Analyse des opinions sous twitter

1 Objectif

L'objectif de ce TP est d'analyser un corpus de tweets en fonction des opinions exprimées (positif/-négatif). Le langage à utiliser est Python.

Rendu attendu : un notebook (aux deux formats .ipynb et .pdf) qui, en plus du code et de ces résultats, présente la méthode et les choix que vous avez faits ainsi qu'une discussion des résultats.

2 Documentation : Python et NLTK - Natural Language processing Toolkit

Vous aurez sûrement besoin d'une documentation sur NLTK :

— Le site de NLTK : http://www.nltk.org/

— La notice d'utilisation de l'interface de WordNet pour NLTK : http://www.nltk.org/howto/wordnet.html

— La notice d'utilisation du PoS tagger : http://www.nltk.org/book/ch05.html

... et sur WordNet et SentiWordNet :

— Le site web de WordNet : http://wordnet.princeton.edu/

— Le site web de SentiWordNet : http://sentiwordnet.isti.cnr.it/

— Le site de Christopher Potts, qui explique comment utiliser WordNet avec NLTK et fournit un script pour pouvoir utiliser SentiWordNet avec NLTK : http://compprag.christopherpotts.net/wordnet.html

3 Implémentation d'un système de détection d'opinions dans les tweets

3.1 Matériel : Présentation du corpus de tweets

Les tweets à analyser sont disponibles à l'adresse suivante : https://clavel.wp.imt.fr/files/2018/05/testdata.manual.2009.06.14.csv_.zip. Cette base (Sentiment140) a été obtenue sur le site de l'université de Stanford http://help.sentiment140.com/for-students. Un extrait en est donné dans le tableau 1. La base contient 498 tweets annotés manuellement. La base propose 6 champs correspondant aux informations suivantes :

1. la polarité du tweet : Chaque tweet est accompagné d'un score pouvant être égal à 0 (négatif), 2 (neutre) ou 4 (positif).
2. l'identifiant du tweet (2087)
3. la date du tweet (Sat May 16 23 :58 :44 UTC 2009)
4. la requête associée (lyx). Si pas de requête la valeur est NO_ QUERY.
5. l'utilisateur qui a tweeté (robotickilldozr)
6. le texte du tweet(Lyx is cool)

Sentiment	Tweet
4	@stellargirl I loooooooovvvvvveee my Kindle2. Not that the DX is cool, but the 2 is fantastic in its own right.
4	Reading my kindle2... Love it... Lee childs is good read.
4	Ok, first assesment of the # kindle2 ...it fucking rocks!!!
0	Fuck this economy. I hate aig and their non loan given asses.
0	@ludajuice Lebron is a Beast, but I'm still cheering 4 the A..til the end.
2	Check this video out – President Obama at the White House Correspondents' Dinner http ://bit.ly/IMXUM
2	need suggestions for a good IR filter for my canon 40D ... got some ? pls DM

Table 1 – Extrait du jeu de données utilisé pour les tests

3.2 Prétraitements

Les tweets contiennent des caractères spéciaux susceptibles de nuire à la mise en place des méthodes d'analyse d'opinions. Ecrire un programme permettant pour chaque tweet de :

- récupérer le texte associé
- segmenter en tokens
- supprimer les urls
- nettoyer les caractères inhérents à la structure d'un tweet
- corriger les abréviations et les spécificités langagières des tweets à l'aide du dictionnaire DicoSlang (fichier `SlangLookupTable.txt` disponible ici : https://clavel.wp.imt.fr/files/2018/06/Lexiques.zip), encodage du fichier : latin1

Algorithme 1 : prétraitement

Data : $tweet = "word1 + word2 + word3 + ... + wordN"$ a string of words ; ; $dicoSlang =$ Dictionary which links abbreviations and their relative words

Result : $preProcessedTweet = [(word1), (word2), ...(wordN)]$

Code python utile

Les packages utiles (pour cette question et celles d'après) :

```
import os
import csv as csv
import nltk as nltk
import re
from nltk.corpus import wordnet as wn
```

Si nécessaire, réalisez la commande suivante sur la console :

```
nltk.download
```

Pour la lecture de fichier csv :

```
    dataFile = open(csvTweetData, 'rb')
    dataReader = csv.reader(dataFile, delimiter=',')
```

La fonction tokenize de nltk

```
words = nltk.word_tokenize(text)
```

Penser à utiliser les dictionnaires python.

Vous préciserez dans le CR le nombre d'occurrences des caractères inhérents à la structure du tweet et le nombre d'occurrences des 'hash-tags' dans le corpus.

3.3 Etiquetage grammatical

Développer une fonction capable de déterminer la catégorie grammaticale (POS : Part Of Speech) de chaque mot du tweet en utilisant la commande suivante de la libraire nltk :

```
taggedData = nltk.pos_tag(tokens)
```

Algorithme 2 : étiquetage grammatical

Data : $preProcessedTweettweet = "word1 + word2 + word3 + ... + wordN"$ a string of words ;
Result : $taggedTweet = [(word1, PoS1), (word2, PoS2), ...(wordN, PoSN)]$

Vous préciserez dans le CR le nombre de mots étiquetés verbes dans le corpus.

3.4 Algorithme de détection v1 : appel au dictionnaire Sentiwordnet

NLTK dispose entre autre d'une interface pour manipuler la base de données *WordNet*. Ainsi, après installation de *NLTK* et du package *WordNet*, un utilisateur peut accéder à l'ensemble des synsets qui sont liés à un mot donné à l'aide d'une commande simple sous Python. Observez son fonctionnement à l'aide des lignes de code suivantes :

```
>>> from nltk.corpus import wordnet as wn

>>> wn.synsets('dog')
```

SentiWordNet est une extension de *WordNet* (fichier SentiWordNet_3.0.0_20130122.txt disponible ici : https://clavel.wp.imt.fr/files/2018/06/Lexiques.zip).

Vous aurez également besoin de télécharger dans votre répertoire de travail le fichier suivant http://compprag.christopherpotts.net/code-data/sentiwordnet.py

Pour cette étape, vous devez développer un programme permettant :

- de récupérer uniquement les mots correspondant à des adjectifs, noms, adverbes et verbes
- d'accéder aux scores (positifs et négatifs) des synsets dans la librairie *NLTK*. Ce script définira dans une classe Python l'objet SentiSynset sur le même modèle que le Synset développé dans *NLTK* pour *WordNet*, et permettra de lire le tableau de *SentiWordNet* comme suit.

```
>>> from sentiwordnet import SentiWordNetCorpusReader, SentiSynset
>>> swn_filename = 'SentiWordNet_3.0.0_20100705.txt'
>>> swn = SentiWordNetCorpusReader(swn_filename)

>>> swn.senti_synset('breakdown.n.03')
breakdown.n.03 PosScore: 0.0 NegScore: 0.25
```

- de calculer pour chaque mot les scores associés à leur premier synset,
- de calculer pour chaque tweet la somme des scores positifs et négatifs des SentiSynsets du tweet,
- de comparez la somme des scores positifs et des scores négatifs de chaque tweet pour décider de la classe à associer au tweet.

Algorithme 3 : Getting sentiment from a preprocessed tweet (Version 1)

Data : $tweet = [(word1, PoS1), (word2, PoS2), ...(wordN, PoSN)]$ where words have been previously preprocessed ; $acceptedPoSList$ = List of PoS words taken into account
Result : $[posScore, negScore, sentiment]$

Vous préciserez dans le CR le nombre de tweets positifs correctement détectés avec cette version de l'algorithme.

3.5 Algorithme de détection v2 : gestion de la négation et des modifieurs

Vous aurez besoin de : la liste des mots en anglais correspondant à des négations (fichier NegatingWordList.txt disponible ici : https://clavel.wp.imt.fr/files/2018/06/Lexiques.zip) et celle correspondant aux modifieurs (fichier BoosterWordList.txt disponible ici : https://clavel.wp.imt.fr/files/2018/06/Lexiques.zip). Pour chaque mot, l'algorithme doit effectuer les opérations suivantes :

- multiplie par 2 le score négatif et le score positif associés au mot si le mot précédent est un modifieur ;
- utilise uniquement le score négatif du mot pour le score positif global du tweet et le score positif du mot pour le score négatif global du tweet si le mot précédent est une négation.

Vous préciserez dans le CR le nombre de tweets positifs, négatifs et neutres correctement détectés avec cette version de l'algorithme et le nombre de termes négatifs contenus dans les tweets positifs.

3.6 Algorithme de détection v3 : gestion des emoticons

Vous avez ici besoin du dictionnaire d'émoticons est disponible (fichier EmoticonLookupTable.txt disponible ici : https://clavel.wp.imt.fr/files/2018/06/Lexiques.zip). Cet algorithme demande en entrée deux listes supplémentaires : une liste d'emoticons positifs et une liste d'émoticons négatifs. Les émoticons positifs rencontrés augmentent de 1 la valeur du score positif du tweet, tandis que les émoticons négatifs augmentent de 1 la valeur du score négatif du tweet.

Vous préciserez dans le CR le nombre de tweets positifs, négatifs et neutres correctement détectés avec cette version de l'algorithme et le nombre d'emoticons présents dans le corpus.

3.7 Votre version : v4

En analysant les sorties des algorithmes proposés précédemment, proposez votre propre algorithme d'analyse des opinions dans les tweets et les performances que vous obtenez.