Projet: Fouille d'opinions dans les commentaires de clients

Cours de fouille de textes - Master 2 MIASHS/SSD - S. Aït-Mokhtar

8 décembre 2021

La date limite pour le rendu du projet est le **15 janvier 2022**.

1 Introduction

Le thème du projet est la fouille d'opinions dans les textes. L'objectif est l'implémentation d'un classifieur qui classifie les phrases d'avis sur des restaurants en 3 classes possibles : « positive » si la phrase exprime une opinion ou un sentiment positif, « negative » si la phrase exprime un sentiment négatif, ou « neutral » si elle ne contient pas d'opinion, ou si elle exprime des sentiments mixtes (positifs et négatifs). Le classifieur doit produire une seule classe par phrase.

Exemples de phrases correctement classifiées :

positive	*Un restau au bord de l'eau c'est sympa.*
negative	*Un tartare de saumon avec trop d'oignons.*
neutral	*C'est presque par hasard que nous avons dîné dans ce restaurant.*
neutral	*Les produits sont bons mais la cuisine reste simple par rapport aux tarifs.*

Le rendu sera évalué en prenant en compte les éléments suivants donnés par ordre d'importance :

- Sa précision (pourcentage de phrase classifiées correctement / nombre de phrase)
- Son efficacité computationnelle (vitesse d'entraînement et de prédiction, mémoire requise).

Des points de pénalités seront appliqués sur la note finale en cas de retard dans l'envoi du rendu de projet (1 point par jour de retard), ou si le programme ne fonctionne pas (au minimum 3 points de pénalités – et la valeur exacte dépendra de l'effort requis et du nombre d'échanges pour le faire fonctionner.)

2 Rendu du projet

Le rendu doit être sous la forme d'**un seul fichier compressé au format zip**. Merci de l'envoyer à l'adresse **sacours@outlook.com** : en pièce jointe si le fichier n'excède pas 5 Mo. Dans le cas contraire, merci de déposer le fichier zip sur un site de stockage en ligne (OneDrive, Google Drive, Dropbox, etc.)

et de m'envoyer par courriel un lien de téléchargement qui ne nécessite pas de s'enregistrer ou de s'identifier sur le serveur de téléchargement.

Lorsque le fichier du rendu est décompressé, le répertoire racine doit contenir les éléments suivants (pour plus de détails, consulter la section 5) :

Elément	Description
README.txt	Un fichier texte pur qui doit contenir les information suivantes : 1. Le(s) nom(s) complet(s) de ou des auteur(s) du rendu (max=2 auteurs) 2. Un ou deux paragraphes décrivant le classifieur (hyper-paramètres, type de représentation, type d'architecture, ressources éventuellement utilisées etc.) 3. La précision moyenne que vous obtenez sur les données de dev.
src	Sous-répertoire contenant TOUS les fichiers de code python nécessaires à l'exécution du projet avec la commande « python tester.py »)
resources	(optionel) : sous-répertoire contenant les ressources additionnelles que vous utilisez (en dehors du fichier des word embeddings distribué)

3 Installations à faire

Remarque : dans le cadre de ce projet, seules sont autorisées les librairies qui sont présentes dans la version de base d'anaconda et celles qui sont listées ci-dessous. Afin de ne pas altérer votre environnement anaconda habituel (si vous avez déjà installé anaconda avant ce projet), il est conseillé de créer un nouvel environnement spécifique et de l'activer avant de procéder aux installations des librairies.

1. Installer anaconda avec python $\geq$ 3.8.x
2. Une fois anaconda installé, **lancer un terminal de commandes *en mode administrateur***. La figure 1 indique comment procéder sur une machine Windows.

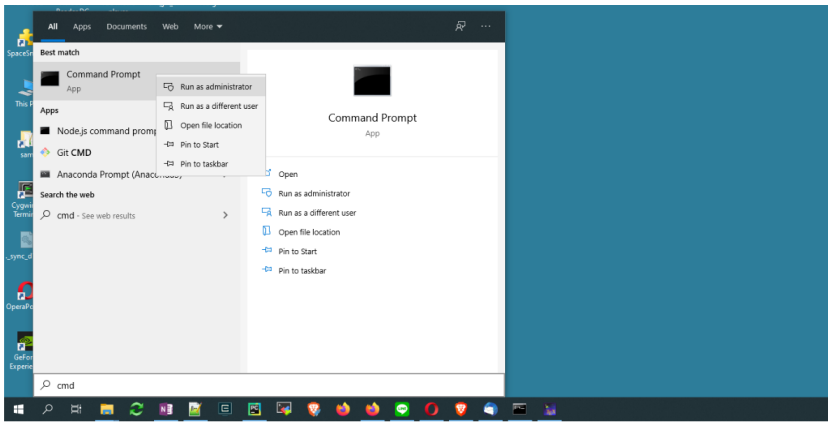

FIGURE 1 – Lancer un terminal de commandes en mode administrateur sur Windows

3. Installer tensorflow 2.3.1 : dans le terminal de commandes, taper la commande suivante :

```
pip install tensorflow==2.3.1
```

4. Installer gensim 3.8.0 :

```
conda install gensim=3.8.0
```

5. Installer la librairie transformers 3.1.0 de huggingface :

```
pip install transformers==3.1.0
```

6. Installer sentencepiece 0.1.91 :

```
pip install sentencepiece==0.1.91
```

7. Installer spaCy 2.3.2 et son modèle pour le français avec les deux instructions suivantes :

```
conda install spacy=2.3.2
python -m spacy download fr
```

4 A télécharger

Dans le répertoire partagé du cours sur OneDrive, le sous-répertoire **tp** contient des éléments à télécharger :

— Ce document (**instructions_projet.pdf**) qui contient les instructions pour réaliser le projet.
— Un sous-répertoire « data » qui contient les données d'entraînement et de développement. Les fichiers de données fournis (**frdataset1_train.csv** (données d'entrainement) et **frdataset1_dev.csv** (données de développement et validation)) ont le format des exemples ci-dessus (section Introduction) : chaque ligne contient une phrase avec sa classe correcte, les deux champs étant séparés par un caractère de tabulation.
— Un sous-répertoire **src** qui contient des fichiers de code sur lesquels votre contribution sera basée.

— Un sous-répertoire **resources** qui contient un fichier de plongements de mots (word embeddings de type word2vec) pour le français que vous pouvez éventuellement exploiter :

`frWac_non_lem_no_postag_no_phrase_200_skip_cut100.bin`

Si votre classifieur utilise d'autres ressources (par exemple liste de mots grammaticaux (stop words), ou liste de mots de polarité), c'est dans ce sous-répertoire **resources** qu'il faudrait les mettre. Notez que c'est **resources** avec un seul *s* (en anglais).

5 Comment procéder

1. Vous pouvez travailler soit individuellement soit en binôme.
2. Créer un répertoire pour le projet (par exemple **tp**) sur votre ordinateur et mettez dedans tout le contenu du sous-répertoire **tp** du dossier partagé du cours (voir section 4 ci-dessus)
3. Le sous-répertoire **data** contient les 2 fichiers de données, l'un pour l'entrainement du modèle et l'autre pour le développement et la validation (voir section 4)
4. Le sous-répertoire **src** contient 2 fichiers de code importants : **tester.py** et **classifier_template.py**
5. Vous ne devez pas modifier **tester.py**, il contient du code pour lancer l'exécution du projet : l'entraînement de votre classifieur sur les données d'entraînement, puis l'évaluation de sa précision sur les données de développement. Si vous le modifiez, votre programme ne fonctionnera pas au moment de l'évaluation du rendu de projet, car seule la version originale de **tester.py** sera utilisée.
6. Faire une copie locale de **classifier_template.py** et appelez-la **classifier.py**.
7. Vous devez travailler sur le fichier de code **classifier.py**, qui, initialement, contient juste le squelette de la classe Classifier (qui représente le classifieur). Ce squelette contient trois fonctions : **__init__** (appelée lors de la création d'une instance de la classe Classifier), **train** (appelée pour effectuer l'entrainement du classifieur avec les données fournies en paramètres) et **predict** (appelée pour faire des prédictions sur des textes). Ces trois méthodes sont appelées par le programme principal **tester.py**
8. Vous ne pouvez pas modifier la signature de ces trois méthodes (sinon, votre programme ne fonctionnera pas), mais vous devez écrire leur code pour que le classifieur fonctionne. Vous pouvez également rajouter d'autre fonctions ou variables à la classe Classifier si nécessaire.
9. Pour exécuter le projet une fois que vous avez terminé de coder le classifieur (dans **classifier.py**), lancez un terminal de commande, allez dans le sous-répertoire **src** du répertoire du projet, et tapez la commande suivante ("python" doit pointer sur le python que vous avez installé pour le projet, voir la section 3) :

```
python tester.py
```

Cette commande lancera par défaut 5 fois le processus d'entrainement-évaluation de votre classifieur et calculera à la fin le score moyen (sur les données de développement) ainsi que le temps d'exécution moyen. Une fois votre travail rendu, je l'évaluerai également sur les données de test (non distribuées). Les deux scores ainsi que le temps d'exécution seront pris en compte pour l'évaluation du rendu (avec un plus grand poids pour le score sur les données de test.)

10. **Si la commande précédente ne fonctionne pas, ou qu'elle produit des erreurs**, c'est que votre travail n'est pas encore prêt pour être rendu.
11. Ne pas modifier **tester.py** ! Votre classifieur doit fonctionner sans erreur en tapant simplement la commande "python tester.py"
12. Si vous avez besoin de créer d'autres fichiers de code, mettez-les dans **src** (toujours en vous assurant que l'exécution du projet se fait sans erreur).
13. Votre mission consiste donc à écrire le code de la classe **Classifier** pour obtenir de bonnes performances sur les données de dév. (et de test). Pour ce faire, vous pouvez choisir l'architecture neuronale, la nature et le nombre de couches, le nombre de neurones dans les couches, le taux d'apprentissage (learning rate), le nombre d'itérations (epochs), le type de vectorisation, les paramètres de la vectorisation (taille du vocabulaire, n-grammes, ...), le type de représentation (creuses en sac-de-mots, ou bien continues, ou hybrides), la tokénisation, la normalisation, le filtrage, l'analyse syntaxique avec spaCy, etc. Le modèle doit cependant être un modèle neuronal et écrit avec Tensorflow, comme les exemples vus en cours.
14. Seules les librairies python présentées dans la section 3, et évidemment celles fournies automatiquement par anaconda, sont autorisées.
15. Vous pouvez également utiliser des ressources (listes de mots polarisés, liste de "stopwords", textes non annotés etc.) pour améliorer la performance. Il faut dans ce cas les mettre dans le sous-répertoire "resources" (en anglais, donc un seul 's') et les fournir dans le fichier zippé de votre rendu. En outre, dans le code, **il faut les référencer avec des chemins relatifs** ("../resources/NomDeLaRessource") pour que la commande "python tester.py" fonctionne correctement si on la lance à l'intérieur du répertoire **src**, et ce quelle que soit la structure des répertoires au dessus du répertoire du projet.
16. **Le rendu doit être sous la forme d'un répertoire compressé au format zip** (pas de gz, bz ou autre format svp).
17. Le nom du fichier compressé doit contenir le nom de famille de l'auteur du rendu, ou les deux noms de familles des deux auteurs du rendu.
18. Avant de compresser votre projet et de l'envoyer, merci d'enlever le fichier des embeddings que je vous ai fourni dans le sous-dossier **resources** (il est gros et il n'est pas utile de l'inclure)