DEFT 2018: Attention sélective pour classification de microblogs

Charles-Emmanuel Dias[1] Clara Gainon de Forsan de Gabriac[1] Vincent Guigue[1]
Patrick Gallinari[1]
Sorbonne Université, CNRS,
Laboratoire d'Informatique de Paris 6, LIP6,
F-75005 Paris, France
<prenom.nom-de-famille>@lip6.fr

RÉSUMÉ

Dans le cadre de l'atelier DEFT 2018 nous nous sommes intéressés à la classification de microblogs (ici, des tweets) rédigés en français. Ici, nous proposons une méthode se basant sur un réseau hiérarchique de neurones récurrent avec attention. La spécificité de notre architecture est de prendre en compte –via un mecanisme d'attention et de portes– les *hashtags* et les mentions directes (*e.g.,* @*user*), spécifiques aux microblogs. Notre modèle a obtenu de très bon résultats sur la première tâche et des résultats compétitifs sur la seconde.

ABSTRACT

DEFT 2018 : Selective Attention for Microblogging Classification

The 2018 DEFT challenge allowed us to investigate sentiment analysis and document classification applied to microblogs (here, twitter) written in French. We hereby present a method based on a hierarchical attentional recurrent neural network. Our architecture is specifically engineered to take advantage of hashtags and direct mentions – specificic of microblogs – by the mean of an attention and gate mechanism. Our model scored very good results on the first task and competitve ones on the second task.

MOTS-CLÉS : Classification, Analyse de Sentiments, Réseaux de Neurones, Attention..

KEYWORDS: Classification, Sentiment Analysis, Neural Networks, Attention..

1 Introduction

L'atelier DEFT 2018 (Paroubek *et al.*, 2018) proposait différentes tâches autours de l'extraction d'informations de tweets francophones. Ici, nous nous sommes focalisés sur les deux premières tâches de classification. Pour résoudre ces deux problèmes, nous proposons ici de nous inspirer d'un modèle neuronal hiérarchique issu de l'analyse de sentiments (Yang *et al.*, 2016) que nous modifions pour proposer un modèle de représentation prenant en compte les diverses spécificités des microblogs.

En effet, les tweets ont souvent une orthographe approximative (lettres répétées, ponctuation excessive ou manquante, smileys...), un vocabulaire spécifique (acronymes, mentions, hashtags...) et même des lettres particulières (emojis en unicode). Aussi, ils regorgent de mots-clés (dits *hashtags*) et de mentions directes (@*user*). Ces mots spécifiques, précédés de marqueurs (# et @), ont souvent une forte valeur informative qu'il convient de proprement intégrer au sein du modèle d'encodage. Pour

toutes ces raisons, représenter efficacement un tweet est particulièrement compliqué, surtout sur des petits corpus. C'est généralement au prix d'une multitude d'heuristiques complexes qu'il est possible de dépasser toutes ces contraintes pour atteindre des performances proches de celles des modèles neuronaux. À l'inverse, en apprenant des représentations latentes du texte (ou en utilisant des représentation pré-apprises (Mikolov *et al.*, 2013; Pennington *et al.*, 2014)), les réseaux de neurones profonds sont moins impactés par ces problèmes syntaxiques et sémantiques. Ils sont capables –sans aucun pré-traitement– de détecter des constructions textuelles avancées (comme les doubles négations) et peuvent même encoder une certaine information sémantique (pluriel, féminin-masculin,...). De ce fait, les machines à vecteur de support se basant sur des représentations de type *sac de mots* ou de *N-grams*, qui furent longtemps les modèles les plus performants pour résoudre les tâches de classification de documents, sont désormais majoritairement surpassés par les modèles neuronaux avec leurs représentations continues.

Ici, nous nous inspirons des travaux de (Yang *et al.*, 2016) pour construire notre modèle de représentation et de classification de microblogs. Ils proposent d'encoder hiérarchiquement le texte (mot par mot puis phrase par phrase) tout en apprenant conjointement –via un mécanisme d'attention– quels sont les éléments discriminants. Contrairement à eux, nous faisons l'hypothèse qu'encoder les tweets caractère par caractère est plus pertinent que mot à mot. Nous postulons qu'un tel encodage permet de nous abstraire de chacun des problèmes grammaticaux énoncés précédemment. Aussi, pour prendre en compte les hashtags et les mentions, nous y ajoutons un mécanisme d'attention et d'interpolation spécifique.

Cet article est organisé de la façon suivante : nous détaillons dans un premier temps les principaux éléments de notre modèle avant de présenter son architecture globale (section 2). Ensuite, nous évaluons notre modèle quantitativement et commentons les résultats obtenus lors de la phase d'évaluation (section 3). Enfin, nous proposons certaines pistes pour l'amélioration de notre méthode de représentation de tweet (section 4).

2 Modèle de classification de microblogs

Généralement, un tweet est un texte court –souvent écrit sur mobile– dont l'orthographe peut être approximative. Ces textes contiennent souvent des mots inexistants, abrégés ou argotiques. Nous considérons que seul l'espace est un caractère fiable, et qu'il agit comme un séparateur entre les mots. Aussi, nous faisons le choix de ne pas prendre les phrases en compte mais uniquement les caractères et les mots (séparés par les espaces). Ici, nous voyons donc un tweet comme une suite de caractères, divisée en plusieurs mots.

Pour notre modélisation nous nous inspirons du modèle hiérarchique d'analyse de sentiment de (Yang *et al.*, 2016) qui est composé de deux sous-entités similaires : des modules bi-directionnels attentifs (nous les appelons RBA). Leur rôle est d'encoder tour à tour les mots et les phrases pour représenter un texte avant d'en prédire sa polarité. Ici, ne prenant pas en compte les phrases, nous descendons d'un niveau hiérarchique et nous proposons d'encoder séquentiellement les mots –caractère par caractère– puis le message mot par mot.

Dans un premier temps nous explicitons la construction d'un module bi-directionnel attentif. Ensuite, nous détaillons comment le modèle prend en compte les mots clés et les mentions avant de décrire l'intégralité du modèle.

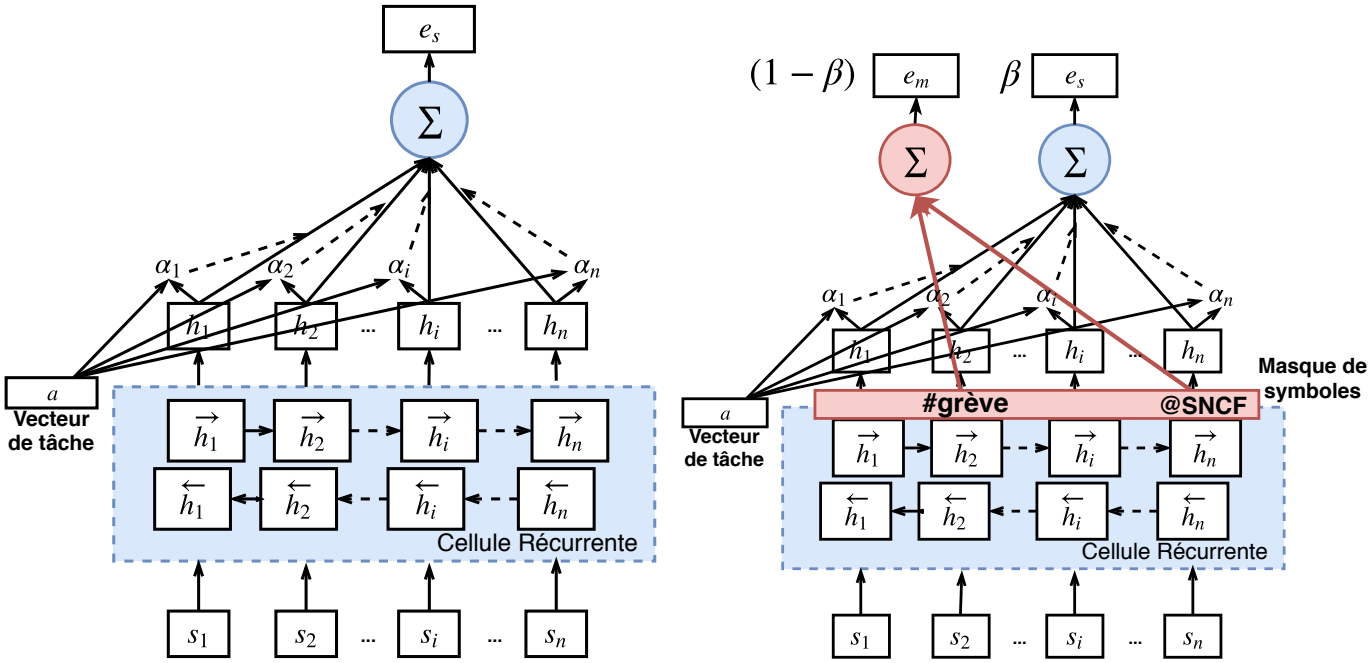

FIGURE 1 – Module **R**écurrent **B**i-directionnel avec **A**ttention **(RBA)** : Sans attention personnalisé (gauche) – Avec attention personnalisé (droite)

2.1 Module Récurrent Bi-directionel avec attention : le RBA

Ce module est le principal bloc de notre modèle de classification. Il prend en entrée une séquence et retourne une représentation transformée et pondérée de celle-ci. Dans le reste de l'article, nous appelons ces sous-modules RBA pour *Module **R**eccurent **B**i-directionnel avec **A**ttention*.

Encodage d'une séquence avec un RBA

Formellement, soit une séquence $seq = \{\mathbf{s}_1, \ldots, \mathbf{s}_i, \ldots, \mathbf{s}_n\}$ composée de n éléments. Pour obtenir sa représentation $\mathbf{e}_s$, la séquence est d'abord passée par un réseau de neurones récurrent bi-directionnel $RB = \{\overrightarrow{RB}, \overleftarrow{RB}\}$ qui, en parcourant la séquence dans les deux sens, encode le contenu intra-séquence. Les sorties du réseau récurrent sont concaténées à chaque pas de temps pour obtenir l'ensemble des représentations cachées $\mathbf{h}_i$ (eq. 1). Ici, nous utilisons une cellule GRU (Chung *et al.*, 2014) comme cellule récurrente.

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}], \qquad \overrightarrow{\mathbf{h}_i} = \overrightarrow{RB}(\mathbf{s}_i), \qquad \overleftarrow{\mathbf{h}_i} = \overleftarrow{RB}(\mathbf{s}_i), \tag{1}$$

Ensuite, chaque élément $\mathbf{h}_i$ est projeté de manière non linéaire dans un espace d'attention afin de calculer son affinité α_i avec un vecteur de tâche $\mathbf{a}$ –qui est lui-même appris lors de l'optimisation– selon la formule suivante :

$$\mathbf{t}_i = \tanh(W^{tu}\mathbf{h}_i + b_u), \qquad \alpha_i = \frac{\exp(\mathbf{a}^\intercal \mathbf{t}_i)}{\sum_i \exp(\mathbf{a}^\intercal \mathbf{t}_i)} \tag{2}$$

Ces affinités α_i sont normalisées à l'aide d'une fonction *softmax* afin qu'elles somment à 1. Ce vecteur de tâche $\mathbf{a}$ correspond au point optimal dans l'espace d'attention. (Yang *et al.*, 2016) a

montré qu'un tel vecteur de tâche –appris comme un paramètre– permettait au modèle de se focaliser automatiquement sur les éléments discriminants d'une séquence en fonction de la tâche.

$$\mathbf{e}_s = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i \tag{3}$$

Enfin, la représentation finale e_s de la séquence d'entrée est la somme des représentations cachés h_i, pondérée par l'attention α.

Attention personnalisé sur les mots-clés #hastags et les mentions @users

Si l'idée de pondérer les éléments d'une séquence pour dénoter leur importance relative est très répandue, l'originalité de notre approche réside dans une prises en compte particulière des éléments propres aux tweets : les hashtags et les mentions. En effet, nous partons du postulat que ces deux éléments fournissent des indices important pour les tâches subséquentes de classification.

Pour mieux prendre en compte ces mots spécifiques, nous proposons un système d'ajouter un système d'attention-interpolation au RBA qui servira à encoder les représentations de mots en une représentation du tweet. Nous travaillons donc sur le second niveau hiérarchique.

Nous proposons dans un premier temps de sélectionner les termes importants via un processus de masquage : soit la matrice –encodant m mots sur n dimensions– de l'ensemble des états cachés $H \in \mathcal{R}^{m,n}$ et $\mathbf{M} \in \{0^n, 1^n\}^m$ une matrice avec $m_{i,*} = 1^n$ si le *i*-eme mot commence par un caractère prédéfini –ici le dièse # ou l'arobase @– et $m_{i,*} = 0^n$ sinon. La représentation de ces mots spécifique est simplement leurs somme. $\otimes$ est le produit terme à terme.

$$e_m = \sum_i (H \otimes M) \tag{4}$$

Finalement, la représentation du tweet e_t est l'interpolation entre cette somme de mots sélectionnés e_m et la somme pondérée par l'attention classique e_s obtenue sur l'intégralité des mots. Cette interpolation est linéaire, de coefficient β, fonction des deux représentations.

$$e_t = (1 - \beta) \times e_m + \beta \times e_s, \qquad \beta = \sigma(W^b[e_s; e_t] + b_b), \qquad \sigma = sigmoid(x) \tag{5}$$

Ce système d'attention permet de sur-pondérer les termes importants dans la représentation finale. Cela permet à l'attention classique de se focaliser sur les marqueurs traditionnels.

2.2 Architecture globale du modèle

Notre modèle fonctionne de manière hiérarchique. Il prend en entrée une liste de liste de caractères et prédit une classe. Tout d'abord, n représentations latentes de mots $e_w(k)$ sont construites au moyen d'un premier RBA (RBA_c) qui encode chaque k mot de m lettres caractère par caractère. Ensuite, à l'aide d'un second RBA (RBA_m), la représentation finale du tweet e_t est construite à partir toutes les représentations de mots précédemment obtenues.

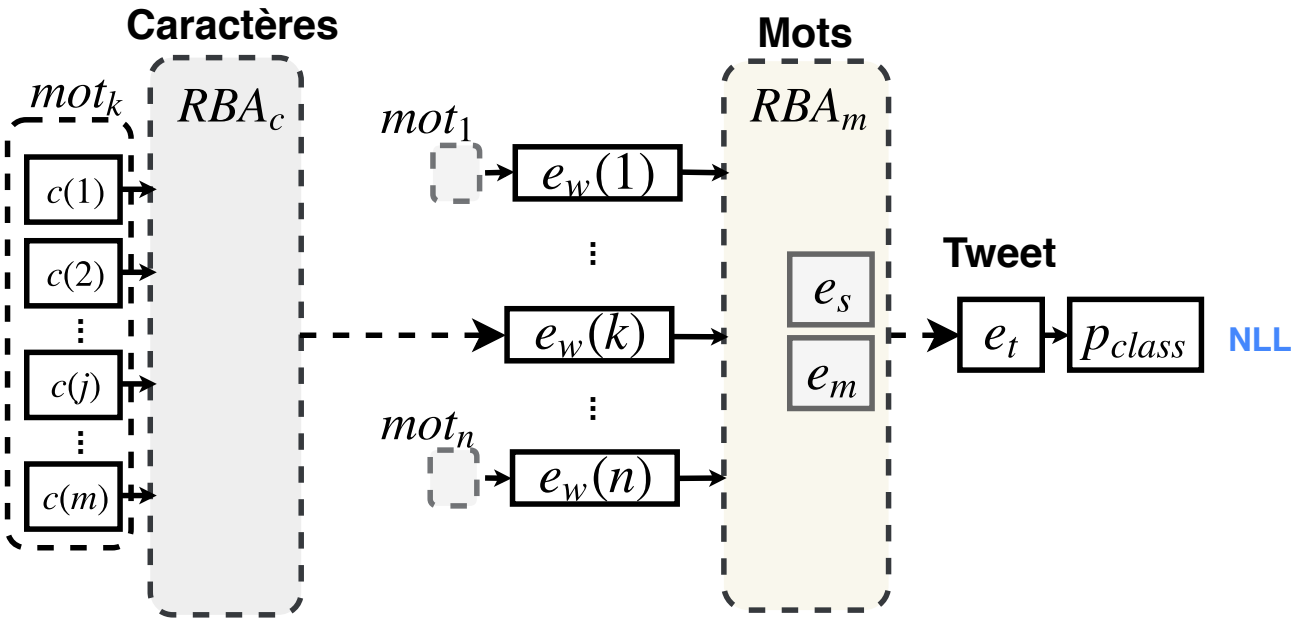

FIGURE 2 – Modèle hiérarchique de classification de microblog complet, composé de deux RBA. RBA_w encode d'abord chaque mots caractère par caractère puis RBA_m encode le tweet mot à mot en utilisant le système d'attention personnalisée

$$e_w(k) = RBA_c([c_0, \cdots, c_m]), \qquad e_t = RBA_m([e_w(1), \cdots, e_w(k), \cdots, e_w(n)]) \tag{6}$$

Enfin, une couche de classification softmax permet la classification finale.

$$p_{class} = softmax(W^{tc} e_t + b_c) \tag{7}$$

Pour entraîner le modèle, nous minimisons l'entropie croisée par descente de gradient par mini-batch. Ici, nous proposons le même modèle pour les deux tâches, la taille de la couche finale varie donc en fonction du nombre de classes à prédire.

3 Evaluation

Dans le cadre de cet atelier, nous présentons une modification du modèle de (Yang *et al.*, 2016) prenant en compte certaines spécificités des microblogs. Proposant un modèle de classification, nous participons aux tâches 1 et 2. L'une étant bi-classes et l'autre multi-classes.

Dans cette section, nous présentons dans un premier temps les données ainsi que les deux tâches que nous avons considérés. Puis, nous reportons les évaluations de notre modèle. Enfin, nous commentons les résultats obtenus.

3.1 Données, pré-processing et tâches

Les données de l'atelier DEFT2018 (Paroubek *et al.*, 2018) sont –modulo les erreurs d'annotation– des tweets annotés selon leur polarité si et seulement s'ils traitent des transports. Les étiquettes sont

aux nombre de cinq : INCONNU, NEUTRE, NEGATIF, POSITIF, MIXPOSNEG.

Tâche 1 – Transport/non-transport : Cette première tâche consiste à faire de la classification thématique. L'enjeu est de séparer les tweet traitant du transport de ceux parlant d'autres choses. Formellement, les tweets avec l'étiquette INCONNU sont considéré comme ne traitant pas de transports alors que ceux avec n'importe quelle autre étiquette sont considérés comme traitant de transports.

Tâche 2 – Analyse de sentiment : Cette deuxième tâche est orientée analyse de sentiments. L'objectif est de classifier les tweets selon leur polarité : NEUTRE, NEGATIF, POSITIF, MIXPOSNEG

Pré-traitement : Chaque tweet est divisé en mots en utilisant l'espace comme séparateur. Ensuite chaque mot est transformé en liste de caractères. Tous les caractères sont utilisés, même ceux n'apparaissant qu'une seule fois dans le corpus d'entraînement. Pour nous auto-évaluer, les données d'entraînement sont séparées en cinq ensembles égaux pour effectuer de la validation croisée. Enfin, pour chaque runs d'évaluation, nous en utilisons quatre (80% des données) pour l'entraînement, et un –divisé en deux– pour la validation (10%) et l'évaluation (10%).

3.2 Résultats

Le premier tableau compare nos résultat en auto-évaluation avec nos résultat sur le corpus d'évaluation non annoté. On peut voir que nos résultats sont cohérents, notre modèle n'a a priori pas sur-appris sur le corpus d'entraînement. Lorsque l'on s'intéresse au classement. On peut voir que sur la tâche 1 notre modèle est compétitif puisqu'un de nos runs arrive $2^{ème}$ en terme de performance. Sur la tâche 2 en revanche, notre modèle est bien en dessous des autres puisque nous nous classons $6^{ème}$ au mieux [1].

	Partie 1	Partie 2	Partie 3	Partie 4	Partie 5	Moyenne	Evaluation
Tâche 1	83.67	83.88	83.75	85.40	85.39	84.42	83.048
Tâche 2	65.83	66.07	64.65	67.47	67.41	66.28	65.556

TABLE 1 – Précision de classification de notre modèle. Les valeurs présentés à gauche sont celles obtenues en auto-évaluation par validation croisées sur cinq ensembles. A droite, les valeurs présentés sont les résultats obtenus sur les données tenues secrètes

4 Discussion

Ici, pour répondre aux tâches 1 et 2 de l'atelier DEFT2018, nous avons proposé un modèle de représentation hiérarchique des tweets, dérivé des travaux de (Yang *et al.*, 2016). L'originalité de notre modèle est de prendre en compte les diverses spécificités qui font que les tweets sont souvent compliqués à traiter. Ici, nous présentons un modèle avec des pré-traitement minimes et ayant des résultats compétitifs.

1. Après vérification, nous avions entraîné notre modèle à classifier les tweets en INCONNU en plus des sentiments (une classe en plus), de ce fait nous affichons une performance moindre en évaluation. S'il on soustrait les 190 tweets classés comme INCONNU, la performance ce même modèle est $\tilde{6}8\%$ de précision

Equipe	Tâche 1	Rang	Tâche 2	Rang
Lip6 (Nous)	83.048	2	65.824	6
EDF	82.293	5	67.013	4
CLaC	77.955	10	33.494	10
Tweetaneuse	**83.124**	**1**	67.699	3
IRIT	82.433	4	69.906	2
IRISA	82.702	3	**70.258**	**1**
ELOQUANT	81.408	6	66.684	5
EPITA	80.502	8	64.172	7
UTTLM2S	79.580	9	63.004	8
SYLLABS	80.604	7	–	12
ADVteam	70.509	11	29.778	11
LISlab	–	12	47.628	9

TABLE 2 – Performance en précision et rang de l'ensemble des équipes sur les tâches 1 et 2. Les meilleures performances sont en gras.

Pour aller plus loin dans la prise en compte des particularités afférentes aux tweets, les mentions et les hashtags pourraient également avoir des représentations spécifiques, non-liés à celles des lettres. En effet, l'utilisation de modèles hybrides utilisant à la fois les mots et les lettres permettent parfois des gains de performances.

Remerciements

Ce travail a été réalisé en partie avec le soutien du FUI-BIND

Références

CHUNG J., GÜLÇEHRE Ç., CHO K. & BENGIO Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, **abs/1412.3555**.

MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. & DEAN J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, **abs/1310.4546**.

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L. & TORRES-MORENO J.-M. (2018). Deft2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en île de france. In *Actes de DEFT*, Rennes, France.

PENNINGTON J., SOCHER R. & MANNING C. D. (2014). Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, p. 1532–1543.

YANG Z., YANG D., DYER C., HE X., SMOLA A. J. & HOVY E. H. (2016). Hierarchical attention networks for document classification. In *HLT-NAACL*.