de classification

Exercice 1 – BOW

1. Importer les dépendances

```python
import pandas as pd
import nltk
import numpy as np
from nltk.corpus import stopwords
from nltk.tokenize import sent_tokenize as st
from nltk.stem import WordNetLemmatizer as wordnet
import re
```

2. Importer les données et déclarer quelques variables

```python
## reading the file
df = pd.read_csv('./data/spam.csv', encoding = 'ISO-8859-1',usecols=['v1','v2'])
corpus = [] #empty list
wordnet = wordnet() #object instantiation
length = len(df['v2']) #finding total number of rows

df.head(10)
```

3. Prétraitement des données

```python
for i in range(length):
    #substitute characters at the beginning of the phrase
    rev = re.sub('[^a-zA-Z]',' ',df['v2'][i])

    #text to lowercase
    rev = rev.lower()

    #each word of the sentence becomes the element of a list
    rev = rev.split()

    #lemmatization via list comprehension
    rev = [wordnet.lemmatize(word) for word in rev if word not in stopwords.words('english')]

    #from list to string
    rev = ' '.join(rev)

    #from list to string
    corpus.append(rev) #appending to the list
```

4. Implémenter BOW, avec sklearn
   Qu'est-ce que cela signifie que le paramètre max_features = 2500 ? Quel est l'avantage qu'il soit plus grand ou plus petit ?

```python
from sklearn.feature_extraction.text import CountVectorizer

#to take max features(columns), 2500
cv = CountVectorizer(max_features=2500)

#converting to array
x = cv.fit_transform(corpus).toarray()

#dependent variable
y = df['v1']
```

5. Transformation de la variable cible

```
## y is a categorical variable so will encode it
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

Exercice 2 – Modélisation

1. Définition de la base d'apprentissage et test

```
## split into train and test set
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

2. Apprentissage d'un modèle de classification
   Quel autre modèle pourriez-vous proposer ?

```
## training the model
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB() # using naive bayes classification algorithm
model.fit(x_train,y_train) # fitting the model
```

3. Prédiction et calcul des performances

```
## predicting the values
y_pred = model.predict(x_test)

#score of the model
model.score(x_test,y_test)

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
```

4. Afficher la performance

```
accuracy_score(y_test, y_pred)
```