Projet ATAL 2023

Retour des étudiants

- Manque de connaissances sur BERT
- Plus de guidage sur comment présenter les résultats
- Pas assez de créneaux dédiés (à cause des contraintes de planning)
- Les points hebdomadaire en groupes étaient utile

Pour le prochain sujet

- Points de cours dans le rdv hebdomadaire (transformers, evaluation)
- Plus de retour sur les informations importantes à écrire dans le rapport (reproductibilité, savoir ce qui es intéressant ou non)
- Contraindre les étudiants à utiliser le codalab (utiliser kaggle ?)

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 LegalEval qui a eu lieu lors de l'atelier SemEval2023.

Pitch

Vous participez à la compétition LegalEval, pour laquelle vous proposerez une approche à base de traits et une approche à base de Transformer. A chaque fois, vous utiliserez une approche de base (baseline) 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.

Vous pourrez comparer vos résultats avec les autres groupes grâce au Codalab créé pour l'occasion (plateforme de gestion de compétitions).

Article de présentation de la tâche : SemEval-2023 Task 6: LegalEval - Understanding Legal Texts, (Modi et al., SemEval 2023)

Télécharger les données : Corpus BUILD Corpus for Automatic Structuring of Legal Documents, (Kalamkar et al., LREC 2022)

Planning

Semaine 39

1. Implémentation d'un classifieur à base de traits
2. Amélioration du classifieur
3. Choix et lecture d'un articles connexe à la tâche

Semaine 40

1. Présentation de l'article choisi lors d'un groupe de lecture informel
2. Prise en main de la bibliothèque transformers (hugging face)
3. Comparaison de différents modèles de transformer

Semaine 4

1. Analyse d'erreur (en priorité)
2. Amélioration de la méthode transformer (si possible en fonction du temps)
   a) Augmentation de données
   b) Post-traitements
   c) Ensemble learning
3. Rédaction et préparation de la présentation finale

Consignes

Tout au long de ce projet vous utiliserez le corpus BUILD (à télécharger ici). Il est découpé en 2 ensembles: `train` pour l'entraînement de vos modèles, `dev` que vous utiliserez pour évaluer vos modèles. Un ensemble de `test` servira a évaluer vos systèmes (il est gardé secret jusque là).

Classifieur à base de traits

Appropriez vous le code de la méthode de base fourni. Puis, pour améliorer cette méthode, choisissez au moins deux descripteurs (features) en plus des tokens, bi-grammes et tri-grammes dans l'article de Teufel et Kan "Robust Argumentative Zoning for Sensemaking in Scholarly Documents". Implémentez ces descripteurs dans le code et évaluez leur impact sur les performances.

a. Faire tourner le code fourni
b. Identifier les différents descripteurs (features) mentionnés dans l'article et en choisir au moins 2
c. Écrire les fonctions permettant d'extraire ces différents descripteurs. 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 les modèles à l'aide de la précision, du rappel et de la f-mesure sur l'ensemble de validation
f. Essayez d'ajouter des descripteurs (entités nommés, word embeddings, …). La bibliothèque spacy offre de nombreuses annotations.

Décrivez l'implémentation de vos descripteurs et justifiez vos choix dans le rapport. Rapportez aussi les scores obtenus par vos classifieur pour les comparer.

Lecture d'article

Chaque groupe choisis un des article de la liste suivante, pour le **mercredi 27/09**. Vous le présenterez à vos camarades lors d'un groupe de lecture **semaine 40 (mercredi 04/10 à 14h dans la salle 210 du bâtiment 34)**.

- AntContentTech at SemEval-2023 Task 6: Domain-adaptive Pretraining and Auxiliary-task Learning for Understanding Indian Legal Texts : Meilleur système de la compétition
- LEGAL-BERT: The Muppets straight out of Law School : Modèle BERT pour le domaine légal
- Cross-Domain Multi-Task Learning for Sequential Sentence Classification in Research Papers : Méthode état de l'art pour la prédiction de rôles rhétoriques

- Combining WordNet and Word Embeddings in Data Augmentation for Legal Texts : Augmentation de données dans les textes légaux
- Automated Extraction of Sentencing Decisions from Court Cases in the Hebrew Language : Extraction d'information dans les textes légaux
- Detecting Relevant Differences Between Similar Legal Texts

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 transparents[1] (il peut y en avoir moins !) et doit décrire ce que vous en avez compris (quelle tâche on essaye de résoudre, quelles contributions, pourquoi, comment, quelle évaluation, quels résultats) et votre avis sur l'article.

Comparaison modèles transformer

Choisissez et comparez les performances de 3 modèles transformers sur le corpus BUILD avant (si pertinent) et après affinage. Faites des hypothèses à priori sur les scores que 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 un panel de modèles variés pour que la comparaison soit intéressante, en variant par exemple le domaine des données d'entraînement utilisées pour l'entraînement de ce modèle.

Analyse d'erreur

L'objectif ici, est d'étudier les sorties d'un système pour comprendre les erreurs qui sont faites, formuler des hypothèses quand à la mauvaise classification et ainsi proposer une expérience qui permettra d'améliorer le système. Vous trouverez dans le dossier du corpus les prédictions de plusieurs modèles transformers affinés sur le corpus BUILD. Pour votre analyse choisissez un sous ensemble **cohérent** et justifié de modèles à étudier (différentes architectures, différents nombres de paramètres, différents domaine de pré-entrainement, …). Vous n'êtes pas obligé de comparer tout les modèles !

Exemple d'analyse à effectuer : (1 ou 2)

- Tous les systèmes font-il les mêmes erreurs ?
- Y a t-il des phrases plus dures que d'autres ?
- Quelles étiquettes sont les mieux/moins bien classifiés ?
- Pour quels documents/étiquettes le modèle est-il le plus confiant (probabilité de l'étiquette) ?

Pour effectuer ces analyses vous pourrez, selon le modèle utilisé , tirer parti des poids du modèle (bertviz ou autre). Vous pourrez aussi par exemple essayer de trouver des catégories d'erreur ou encore des similarités dans les exemples mal classifiés.

Votre analyse doit résulter en un graphique, un tableau de chiffre ou quelques exemples bien choisis. Cela permettra de justifier votre réponse.

---

[1] Par exemple: titre, tâche, contributions, méthode utilisée, tableau de résultat commenté, avis, conclusion.

Dans le rapport : décrivez la question a laquelle vous essayez de répondre, décrivez votre démarche pour y répondre, puis présentez votre analyse.

Amélioration de la méthode transformer (au choix)

Utilisation de données externes (self-learning)

1. Récolter des documents similaires non annotés
2. Appliquer la prédiction d'un modèle
3. Choisir les prédictions les plus sûres
4. Ajouter les documents et leur annotation au corpus d'entraînement
5. Affiner le modèle

Ensemble de classifieurs (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 entraîné avec un sous-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

Prompt engineering avec ChatGPT (ou autre)

Essayer plusieurs techniques de prompting avec ChatGPT (ou autre modèle génératif) pour prédire l'ensemble des documents ou bien refaire une prédiction lorsqu'un autre classifieur est peu confiant.
Essayez différentes manières de préparer le modèle à répondre aux questions. Donnez lui quelques exemples ou non. Essayez de le corriger ou non.
En testant plusieurs stratégies avec le même ensemble de document, vous devriez pouvoir évaluer ces différentes stratégies et les comparer.

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.

Présentation finale

Vous aurez 15 minutes de présentation et 5 minutes de questions/discussion (prévoyez entre 7 et 9 transparents maximum).

Ce que j'attends de votre présentation:

- contextualiser la tâche (avec un exemple d'input et d'output) (1 transparent)
- présenter succinctement votre classifieur (~1-2 transparent)
  - quelques features et le modèle utilisé
  - quelles améliorations vous avez apportées (ou pensés) et pourquoi
  - quels résultats ont été obtenus
- quels modèles vous avez choisis de comparer (~1-2 transparent)
  - quelles sont leur particularités
  - pourquoi ont-ils été choisis et quels sont leur résultats
- vos analyse / tentatives d'améliorations (~1-2 transparent))
  - pourquoi cette amélioration et les résultats obtenus (même négatifs)
  - cadre de l'analyse ; résultats
- conclusion

Pensez bien à contextualiser votre présentation : quelle est la tâche que l'on essai de résoudre? Montrer quelques exemples de messages et de prédiction pour avoir une idée des données manipulées.

Lorsque vous affichez des chiffres préférez afficher 56,5 au lieu de 0,5647725 (vous pouvez utiliser plus de chiffres significatif **si pertinent**).

Le but de la présentation est de montrer le travail que vous avez fournis, il faut qu'elle soit accessible et ne pas perdre l'auditoire dès le deuxième transparent.

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 avant 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)

Dates importantes

Présentations finales : Vendredi 26 janvier à 14h au LS2N
Rendu rapport : Mercredi 31 janvier à 18h par mail à l'adresse ygor.gallina@univ-nantes.fr avec l'objet : « [M2ATAL][UE Projet] Rapport ».

Liste d'outils utiles

spacy, nltk, textblob