Prof. Amal Zouaq – INF8460 (Automne 2020)
Chargé de laboratoire : Antoine Lefebvre-Brossard et Dylan Favacque

École Polytechnique de Montréal

Département Génie Informatique et Génie Logiciel

INF8460 – Traitement automatique de la langue naturelle

Objectifs d'apprentissage

- Savoir accéder à un corpus, le nettoyer et effectuer divers pré-traitements sur les données
- Savoir effectuer une classification automatique des textes pour l'analyse de sentiments
- Évaluer l'impact des pré-traitements sur les résultats obtenus

Logiciels

Anaconda (https://www.anaconda.com/products/individual), Jupyter notebook; NLTK; ScikitLearn

Lectures/ Ressources

http://www.nltk.org/

https://scikit-learn.org/stable/

Modalités de remise du TP

La date et heure de remise sont spécifiées sur Moodle. La remise se fait généralement à midi le lundi précédant votre prochain labo. Veuillez lire **attentivement** la politique de retard dans le plan de cours.

Vous devez soumettre votre code Python sous forme de ipython notebook.

Critères d'évaluation

- La réponse correcte à chaque question (code, sortie)
- La qualité du code incluant les commentaires

Corpus et code

Dans le fichier archive `inf8460_A20_TP1.zip`, vous trouverez :

- Le squelette de code à compléter: INF8460_A20_ TP1.ipynb
- Les données dans le répertoire `corpus`: RtGender: https://nlp.stanford.edu/robvoigt/rtgender/

Les données sont séparées en deux fichiers `train.csv` et `test.csv`, tous deux tirés du corpus que vous pouvez trouver sur le lien ci-dessus. Il s'agit de textes provenant de réseaux sociaux qui sont annotés avec les labels «Positive » et « Negative » pour indiquer un sentiment positif ou négatif.

Travail à faire (/100 points)

1. Pré-traitement et exploration des données (40 points)

Lecture et pré-traitement (14 points)

Dans cette section, vous devez compléter la fonction `preprocess_corpus` qui doit être appelée sur les fichiers `train.csv` et `test.csv`. La fonction `preprocess_corpus` appellera les différentes fonctions créées ci-dessous. Les différents fichiers de sortie doivent se retrouver dans le répertoire `output`. Chacune des sous-questions suivantes devrait être une ou plusieurs fonctions.

1) *(2 points)* Segmentez chaque corpus en phrases, et stockez-les dans un fichier `<nomcorpus>_phrases.csv` (une phrase par ligne)
2) *(2 points)* Normalisez chaque corpus au moyen d'expressions régulières en annotant les négations avec `_Neg` L'annotation de la négation doit ajouter un suffixe _NEG à chaque mot qui apparait entre une négation et un signe de ponctuation qui identifie une clause. Exemple :
   No one enjoys it. no one_NEG enjoys_NEG it_NEG .
   I don't think I will enjoy it, but I might. i don't think_NEG i_NEG will_NEG enjoy_NEG it_NEG, but i might.
3) *(2 points)* Segmentez chaque phrase en mots (*tokenisation*) et stockez-les dans un fichier `<nomcorpus>_mots.csv.` (Une phrase par ligne, chaque token séparé par un espace, il n'est pas nécessaire de stocker la phrase non segmentée ici) ;
4) *(2 points)* Lemmatisez les mots et stockez les lemmes dans un fichier `<nomcorpus>_lemmes.csv` (une phrase par ligne, les lemmes séparés par un espace) ;
5) *(2 points)* Retrouvez la racine des mots (*stemming*) en utilisant `nltk.PorterStemmer()`. Stockez-les dans un fichier `<nomcorpus>_stems.csv` (une phrase par ligne, les racines séparées par une espace) ;
6) *(2 points)* Ecrivez une fonction qui supprime les mots outils (stopwords) du corpus. Vous devez utiliser la liste de stopwords de NLTK ;
7) *(2 points)* Écrivez une fonction `preprocess_corpus(corpus)` qui prend un corpus brut stocké dans un fichier `.csv`, effectue les étapes précédentes, puis stocke le résultat de ces différentes opérations dans un fichier `corpus _norm.csv`

Exploration des données (26 points)

1) Complétez les fonctions retournant les informations suivantes (une fonction par information, chaque fonction prenant en argument un `corpus` composé d'une liste de phrases segmentées en jetons (tokenization)) ou une liste de genres et une liste de sentiments:
   - **a)** *(2 point)* Le nombre total de jetons (mots non distincts)
   - **b)** *(2 point)* Le nombre total de types
   - **c)** *(2 point)* Le nombre total de phrases avec négation
   - **d)** *(2 point)* Le ratio token/type
   - **e)** *(2 point)* Le nombre total de lemmes distincts
   - **f)** *(2 point)* Le nombre total de racines (stems) distinctes
   - **g)** *(2 points)* Le nombre total de documents (par classe)
   - **h)** *(2 points)* Le nombre total de phrases (par classe)
   - **i)** *(2 points)* Le nombre total de phrases avec négation (par classe)

j) *(2 points)* Le pourcentage de réponses positives par genre de la personne à qui cette réponse est faite (op_gender)

2) *(1 point)* Écrivez la fonction `explore(corpus, sentiments, genders)` qui calcule et affiche toutes ces informations, précédées d'une légende reprenant l'énoncé de chaque question (a,b, ….j).

3) *(5 points)* Calculer une table de fréquence (lemme, rang (le mot le plus fréquent a le rang 1 etc.) ; fréquence (le nombre de fois où il a été vu dans le corpus). Seuls les N mots les plus fréquents du vocabulaire (N est un paramètre) doivent être gardés. Vous devez stocker les 1000 premières lignes de cette table dans un fichier nommé `table_freq.csv`

2. Classification automatique (30 points)

Vous allez maintenant procéder à l'analyse de sentiments.

a) Classification automatique avec un modèle sac de mots (unigrammes), Naive Bayes et la régression logistique (15 points)

(7.5 points) Naive Bayes
(7.5 points) Régression logistique

En utilisant la librairie scikitLearn et l'algorithme *Multinomial Naive Bayes* et *Logistic Regression*, effectuez la classification des textes avec un modèle sac de mots unigramme pondéré avec TF-IDF. Vous devez entrainer chaque modèle sur l'ensemble d'entrainement et le construire à partir de votre fichier `corpus_train.csv`.

Construisez et sauvegardez votre modèle sac de mots avec les données d'entrainement en testant les pré-traitements suivants (séparément et en combinaison): tokenisation, lemmatisation, stemming, normalisation des négations, et suppression des mots outils. Vous ne devez garder que la combinaison d'opérations qui vous donne les meilleures performances sur le corpus de test.
Indiquez dans un commentaire les pré-traitements qui vous amènent à votre meilleure performance (voir la section 3 – évaluation). Il est possible que la combinaison optimale ne soit pas la même selon que vous utilisiez la régression logistique ou Naive Bayes.
On s'attend à avoir deux modèles optimaux, un pour Naive Bayes, et un avec régression logistique.

b) Autre représentation pour l'analyse de sentiments et classification automatique (15 points)

On vous propose maintenant d'utiliser une nouvelle représentation de chaque document à classifier. Vous devez créer à partir de votre corpus la table suivante :

Vocabulaire	**Freq-positive**	**Freq-négative**
happy	10	1
…	…	…

Où :

- Vocabulaire représente tous les types (mots uniques) de votre corpus d'entrainement

- Freq-positive : représente la somme des fréquences du mot dans tous les documents de la classe positive
- Freq-négative : représente la somme des fréquences du mot dans tous les documents de la classe négative

Notez qu'en Python, vous pouvez créer un dictionnaire associant à tout (mot, classe) une fréquence. Ensuite il vous suffit de représenter chaque document par un vecteur à 3 dimensions dont le premier élément représente un biais (initialisé à 1), le deuxième élément représente la somme des fréquences positives (freq-pos) de tous les mots uniques (types) du document et enfin le troisième élément représente la somme des fréquences négative (freq-neg) de tous les mots uniques du document.

En utilisant cette représentation ainsi que les pré-traitements suggérés, trouvez le meilleur modèle possible en testant la régression logistique et Naive Bayes.
Vous ne devez fournir que le code de votre meilleur modèle dans votre notebook.

3. Evaluation et discussion (20 points)

a) *(10 points)* Pour déterminer la performance de vos modèles, vous devez tester vos modèles de classification sur l'ensemble de test et générer vos résultats pour chaque modèle dans une table avec les métriques suivantes : Accuracy et pour chaque classe, la précision, le rappel et le F1 score. On doit voir cette table générée dans votre notebook avec la liste de vos modèles de la section 2 et leurs performances respectives.
b) *(6 points)* Générez un graphique qui représente la performance moyenne (mean accuracy – 10 Fold cross-validation) de vos différents modèles par tranches de 500 textes sur l'ensemble d'entrainement.
c) *(4 points)* Que se passe-t-il lorsque le paramètre de régularisation de la régression logisque (C) est augmenté ?

4. Analyse et discussion (10 points)

En considérant les deux types de représentations, répondez aux question suivantes en reportant la question dans le notebook et en inscrivant votre réponse:

a) *(3 points)* Quel est le meilleur modèle de classification en général? Pour la classe positive et négative ?
b) *(2 points)* Quel est l'impact de l'annotation de la négation ?
c) *(2 points)* La suppression des stopwords est-elle une bonne idée pour l'analyse de sentiments ?
d) *(3 points)* Le stemming et/ou la lemmatisation sont-ils souhaitables dans le cadre de l'analyse de sentiments ?

5. Contribution (0 points)

Complétez la section en haut du notebook indiquant la contribution de chaque membre de l'équipe en indiquant ce qui a été effectué par chaque membre et le pourcentage d'effort du membre dans le TP.