Data Jay Colt

INSTRUCTIONS D’INSTALLATION

Installer anaconda qui contient python et les principales librairies de data science

<https://www.anaconda.com/download>

Nous allons créer un environnement virtuel puis installer toutes les librairies nécessaires au bon fonctionnement des fonctions.

Pour ça :

Ouvrir un anaconda prompt

se déplacer via ‘cd’ pour aller dans un dossier, ‘..’ pour revenir dans le dossier parent ou ‘dir’ pour visualiser les dossiers et fichiers contenus dans le dossier courant.

Aller dans le dossier “DJC-Wamiz\Code\”.

Créer un environnement virtuel avec la commande :

python -m venv venv\_name

Activer le en utilisant la fonction suivante :

venv\_name\Scripts\activate

Pour installer les librairies, il faut accéder au fichier requirement.txt

Pour cela, aller dans le dossier “DJC-Wamiz\Code\DJC-Wamiz-code\”.

Une fois dans le dossier utiliser la commande :

pip install -r requirements.txt

L’installation peut être assez longue car il y a beaucoup de librairies à installer.

Normalement vous êtes opérationnels pour vous servir des 5 fonctions relatives à l’analyse du forum.

Après utilisation, vous pouvez quitter l’environnement virtuel en tapant “deactivate”

Data Jay Colt

DESCRIPTION DES FONCTIONS

Pour ce qui est du clustering, j’ai fait 3 fonctions. La première sert à lancer un nouvel apprentissage ou un apprentissage sur d’autres catégories. Attention à ne pas effacer l’apprentissage actuel car chaque apprentissage nécessite un labelling manuel des clusters obtenus. Le travail a déjà été fait sur la catégorie “nourriture pour chats”, ce serait dommage de devoir recommencer. Pour une nouvelle catégorie, j’ai créé une fonction qui renvoie 15 messages du cluster afin de pouvoir faire ce labelling manuel sur les clusters obtenus pour les autres catégories testées. Une troisième fonction permet de trouver les clusters pour de nouvelles données.

La première fonction s’appelle Train\_Clustering\_Model.py

Elle prend en argument les chemins vers les fichiers de base de données (catégories, topics et posts), le numéro de la catégorie (259 pour la nourriture du chat) et un argument de type chaîne de caractères qui sera indiqué dans tous les fichiers générés pour ne pas confondre les apprentissages.

Exemple d’utilisation :

python Train\_Clustering\_Model.py -c ../../Data/community\_forum\_category.csv -t ../../Data/community\_forum\_topic.csv -p ../../Data/community\_forum\_post.csv -n 259 -l "on\_part\_1"

La seconde fonction permet à partir des fichiers générés par la première fonction d’observer les 15 premiers messages des clusters en vue de les annoter.

Elle s’appelle Look\_At\_Messages.py

Elle prend en argument les chemins vers les fichiers de base de données (catégories, topics et posts), le numéro de la catégorie, le numéro du cluster à observer, et un argument de type chaîne de caractères qui signe les fichiers générés par la première fonction (celle d’apprentissage).

Exemple d’utilisation :

python Look\_At\_Messages.py -c ../../Data/community\_forum\_category.csv -t ../../Data/community\_forum\_topic.csv -p ../../Data/community\_forum\_post.csv -n 259 -cn 113 -l "on\_part\_1"

La troisième fonction permet de catégoriser de nouveaux messages. Il suffit d’un fichier xlsx à mettre dans le dossier des données et il faut une colonne “messages”. La fonction associe les messages aux différents clusters et écrit un nouveau fichier. Il faut aussi la signature du modèle à utiliser.

Exemple d’utilisation :

python Get\_Cluster\_From\_Data.py -p ../../Data/mini\_new\_messages.xlsx -l "on\_part\_1"

Pour ce qui est de l’analyse des tendances, j’ai fait une fonction qui crée les courbes temporelles des messages de chaque cluster et les mets dans un dossier. On peut choisir l’année à laquelle on débute le calcul. Le premier point correspondra alors à 3 mois et finira à la date de fin du fichier moins 3 mois car la fenêtre temporelle gaussienne utilisée est de 6 mois. En dessous le bruit était trop important.

On peut utiliser cette fonction à l’aide de la commande suivante :

python Plot\_Temporal\_Curves.py -c ../../Data/community\_forum\_category.csv -t ../../Data/community\_forum\_topic.csv -p ../../Data/community\_forum\_post.csv -n 259 -ys 2020 -pi ../../Data/images/ -pl ../../Data/CluFin.xlsx -l ""

Le dossier “images” doit être créé dans le dossier data avant d’utiliser la fonction. S’il n’est pas déjà créé, il faut en créer un. Les arguments de cette fonction sont les csv de database de catégorie, topic et post, le numéro de la catégorie, l’année à partir de laquelle on commence l’analyse, le chemin vers le dossier images et le fichier avec les clusters annotés. Et aussi l'identifiant du modèle à utiliser.

La dernière fonction est une fonction d’analyse d’émotion d’un message. Elle s’utilise comme ça :

python Get\_Emotion.py -p ../../Data/mini\_new\_messages.xlsx -l ""

Elle a pour argument le chemin vers un fichier un fichier excel contenant une colonne messages. Elle crée un fichier avec en plus les émotions détectées et la probabilité ainsi que si oui ou non la probabilité est assez haute (le seuil n’est pas le même pour les messages positifs et négatifs). Le reste des messages est considéré comme neutre.