

Louis DESEURE | Paul WAULTRE

Junia isen | pROMO 64

Rapport de NLP

Table des matières

[Présentation du projet 2](#_Toc98257555)

[Notre approche 3](#_Toc98257556)

[Notre solution 4](#_Toc98257557)

[Axes d’amélioration 5](#_Toc98257558)

## Présentation du projet

Le projet de NLP de cette année consistait à déterminer la catégorie de toxicité d’un commentaire laissé par un utilisateur. Les applications d’un tel projet sont nombreuses mais on peut penser que les usagers pourraient s’en servir pour filtrer les commentaires à caractère injurieux d’un vidéo, d’un site internet ou de tout autre support pouvant accueillir un espace dédié aux commentaires. Ainsi, nous avions à notre disposition un dataset de 159571 commentaires pouvant rentrer dans une ou plusieurs des six catégories suivantes :

* Toxicité
* Toxicité sévère
* Obscène
* Menace
* Insulte
* Haine d’identité

Il existe dans ce dataset des commentaires ne faisant partie d’aucune catégorie comme d’autres qui peuvent en faire partie de plusieurs à la fois.

Notre objectif était d’entrainer une intelligence artificielle afin de prédire, par la suite une classification pour de futurs commentaires.

## Notre approche

Après avoir visualisation rapidement le dataset, nous avons remarqué plusieurs choses. Premièrement malgré la quantité nous n’avons que très peu de commentaires classés comme menaçant (0,3%). Par conséquent, il y a peu de chance que noter solution soit bien entrainée pour ce cas de figure.

Deuxièmement, nous avons remarqué que près de 89,5% des commentaires du dataset se sont dans aucune des six catégories citées en amont. Il va falloir que nous fassions preuve de vigilance pour que notre intelligence artificielle ne se contente pas de mettre tous les commentaires comme neutre car cela lui semblerait être la réponse la plus probable.

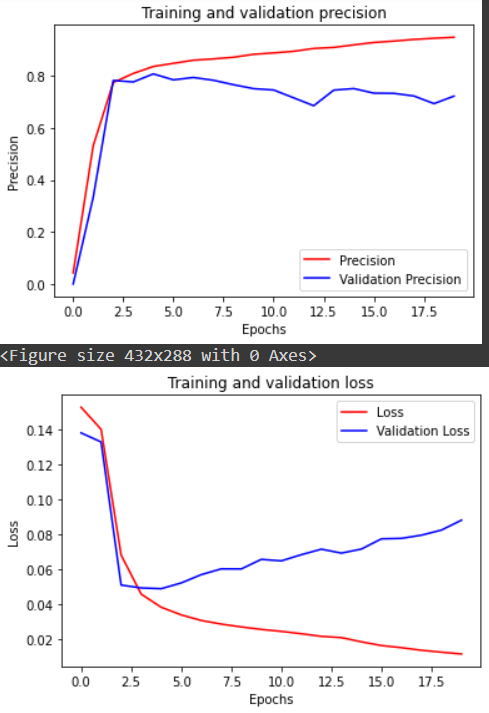
Troisièmement, il s’agit de commentaire donc nous ils contiennent très probablement des abréviations et des concaténations. Pour pallier ce problème nous avons trouvé judicieux de reprendre l’approche du TD sur l’analyse de sentiment et nous avons décider de nettoyer les commentaires grâce aux fonctions pour nettoyer les tweets.

Ainsi, les contractions sont séparées, les caractères spéciaux sont supprimés, ainsi que les URL, les mentions, les mots trop courts, les mots vides, les abréviations seront remplacées par leur équivalent complet grâce à ce dictionnaire d'abréviations et à la fonction « convert\_abbrev\_in\_text ».  On évite ainsi les différences de poids entre un mot ou un groupe de mots et son abréviation.

## Notre solution

Nous commençons déjà par prévisualiser grâce à un RandomForest pour connaître la difficulté de la réalisation du projet. On peut considérer cela comme une ébauche. Nous commençons par tokenizer nos commentaires en supprimant les abréviations, les stopwords (les petits mots de liaisons) et en supprimant aussi la ponctuation. En appliquant la RandomForest, on remarque que 89% des réponses proposées par le modèle sont justes. Cependant, cela correspond fort au nombre de commentaires non toxiques. Nous pouvons donc déduire que ce modèle ne suffit pas à résoudre ce problème.

Nous allons tenter alors avec un RNN où nous avons une couche d’Embedding, une couche de LSTM et une Dense. Dans ce modèle, nous structurons les données sous forme d’indexation en effet nous créeons un bag of word en ayant tokenisé, chaque mot est caractéisé par un label, ainsi chaque commentaire est codé en nombres. Nous utilisons en metrics de l’Embedding que nous prenons Précision car en effet après des tests avec les valeurs de Accuracy, ACU et recall donnaient de moins résultats. Nous en sommes arrivés à la suite de 20 époques du modèle à une précision de 94% de réponses juste du modèle.

Voici un exemple de l’amélioration du taux de précision et du training loss. 

Nous avons créé ensuite une fonction de prédilection manuelle où l’utilisateur rentre la phrase de son choix afin de savoir si cette phrase est toxique ou non. Afin d’interpréter les données, nous supposons que la donnée est significative lorsqu’elle dépasse le seuil de 30 % du label.

Nous nous sommes inspirés des TP vus en cours à propos tel que celui sur le Sentiment Analysis avec TensorFlow.

## Axes d’amélioration

Certains points peuvent apporter une amélioration notable pour notre réseau de neurones mais que nous n’avons pas mis en place pour respecter les délais.

Par exemple, nous aurions pu augmenter la variable max\_length dans notre code qui est fixée à 150. Cela aurait eu pour conséquence de prendre davantage de caractères en considération mais aurait grandement allonger le temps de calcul.

Nous aurions pu utiliser les notions de context\_word et de stemming. Les context\_word nous auraient servi à contextualiser le commentaire et ainsi avoir un début de réponse quant à sa classification. Le stemming de son côté aurait d’une part éviter les différences de poids entre deux mots similaires tels que « has » et « had ».

Puis, nous aurions pu différencier les mots en majuscule et les mots en minuscule. Par exemple, le mot « shut » est assez ambigu car il est utilisé comme insulte comme dans « SHUT UP » ou comme simple verbe comme dans « shut the door ». Une différenciation entre ces deux écritures aurait permis de ne pas se tromper sur la prédiction de classification des phrases le contenant.

Enfin, nous n’avons pas beaucoup jouer avec les variations de notre réseau de neurones et de couches ses couches. Nous pensons qu’il existe une configuration plus efficace mais nous avons estimé que la nôtre était relativement suffisante.