Projet ATAL 2022

Objectif

Mettre en œuvre différentes approches et outils vus en cours au travers de la tâche de détection de tweets offensif en utilisant les données de la compétition OffensEval qui a eu lieu lors de l'atelier SemEval2019.

Pitch

Vous participez à la compétition OffensEval2019, pour laquelle vous proposerez une approche à base de traits et une approche à base de Transformer. A chaque fois, vous utiliserez une approche état de l'art puis essaierez de l'améliorer. Vous rédigerez un rapport sous la forme d'un article scientifique qui contextualise la tâche, explique les choix que vous avez fait pour améliorer les approches ainsi que vos résultats et une analyse d'erreur de vos différents systèmes.

Planning

Semaine 39/40

1. Implémentation d'un classifieur à base de traits
2. Amélioration du classifieur
3. Choix et lecture d'un des articles des équipes participantes à OffensEval2019 ou 2020

Semaine 43/44

1. Présentation de l'article choisi lors d'un groupe de lecture informel
2. Utilisation de la bibliothèque transformers (hugging face)
3. Comparaison de différents modèles de transformer
4. Analyse d'erreur

Semaine 49/50

1. Amélioration de la méthode transformer
    a. Augmentation de données
    b. Post-traitements
    c. Apprentissage d'ensembles (Ensemble Learning)
2. Rédaction (+ Vérification des scores et expériences)

Consignes

Tout au long de ce projet vous utiliserez le corpus OLID (a télécharger tout en bas a droite de la page). Pensez à créer un **ensemble de validation** (10% de l'ensemble d'entraînement) que vous utiliserez au long de vos expériences.

Classifieur à base de traits

Lisez l'article "Classiffier Automated Hate Speech Detection and the Problem of Offensive Language" puis écrivez le code qui reproduit la méthode décrite à l'aide de SKLearn (vous utiliserez seulement la régression logistique)

a. Charger les données dans un dataframe pandas
b. Identifier les différents traits (features) utilisés dans l'article
c. Écrire les fonctions permettant d'extraire ces différents traits. Utilisez les fonctions de sklearn.pipeline et de sklearn.preprocessing (FunctionTransformer).
d. Entraîner le modèle à l'aide de l'ensemble d'entraînement
e. Évaluer le modèle à l'aide de la précision, du rappel et de la f-mesure sur l'ensemble de validation

Amélioration du classifieur

A l'aide de vos connaissances de cours testez une autre approche pour résoudre la tâche. (ex: ajouter des traits, utiliser des plongements de mots, modification des pré-traitements, autre modèle). Voir https://aclanthology.org/W17-1101.pdf pour une liste de traits communément utilisés dans la tâche de classification de discours de haine.
Décrivez et justifiez vos choix dans le rapport. Rapportez aussi les scores obtenus par vos deux classifieur dans le rapport.

Lecture d'article

Choisir un des article décrivant la participation d'une équipe à OffensEval2019 ou 2020. Vous le présenterez aux autres groupes lors d'un groupe de lecture **semaine 43 (lundi 24/10 à 11h dans la salle 113 du bâtiment 11)**.

Pour trouver ces articles identifiez les noms des équipes dans les Tables 5 et 5,6,7. Puis cherchez sur google scholar "NOM EQUIPE semeval ANNEE task NUMTACHE".

Le but du groupe de lecture est de partager votre lecture aux autres groupes, il faut donc que votre présentation soit accessible et compréhensible. Pour cela choisissez les informations les plus pertinentes à partager. Votre présentation doit comporter 7 slides maximum (il peut y en avoir moins !) et doit décrire **pourquoi vous l'avez choisi** et **ce que vous avez compris de l'article** : quelle méthode, pourquoi, quels résultats ainsi que votre avis sur l'article, l'évaluation, la méthode utilisée, etc...

Classifieur basé sur les transformers

Écrivez le code qui reproduit la méthode utilisant BERT dans l'article « NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection using Bidirectional Transformers » à l'aide de la bibliothèque transformers et des tutoriels fournis dans la documentation.

Comparaison modèles transformer

Choisissez et comparez les performances de 3 modèles transformers sur le corpus OLID avant (si pertinent) et après affinage. Faites des hypothèses à priori sur les scores qui vont obtenir ces modèles. Vous rapporterez ensuite les scores dans le rapport puis discuterez des

(éventuelles) différences de scores (en fonction du domaine des données d'entraînement, du nombre de paramètres, de la langue du modèle, ...).

Choisir des un panel de modèles variés pour que la comparaison soit intéressante.

Analyse d'erreur

Étudiez quelques exemples mal classifiés par le meilleur modèle transformer et votre classifieur amélioré et essayez de les regrouper en catégories. Décrivez ensuite votre analyse dans le rapport (**avec quelques exemples**).

Selon le modèle utilisé vous pouvez tirer parti des coefficients appris pour tenter d'expliquer certaines erreurs de classification.

Amélioration de la méthode transformer (au choix)

Utilisation de données externes (data augmentation)

1. Récolter des tweets non annotés (cf. Kaggle ou Fortuna et al. 2020)
2. Appliquer la prédiction
3. Choisir les prédictions les plus sûres du modèle
4. Choisir les documents à ajouter au corpus d'entraînement
5. Affiner le modèle

Ensemble de classifieur (ensemble learning)

L'idée est que chaque classifieur va faire des erreurs différentes. Il « suffit » ensuite de choisir le bon classifieur.

1. Utiliser le vote majoritaire
2. Utiliser le stacking : un meta classifieur utilise les prédiction des autres classifieurs pour faire sa prédiction
3. Utiliser le bagging (bootstrap aggregating) : chaque classifieur est entraîner avec un sou-ensemble d'entraînement différent
4. Utiliser le boosting : les classifieurs sont entraînés les uns après les autres en mettant l'accent sur les exemples mal classés

Post-traitements

1. Choisir un seuil de prédiction (courbe ROC)
2. Utiliser un autre classifieur lorsque le classifieur est peu confiant
3. Créer des règles à partir de l'analyse d'erreur

Rapport

Vous écrirez votre rapport au format LaTeX en utilisant le style des articles ACL (ici).

Il devra contenir a minima :

- un résumé,
- une introduction qui décrit succinctement votre travail et la tâche que vous tentez de résoudre avec des références bibliographiques aux travaux précédents,

- une description des données que vous utilisez,
- le cadre expérimental de vos expériences,
- leurs résultats et les analyses d'erreur,
- une conclusion.

Inspirez vous de la forme et de la structure des articles que vous aurez lu jusque là.

Modalités d'évaluation

Rapport : clarté du rapport ; justification des choix effectués

Code : organisation du code ; lisibilité du code

Soutenance : clarté de l'exposé et des transparents ; qualité de la réponse aux questions

Rendus (chaque vendredi à 17h)

A envoyer à ygor.gallina@univ-nantes.fr

Semaine 39 : Envoyez moi votre code tel qu'il est
Semaine 40 : Envoyez moi votre code mis au propre (un notebook lisible et déjà exécuté)

Semaine 43 : Envoyez moi votre code tel qu'il est + un pdf de votre rapport en l'état
Semaine 44 : Envoyez moi votre code mis au propre (un notebook lisible et déjà exécuté)

Semaine 49 : Envoyez moi votre code tel qu'il est + un pdf de votre rapport en l'état
Semaine 50 : Envoyez moi votre code mis au propre (un notebook lisible et déjà exécuté) + un pdf de votre rapport en l'état

Date soutenance

19/01/2023 Après-midi

Liste d'outils utiles

spacy, nltk, textblob, Lexique hatebase