Participation de l'IRISA à DeFT 2018 : classification et annotation d'opinion dans des tweets

Anne-Lyse Minard[1] Christian Raymond[1,2] Vincent Claveau[1]
(1) CNRS, IRISA, Univ Rennes
(2) INSA Rennes, Rennes
Campus de Beaulieu, 35042 Rennes, France
prenom.nom@irisa.fr

RÉSUMÉ

Cet article décrit les systèmes développés par l'équipe LinkMedia de l'IRISA pour la campagne d'évaluation DeFT 2018 portant sur l'analyse d'opinion dans des tweets en français. L'équipe a participé à 3 des 4 tâches de la campagne : (i) classification des tweets selon s'ils concernent les transports ou non, (ii) classification des tweets selon leur polarité et (iii) annotation des marqueurs d'opinion et de l'objet à propos duquel est exprimée l'opinion. Nous avons utilisé un algorithme de boosting d'arbres de décision et des réseaux de neurones récurrents (RNN) pour traiter les tâches 1 et 2. Pour la tâche 3 nous avons expérimenté l'utilisation de réseaux de neurones récurrents associés à des CRF. Ces approches donnent des résultats proches, avec un léger avantage aux RNN, et ont permis d'être parmi les premiers classés pour chacune des tâches.

ABSTRACT

IRISA at DeFT 2018: classifying and tagging opinion in tweets

This paper describes the systems developed at IRISA by the LinkMedia team for the challenge DeFT 2018. The challenge focuses on opinion mining in French tweets about transports. The team has participated in 3 out of the 4 tasks: (i) classification of the tweets whether they are about transports or not, (ii) classification of the tweets according to their polarity and (iii) fine grained annotation of the sentiment expression and the object about which an opinion is given. For the tasks 1 and 2, we have used a boosting algorithm as well as recurrent neural networks (RNN). For the 3rd task, we have experimented the use of recurrent neural networks combined with some CRF. All the approaches give close results, with a slight advantage when using RNN, and yields among the best results for every tasks.

MOTS-CLÉS : analyse d'opinion, boosting, arbres de décision, réseau de neurones récurrents, plongement de mots, CRF.

KEYWORDS: opinion mining, boosting, decision trees, recurrent neural networks, word embedding, CRF.

1 Introduction

Cet article présente les systèmes que l'IRISA a développés dans le cadre de sa participation à la campagne d'évaluation DeFT 2018. Cette campagne porte sur l'analyse de sentiments dans des tweets qui concernent les transports. Elle fait suite à la campagne DeFT 2017 portant elle-aussi sur la

	# tweets	transport	positif	negatif	neutre	mixposneg
simple	54 638	35 468	7 328	13 109	12 611	2 420
batch_b	14 278	0	-	-	-	-
total	68 916	35 468	7 328	13 109	12 611	2 420

TABLE 1: Distribution des annotations dans le corpus d'entraînement pour les tâches 1 et 2.

fouille d'opinion. Nous avons donc repris et adapté certaines techniques développées précédemment (Claveau & Raymond, 2017). Notre équipe a participé à trois des quatre tâches proposées par les organisateurs :

— tâche 1 : classification des tweets selon qu'ils concernent les transports ou non ;
— tâche 2 : classification des tweets concernant les transports selon leur polarité (POSITIF, NEGATIF, NEUTRE, MIXPOSNEG) ;
— tâche 3 : identification des marqueurs de sentiments et de la cible du sentiment correspondant (annotation d'empans de texte et de relations).

La tâche 4, à laquelle nous n'avons pas participé, consiste à déterminer l'entité qui exprime le sentiment (source), les négations et les modifieurs, ainsi que les relations entre ces éléments.

Des données d'entraînement étant fournies par les organisateurs du challenge, nous avons classiquement adopté une approche d'apprentissage supervisé. Nous avons expérimenté deux méthodes d'apprentissage reposant sur des fondements différents, et sur des représentations différentes des données : le boosting d'arbres de décision et les réseaux de neurones récurrents.

Dans la suite de l'article, nous présentons dans la section 2 notre participation aux tâches 1 et 2, puis dans la section 3 nous détaillons nos expérimentations dans le cadre de la tâche 3.

2 Classification de tweets : tâche 1 et 2

La tâche 1 consiste à classer les tweets selon qu'il y soit question des transports ou non. Pour cela nous avons à disposition un corpus de 68 916 tweets dont 35 468 sont classés TRANSPORT (voir le tableau 1). Le corpus distribué par les organisateurs est composé de deux sous-corpus : "simple" et "batch_b". Dans la sous-partie "batch_b" du corpus, aucun tweet n'est classé TRANSPORT, même si un grand nombre de tweets concerne les transports. Nous avons donc décidé de n'utiliser que le sous-corpus "simple" pour entraîner nos modèles.

2.1 Bonzaiboost

Bonzaiboost est une implémentation de l'algorithme de boosting adaboost.MH (Laurent *et al.*, 2014) sur des arbres de décision. Cet algorithme est connu pour sa pertinence dans le domaine du traitement des langues et de l'apprentissage en général. Son utilisation constitue pour nous une solide référence sur laquelle se comparer afin d'expérimenter des systèmes plus sophistiqués. Cet algorithme appliqué sur des arbres de décision très faibles (2 feuilles) nous permet en outre de facilement interpréter le modèle appris et d'obtenir un retour d'information très intéressant. Son point faible, lié à l'algorithme de boosting lui-même, est de booster les exemples mal classés au long des itérations

lors de l'apprentissage : dans le cas de corpus bruités (avec la présence d'annotations erronées ou non cohérentes) l'algorithme insiste vainement à vouloir les classer. C'est notamment le cas, ici, où l'annotation d'opinions exprimées dans des tweets est relativement subjective et difficile.

Un modèle assez simple a été appris, où les caractéristiques extraites sont uniquement des sacs de mots convertis en minuscules. Le tableau 2 illustre les opinions marquées ainsi que la classe transport [1] par leur 13 règles les plus caractéristiques selon ce modèle. Afin de s'affranchir des variations orthographiques et de proposer des patrons plus généraux, nous avons testé d'apprendre un modèle sur des Ngrammes de caractères avec $N \in [3, 5]$ mais le système résultant est équivalent.

sncf	-2.849
@rera_ratp	-2.840
@rerc_sncf	-2.827
aéroport	-2.773
@rerb	-2.691
rer	-2.612
#sncf	-2.549
@sncf	-2.453
#ratp	-2.425
navigo	-2.277
trafic	-2.069
métro	-2.065
tramway	-2.048

(a) TRANSPORT

2016™	1.242
plaisir	1.149
ptdrrr	1.148
mdrrr	1.071
ptdrr	1.054
mdrr	1.042
mdrrrr	1.029
adore	1.020
sympa	1.000
beau	0.996
bravo	0.982
cool	0.980
rire	0.946

(b) POSITIVE

puent	1.423
fdp	1.161
accident	1.135
pue	1.112
gênant	1.074
pute	1.015
marre	0.993
flemme	0.989
honte	0.977
chier	0.939
merde	0.928
bordel	0.921
pire	0.899

(c) NÉGATIVE

TABLE 2: Treize mots les plus caractéristiques des tweets évoquant les TRANSPORTS et des opinions marquées : POSITIVES ou NÉGATIVES, accompagné de leur score de vote donné par l'algorithme de boosting.

2.2 BiLSTM+softmax

Pour résoudre ces deux tâches nous avons également expérimenté des approches à base de réseaux de neurones récurrents. La première méthode utilise une couche de LSTM bidirectionnelle (Graves *et al.*, 2013). La figure 1 décrit la méthode utilisée. La couche d'entrée prend une représentation des mots : concaténation des plongements des mots qui composent le tweet à classer et des one-hot vecteurs des mots (c'est-à-dire des vecteurs utilisés pour distinguer chaque mot dans un lexique). La couche de LSTM bidirectionnelle qui suit permet de prendre en compte l'aspect séquentiel des mots du tweet. Pour finir nous avons une couche cachée dense avec une activation softmax. En plus des couches décrites nous avons inséré des couches de Dropout pour éviter le sur-apprentissage. L'apprentissage est fait en 3 itérations et la taille du batch est celle par défaut, soit 32.

Les tweets sont prétraités de la façon suivante :

1. En l'occurrence pour la classe TRANSPORT, les mots caractéristiques sont ceux de l'absence de NON-TRANSPORT.

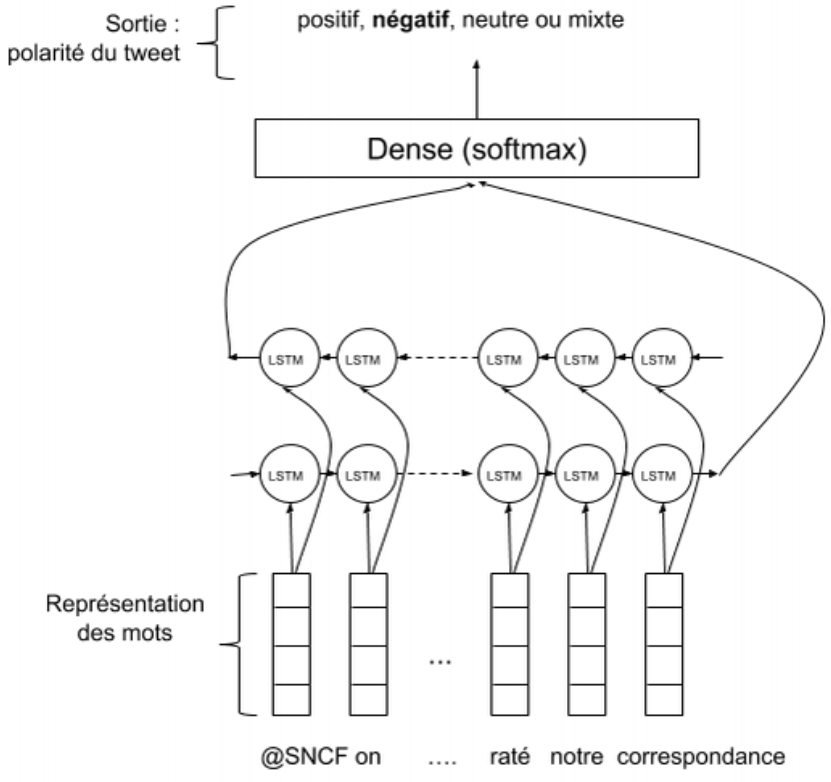

FIGURE 1: BiLSTM pour la classification des tweets.

— tokenization [2] avec le module tokenize de NLTK [3] ;
— remplacement des emojis par leur code [4] (par exemple l'emoji qui représente des mains qui applaudissent est remplacé par ":clap:") ;
— remplacement des nombres et des URLs respectivement par _num_ et _http_ ;
— tous les caractères sont convertis en minuscule ;
— tous les accents sont supprimés ainsi que la cédille.

Les plongements de mots utilisés pour représenter les mots du texte sont entraînés avec l'outil fasttext (Bojanowski *et al.*, 2017) sur des données de wikipedia et common crawl, et mis librement à disposition par Grave *et al.* (2018).

2.2.1 Expérience 1 : utilisation des données de DeFT 2017

La campagne d'évaluation DeFT 2017 s'est intéressée également à l'analyse de sentiment, avec un intérêt particulier porté aux tweets figuratifs. Les tâches 1 et 3 consistaient à classer les tweets selon leur polarité (objective, positif, négatif ou mixte). Pour la tâche 1, seuls les tweets non figuratifs étaient considérés et pour la tâche 3 à la fois des tweets figuratifs et non figuratifs. Nous avons évalué l'impact de l'utilisation des données d'entraînement et de test de DeFT 2017 sur la classification des tweets de DeFT 2018. Les résultats obtenus en validation croisée de 5 plis sont présentés dans le tableau 3. Nous observons qu'en utilisant les données de la tâche 1 de DeFT 2017 (3 906 tweets) en plus des données de DeFT 2018 nous n'améliorons pas les performances de notre système. Et

2. Les tweets fournis par les organisateurs avaient déjà été tokenisés, mais le module tokenize nous a permis d'améliorer le découpage.

3. `http://www.nltk.org/api/nltk.tokenize.html`

4. Pour remplacer les emojis par leur code nous utilisons le module python emoji disponible à l'adresse `https://github.com/carpedm20/emoji`.

DEFT 2018	+	+	+
DEFT 2017	-	T1	T3
POSITIF	64,92	64,98	63,11
NEGATIF	73,70	73,96	73,06
NEUTRE	74,24	74,07	73,70
MIXPOSNEG	31,37	31,58	25,40
Micro F-mesure	**69,93**	**69,94**	69,11

TABLE 3: Résultats obtenus en utilisant différents ensembles de données d'entraînement pour la tâche 2.

en utilisant les données de la tâche 3 de DeFT 2017 (5 118 tweets) les performances diminuent légèrement.

Les deux corpus contiennent des tweets sur des sujets différents : sujets d'actualité pour DeFT 2017 et transports pour DeFT 2018. Cette différence peut expliquer pourquoi l'utilisation des données de DeFT 2017 ne permet pas d'améliorer les performances de notre classifieur. En particulier en observant les mots du tableau 2 et ceux du tableau 5 dans l'article de Claveau & Raymond (2017), nous remarquons une grande différence dans le type des mots exprimant des opinions (par exemple pour la polarité négative "puent", "fdp", "accident", "pue" versus "pauvre", "nul", "sarko", "plein", "gvt").

Dans la suite de nos expériences nous avons donc utilisé uniquement les tweets de DeFT 2018.

2.2.2 Expérience 2 : variation de la quantité des données d'apprentissage utilisées

Nous nous sommes également intéressés aux performances de notre système en fonction de la quantité de données d'apprentissage utilisées. La figure 2 présente l'évolution de la micro F-mesure (générale et pour chaque valeur de polarité) en fonction du nombre de données d'apprentissage. Nous observons que les performances maximales du système sont atteintes avec 80% du jeu d'entraînement (soit environ 28 400 tweets). Nous pouvons faire les mêmes observations pour les classes POSITIF, NÉGATIF et NEUTRE. En revanche pour la classe MIXPOSNEG les performances augmentent encore lorsque la totalité des données d'entraînement est utilisée. Cette évaluation a été faite en validation croisée de 5 plis.

2.3 Variantes autour des RNN

Sur la base de l'approche précédente, nous avons exploré de nombreuses variantes de structure du réseau. Nous avons en particulier étudié l'apport de modèles d'attention. Ces modèles, très populaires, permettent dans le cas de données séquentielles comme le sont nos tweets, de fonder la décision du réseau sur la base de certains mots. C'est-à-dire que le réseau va être entraîné à donner beaucoup de poids aux mots de l'entrée pertinents pour prédire la classe attendue, et très peu de poids aux autres mots. En pratique, ces modèles d'attention sont implémentés sous la forme d'une couche de neurones supplémentaire avec une activation softmax et dont les poids sont ensuite multipliés à la sortie de la couche BiLSTM (ou BiGRU).

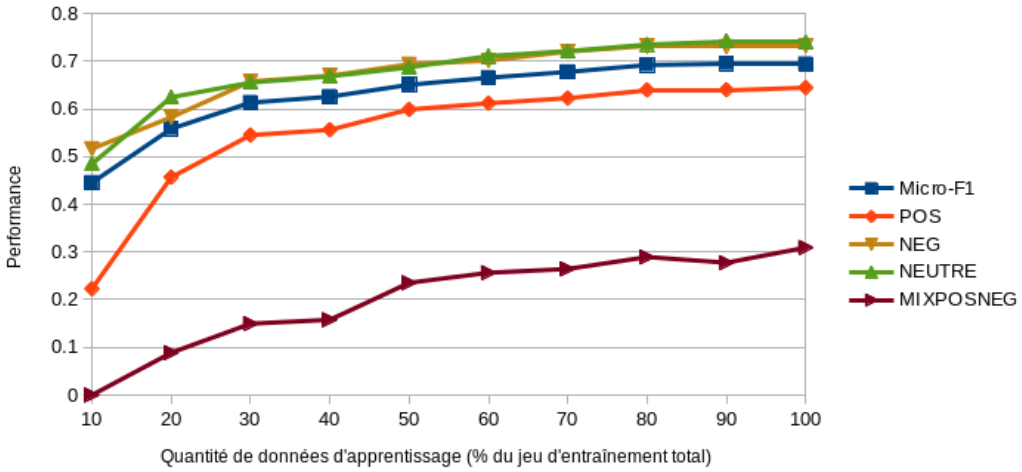

FIGURE 2: Évolution des performances du système en fonction du nombre de données d'apprentissage utilisées.

Nous avons ainsi proposé cette simple variante du système de la sous-section précédente pour la tâche 1. Pour limiter les effets du sur-apprentissage, ces modèles sont appris avec peu d'itérations, nous avons proposés un *run* avec un arrêt après trois itérations (noté RNN3) et un autre après cinq itérations (RNN5).

Pour la tâche 2, nous avons modifié plus profondément le réseau présenté dans la sous-section précédente. Nous avons deux branches avec la même architecture (Embedding, Bi-LSTM, couche d'attention, et un neurone de sortie par branche). Une branche apprend si le tweet est POSITIF ou non, et l'autre s'il est NEGATIF ou non. La combinaison des deux sorties permet bien d'avoir les quatre classes possible (NEUTRE quand les deux branches renvoient 0, MIXPOSNEG quand les deux renvoient 1). Là encore, l'apprentissage des modèles est limité à 3 et 5 itérations.

Enfin, d'autres variantes ont également été testées, portant soit sur la préparation des données, soient sur l'architecture du réseau. Leurs résultats en validation croisée étant identiques aux approches décrites ci-dessus, elles n'ont pas été soumises à l'évaluation finale. Nous avons par exemple étudié l'impact des données textuelles utilisées pour apprendre les plongements de mots, sans constater de différences importantes entre des tweets et du texte propre (wikipedia, journaux). Nous avons également testé des implémentations de word2vec et fasttext avec différents paramètres. Nous avons également essayé de diminuer l'importance des mots thématiques pour la tâche 2. Pour cela nous avons récupéré les poids de ces mots dans la couche d'attention du réseau utilisé pour la tâche 1. En effet, ces mots marqueurs thématiques du transport reçoivent un poids important pour la tâche 1. En inversant ces poids et en les intégrant à une couche, que l'on pourrait appelé couche d'inattention, dans le réseau de la tâche 2, nous espérions que le réseau se focaliserait sur les mots marqueurs d'opinion. Mais nous n'avons pas constaté de différences importantes, pour un réseau plus complexe et plus long à entraîner.

2.4 Résultats

Dans le tableau 4, nous présentons les résultats obtenus par nos 4 systèmes pour les tâches 1 et 2. Ces résultats ont été fournis par les organisateurs de la campagne d'évaluation.

En entrée des systèmes pour la tâche 2 nous utilisons les données obtenues en sortie de la tâche 1.

		Tâche 1	Tâche 2		
		P	P	R	F
Bonzaiboost	run2	82,06	67,95	94,14	78,93
BiLSTM+softmax	run1	82,66	70,23	96,50	**81,31**
RNN3	run3	82,51	69,82	95,66	80,72
RNN5	run4	**82,70**	69,81	96,23	80,92

TABLE 4: Résultats obtenus par nos systèmes pour les tâches 1 et 2 de la campagne d'évaluation DeFT2018.

Les tweets classés par erreur comme TRANSPORT (faux positif) sont ignorés lors de l'évaluation de la tâche 2. En revanche les tweets qui concernent les transports et qui ont été classés par erreur dans la classe NON-TRANSPORT (faux négatif) sont comptés comme faux négatif dans la tâche 2.

Pour la tâche 1, notre meilleur système (RNN5) obtient une précision de 82,70%, 0,42 points en dessous du meilleur système de la compétition (différence non statistiquement significative d'après un t-test pairé avec p=0.05). Pour la tâche 2, les meilleures performances sont obtenues par le système BiLSTM+softmax avec une F-mesure de 81,31%, 0,98 points de moins que le meilleur système de la compétition.

3 Annotation fine d'opinion/sentiment/émotion : tâche 3

La tâche 3 consiste à annoter les empans de texte exprimant une opinion, un sentiment ou une émotion, ainsi que la cible du sentiment, c'est-à-dire l'objet [5] à propos duquel est exprimée une opinion. Les marqueurs qui expriment un sentiment, une émotion ou une opinion (OSEE) sont associés à 20 types différents. Dans le tableau 5, nous présentons les différents types de marqueurs et leur distribution dans les données d'entraînement. Il nous était ensuite demandé de relier la cible à l'expression de l'opinion.

Cette tâche se rapproche de la tâche "Aspect-Based Sentiment Analysis" qui a eu lieu pour la première fois à SemEval 2014 (Pontiki *et al.*, 2014). Elle consistait en quatre sous-tâches, les deux premières étant les plus proches de la tâche 3 de DeFT2018 : (i) extraction de l'aspect, c'est-à-dire l'attribut de l'objet sur lequel une opinion est donnée ; (ii) identification de la polarité associée à l'aspect. À SemEval 2016 (Pontiki *et al.*, 2016), la tâche est devenue multilingue avec entre autres des données pour le français (Apidianaki *et al.*, 2016). Les deux meilleurs systèmes pour le français étaient basés sur des CRF (Conditional Random Field) (Brun *et al.*, 2016; Kumar *et al.*, 2016). Pour l'anglais, Toh & Su (2016) ont expérimenté l'utilisation de réseaux de neurones récurrents (bidirectionnel Elman-type RNN) associés à des CRF et obtiennent les meilleurs résultats de la tâche "Aspect-Based Sentiment Analysis" de SemEval 2016. Ils ont observé que l'utilisation d'un Elman-type RNN bidirectionnel associé à des CRF améliore les performances du système par rapport à des CRF seuls. L'association d'une couche de RNN bidirectionnel et de CRF a été testé sur plusieurs tâches de *sequence labelling* ces dernières années et a souvent permis de dépasser les résultats état-de-l'art. Les BiLSTM et BiGRU sont souvent employés (Huang *et al.*, 2015; Ma & Hovy, 2016; Dalloux *et al.*, 2017).

5. Le terme "objet" utilisé par les organisateurs est à prendre au sens large, en effet il inclut également des situations ou événements, des personnes, etc.

	NEGATIF		POSITIF	
émotion	déplaisir	631	plaisir	4 919
	dérangement	2 061	apaisement	541
	mépris	2 359	amour	560
	surprise négative	140	surprise positive	118
	peur	961		
	colère	2 090		
	ennui	89		
	tristesse	1 042		
sentiment	insatisfaction	1 484	satisfaction	1 991
opinion	désaccord	1 785	accord	580
	dévalorisation	2 826	valorisation	6 982
type générique	négatif	9 949	positif	2 838

TABLE 5: Distribution des types de marqueurs d'opinion, sentiment et émotion dans le corpus d'entraînement.

Nous avons traité cette tâche comme une tâche de *sequence labeling*. Nous avons expérimenté une méthode basée sur des réseaux de neurones récurrents et des CRF. Les relations entre la cible et l'OSEE ont été extraites avec une simple règle de proximité. Dans cette partie, nous décrivons dans un premier temps le corpus, puis la méthode utilisée, les expériences effectuées et enfin les résultats obtenus.

3.1 Corpus

Les données distribuées par les organisateurs contiennent 68 916 tweets (voir tableau 1). Pour 44 742 de ces tweets, une annotation pour la tâche 3 est disponible. Pour cette tâche sont considérés uniquement les tweets qui concernent les transports et qui ont une polarité positive, négative ou mixte, ce qui ne concernent en théorie que 22 857 tweets. Nous avons donc à disposition plus d'annotations que celles répondant aux critères de la tâche 3. Dans le tableau 6, nous présentons le nombre d'annotations de CIBLE (objet à propos duquel est exprimée une opinion) et de OSEE (expression d'opinion, sentiment et émotion) pour différents sous-corpus. La colonne "distribué" indique le nombre total d'annotations disponibles. La colonne "transport" contient les informations concernant les tweets TRANSPORT [6], quelque soit leur polarité. La colonne "POS/NEG" indique le nombre d'annotations disponibles dans l'ensemble des tweets répondant aux critères de la tâche.

Nous avons converti les données au format IOB2 [7]. Malheureusement certains offsets étaient erronés et nous n'avons donc pu utiliser que 89% du corpus avec une incertitude sur la qualité des données. Nous donnons ci-dessous un exemple de tweet annoté avec les offsets fournis et qui illustre le problème rencontré :

<valorisation>Fort de</valorisation> mon <positif>talent</positif> <source>, j</source>' a<insatisfaction>i ra</insatisfaction>t<cible>é le voyage en bu</cible>s.

6. Pour le sous-corpus "batch_b", nous avons effectué une classification automatique des tweets pour distinguer ceux qui concernent les transports des autres.

7. Dans le format IOB2, B- indique le début d'un chunk, I- indique qu'un token est à l'intérieur d'un chunk et O qu'un token ne fait pas partie d'un chunk.

	distribué			transport			POS/NEG		
	CIBLE	OSEE	rel	CIBLE	OSEE	rel	CIBLE	OSEE	rel
simple	34 772	43 946	43 641	30 400	38 622	38 218	26 198	36 563	32907
simple IOB2	30 488	62 214	-	30 488	62 214	-	23 063	51 006	-
batch_b	83 087	48 425	0	59 033	30 198	0	-	-	-
batch_b IOB2	69 969	72 129	-	47 722	45 556	-	-	-	-

TABLE 6: Distribution des annotations dans le corpus d'entraînement pour la tâche 3.

Les OSEE sont classifiées en 20 classes (voir tableau 5). Dans le cas où il serait difficile de classer une expression dans une des 18 premières classes, le marqueur est typé uniquement comme "négatif" ou "positif". Nous remarquons qu'un grand nombre de marqueurs est associé à une de ces deux classes (30%).

3.2 Méthodes

Pour résoudre cette tâche nous avons expérimenté une méthode basée sur des réseaux de neurones récurrents et des CRF (Lafferty *et al.*, 2001). La méthode est illustrée dans la figure 3. Chaque mot du tweet est représenté par une concaténation d'un vecteur issu d'un plongement et d'un one-hot vecteur. Ces vecteurs sont fournis en entrée à une couche de GRU bidirectionnelle (BiGRU) (Cho *et al.*, 2014). La couche de sortie est une couche de CRF qui prédit de façon séquentielle les étiquettes des mots du tweet. Des couches Dropout sont ajoutées pour éviter le sur-apprentissage. L'apprentissage est fait en 3 itérations et la taille du batch est celle par défaut, soit 32.

Dans la phase d'expérimentation nous avons testé à la fois l'utilisation de BiLSTM et de BiGRU. Les résultats obtenus avec une couche BiGRU étaient légèrement meilleurs qu'avec une couche BiLSTM.

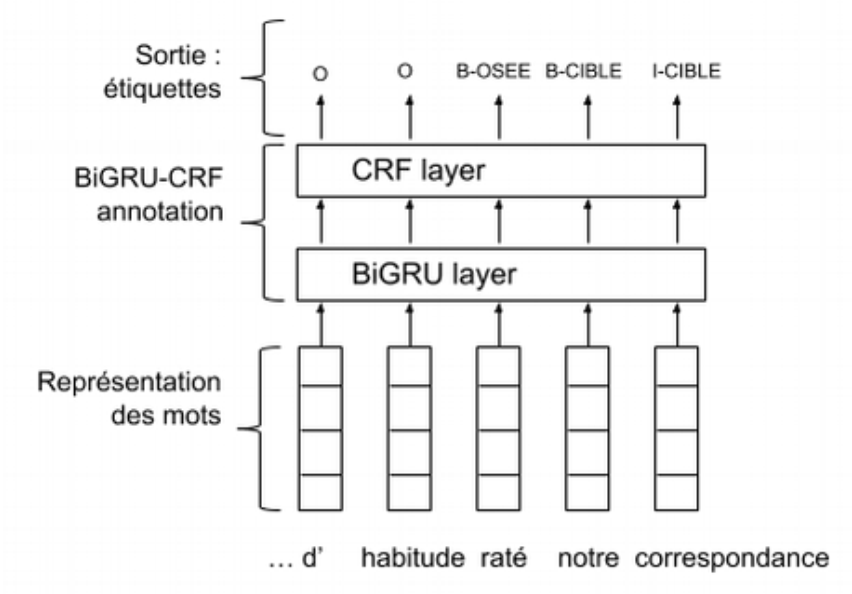

FIGURE 3: BiGRU-CRF pour l'annotation des expressions de sentiment et des cibles.

Nous avons effectué les prétraitements suivants sur les données :

— remplacement des nombres, des URLs, des hashtags et des alias respectivement par _num_, _http_, _hashtag_ et _alias_ ;
— tous les caractères sont convertis en minuscule ;

— tous les accents sont supprimés ainsi que la cédille.

Les plongements de mots utilisés ont été appris avec fasttext sur une corpus composé de wikipedia et common crawl Grave *et al.* (2018).

Les relations entre les cibles et les OSEE sont extraites en utilisant une simple règle de proximité, c'est-à-dire qu'une cible est reliée aux marqueurs d'opinion les plus proches en termes de nombre de mots. Nous avons défini cette règle après observation d'exemples du corpus mais n'avons pas réalisé d'expériences pour vérifier sa validité.

3.3 Expériences

Nous avons expérimenté trois approches qui se différencient par le nombre de labels appris par un modèle et le nombre de modèles :

— **2 branches** : un modèle entraîné pour l'annotation des cibles (B-CIBLE, I-CIBLE et O) et un pour l'annotation des OSEE (41 étiquettes : O, B-PEUR, I-PEUR, B-AMOUR, etc.) ;
— **2 étapes** : un modèle entraîné pour annoter les cibles et les OSEE sans y associer un type précis (B-CIBLE, I-CIBLE, B-OSEE, I-OSEE et O) et un pour typer les OSEE (41 étiquettes) ;
— **1 étape** : un modèle entraîné pour l'annotation des 43 étiquettes.

	CIBLE	OSEE
2 branches	31,19	58,12
2 étapes	29,50	57,71
1 étape	27,47	57,24

TABLE 7: Évaluation des trois approches proposées (en termes de micro F-mesure).

L'évaluation a été effectuée en utilisant le scorer de CoNLL 2003. Une séquence est considérée comme correcte si son empan correspond exactement à un empan dans l'annotation de référence (strict match) et que l'étiquette qui leur est associée est identique. L'évaluation est faite en validation croisée à 5 plis sur les données décrites dans la colonne "POS/NEG" du tableau 6. Les résultats sont donnés dans le tableau 7 en termes de micro F-mesure. Les meilleures performances sont obtenues avec la méthode "2 branches" avec une mico F-mesure de 31,19% pour l'annotation de la cible et de 58,12% pour l'annotation et la classification des OSEE.

	simple			simple POS/NEG			simple + batch_b		
	P	R	F	P	R	F	P	R	F
CIBLE	42,03	26,55	**32,51**	38,34	26,41	31,19	25,30	22,70	23,32
OSEE	64,01	54,70	**58,97**	61,31	55,27	58,12	51,94	53,32	52,60

TABLE 8: Évaluation de l'impact du corpus d'entraînement utilisé avec l'approche "2 branches" en termes de précision (P), rappel (R) et F-mesure (F).

Dans la section 3.1, nous avons décrit le corpus et indiqué que les données distribuées contenaient plus de tweets annotés que ceux répondant aux critères de la tâche. Pour évaluer l'impact de l'utilisation de ces données disponibles, nous avons entraîné notre meilleur système ("2 branches") avec différents sets de données d'entraînement. Dans le tableau 8, nous présentons les résultats obtenus en validation croisée à 5 plis. Nous observons que les meilleures performances sont atteintes en utilisant le sous-corpus "simple" en entier (c'est-à-dire à la fois les tweets positifs, négatifs mixtes et les tweets

		micro F-mesure	CIBLE	OSEE
2 branches	run1	40,00	26,25	49,64
2 étapes	run2	39,69	23,11	51,43
1 étape	run3	44,02	22,63	56,24

TABLE 9: Résultats obtenus par nos systèmes pour la tâche 3 de la campagne d'évaluation DeFT2018 (résultats non officiels).

neutres). Nous remarquons que lorsque nous utilisons le corpus "batch_b" en plus du corpus "simple" la précision chute de plus de 10 points, ce qui montre la basse qualité de ces annotations.

3.4 Résultats

Nous présentons dans cette partie les résultats obtenus pour la tâche 3 sur les données d'évaluation. Les résultats obtenus pour chaque run sont présentés dans le tableau 9. Ces scores ont été calculés par nos soins, en utilisant le scorer de CoNLL 2003. Aucune évaluation des relations entre cible et marqueur de sentiment n'a été faite. Comme pour les évaluations présentées dans la sous-section précédente, la comparaison entre la référence et la sortie du système est faite en "strict match". L'évaluation ne porte que sur les tweets annotés à la fois dans la référence et dans la sortie du système.

Les modèles ont été entraînés en utilisant toutes les données disponibles ("simple + batch_b"). Nous avons montré dans la sous-section précédente que l'utilisation des données du sous-corpus "batch_b" faisait diminuer les performances du système, mais cette observation a été faite après la phase d'évaluation.

Dans le tableau 9, nous observons que les meilleures performances pour la détection et la classification des marqueurs de sentiment sont obtenues avec l'approche "1 étape" avec une F-mesure de 56,24, contrairement aux résultats obtenus en validation croisée sur les données d'entraînement. Pour l'annotation des cibles à propos desquelles une opinion est exprimée, l'approche permettant d'obtenir les meilleurs résultats est l'approche "2 branches" avec une F-mesure de 26,25, performances qui restent très basses.

4 Conclusion

Nous avons présenté dans cet article la participation de l'équipe LinkMedia de l'IRISA à la campagne d'évaluation DeFT 2018. Nous avons développé des systèmes basés sur le boosting d'arbres de décisions et sur des réseaux de neurones récurrents pour les deux premières tâches. Nous avons observé que l'algorithme de boosting obtient des résultats comparables aux RNN pour la tâche 1 de classification des tweets en TRANSPORT/NON-TRANSPORT. En revanche pour la tâche 2, classification selon la polarité, les RNN obtiennent des meilleures performances.

Pour la tâche 3, que nous avons traité comme une tâche de *sequence labelling*, nous avons développé une méthode à base de RNN et CRF. Nous sommes les seuls participants de cette tâche, nous n'avons donc pas de points de comparaison avec d'autres approches. Nous espérons pouvoir très prochainement entraîner de nouveau nos modèles sur des données propres (c'est-à-dire pour lesquelles les offsets ont été corrigés).

Références

APIDIANAKI M., TANNIER X. & RICHART C. (2016). Datasets for Aspect-Based Sentiment Analysis in French. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* : European Language Resources Association (ELRA).

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

BRUN C., PEREZ J. & ROUX C. (2016). XRCE at SemEval-2016 Task 5 : Feedbacked Ensemble Modeling on Syntactico-Semantic Knowledge for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 277–281, San Diego, California : Association for Computational Linguistics.

CHO K., VAN MERRIENBOER B., BAHDANAU D. & BENGIO Y. (2014). On the Properties of Neural Machine Translation : Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 103–111 : Association for Computational Linguistics.

CLAVEAU V. & RAYMOND C. (2017). IRISA at DeFT2017 : classification systems of increasing complexity . In *DeFT 2017 - Défi Fouille de texte*, Actes de l'Atelier Défi Fouille de Texte, DeFT, p. 1–10, Orléans, France.

DALLOUX C., CLAVEAU V. & GRABAR N. (2017). Détection de la négation : corpus français et apprentissage supervisé. In *SIIM 2017 - Symposium sur l'Ingénierie de l'Information Médicale*, p. 1–8, Toulouse, France.

GRAVE E., BOJANOWSKI P., GUPTA P., JOULIN A. & MIKOLOV T. (2018). Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

GRAVES A., MOHAMED A.-R. & HINTON G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 6645–6649 : IEEE.

HUANG Z., XU W. & YU K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR*, **abs/1508.01991**.

KUMAR A., KOHAIL S., KUMAR A., EKBAL A. & BIEMANN C. (2016). IIT-TUDA at SemEval-2016 Task 5 : Beyond Sentiment Lexicon : Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 1129–1135, San Diego, California : Association for Computational Linguistics.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, p. 282–289 : Morgan Kaufmann, San Francisco, CA.

LAURENT A., CAMELIN N. & RAYMOND C. (2014). Boosting bonsai trees for efficient features combination : application to speaker role identification. In *InterSpeech*, Singapour.

MA X. & HOVY E. H. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 1064–1074, Berlin, Germany : Association for Computational Linguistics.

PONTIKI M., GALANIS D., PAPAGEORGIOU H., ANDROUTSOPOULOS I., MANANDHAR S., AL-SMADI M., AL-AYYOUB M., ZHAO Y., QIN B., DE CLERCQ O., HOSTE V., APIDIANAKI

M., TANNIER X., LOUKACHEVITCH N., KOTELNIKOV E., BEL N., JIMÉNEZ-ZAFRA S. M. & ERYIĞIT G. (2016). SemEval-2016 Task 5 : Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 19–30, San Diego, California : Association for Computational Linguistics.

PONTIKI M., GALANIS D., PAVLOPOULOS J., PAPAGEORGIOU H., ANDROUTSOPOULOS I. & MANANDHAR S. (2014). SemEval-2014 Task 4 : Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 27–35, Dublin, Ireland : Association for Computational Linguistics and Dublin City University.

TOH Z. & SU J. (2016). NLANGP at SemEval-2016 Task 5 : Improving Aspect Based Sentiment Analysis using Neural Network Features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, p. 282–288, San Diego, California : Association for Computational Linguistics.