Projet: Fouille d'opinions dans les commentaires de clients

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

6 décembre 2022

La date limite pour rendre le projet est le **10 janvier 2023**.
Il est possible de travailler **seul ou au maximum à deux** sur le projet.

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 phrases classifiées correctement / nombre de phrases)
- 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**. Le nom de ce fichier compressé doit être composé du nom ou des noms

de famille des auteurs. 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 site en question.

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 informations 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 (type de représentation, type d'architecture, hyper-paramètres, 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 run_project.py », le fichier python "
resources	(optionnel) : sous-répertoire contenant les ressources additionnelles que vous utilisez (en dehors du fichier des word embeddings distribué)

3 Librairies Python autorisées

La liste des librairies Python autorisées est limitée à celle fournie dans le document "Installations à faire.pdf" distribué au début des séances de TP.

La librairie *datasets* de HuggingFace est également autorisée.

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 (ou validation). Les fichiers de données fournis (**frdataset1_train.csv** (données d'entrainement) et **frdataset1_dev.csv** (données de développement ou 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** : si votre classifieur utilise d'autres ressources (par exemple 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). Si vous avez besoin dans votre code de référencer un fichier dans ce répertoire, il faudrait utiliser un chemin relatif (et non absolu) afin que la commande **python run_project.py** puisse s'exécuter correctement dans le répertoire src.

5 Comment procéder

1. 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)
2. 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)
3. Le sous-répertoire **src** contient 3 fichiers de code importants : **run_project.py**, **review_dataset.py** et **classifier_bow.py**
4. Vous ne devez pas modifier **run_project.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 **run_project.py** sera utilisée.
5. Le fichier **classifier_bow.py** est un exemple de classifieur, utilisant les représentations sac-de-mots. Il définit une classe FTVectorizer dont le rôle est de vectorizer les textes et les labels, ainsi que la classe FTClassifier qui définit le modèle du classifieur en suivant le protocole de Pytorch Lightning. Faire une copie locale de **classifier_bow.py** et appelez-la **classifier.py**. Dans un terminal avec l'environnement python requis, allez dans le répertoire src et tapez la commande "python run_project.py" : le programme devra s'exécuter correctement : entraînement du modèle, puis calcul de la précision sur les données de validation, puis affichage des résultats.
6. Votre travail consiste essentiellement à modifier les classes FTVectorizer et FTClassifier dans le fichier **classifier.py**, et aussi les valeurs des hyperparamètres (classe HyperParameters) pour améliorer les performance du classifieur. Ne modifiez pas les noms, arguments et type de données retournées des méthodes de ces classes : en revanche modifiez les corps des méthodes pour changer l'architecture neuronale du modèle, la vectorization (qui doit être compatible avec le modèle), etc. Par exemple, vous pouvez modifier le modèle de classification en y intégrant un modèle de langage pré-entraîné de type transformer, en vous inspirant du notebook "transformer_classifier.py" distribué lors des séances de TP. Vous pouvez rajouter de nouvelles méthodes si vous en avez besoin, et même de nouvelles classes si nécessaire. Plus généralement, 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 continuesavec les transformers, ou bien une combinaison des deux, etc.), la tokénisation, la normalisation, le filtrage, l'analyse syntaxique avec spaCy, etc. Le modèle doit cependant être un modèle neuronal et écrit avec Pytorch et Pytorch Lightning, comme les exemples vus en cours.

7. Vous ne devez pas modifier **review_dataset.py**. Il définit la classe ReviewDataset adaptée aux données de la tâche : son rôle est de tokénizer et d'encoder en vecteurs les textes à classifier, en utilisant FTVectorizer qui se trouve dans **classifier.py**.
8. Pour exécuter le projet une fois que vous avez terminé de coder le classifieur et son vectoriseur (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) :

   ```
   python run_project.py
   ```

   Cette commande lancera 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.)
9. **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. Il faudra alors corriger les erreurs dans le fichier **classifier.py**.
10. Ne pas modifier **run_project.py** ! Votre classifieur doit fonctionner sans erreur en tapant simplement la commande "python run_project.py"
11. 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).
12. 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 run_project.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.
13. **Le rendu doit être sous la forme d'un répertoire compressé au format zip** (pas de gz, bz ou autre format svp). Le nom du fichier compressé doit être composé du nom de famille de l'auteur du rendu ou des deux noms de familles des deux auteurs du rendu séparés par un caractère _ (souligné).