Détection d'auteurs - NLP

Dataset

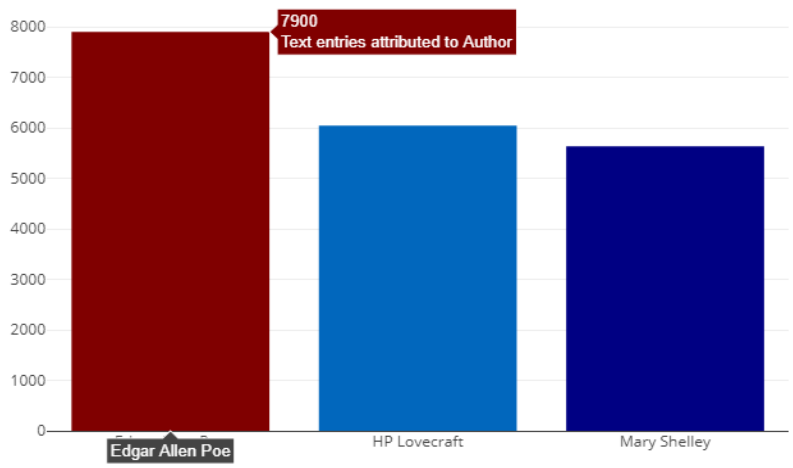
C'est une base de données composé de différents corpus écrit par 4 différents auteurs Edgar Allan Poe, Mary Shelley, et HP Lovecraft.
Tous ces corpus sont des histoires d'horreurs. Notre dataset est de la forme suivante :

	id	text	author
0	id26305	This process, however, afforded me no means of...	EAP
1	id17569	It never once occurred to me that the fumbling...	HPL
2	id11008	In his left hand was a gold snuff box, from wh...	EAP
3	id27763	How lovely is spring As we looked from Windsor...	MWS
4	id12958	Finding nothing else, not even gold, the Super...	HPL

https://www.kaggle.com/c/spooky-author-identification/data

Pre-processing with NLTK

Distribution des données :

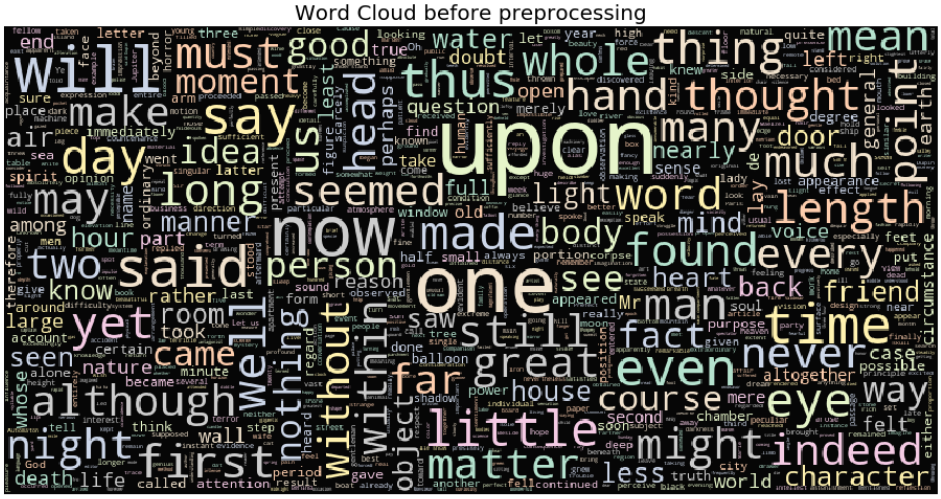


Warichet Sébastien
Hutin Gérémy
Deschamps-Berger Théo

Nous avons mélangé et splitté notre dataset en deux parties :
⅔ en *train_set* (notre training set) et ⅓ en *test_set (*notre validation set).

Nous avons détecté dans notre *train_set* un biais lié aux noms propres, par exemple le nom “Windsor” apparaît 72 fois tous pour l’auteur Mary W Shelley, ou le nom West qui apparaît dans 119 fois pour l’auteur HP Lovecraft.

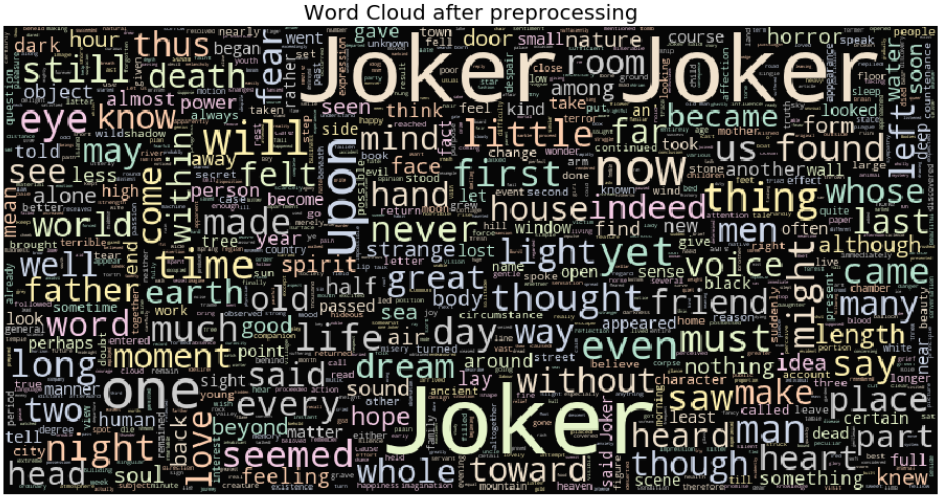
Word Cloud before preprocessing



On a donc décidé de remplacer les noms propres par le mot ‘Joker’. Nous avons décidé de ne pas les supprimer pour ne pas modifier les structures de phrases ce qui aurait induit un biais.

Notre nouveau ‘word cloud’ (mots les plus fréquent) ressemble au suivant :

Word Cloud after preprocessing



Sans surprise nous obtenons le mot ‘Joker’ dans le premier tier.

Warichet Sébastien
Hutin Gérémy
Deschamps-Berger Théo

Encodage des mots

Les mots sont encodés sous forme d' ”embedding”, à noter que cette transformation fait partie de notre modèle et est donc entraîné avec notre training set.
Pour entraîner un embedding on utilise le contexte des mots, les mots qui ont un contexte similaire auront des vecteurs proches. Ceci permet de créer une compression.
On a testé avec l'embedding pré-entraîné GloVe (“Global Vectors for Word Representation”), calculé à partir d'une base de 400 000 mots extraits de Wikipedia en 2014 et produisant une représentation des mots sur 100 dimensions.

Le fait d'encoder les mots sous cette forme a une influence sur la définition de notre problématique, en effet cet encodage est généralement utilisé pour récupérer de l'information dans un texte. Ceci implique que notre modèle est sensible aux sujets abordés dans les différents livres. Comme tous les livres sont sur le même sujet (histoire d'horreur) on peut supposer que les sujets et le vocabulaire est proche entre les auteurs.

Si on avait voulu se limiter au style des auteurs, il aurait fallu choisir un type d'encodage plus basique type “one hot vector”.

Global Average pooling

On prend la moyenne de l'ensemble des vecteurs produit par l'embedding pour chaque phrase.
Le but de cette étape est de renvoyer un vecteur de taille fixe pour chaque phrase qui satisfera l'entrée de notre réseau dense.
On en déduit que l'importance d'un mot sur ce vecteur de moyenne dépend de la longueur de la phrase dans lequel il est.

Réseau Dense

Notre réseau dense comprend trois couches, il prend en entrée notre vecteur de moyenne obtenu avec le Global Average pooling.
Les fonctions d'activation de nos différentes couches sont des RELU. Et la fonction d'activation de notre dernière couche est un softmax.

Premiers résultats

Les premiers résultats étaient bons, nous avons rapidement obtenus avec nos données embedded et un average pooling, 80% d'accuracy après une cinquantaine d'époques.

Warichet Sébastien
Hutin Gérémy
Deschamps-Berger Théo

Amélioration de notre modèle ajout d'un RNN

Le modèle de type RNN fait du lien entre les différents mots d'une phrase. On va donc différencier les auteurs sur la structure de leurs phrases.
Les RNN semble naturel pour traiter du NLP, car ils sont à l'image que l'on se fait d'un langage par exemple les RNN ont un sens comme le langage (de gauche à droite pour les langage européen).
Notre problème étant relativement simple RNN et "Global average pooling" donne des résultats comparables.
Pour tirer avantage des RNN ils faudrait complexifier le problème par exemple ne s'intéresse qu'au style de l'auteur (encodé les mots sans embedding) ou prendre un meilleur "dataset" avec plus d'auteurs et plus de livres.

Résultats finaux

L'ajout d'un RNN n'améliore pas les résultats et est finalement beaucoup plus long à entraîner.

Résultats avec 5 époques :

Epoch 1/5
loss: 1.0699 - accuracy: 0.4241 - validation_loss: 0.9931 - validation_acc: 0.4780
...
Epoch 5/5
loss: 0.3810 - accuracy: 0.8526 - validation_loss: 0.5129 - validation_acc: 0.7870

Nous obtenons des scores de 79 % accuracy.

Conclusion

En conclusion nous pouvons dire que quel que soit le modèle que nous avons utilisé, les résultats sont bons et comparable, ce qui semble indiquer que la problématique est simple malgré le fait que le remplacement des noms propres ait complexifié notre problème..
Les différents auteurs semblent avoir un vocabulaire qui leur est propre (par exemple le mot "snuff" ou "carcass" pour EAP, "nurse" pour MWS ...), ceci explique sans doute la facilité de résolution de ce problème.

Critique : Il semblerait que notre "dataset" ne comprend pas assez de livre différent.

Warichet Sébastien
Hutin Gérémy
Deschamps-Berger Théo

Possible améliorations futures

CNN - Keras
(Non implémenté)

Le modèle CNN 1D va rechercher des patterns courants, ces patterns seront ensuite utilisés pour différencier les auteurs.
Les modèle CNN sont généralement utilisé pour la classification d'image, ils considèrent donc uniquement les mots contenu dans la fenêtre pour détecter les patterns (ne pas oublier d'introduire de l'overlap). Ce qui n'est pas forcément en lien avec le traitement du langage.
Malgré tous les NLP sont utilisé et donne de bon résultat.
L'avantage des CNN est qu'ils sont plus rapides que les RNN.

Warichet Sébastien
Hutin Gérémy
Deschamps-Berger Théo