LIRMM@DEFT-2018 – Modèle de classification de la vectorisation des documents

Waleed Mohamed Azmy[1] Bilel Moulahi[1, 2] Sandra Bringay[1] Jérôme Azé[1], Maximilien Servajean[1]
(1) LIRMM, Université de Montpellier, CNRS, Montpellier, France
(2) IUT de Béziers, Université de Montpellier, France
prenom.nom@lirmm.fr

RÉSUMÉ

Dans ce papier, nous décrivons notre participation au défi d'analyse de texte DEFT 2018. Nous avons participé à deux tâches : (i) classification transport/non-transport et (ii) analyse de polarité globale des tweets : positifs, negatifs, neutres et mixtes. Nous avons exploité un réseau de neurone basé sur un perceptron multicouche mais utilisant une seule couche cachée.

ABSTRACT

LIRMM DEFT-2018 – Document Vectorization Classification model

In this paper, we describe our participation to the DEFT 2018 French Text Mining Challenge. The goal of the challenge is the sentiment analysis of French tweets. We participated to two tasks : (i) Transport/non-transport classification and (ii) Polarity analysis of positive, negative, neutral and mixed sentiments. We explored a neural network based on MultiLayer Perceptron using only one hidden layer.

MOTS-CLÉS : Analyse de polarité, réseaux de neurone, word embedding, doc2vec.

KEYWORDS: Polarity analysis, neural networks, word embedding, doc2vec.

1 Modèle de l'équipe ADVANSE du LIRMM pour l'édition 2018 de DEFT

De nombreux modules de traitement du langage naturel (NLP) commencent par l'extraction de certaines caractéristiques importantes du texte. Ces caractéristiques peuvent être, par exemple, le nombre ou la fréquence de mots spécifiques, des motifs prédéfinis, l'étiquetage grammatical, etc. Ces caractéristiques sont souvent définies manuellement et doivent être choisies avec soin, voire même nécessiter l'intervention d'un spécialiste des données étudiées. Bien que des résultats intéressants puissent être obtenus avec de telles approches, l'un des inconvénients récurrents est souvent la faible capacité à généraliser.

Depuis quelques années, plusieurs approches proposent d'utiliser des méthodes de vectorisation de mots et de documents. Ces stratégies qui convertissent des mots, des phrases ou même des documents entiers en vecteurs prennent en considération l'ensemble du texte et pas seulement certaines de ces parties. Il existe de nombreuses façons de transformer un texte en un espace à haute dimension comme la fréquence des termes et la fréquence inverse des documents (TF-IDF), l'analyse sémantique latente

(LSA), l'allocation de Dirichlet latente (LDA), etc(Maas *et al.*, 2011).

Cette nouvelle approche a été révolutionnée par Mikolov et al(Mikolov *et al.*, 2013a,b) qui a proposé le Continuous Bag Of Words (CBOW) et les modèles de sauts de grammaires connus sous le nom de Word2Vec. Il s'agit d'un modèle probabiliste qui utilise une architecture de réseau de neurones à deux couches pour calculer la probabilité conditionnelle d'un mot compte tenu de son contexte. Sur la base de ces travaux, Le et al. proposent un modèle vectoriel de paragraphe.

L'algorithme, également connu sous le nom de Doc2Vec, apprend les représentations de longueur fixes à partir de textes de longueur variable, tels que des phrases, des paragraphes et des documents(Le & Mikolov, 2014). Les vecteurs de mots et les vecteurs de documents sont formés par les modèles de langage de gradient stochastique et de rétro-propagation des réseaux neuronaux.

Tout d'abord, nous avons formé un modèle Doc2Vec sur l'ensemble du corpus d'entraînement fourni par les organisateurs du défi. Chaque tweet est traité comme un document séparé. Après avoir construit le modèle de vectorisation du document, chaque tweet peut être représenté avec un vecteur de N caractéristiques. Les principaux paramètres de construction d'un tel modèle sont donnés dans le tableau 1. Nous faisons varier le nombre de dimensions de 100 à 400 et essayons d'optimiser le modèle en utilisant la validation croisée et la mesure de précision. Pour les deux tâches, le nombre de dimensions pouvant représenter un tweet était égal à 250.

Paramètre	**Valeur**
Learning Rate	0.001
Nombre de dimensions utilisées	de 100 à 400
Context Window Size	10
Training epochs	20
Loss	Negative Sampling
Minimum word count	2

TABLE 1 – Document Vectorization Main Parameters

Ces vecteurs sont utilisés comme ensemble d'apprentissage pour entraîner un second réseau neuronal basé sur un perceptron multicouches. Pour la première tâche, deux classes ont été utilisées, alors que pour la seconde tâche, nous avons utilisé quatre classes. Nous avons également utilisé la descente de gradient stochastique et le réseau neuronal de rétro-propagation avec une couche cachée de 150 neurones.

2 Résultats

Les résultats des deux tâches, ainsi que quelques statistiques sur les autres soumissions sont présentés dans le tableau 2. La micro moyenne F1-Mesure est utilisée pour évaluer les expérimentations. Il y a 39 soumissions pour la première tâche et 41 pour la deuxième tâche. Les résultats montrent que le modèle n'arrive pas à prédire l'information globale et ne prête pas attention aux sentiments. Le résultat de la première tâche semble être meilleur, mais en général, les modèles devraient être plus profonds pour que nous puissions obtenir de meilleures performances.

	Notre modèle	Moyenne	Déviation Standard	Min	Max
Task-1	0.827	0.89	0.032	0.719	1
Task-2	0.38	0.727	0.162	0.38	1

TABLE 2 – Micro-mean F1 measure for the proposed model and statistics from other models

3 Conclusion et discussion

Notre modèle essaie simplement de faire une classification sans dictionnaires ou caractéristiques spécifiquement créées pour la tâche d'intérêt. Notre travail peut être vu comme une première étape en essayant de donner une réponse à la question ouverte : "Devrions-nous nous préoccuper de la linguistique ?" Nous pensons clairement que la réponse fournie par ce travail préliminaire est "oui".

L'utilisation des modèles CBOW et Skip-grams pour vectoriser le texte pourrait être bénéfique, mais l'inclusion de certaines caractéristiques du dictionnaire ou de signaux d'attention peut aider. Une autre façon est de construire le deuxième modèle en apprenant beaucoup plus en profondeur et non en utilisant seulement un réseau neuronal plat. Certains modèles tels que Convolution Neural Networks (CNN) ou Recurrent Neural Networks (RNN) permettent de pousser le modèle à aller plus loin et à prendre en considération les sentiments.

Remerciements

Nous remercions la région Occitanie et l'Agglomération Béziers Méditerranée qui finance la thèse de Waleed Mohamed Azmy, ainsi que la Fondation FondaMental qui finance le contrat d'ingénieur de recherche de Bilel Moulahi.

Références

LE Q. V. & MIKOLOV T. (2014). Distributed representations of sentences and documents. *CoRR*, **abs/1405.4053**.

MAAS A. L., DALY R. E., PHAM P. T., HUANG D., NG A. Y. & POTTS C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, volume 1.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, **abs/1301.3781**.

MIKOLOV T., YIH S. W.-T. & ZWEIG G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (NAACL-HLT-2013)* : Association for Computational Linguistics.