Natural Language Processing

Fake News Challenge : Stance Detection

Victor Rambaud, Thomas Fournier, Louis Leprince

Abstract

Le Fake News Challenge (FNC) a pour objectif l'utilisation de l'intelligence artificielle, l'apprentissage automatique et le traitement automatique du language pour répondre au problème de détection des fake news. La détection des fake news est problème complexe difficile à résoudre même par des experts. Pour simplifier ce problème, une décomposition du problème en tâche est proposée et la première étape du FNC est la Stance Detection. Cela consiste en l'étude de la position des journaux sur des nouvelles. A partir d'un titre et du corps d'un article l'objectif est de classifier la position.

1 Introduction

Dans ce projet nous proposons une solution pour le problème de la Stance Detection. A partir des données du FNC, nous avons développé un modèle pour prédire la position d'un journal sur une nouvelle à partir du corps d'un article et du titre de la nouvelle.

Pour traiter le texte des articles et des titres nous avons employé des méthodes de traitement automatique du language. On a pour cela utilisé deux Embedder, celui de Google et ELMo. L'Embedder permet la transformation des mots en vecteur à valeurs numériques les représentants. Celui de Google a été conçu pour que des mots similaires aient des vecteurs similaires. ELMo est plus complexe, il produit pour des vecteurs représentant les mots dans leurs contextes.

Pour le modèle, nous nous sommes inspiré d'une partie de la solution proposé par l'équipe Talos Intelligence au cours du FNC. Il s'agit d'un réseau de neuronnes convolutif qui traite le titre et l'article indépendemment, pour ensuite les regrouper et les passer dans un réseau de neuronnes dense pour prédire la position. Nous avons également développé un autre modèle qui s'appuie sur une architecture BiLSTM mais les résultats se sont avérés moins bons.

Pour l'évaluation de la performance des modèles nous avons utilisé deux métriques, la précision de classification et le score du FNC. Pendant l'entrainement on a effectué des sauvegardes séparées pour les deux métriques (meilleur modèle sur l'ensemble de validation).

On propose ainsi un pipeline qui à partir d'un fichier de configuration permet l'entrainement d'un modèle avec l'Embedder souhaité. A partir d'une sauvegarde d'un modèle, il est possible de reprendre l'entrainement du modèle et d'évaluer le modèle sur l'ensemble de test.

2 Data Processing

2.1 Fake News Challenge Dataset

Le dataset FNC se décompose en 2 ensembles, celui d'entrainement et celui de test. On dispose pour chaque ensemble d'un fichier avec les corps des articles (bodies.csv) auxquels sont associé un identifiant et un fichier de titres (stances.csv) auxquels sont associés l'identifiant d'un corps d'article et la position à prédire. Pour générer l'ensemble de validation, on a extrait aléatoirement 20% des titres d'articles du fichier d'entrainement. Les corps d'article pour la validation sont les mêmes que ceux pour l'entrainement mais les couples titre et corps d'article sont différents. Dans le Tableau ci-dessous on propose le détail des données. Une epoch corresponds au parcours de l'ensemble des couples titres et corps présent dans le fichier stances.csv.

Set	**Headlines**	**Bodies**
Train	40123	1683
Validation	9849	1683
Test	25413	904

Concernant les positions à prédire il y en a 4 possibles, unrelated quand l'article ne parle pas de la nouvelle évoquée dans le titre, agree quand l'article confirme la nouvelle du titre, disagree quand la nouvelle contredit la nouvelle du titre et enfin discuss quand l'article ni confirme ni contredit la nouvelle du titre.

FNC Score métrique: Le FNC propose comme métrique d'évaluation d'un modèle le calcul d'un score en fonction des prédictions. On a utilisé cette métrique pour quantifier la qualité de notre modèle pendant l'entrainement. Le score se calcul de la manière suivante, si la position est unrelated et que la position prédite est également unrelated alors on gagne 0.25 points. Si la position n'est pas unrelated et que la position prédite est une des trois autres alors on gagne 0.25 points et si la position prédite est correcte on gagne 0.75 points.

Pour avoir une idée de l'échelle de la métrique sur les ensembles de données, on a calculé le score maximum qui corresponds au cas où toutes les positions sont correctement prédites et le null score qui corresponds au cas où toutes les positions sont prédites à unrelated. On propose les scores pour chaque ensemble dans le tableau ci-dessous.

Set	**Max Score**	**Null Score**
Train	18106.0	7339.0
Validation	4457.25	1797.25
Test	11651.25	4587.25

2.2 Embedder

L'embedder permet la transformation du texte en valeur numérique. Sous forme de vecteur la représentation du texte peut être utilisé dans notre modèle pour résoudre le problème de la Stance Detection. On a utilisé deux types d'Embedder qui permettent une vectorisation de chaque mot d'un texte.

Comme on va travailler avec des batchs pendant l'apprentissage, chaque Embedder produit une sequence de vecteur de la taille de la plus grande sequence dans le batch. C'est à dire, si dans un batch le plus grand texte contient n tokens après traitement alors on aura n représentations vectorielles pour chaque élement dans le batch (les tokens vide sont rajoutés pour les éléments de plus petites tailles dans le batch).

Google News Vector: L'Embedder Google News propose un embedding des mots par un vecteur de taille 300 et il dispose d'un vocabulaire de 3 millions de mots anglais. Il a été entrainné sur un corpus de 3 milliards de mot issus d'article. L'Embedder Google est très intéressant pour notre problème puisqu'on travaille également sur des articles.

Le vocabulaire se fait sur les mots, nous avons donc supprimé la ponctuation et les caractères spéciaux du texte avant d'efféctuer la séparation par mot. En appliquant un 3/2 Grams sur le texte, on a obtenu un vecteur avec les tokens correspondant à chaque mot du texte. L'utilité du 3/2 Grams est de repérer, par exemple, les mots composés ou les associations prénom nom. Le vecteur avec les tokens est ensuite traité par l'Embedder Google pour obtenir le vecteur de taille 300 pour chaque mot du texte (chaque token corresponds à un vecteur de taille 300 fixe).

ELMo : L'Embedder ELMo fait partie des meilleurs modèle actuellement, il permet une vectorisation contextualisé du texte. ELMo repose sur une architecture BiLSTM qui permet le traitement des mots et de leur contexte d'utilisation. De plus il utilise une tokenisation au niveau des caractères ce qui permet de garder l'information liée à la ponctuation. Pour obtenir le vecteur de mot à fournir à ELMo, on a simplement séparé le texte en utilisant les espaces comme séparateur.

Dans notre pipeline on peut choisir 3 versions de ELMo, Small, Medium et Big qui produisent respectivement pour chaque mot du texte des vecteurs de taille 256, 512 et 1024. Les versions Small et Medium ont été entrainé sur un corpus de 1 milliard de mot et la version Big sur un corpus de 5.5 milliards.

A la différence de l'Embedder Google, ELMo utilise le contexte dans lequel est employé le mot, ainsi la vecteur final n'est pas fixe pour un mot et il dépendra des mots qui l'entourent. Avec l'Embedder ELMo le temps de calcul augmente

grandement puisque le texte passe dans un modèle complexe pour produire son vecteur. Dans ce projet, on a utilisé la version Big.

3 Models

Dans cette partie on va présenter le modèle utilisé pour la Stance Detection. A partir de la représentation vectorielle des titres et des corps d'articles obtenues par l'Embedder, on souhaite prédire la position. On a développé deux modèles dont un BiLSTM qu'on ne présentera pas dans le rapport puisque les résultats obtenus sont moins bons que ceux du CNN.

3.1 FakeNewsCNN

Notre FakeNewsCNN se décompose en deux parties, la première corresponds à 5 couches de convolution à 1 dimension et la deuxième corresponds à quatre couches denses. Le nombre de neuronnes dans la couche d'entrée est égale à la dimension des vecteurs obtenus par l'Embdder, et la couche de sortie est composée de 4 neuronnes (un pour chaque position à prédire). On applique un LogSoftMax sur les activations de la couche de sortie pour obtenir une probabilité de prédiction pour chaque position. Pour la dimension des couches du réseaux, on utilise un paramètre auquel on applique un facteur. La version qu'on a utilisé a pour paramètre 256. Le détail des dimensions est visible directement dans le code.

Pour traiter un couple titre et corps d'article, on passe séparemment les représentations vectorielles du titre et du corps d'article dans la première partie du FakeNewsCNN (partie convolution) puis on concatène les features maps obtenues pour le titre et pour le corps. Les features map concaténnée sont ensuite passées dans la deuxième partie du FakeNewsCNN (partie dense).

En procédant de cette façon, les mêmes features sont appliquées sur le titre et sur le corps de l'article et on doit obtenir des features map similaires. Enfin l'utilisation des features map concaténnées dans un réseau dense permet la detection des similarités entre la feature map du titre et celle du corps (repérer si une partie du contenu du titre est présente dans le corps de l'article).

3.2 Learning process

Pour l'entrainement on a utilisé l'optimiseur Adam avec un learning rate de 0.0001 pour les 30 premières epoch puis en repartant du meilleur modèle obtenu on a effectué 30 nouvelles epochs avec un taux de 0.00005 (le meilleur modèle est en epoch 29, après il y a eu overfitt). On a utilisé des batchs de taille 16 avec l'Embedder Google et de taille 8 avec ELMo. On présente dans la Figure 1 l'évolution des métriques au cours des epochs.

Results : On présente dans le Tableau ci-dessous les performances (métriques) des différents modèles sur l'ensemble de test.

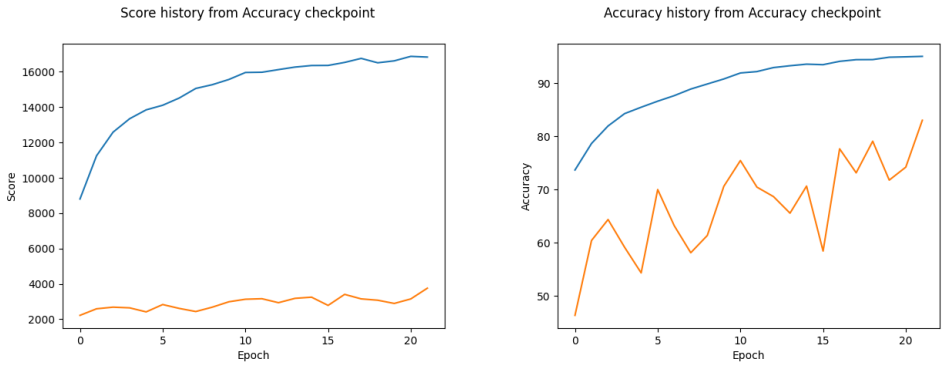

Figure 1: GOOGLE CNN - A gauche l'historique de la métrique score et à droite de la métrique accuracy (sur le train en bleu et sur la val en orange)

Model	**Embedder**	**Checkpoint**	**Score**	**Acc**	**Max**	**Null**
CNN	GOOGLE	Score	**5926.75**	**42.29**	11651.25	4587.25
CNN	GOOGLE	Acc	**5926.75**	**42.29**	11651.25	4587.25
CNN	ELMo	Score			11651.25	4587.25
CNN	ELMo	Acc			11651.25	4587.25

Remarque: Comme mentionné dans le mail, on vous prie de nous excuser pour le retard et on vous envoie prochaînement les résultats avec l'Embedder ELMo.