Doron Israel Elliot Muller Nicolas Wagner
doron.israel@dauphine.eu elliot.muller@dauphine.eu nicolas.wagner@dauphine.eu

Rapport projet final : Toxic Comment Classification

Intro :

Pour ce projet, on dispose d'un grand nombre de commentaires Wikipédia labellisés selon 6 types de toxicité par des évaluateurs humains.
Le but de ce challenge est de créer un modèle qui prédit une probabilité de chaque type de toxicité pour chaque commentaire.
La particularité est qu'il s'agit d'une tâche de classification multi-label : en effet, un même commentaire peut être classé dans plusieurs catégories.
Nous avons envisagé 2 méthodes pour réaliser cette tâche.

Preprocessing :

Nous nous sommes inspirés d'un TP abordé en cours utilisant la librairie re pour travailler avec expressions rationnelles. Nous avons créé une fonction appelée "clean_text" qui applique successivement les opérations usuelles de pre-processing de texte. La fonction convertit les commentaires en minuscules (lowercase), enlève la ponctuation, les nombres, les éventuels liens, etc.
Pour la tokenisation nous avons ensuite utilisé le Tokenizer de Pytorch (qui utilise spacy) pour construire notre vocabulaire. Nous avons volontairement décidé de ne pas procéder aux tâches de lemmatisation et au retrait des "stops words" qui dans certains cas peuvent se révéler utiles mais qui ici présentent peu d'intérêt et peuvent même affecter la performance du modèle car il s'agit d'une tâche de classification avec beaucoup de données. En effet, beaucoup d'articles sur le Machine Learning mentionnait cet aspect.

Modèle :

Le classifieur a la structure suivante :

- Une couche de Word Embedding, suivi d'un BI-LSTM, puis concaténation des deux états cachés finaux, suivi d'un Dropout, et d'une couche linéaire.

La couche de Dropout est là pour éviter l'overfitting en abandonnant certaines couches du réseau lors de l'entraînement du modèle.

Comme il s'agit d'un problème de classification multi label, nous avons tout d'abord pensé qu'après avoir concaténé les deux états finaux du BI-LSTM, nous les enverrions chacun dans 6 couches linéaires différentes afin de prédire chacun des labels. Cependant, nous avons vite abandonné cette idée à cause du risque d'overfitting. De plus, les labels n'étant pas indépendants, il n'est pas absurde de considérer un seul réseau linéaire pour les traiter.
L'embedding dimension est de 100 tout comme la dimension des états cachés du BI-LSTM.
Le dropout rate est à 0.5

Entraînement du modèle et résultats :

La loss utilisée pour entraîner notre modèle est la “binary cross entropy loss” avec l’algorithme d’optimisation ADAM avec un pas de 1e-3.
Le score pour sélectionner les paramètres du modèle est la moyenne du roc auc obtenu pour chaque label.

Dans un premier temps, le classifieur a été entraîné et évalué sur 5 epochs avec une taille de batch de 50. Nous avons alors sélectionné les paramètres donnant le plus grand score pour ce modèle.

Remarquant qu’après les 5 epochs le score du modèle était toujours croissant, nous avons créé un second modèle reprenant en grande partie les bases du premier et l’avons entraîné pour 2 epochs supplémentaires.

Cependant, le score n’ayant pas augmenté avec ces 2 nouvelles epochs, nous avons décidé d’arrêter l’entraînement. Finalement, nous avons donc obtenu respectivement un score de 0.981 et de 0.982 sur notre ensemble de validation pour ces 2 modèles.

Sur le test set, nous avons obtenu les scores suivants :

	Public Leaderboard	Private Leaderboard
BI-LSTM 1 (paramètres = meilleur des 5 epochs)	0.96726	0.96723
BI-LSTM 2 (paramètres = meilleur des 7 epochs)	0.97108	0.96831

Conclusion:

Suite à l’avancé du rendu du projet d’une semaine, nous nous sommes concentrés sur la construction d’un modèle simple et fonctionnel. Au cours de sa réalisation, nous avons pensé à plusieurs idées permettant de l’améliorer : ajout de couches convolutionnelles entre la couche d’embedding et la couche du BI-LSTM, création d’une couche d’attention, modification du preprocessing, etc.
Nous avons finalement obtenu des résultats assez satisfaisants avec ces 2 modèles et espérons vous avoir montré notre bonne compréhension des enjeux du traitement naturel du langage.

Doron Israel, Elliot muller, Nicolas Wagner | Master Mash