Projet : Participez à une compétition Kaggle

Antoine Maby - 01/02/2022

Problématique

Kaggle, une plateforme qui organise des compétitions en data science et qui récompense les meilleurs analystes internationaux

Kaggle est conçue comme une plateforme collaborative. Ce projet est l'occasion de participer à cette communauté

Compétition choisie : Jigsaw Rate Severity of Toxic Comments : Rank relative ratings of toxicity between comments

Déroulement

Déroulement du problème :

- Comprendre la problématique
- Choisir un Dataset
- Trouver une bibliographie
- Trouver une Baseline
- Améliorer les résultats
- Partager les résultats à la communauté

Problématique de la compétition

- Examiner des commentaires, sans contexte, les plus toxiques est difficile

- Chaque individu peut avoir sa propre barre de toxicité

- Une idée pourrait être de faire un vote à la majorité pour décider mais il y a une perte d'information

Problématique de la compétition

- Les auteurs du concours ont choisi de demander à des individus de choisir entre deux commentaires lequel est le plus toxique

- Résout une partie du problème mais pas totalement

- Lorsque les deux commentaires ne le sont pas, le choix semble se porter vers le hasard

- Lorsqu'un commentaire est significativement plus toxique, les résultats entre les individus devraient concorder

Problématique de la compétition

- Réaliser un algorithme de scoring sur la toxicité d'un commentaire

- Tester sur des exemples de paires commentaires

- Scorer des commentaires et les transmettre aux concours

Choix des Datasets

- Il n’y a pas de Datasets fourni par les auteurs pour entraîner nos algorithmes

- 2 Datasets fournis : un de Validation et un de commentaires à scorer

- Mais des compétitions précédentes sur des sujets proches

Choix des Datasets

- Le choix s'est porté sur Toxic Comment Classification Challenge pour le train

- 150 000 commentaires avec 6 classes différentes

- Les classes sont : Toxic, Severe Toxic, Obscene, Insulte, Identity Hate et Menace

Choix des Datasets

- Un deuxième choix s'est porté sur Jigsaw Unintended Bias in Toxicity Classification

- 2 000 000 commentaires mais une description différente de la toxicité

- La toxicité pour les commentaires est une note entre 0 et 1. On le modifie en le transformant en classe pour une toxicité non nulle.

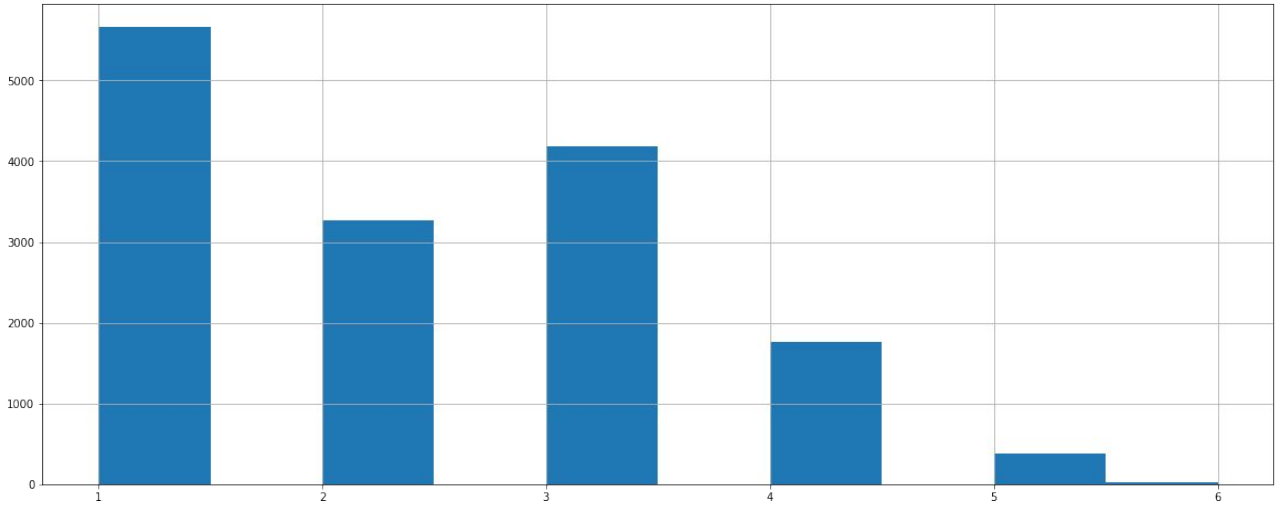
Choix des Datasets

- Finalement, 4 Datasets différents composent notre projet

	Validation(paire de commentaires)	Comments	Train	Augmentation
Nombre de commentaires	30108	7537	159571	1999516

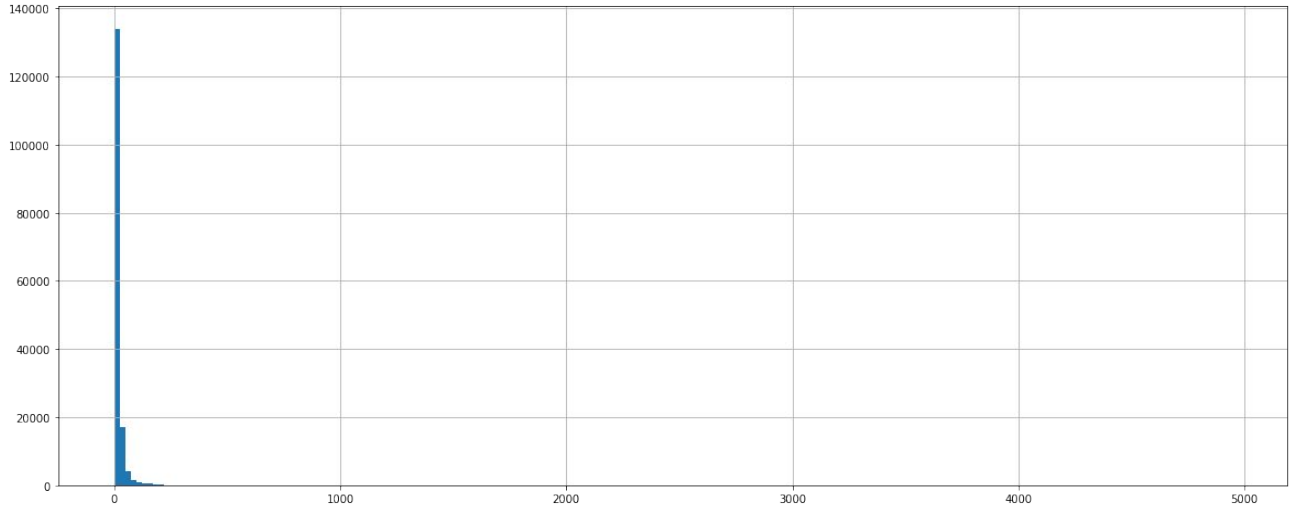
Analyse Exploratoire

Distribution du nombres de classes pour les commentaires



Analyse Exploratoire

Distribution de la ponctuation dans les commentaires



Nettoyage du corpus

Pour nettoyer le corpus, nous allons supprimer

- Les bannières HTML
- Mettre tout le texte en minuscules
- Supprimer les caractères Unicode
- Suppression des espaces supplémentaires
- Suppression de la ponctuation
- Suppression des liens
- Supprimer les nombres
- Changement des contractions connues d’Internet

Nettoyage du corpus

Deux processus différents en fonction de l'encoder que nous utiliserons.

- Si Tf-idf :
  - Supprimer les Stopwords
  - Tockeniser
  - Lemmatiser
  - Tf-idf

- Si Sentence Encoder
  - Sentence Encoder

Choix de la Baseline

- Nous devons maintenant choisir une méthode pour mettre en place notre projet

- Nous les testerons toutes avec la méthode de Tf-idf

- Random Forest Classifier, Régression Lightgbm et Classification binaire Lightgbm

Random Forest Classifier

- Random Forest CLassifier multi task sur les 6 classes du dataset de Train

	Random Forrest	Lightgbm Regression	Lightgbm Classification
Score	0.46	0	0

Lightgbm Regression

- Regression sur la somme des 6 classes du dataset de Train

	Random Forrest	Lightgbm Regression	Lightgbm Classification
Score	0.46	0.65	0

Lightgbm Classification

- Classification binaire Toxic/Non Toxic sur les Datasets Train et Augmentation

	Random Forrest	Lightgbm Regression	Lightgbm Classification
Score	0.46	0.65	0.57

Choix de la Baseline

- Le choix se porte sur la régression Lightgbm sur la somme des 6 classes

- Améliorons les performances tout d'abord en testant de nouveaux Sentence Encoder

- Puis nous testerons des améliorations de l'algorithme

Choix du Sentence Encoder

- Après avoir tester toutes nos Baseline avec Tf-idf, testons d'autres Sentences Encoder

- Nous testerons Universal Sentence Encoder et Sentence Transformer

- Nous testerons la concaténation de deux encoder

Universal Sentence Encoder

- Régression Lightgbm sur le somme des 6 classes

	USE	All Mini L12	All Mini L12 pré-train	Tfidf et All Mini L12	USE et All Mini L12
Score	0.6858	0	0	0	0

Sentence Transformer

- Régression Lightgbm sur le somme des 6 classes
- All Mini L12 v2 est le Sentence Transformer
- Compromis entre Rapidité et Performance

	USE	All Mini L12	All Mini L12 pré-train	Tfidf et All Mini L12	USE et All Mini L12
Score	0.6858	0.6857	0	0	0

Sentence Transformer pré-train

- Sentence Transformer permet un pré-entraînement sur des paires de commentaires
- Paires Nulle (Toxic/Non Toxic) et Paire Positive (Toxic/Toxic)
- Utilisation du Dataset Augmentation : 240 000 paires de commentaires

Sentence Transformer pré-train

- Régression Lightgbm sur le somme des 6 classes

	USE	All Mini L12	All Mini L12 pré-train	Tfidf et All Mini L12	USE et All Mini L12
Score	0.6858	0.6857	0.6824	0	0

Tf-idf et Sentence Transformer

- Concaténation des deux méthodes puis Régression Lightgbm sur la somme des classes

	USE	All Mini L12	All Mini L12 pré-train	Tfidf et All Mini L12	USE et All Mini L12
Score	0.6858	0.6857	0.6824	0.6815	0

Universal Sentence Encoder et Sentence Transformer

- Concaténation des deux méthodes puis Régression Lightgbm sur la somme des classes

	USE	All Mini L12	All Mini L12 pré-train	Tfidf et All Mini L12	USE et All Mini L12
Score	0.6858	0.6857	0.6824	0.6815	0.6868

Choix du Sentence Encoder

- Le choix se porte sur Universal Sentence Encoder et Sentence Transformer

- Passons aux tests de possible amélioration

- Les améliorations seront : Fine Tuning, Ajout de Features et Poids sur les classes

Fine Tuning

Fine Tuning principal sur num leaves, max depth et min data in leaf

	Base	Finetune	Feature	Poids 1	Poids 2
Score	0.6858	0.6904	0	0	0

Features

Création de 6 features et intégration dans le modèle

- le nombre de mots
- le nombre de mots uniques
- le nombre de stopword
- le nombre de ponctuations
- la longueur moyenne des mots
- le nombre de caractères

Features

Création de 6 features et intégration dans le modèle

	Base	Finetune	Feature	Poids 1	Poids 2
Score	0.6858	0.6904	0.6872	0	0

Poids sur les classes

Nous testons différents poids sur les classes

	Toxic	Severe Toxic	Obscene	Threat	Insult	Identity Hate
Poids	0.32	1.5	0.16	1.5	0.64	1.5

	Base	Finetune	Feature	Poids 1	Poids 2
Score	0.6858	0.6904	0.6872	0.6821	0

Poids sur les classes

Nous testons différents poids sur les classes

	Toxic	Severe Toxic	Obscene	Threat	Insult	Identity Hate
Poids	0.5	2	0.5	2	0.5	2

	Base	Finetune	Feature	Poids 1	Poids 2
Score	0.6858	0.6904	0.6872	0.6821	0.6833

Conclusion

- Test de plusieurs méthodes, choix de Régression Lightgbm

- Test de plusieurs Sentence Encoder, choix de Universal Sentence Encoder et Sentence Transformer

- Amélioration des algorithmes : pas d'amélioration sur la validation mais amélioration sur les performances finales

Merci pour votre attention