PARTIE I :

L'objectif du pipeline est de pouvoir regrouper l'ensemble des prétraitements et de pouvoir les faire suivre par le classifieur. Le principe consiste d'abord à mettre la chaîne de pré-traitement, ensuite mettre le classifieur et enfin utiliser directement le pipeline.

Il peut arriver que différentes combinaisons de pré-traitements puissent être utilisées. Les pipelines sont très importants lorsque l'on sauvegarde un modèle. En effet, comme ils prennent en compte les pré-traitements tout est sauvegardé et une fois un modèle appris, il est possible de le sauvegarder pour pouvoir lui appliquer d'autres données à prédire (deux librairies existent : pickle et joblib) et il est impératif que les nouvelles données suivent le même traitement. Cela veut dire que dans le cas de nouvelles données à évaluer avec un modèle lors de la prédiction, les données seront automatiquement transformées.

Pour une toute première classification, une première évaluation de la qualité de la prédiction peut se faire avec le calcul de l'*accuracy* (pourcentage de prédictions correctes). Il est indispensable de créer un jeu d'apprentissage sur lequel un modèle est appris et un jeu de test pour évaluer le modèle.

Dans le cas des réseaux sociaux, nous supposons qu'un utilisateur laisse un commentaire qui est non étiqueté, composé d'un certain nombre de mots. En considérant que les langages abusifs à détecter sont (sexisme, xénophobie et islamophobie) et un commentaire neutre. Soit S, X, I, N, quatre classes désignant respectivement sexisme, xénophobie, islamophobie et neutre dans un contenu textuel.

Les membres de ces classes sont les publications dont le contenu est classé comme appartenant à la classe correspondante. De plus, étant donné que l'utilisateur a un historique de messages postés, la sauvegarde du pipeline permettra de reconnaitre toutes commentaires précédents comme appartenant à l'une des classes S, X, I, N. De même, d'autres publications d'autres utilisateurs ont également été étiquetés en conséquence, formant leur historique précédent. Sur la base de ces faits, le problème est d'identifier la classe à laquelle appartient le commentaire non étiqueté de l'utilisateur.

Pour le prétraitement[1] : Normalisation (pour supprimer les ponctuations, majuscule, espace et les stopwords, ensuite vient la tokenisation qui consiste d'abord à séparer les mots, à lemmatiser ou standardiser selon la situation et la langue qu'on utilise puis vient la vectorisation qui consiste à traduire les mots en vecteur avec les bag of words.

[1] https://github.com/elhadj03/Machine-Learning/blob/master/Nettoyage.py.ipynb

Figure 1 : Processus de classification du scraping à la mise en place d'un pipeline

En appliquant notre modèle (le cas des réseaux sociaux) avec comme classifieur le LSTM, nous aurons après avoir fait le prétraitement le graphique suivant :

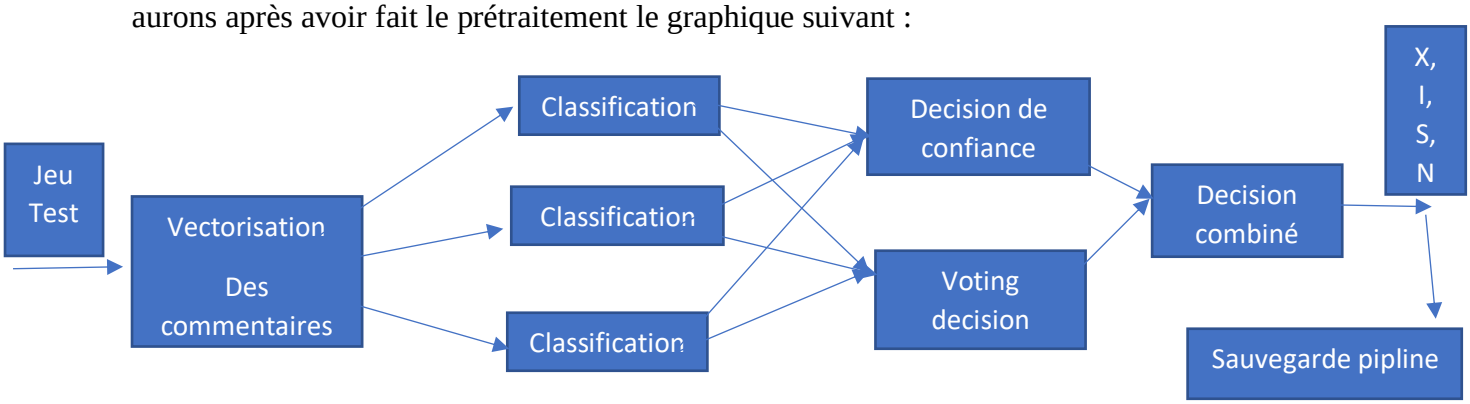

Figure 2 : Pipeline processus de détections de langages abusifs

Nous prenons le fichier test et en sortie il donne les résultats comme : Xénophobie ou Islamophobie ou Sexisme ou Neutre et qu'on enregistre dans le pipeline.

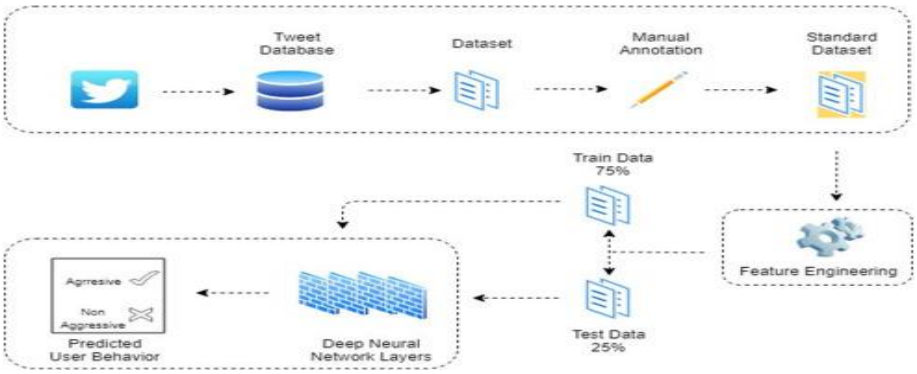

Figure 3 : Résumé d'un Processus de Depp Learning pour un tweet

PARTIE II

L'architecture du classificateur hybride se compose principalement de deux composants, à savoir : le réseau de neurones convolutifs (CNN) et la mémoire à long terme (LSTM). CNN est appliqué pour extraire des séquences de haut niveau de caractéristiques de mots. Alors que LSTM est embarqué pour capturer des séquences à long terme et réduit également le temps d'apprentissage du modèle proposé (il suppose que les éléments connus dans le passé ne le sont plus et qu'ils représentent le présent). Les données traitées (textuelles et non textuelles) sont introduites dans le modèle hybride CNN-LSTM où les hyperparamètres sont réglés dans une division standard. Une nouvelle approche est proposée qui fonctionne sous une structure en deux phases. Dans la première phase, les éléments textuelles et non textuelles sont introduits dans le modèle CNN-LSTM avec l'utilisation de vecteurs pré-entraînés pour identifier si le message choisi est abusif ou non. Dans la deuxième phase, les messages abusifs classés sont envoyés au classifieur comme par exemple le modèle de sujet Biterm (BTM) qui est un type de classificateur de texte court. Il utilise une co-occurrence de biterm de mots pour regrouper messages abusifs similaires afin d'identifier les langages abusifs (i.e : xénophobie, sexisme, islamophobie…).

Ainsi, après avoir extrait les données (commentaire, image, titre …) dans les sites (Facebook, LinkedIn, Instagram, Twitter, YouTube…) : le prétraitement des données textuelles consistera à normaliser le texte en utilisant les expressions régulières pour supprimer les ponctuations, les espaces, la conversion de tous les caractères majuscules en minuscules est aussi traitée. Alors que les données non textuelles sont traitées à l'aide de l'outil de reconnaissance optique de caractères (OCR) pour récupérer le contenu textuel, puis l'étape de prétraitement y est appliquée.

Une fois terminé le pipeline de prétraitement, l'étape suivante consiste à extraire les caractéristiques des données collectées. L'extraction d'entités à partir des données est appelée ingénierie d'entités. Cela se fait à l'aide de modèles d'intégration pré-formés. Pendant le processus d'incorporation, divers mots sont représentés à l'aide d'une représentation vectorielle dense. L'incorporation de mots est une amélioration par rapport à l'approche du bag of words (BOW) où chaque mot est représenté à l'aide de vecteurs pour représenter l'ensemble du vocabulaire. Les représentations sont rares car chaque mot est représenté par une ligne composée de valeurs nulles. Alors que dans une couche d'intégration, chaque mot est représenté par des vecteurs denses. Les vecteurs denses représentent la projection d'un mot dans un espace

vectoriel continu. La position d'un mot est représentée dans ces espaces vectoriels appris à partir du texte court du titre et est appelée comme son incorporation.

Les incorporations de mots sont utilisées pour fournir une meilleure représentation des caractéristiques vectorielles des mots. Gensim (bibliothèque PNL) fournit un wrapper étonnant pour adopter différents modèles d'incorporation de mots pré-entraînés, notamment GloVe et Word2vec. Nous pouvons également considérer que la couche d'intégration spécifie trois arguments qui sont définis pour la première couche cachée du modèle hybride CNN-LSTM : dimension d'entré (précise la taille du vocabulaire dans notre corpus collecté), dimension de sortie (spécifie la taille de l'espace vectoriel où les mots sont incorporés) et longueur d'entrée (spécifie la longueur des séquences d'entrée, c'est -à- dire la limite maximale de mots pour chaque titre ou commentaire).

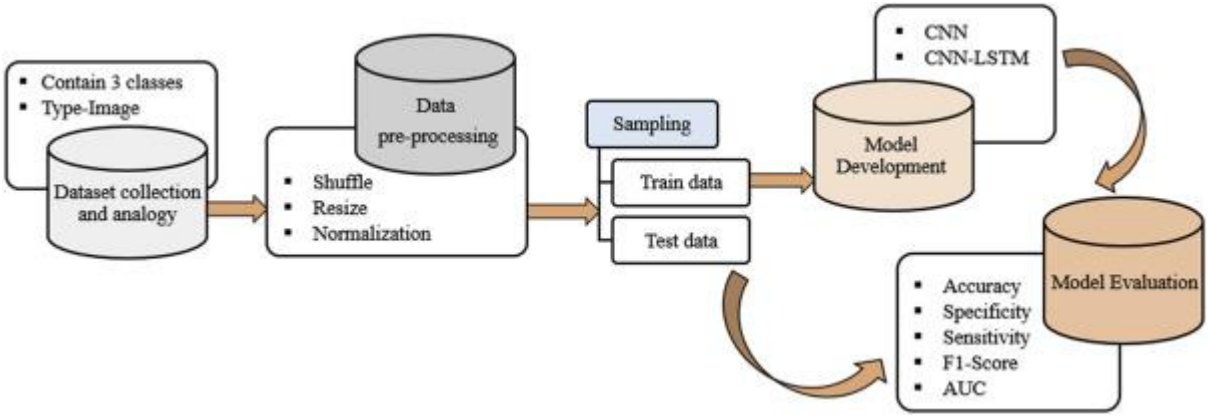

Figure 4 : Processus de détection d'un langage abusif avec le modèle CNN-LSTM

Bibliographie

Liu, T., Bao, J., Wang, J., & Zhang, Y. (2018). A Hybrid CNN–LSTM Algorithm for Online Defect Recognition of CO_2 Welding. *Sensors (Basel, Switzerland), 18*(12), 4369.

Mishra, P., Del Tredici, M., Yannakoudakis, H., & Shutova, E. (2019). Abusive language detection with graph convolutional networks. *arXiv preprint arXiv:1904.04073*.

Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016, August). Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)* (pp. 225-230).

Liens Utiles :

https://www.youtube.com/watch?v=6ciGTSrL-l4&ab_channel=PyConSG

https://github.com/karpathy/char-rnn

https://www.frontiersin.org/files/Articles/463661/fdata-02-00008-HTML/image_m/fdata-02-00008-g001.jpg