Cours : Traitement Automatique du Langage Naturel

Promo : Master 2 Polytech

Auteur : Amir HAZEM

Date : Novembre 2018

TP 1 : Prétraitements et Expressions Régulières

Préambule :

Le but de l'ensemble des 6 séances de TP est de réaliser une chaîne complète de traitement automatique de documents écrits. Le cas d'usage sera l'analyse de sentiments et plus particulièrement la tâche de détection de langage figuratif et non figuratif dans des tweets en français. Pour ce faire, on s'appuiera sur le défi fouille de textes (Deft2017) https://deft.limsi.fr/2017/. **Votre code source devra impérativement être commenté !!!**

Suivez le modèle fourni ici : http://www.amirhazem.ovh/teaching.html

Travail à faire :

Dans ce TP, nous nous intéresserons dans un premier temps à l'étape de prétraitement des données. Il vous est demandé de réaliser un script python (version 2.7.X) qui permettra de générer à partir d'un corpus brut, une version tokenisée. À partir de cette version tokenisée, vous devrez ajouter les fonctions de lemmatisation, de racination et d'étiquetage morphosyntaxique.

Dans un second temps, nous nous intéresserons à l'analyse du contenu des corpus. Ainsi, il vous est demandé de remplir un tableau de statistiques sur les données du corpus tel que : la fréquence des n-grammes (n=[1, 5]), le nombre de noms, de verbes, d'adjectifs, la longueur des tweets en nombre de mots et de caractères, etc. Certaines de ces statistiques, comme la longueur des tweets seront à mettre en correspondance avec les étiquettes des trois tâches. Par exemple, la longueur moyenne des tweets positifs, etc. Enfin, nous nous intéresserons aux expressions régulières pour extraire certaines informations, par exemple : les urls, les émoticônes, les hashtags (#), les dates, la ponctuation (!, ?), les abréviations, etc. Ces informations seront à rajouter à votre tableau de statistiques. Le but de l'observation du corpus et de l'extraction des statistiques est de vous donner des pistes pour choisir les traits discriminants pour vos classifieurs.

Exercice1 :

Récupérer et consulter le corpus DEFT2017 (http://www.amirhazem.ovh/teaching.html)
Vous devrez avoir à votre disposition les fichiers suivants : *task1-train.csv, task2-train.csv et task2-train.csv.* Une fois le corpus téléchargé et consulté, effectuez les tâches suivantes :

1- Tokenisation

Chaque phrase du corpus d'entraînement devra être tokenisée.
Votre script prendra en entrée un fichier d'entraînement (par exemple : task1-train.csv) et produira en sortie un fichier tokenisé (par exemple : task1-train.csv**.tok**)
Vous devrez utiliser deux fonctions de tokenisation en vous appuyant sur la librairie NLTK.
Une première tokenisation classique (*nltk.word_tokenize(phrase)*) et une seconde dédiée aux tweets (*nltk.tokenize.TweetTokenizer(phrase)*). En tout, vous aurez à générer 6 fichiers tokenisés.

2- Lemmatisation, Racination (Stemming) et Étiquetage Morphosyntaxique

2-1 Lemmatisation et Racination

Pour chaque fichier du corpus d'entraînement et à partir du corpus préalablement tokenisé, fournir une version lemmatisée et une version racinée de chaque corpus.
Votre script prendra en entrée par exemple le fichier task1-train.csv**.tok** et produira en sortie les fichiers task1-train.csv**.tok.lem** et task1-train.csv**.tok.stem.**
Pour visualiser les résultats de certains algorithmes de racination, vous pouvez vous appuyer sur le lien suivant : https://text-processing.com/demo/stem/

2-2 Étiquetage Morphosyntaxique

Effectuer l'étiquetage morphosyntaxique sur les fichiers tokenisés du corpus d'entraînement. Pour ce faire, vous vous appuierez sur la fonction pos_tag de la librairie NLTK.
Vous constaterez que cette fonction n'est pas adaptée pour le français. De plus, python 2 ne permet pas une lemmatisation efficace du Français. c'est pourquoi vous aurez besoin d'installer TreeTagger (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/), un outil qui permet entre autres la tokenisation et l' étiquetage morphosyntaxique sur plusieurs langues.

Prérequis :

Installer le wrapper treetagger pour python en utilisant la commande suivante :

- *pip install treetaggerwrapper*
- Indiquer les paramètres du wrapper (TAGDIR,TAGLANG,TAGINENC,TAGOUTENC)
- Renommer french.par et french-abbreviations par french-utf8.par et french-abbreviations-utf8

Appuyez vous sur la documentation qui se trouve ici :

https://treetaggerwrapper.readthedocs.io/en/latest/

Votre script prendra en entrée par exemple le fichier task1-train.csv.tok et produira en sortie le fichier task1-train.csv**.tok.postag.**

Chaque opération tiendra compte des deux types de tokenisation (classique et celle dédiée aux tweets). Ainsi vous aurez 6 fichiers lemmatisés, 6 fichiers racinés et 6 fichiers accompagnés de leur étiquetage morphosyntaxique.

3- Filtrage des mots Outils et Extraction de Lexiques de Sentiments

3-1 Filtrage

À partir des fichiers .csv, .tok, .lem, .stem et .postag, générer une version où chaque phrase aura été préalablement filtrée, c'est-à-dire, chaque phrase ne devra contenir que les mots pleins (porteur de sens). Vous vous appuierez sur le fichier stopwords_fr.txt que vous trouverez ici : http://www.amirhazem.ovh/teaching.html

3-2 Extraction d'un lexique de sentiments

À l'aide des étiquettes de chaque tâche , extraire automatiquement un lexique de sentiments.

Pour ce faire, considérer comme lexique positif les mots qui ont une forte probabilité d'apparition dans des tweets étiquetés comme positifs, etc. Ainsi, un calcul de probabilités pour chaque paire : (mot, étiquette) est préalablement requis.

Exercice2 :

Afin d'appréhender au mieux la tâche de classification, l'observation des données constitue un élément-clé, c'est pourquoi il vous est demandé dans ce qui suit de produire un ensemble d'observations statistiques sur les données d'entraînement.

1- Statistiques sur les données

Les statistiques devront être d'abord fournies par tâche, puis dans un tableau global pour les traits qui peuvent être généralisés. Extraire à partir du corpus d'entraînement les informations suivantes :

- la fréquence des n-grammes (n=[1, 5]) après filtrage des mots outils.
- le nombre de noms, de verbes, d'adverbes et d'adjectifs.
- les mots pleins qui sont utilisés dans chaque type de langage, à savoir : quels sont les noms, les verbes et les adjectifs que l'on retrouve le plus souvent dans un langage figuratif par exemple. Donner la fréquence pour chaque étiquette.
- le nombre de tweets contenant des points d'exclamation et d'interrogation ainsi que la fréquence pour chaque étiquette.
- la longueur des tweets en nombre de mots et de caractères.
- le TfxIdf sur les n-grammes.

Libre à vous d'ajouter des traits que vous jugerez pertinents pour les trois tâches.

2- Expressions régulières

Extraire à l'aide d'expressions régulières les informations suivantes :

- les urls
- les émoticônes
- les hashtags (#) et arobases (@)
- les dates
- les abréviations
- des ponctuations consécutives (!!!, ??, etc.)
- les mots dont certains caractères sont répétés plus de deux fois (Noooon par exemple)
- à l'aide de certains marqueurs tels que la majuscule ou l'étiquetage morphosyntaxique, extraire les entités nommées du corpus d'entraînement.
- dégager des corrélations entre l'utilisation des entités nommées et le type de langage.

Compléter vos tableaux de statistiques avec les informations extraites grâce aux expressions régulières.

Libre à vous d’ajouter des traits que vous jugerez pertinents pour les trois tâches.

À rendre **(travail à effectuer par binôme en présentiel pendant les séances de TP)**

- Un script contenant les différentes fonctions implémentées.
  Chaque fonction devra être commentée et accompagnée d’un exemple d’exécution.
- Un compte rendu succin contenant les résultats statistiques extraits à partir de votre script.
  Chaque résultat mis dans le tableau doit pouvoir être recalculé via votre script.
  Veillez à ce que chaque fonction réponde à une seule question.
- Complétez votre script par un menu qui permettra de choisir l’action à effectuer.