"L'important c'est de participer" : positive #ironie. Analyse de sentiments et détection de l'ironie Les systèmes du LIUM et d'OCTO.

Amira Barhoumi[1,4] Vincent Levorato[2,3] Nicolas Dugué[1] Nathalie Camelin[1]
(1) LIUM, Université du Maine, 72000 Le Mans - prenom.nom@univ-lemans.fr
(2) OCTO Technology, 34 avenue de l'Opéra, 75002 Paris - vlevorato@octo.com
(3) LIFO, Université d'Orléans, 45100 Orléans - vincent.levorato@univ-orleans.fr
(4) MIRACL, Université de Sfax, Tunisie - amirabarhoumi29@gmail.com

RÉSUMÉ

Nous présentons le travail de l'équipe LIUM-OCTO pour DEFT 2017 sur les tâches de détection de polarité (tâche 1), et d'identification du langage figuratif (tâche 2). En tâche 1, nous produisons des systèmes à base de règles et plusieurs systèmes statistiques, que nous combinons pour obtenir notre meilleur système. Ces systèmes travaillent sur des descripteurs très variés, obtenus à partir d'une étude approfondie du corpus pour les règles, et d'un grand nombre d'indicateurs spécifiques à la description du texte pour les approches statistiques. En tâche 2, à partir d'un état de l'art détaillé, nous combinons et évaluons les différents descripteurs connus de la littérature. Nous utilisons également les indicateurs extraits en tâche 1 pour apprendre nos classifieurs statistiques.

ABSTRACT

LIUM-OCTO Results - DEFT 2017.

We detail the LIUM-OCTO results on DEFT 2017, considering polarity detection (task 1), and figurative language identification (task 2). About task 1, we designed rules-based systems and several statistical systems that we stacked. The rules-based system exploits features extracted after a deep insight in the corpus. The features used in the statistical classifiers allow a fine-grain description of tweets. About task 2, we combine and evaluate features described in the literature. We also used the features extracted for task 1 to train our classifiers.

MOTS-CLÉS : Détection de polarité, langage figuratif, classification, tweets, tal, sna.

KEYWORDS: Polarity detection, figurative language, classification, tweets, nlp, sna.

1 Introduction

Pour la première année, le LIUM a choisi de participer au défi DEFT2017 sur la classification de textes. Pour cela, il s'est associé à un consultant et chercheur d'OCTO Technology, une société spécialisée dans les technologies de l'information.

Nous avons choisi de partiper aux tâches 1 et 2 présentées dans les sous-sections suivantes. Nous détaillons ensuite dans la section 2, l'ensemble des descripteurs que nous avons choisi afin de représenter le tweet en entrée de nos systèmes, présentés eux en section 3. La section 4 présente nos résultats pour terminer par une discussion section 5.

1.1 Présentation de la tâche 1

L'analyse d'opinion est un domaine de recherche en plein essor. Il s'agit principalement d'identifier la polarité d'un texte donné : positif ou négatif. On peut l'appliquer au niveau du document, de la phrase ou d'un groupe de mots (Wilson *et al.*, 2004). La majorité des méthodes d'analyse peuvent être classées en trois approches : symbolique, statistique et hybride. L'approche symbolique utilise un lexique de sentiments ou des règles linguistiques pour prédire la polarité d'un énoncé donné (Turney, 2002). La deuxième est basée sur les techniques d'apprentissage automatique. Et finalement, l'approche hybride constitue une combinaison des deux approches précédentes. Les machines à vecteurs de supports (SVM) (Gaurangi *et al.*, 2014; Zainuddin & Selamat, 2014) et Naïves Bayes (Tripathy *et al.*, 2015) représentent les classifieurs les plus répandus dans ce domaine. En outre, des travaux récents font recours à l'apprentissage profond et utilisent des réseaux de neurones tels que les réseaux convolutionnels (Rouvier *et al.*, 2015; Deriu *et al.*, 2017).

L'originalité des approches de la littérature provient moins des algorithmes de classification que des descripteurs utilisés sur lesquels nous nous concentrons donc. Classiquement, les textes à classer sont représentés par un ensemble de descripteurs obtenus via des outils de traitement automatique des langues naturelles tels que les lemmes, les stemmes, les étiquettes morphosyntaxiques. Par ailleurs, un ensemble de descripteurs revient régulièrement dans la littérature comme les aspects lexicaux ou liés au style de rédaction du tweet (Cossu *et al.*, 2016) : la ponctuation, la taille du tweet, la présence d'URLs, hashtags et mentions, de majuscules, *etc.* La ponctuation et surtout sa répétition semble particulièrement importante (Barbieri & Saggion, 2014; Reyes & Rosso, 2014). Des descripteurs considérant la polarité du vocabulaire sont également communément utilisés : lexiques de polarité pour détecter la présence ou la fréquence de mots ou expressions polarisés, l'existence de termes de négation et d'intensificateurs (Chatzakou & Vakali, 2015), la présence d'émoticones (González-Ibánez *et al.*, 2011; Reyes & Rosso, 2014; Karoui *et al.*, 2015), *etc.*

La première tâche à laquelle nous avons participé s'inscrit dans ce contexte et consiste à classer un ensemble de tweets selon 4 niveaux de polarité : *objective*, *negative*, *positive* ou *mixed*. Les organisateurs ont mis à disposition un corpus d'apprentissage composé de 3906 tweets et un corpus de test composé de 976 tweets. Ils ont été annotés manuellement selon ces 4 polarités et la répartition des *tweets* dans ces classes au sein du corpus d'apprentissage est décrite dans la Table 1.

Classe	objective	negative	positive	mixed	**Total**
Nombre de tweets	1643	1268	494	501	3906

TABLE 1 – Tâche1 : Répartition des tweets dans le corpus d'apprentissage.

On remarque que plus d'un tiers de corpus est étiqueté avec la polarité objective, un autre tiers avec la polarité négative et le dernier tiers est composé de tweets à polarité positive ou mixed équitablement.

1.2 Présentation de la tâche 2

La tâche 2 consiste à séparer les *tweets* contenant du langage figuratif -ironie, sarcasme et humour- des autres. Le corpus fourni est composé de 7317 *tweets* annotés manuellement. Le corpus d'apprentissage constitué de 80% des *tweets* du corpus global contient 3906 *tweets* au label *non figuratif*, et 1947 *tweets* étiquetés *figuratif*.

La détection de l'ironie et du sarcasme dans des *tweets* a fait l'objet d'un intérêt récent, notamment avec des corpus en langue anglaise. Cette tâche a donc pour originalité de se concentrer sur des *tweets* en français, et de considérer le langage figuratif en général. Par ailleurs, dans la littérature, le corpus étudié est souvent étiqueté en utilisant les *hashtags* - *e.g. #sarcasm*, pour travailler sur un corpus de grande taille (González-Ibánez *et al.*, 2011; Riloff *et al.*, 2013). Les auteurs prennent néanmoins en compte ce biais pour leur apprentissage, dit *distant*, et testent leur système sur un sous-ensemble réétiqueté manuellement. Les accords entre étiquetages manuel et distant sont discutés dans Liebrecht *et al.* (2013).

Les descripteurs utilisés dans ces travaux [1] ainsi que les outils de classification sont majoritairement indépendants de la langue et de la taille du corpus. Outre les descripteurs décrits dans la section précédente également utilisables pour cette tâche, il existe des traits caractéristiques à cette tâche. Ainsi, par exemple, les descripteurs de la polarité sont parfois combinés à la présence d'intensifieurs, ou de négation qui sont des marqueurs d'ironie (Liebrecht *et al.*, 2013; Barbieri & Saggion, 2014; Karoui *et al.*, 2015). En effet, le sarcasme peut naître du contraste entre un sentiment positif et une situation négative (Riloff *et al.*, 2013). On retrouve également des descripteurs plus complexes. On note ainsi les *pattern* décrits par Tsur *et al.* (2010), morceaux de phrase qui servent à la structurer, et qui peuvent être caractéristiques de la présence de sarcasme. Barbieri & Saggion (2014) proposent d'étudier la fréquence des mots dans un corpus externe afin d'étudier les différents registres de langue ou les effets de surprise qui peuvent intervenir en utilisant des mots rares proches de mots fréquents dans une phrase. Ils considèrent également le nombre de synonymes des mots choisis : plus un mot a de synonymes, plus il est susceptible de créer une ambiguïté, marqueur possible d'ironie. Enfin, Karoui *et al.* (2015) se concentrent sur la contextualisation du *tweet* en allant chercher des informations externes pour confirmer ou infirmer sa nature ironique.

Nous tentons ainsi de reproduire les résultats de la littérature en combinant certains de ces descripteurs et en les associant à nos propres descripteurs, détaillés dans la section suivante.

2 Les différents descripteurs

2.1 Descripteurs communs

Les *tweets* ont leurs spécificités linguistiques et particularités stylistiques (Balahur, 2013) qui peuvent influencer la classification. Nous avons effectué sur les *tweets* les prétraitements suivants :

- remplacer les liens hypertextes par "URL" ;
- séparer le symbole # dans les hashtags et le remplacer par "_HASHTAG_" ;
- séparer le symbole @ dans les pseudonymes et le remplacer par "_PSEUDO_" ;
- séparer les signes de ponctuation du texte.

Nous avons ensuite extrait du *tweet* de nombreux descripteurs détaillés brièvement ci-après.

2.1.1 Statistiques basiques

- **Ponctuation** : présence de points d'exclamation, d'interrogation ou les deux, d'une citation [...], de trois points de suspension.

1. Plus de détails - https://git-lium.univ-lemans.fr/camelin/DEFT2017/blob/master/Revue/RevueSarcasme.md

— **Enumération de caractères** : nombres de caractères, de mentions, d'URLs, de lettres, de lettres minuscules et majuscules, de caractères numériques, de ponctuation, de points d'exclamation et d'interrogation.
— **Ratio de mots non français** basé sur le dictionnaire Français-Gutenberg (Pythoud, 1998).
— **Distribution de la taille des mots** : nous avons construit un descripteur par taille de mots de 1 à 10, puis toutes les dizaines (mots de taille entre 10 et 20, 20-30, 30-40, et > 40).
— **Entropie** : au sens entropie de Shannon, on calcule le ratio de compression du texte avec *zlib* en approximant la complexité de Kolmogorov (Grünwald & Vitányi, 2003).

2.1.2 Polarité

— **Enumération de termes polarisés :** nous avons utilisé les lexiques *Polarimots*[2] et *Emotaix* (Karoui *et al.*, 2015). A partir de ces lexiques, nous calculons les nombres de mots, expressions, hashtags positifs ou négatifs figurant dans le tweet.
— **Négation :** présence ou absence d'un terme de négation. Nous avons suivi Morlane-Hondère & D'hondt (2015) : si un terme de négation figure dans une fenêtre de deux mots avant ou après un mot polarisé, alors on inverse sa polarité. Cette opération intervient dans le calcul du nombre de mots positifs et négatifs.
— **Emoticones :** présence ou absence d'émoticônes dans le tweet. Nous avons construit un lexique d'émoticônes qui regroupe des smileys classiques tels que :), <3, ‘, *etc* et d'autres codés en hexadécimale tels que &#x1f636 ; &#x2729 ; *etc.*
— **VADER** : scores générés par cet outil d'analyse du sentiment basé sur un lexique et sur des règles, spécifiquement adapté aux sentiments exprimés dans les médias sociaux (Hutto & Gilbert, 2014).
— **AFINN** : un autre outil d'analyse de sentiment (Årup Nielsen, 2011).

2.1.3 Analyse textuelle

— **Tokenization** : nombre d'unigrammes en majuscules, nombre d'unigrammes total, nombre de stopwords, nombre d'unigrammes non stopwords.
— **PoS Tagging** : 30 étiquettes morpho-syntaxiques basées sur l'analyse du Stanford Log-linear Part-Of-Speech Tagger (Toutanova *et al.*, 2003).
— **Stemming** : nombre de racines uniques, nombre de racines uniques non stopwords.

2.2 Descripteurs spécifiques à la tâche 2

Tous les descripteurs communs aux deux premières tâches décrits précédemment ont été utilisés pour la seconde tâche. Nous utilisons également les traits décrits dans la Table 2.

2. http://polarimots.lif.univ-mrs.fr/

Polarité		
Emotaix	Descripteurs issus du lexique de polarité *Emotaix*	Karoui *et al.* (2015)
FEEL	Descripteurs issus du lexique de polarité *FEEL*	González-Ibánez *et al.* (2011)
Lexique manuel	Lexique manuel annotant les mots les plus fréquents : polarité et thématique	
Emoticones	Descripteurs issus d'une expression régulière de détection d'émoticones	González-Ibánez *et al.* (2011); Karoui *et al.* (2015)
Negation	Présence de négation dans le tweet	Riloff *et al.* (2013); Karoui *et al.* (2015)
URLs		
Urls	Ratio de la fréquence d'apparition d'un domaine dans les deux classes	
Style/Structure		
Ponctuation	Ponctuation et répétition de ponctuation	Karoui *et al.* (2015)
Majuscules	Ratio de majuscules dans le tweet	
Patterns	Structures de phrases fréquentes dans une seule classe	Tsur *et al.* (2010)
Suprise/Ambiguïté		
Wordnet	Ratio du nombre de synonymes dans *wordnet* et du nombre de mots	(Barbieri & Saggion, 2014)
Corpus externe	Fréquence moyenne des mots du tweet dans le corpus *FRWAC* (Baroni *et al.*, 2009), et du mot le plus rare	Barbieri & Saggion (2014)

TABLE 2 – Descripteurs spécifiques à la tâche 2 et leurs références.

3 Les différentes méthodes de classification

3.1 Classifieur symbolique : Méthode à base de règles.

Le système à base de règles permet de classer un tweet comme étant objectif ou non. Il se compose de trois règles (voir Table 3) élaborées après une phase d'analyse du corpus d'apprentissage faisant recours au calcul de la pureté $\bar{p}(regle)$ de chaque règle selon l'équation 1 :

$$\bar{p}(regle) = \frac{1}{N(regle)} \sum_{c \in \{classes\}} \frac{n_c^2(regle)}{N(regle)} \quad (1)$$

avec :

— $N(regle)$: le nombre total de documents vérifiant la règle ;
— c : une classe parmi les 4 polarités ;
— $n_c(regle)$: le nombre de documents de la classe c vérifiant la règle.

Lors de cette étape d'analyse du corpus, nous nous sommes focalisés sur un ensemble de descripteurs parmi lesquels nous citons : l'existence des urls, des pseudonymes, des hashtags et leurs positions

dans les tweets, l'existence des signes de ponctuation (! ? : " et «), et leurs combinaisons.

	Description	Pureté
règle1	le tweet commence par une url	1.0
règle2	le tweet contient : et contient une url	0.86
règle3	le tweet contient : et des guillemets (« ou ")	0.67

TABLE 3 – Tâche 1 : les règles et leurs mesures de pureté.

Sur le corpus d'apprentissage, ce système permet de classifier 24% du corpus avec une précision de 81.58%. Afin de classifier tous les *tweets*, ce système est associé en cascade avec un système basé sur l'algorithme de boosting que nous décrivons dans la section suivante.

3.2 Classifieur basé sur le boosting de petits arbres de décision.

Le *boosting* est un processus d'apprentissage itératif qui associe plus de poids aux exemples mal classés à chaque étape, ceci afin de renforcer leur prise en compte par le classifieur à l'itération suivante.

Nous avons utilisé l'implémentation *Bonzaiboost* [3] qui permet notamment de contraindre la profondeur de l'arbre de décision. Il a été montré par Laurent *et al.* (2014) que l'utilisation de petits arbres (profondeur 3) permet la combinaison avec succès à chaque itération de descripteurs variés.

3.3 Régression logistique

L'ensemble des variables étant quantitatives, nous n'avons pas effectué d'adaptation des données d'entrée. Classiquement, on cherche à estimer la probabilité p de la réalisation de la variable à expliquer Y, ici une appartenance à une classe selon l'équation 2

$$g(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{2}$$

Nous nous sommes placés dans un cas de régression logistique binaire où on traite chaque classe séparément (stratégie *One-vs-Rest*). La résolution est faite par le solveur *LIBLINEAR* (algorithme à directions de descente) (Fan *et al.*, 2008) qui minimise la fonction de coût pénalisée suivante (régularisation L2) :

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1) \tag{3}$$

3.4 Méthodes ensemblistes

L'objectif de ces méthodes est de combiner la prédiction de plusieurs estimateurs de base afin d'améliorer la robustesse de la prédiction qu'on aurait obtenue sur un seul estimateur. Deux familles de méthodes ensemblistes sont généralement distinguées :

3. http://bonzaiboost.gforge.inria.fr/

— *méthodes d'établissement de la moyenne* : il s'agit de construire plusieurs estimateurs indépendants, puis de faire la moyenne de leurs prédictions. L'estimateur combiné est généralement meilleur que n'importe quel estimateur de base unique car sa variance est réduite (*Random Forest* (Breiman, 2001), *Extremely Randomized Trees* (Geurts *et al.*, 2006)).
— *méthodes de renforcement* : les estimateurs de base sont construits séquentiellement et on essaie de réduire le biais de l'estimateur combiné. L'intérêt est de combiner plusieurs modèles faibles pour produire un modèle ensembliste puissant (*Gradient Tree Boosting* (Hastie *et al.*, 2001), *Extreme Gradient Boosting* (Chen & Guestrin, 2016)).

Dans notre cas, le classifieur qui donne les meilleurs résultats utilise le modèle *Extreme Gradient Boosting* (XGBoost [4]) qui approxime la construction des arbres pour gagner en efficacité et ainsi pouvoir passer à l'échelle plus facilement dans un environnement distribué.

3.5 Réseaux de neurones

Nous nous sommes intéressés à deux modèles en particulier :
— *MultiLayer Perceptron* : ce modèle supervisé apprend une fonction $f(X)$ avec X l'ensemble des variables en entrée pour en ressortir un nombre de dimensions en fonction du problème (4 pour la tâche 1 et 2 pour la tâche 2). Il y a donc une couche de $|X|$ neurones en entrée et une couche de m neurones qui aggrège les résultats où m est la dimensionalité choisie en sortie (*i.e.* nombre de classes). La non-linéarité du modèle est obtenue en insérant des couches cachées de neurones utilisant des fonctions d'activation non-linéaires sur la somme pondérée des sorties de la couche précédente (Collobert & Bengio, 2004). L'ensemble des couches est de type "FC" (Fully Connected).
— *Long Short-Term Memory* : ce type de réseau de neurones appartient à la famille des réseaux de neurones récurrents. Contrairement au perceptron, ceux-ci intègrent une mémoire interne dans une couche cachée. Un neurone est dit récurrent lorsqu'il utilise en entrée à l'instant t sa sortie à $t-1$. En plus de recevoir les sorties de la couche précédente, chaque couche cachée reçoit également sa propre sortie. L'aspect séquentiel des données peut ainsi être mieux capturé avec ce type de modèle. Pour les 2 tâches du challenge, nous avons encodé l'entrée de ce modèle sous forme de séquences sur la base de l'approche de Graves (2012). Nous avons tout d'abord appliqué une stemmatisation sur chaque tweet, chaque *stem* ayant un id unique dans l'ensemble du corpus, et après normalisation de la taille de la séquence à k entiers, nous obtenons pour chaque tweet une séquence correspondant à l'ordre des mots sous leur forme racine du type $(s_{id_1}, s_{id_2}, ..., s_{id_k})$. Nous avons donc une couche LSTM de 32 neurones suivi d'une couche FC de c neurones, c étant le nombre de classes de la tâche, avec une fonction d'activation finale de type *softmax*.

3.6 Méthode de combinaison : le *stacking*

La technique du *Stacking* s'apparente à une technique ensembliste (Wolpert, 1992). Un classifieur prend en entrée l'ensemble des prédictions d'autres classifieurs. Dans notre cas, chacun de nos classifieurs a été entraîné avec un processus de validation croisée (90 pour l'apprentissage, 10 pour

4. https://github.com/dmlc/xgboost

le test), puis nous avons appris un meta-classifieur pour prédire la cible finale. Pour la tâche 1 nous avons utilisé un simple arbre de décision, et un modèle Extreme Gradient Boosting pour la tâche 2.

4 Expériences et résultats

En s'appuyant sur l'ensemble des descripteurs et méthodes décrits dans les 2 sections précédentes, nous avons mis en oeuvre plusieurs systèmes afin de résoudre les tâches 1 et 2. Lors de la phase d'apprentissage, nous avons choisi d'utiliser la validation croisée en *10 folds*. Cette méthode présente les avantages suivants :

— le corpus, dans les 2 tâches, est de petite taille, notamment pour des méthodes statistiques comme les réseaux de neurones. Cette méthode présente l'avantage d'apprendre les modèles sur 90% du corpus tout en évaluant leur pouvoir de généralisation sur l'ensemble du corpus d'apprentissage.

— nous avons ainsi pu comparer facilement nos résultats et également les combiner, chacun des membres de l'équipe réalisant ses prédictions sur la totalité du corpus (pas de découpage apprentisage/test à définir).

Nous avons ensuite comparé, pour chaque tâche, les résultats obtenus par nos différents systèmes sur le corpus d'apprentissage et choisi de soumettre les 2 systèmes les plus performants et en troisième système, le *stacking* le plus performant combinant plusieurs de nos systèmes.

Notre travail a ainsi donné lieu à de nombreuses expérimentations. Nous reportons dans les sous-sections suivantes, les résultats qui nous semblent les plus pertinents.

4.1 Tâche 1

Les résultats sur le corpus d'apprentissage pour la tâche 1 sont reportés dans la Table 4.

Syst	Macro F-Score
LSTM-NN	0,48
MLP-NN	0,51
Boost	0,54
LogReg	0,54
Rules-Boost	0,59
Stack-Tree	0,61

TABLE 4 – Résultats obtenus sur le corpus d'apprentissage pour la tâche 1.

Pour cette tâche, les solutions proposées avec les réseaux de neurones (MLP-NN et LSTM-NN) parviennent péniblement à un score de 50%. Cela est certainement dû à la taille du corpus qui est trop faible pour apprendre correctement les 4 classes en présence. Les systèmes à base de régression logistique (LogReg) ou de Boosting (Boost) obtiennent de meilleures performances avec une macro-fmesure de 0,54. Le système le plus performant est le système hybride faisant d'abord la classification des tweets selon le système de règles puis classant les tweets restants avec le système à base de boosting d'arbres de profondeur 3. Dans la Table 5, nous présentons les résultats détaillés de ce système.

concept	précision	rappel	Macro F-Score	proportion
mixed	0,34	0,17	0,23	0,13
negative	0,62	0,71	0,66	0,32
objective	0,78	0,85	0,82	0,42
positive	0,65	0,57	0,61	0,13
all	0,60	0,57	0,59	1

TABLE 5 – Détails du système Rules-Boost sur la tâche 1.

Nous remarquons que les plus faibles résultats sont obtenus sur les classes les moins représentées dans le corpus. Notamment, la classe *mixed* est très compliquée à prédire avec une précision de seulement $0,34$ pour un rappel de $0,17$. Nous ne savons pas si cela est seulement dû à la sous-représentation de cette classe ou également au fait que la classe *mixed* est par définition difficile à définir puisqu'elle porte à la fois des marqueurs de classe positive et négative.

Pour la tâche 1, nous avons donc soumis trois systèmes : celui à base de règles et de boosting (Rules-Boost), celui à base de régression logistique (LogReg) et le *stacking* qui est une combinaision des deux précédents (Stack-Tree). La Table 6 donne une vue d'ensemble des résultats obtenus pour la tâche 1 sur le corpus de Test en précisant les f-mesures de nos systèmes.

Syst	Macro F-Score
Minimum	0,23
Moyenne	0,47
Mediane	0,52
Rules-Boost	0,53
LogReg	0,53
Stack-Tree	0,54
Maximum	0,64

TABLE 6 – Résultats obtenus sur le corpus de test pour la tâche 1.

Nous observons que les systèmes Rules-Boost et LogReg obiennent des performances similaires. Si le système LogReg obtient des résultats proches de ceux obtenus sur le corpus d'apprentissage, le système Rules-Boost obtient de moins bons résultats avec une perte de 0,06 points par rapport au corpus d'apprentissage. Néanmoins la combinaison des deux systèmes par *stacking* permet toujours de gagner en performance.

4.2 Tâche 2

Les résultats obtenus sur le corpus d'apprentissage par les systèmes mis en oeuvre sur la tâche 2 sont présentés dans la Table 7.

Sur cette tâche, les systèmes qui donnent les meilleurs résultats sont les réseaux de neurones basés sur un *MultiLayer Perceptron* (MLP-NN) et le boosting d'arbres de décision de profondeur 3 (Boost). Les méthodes ensemblistes, notamment le XGBoost, les réseaux de neurones de type LSTM et la régression logistique obtiennent de moins bons résultats.

Pour la tâche 2, nous avons soumis le système à base de boosting (Boost), le système basé sur un

Syst	Macro F-Score
XGBoost	0,63
LSTM-NN	0,63
LogReg	0,66
MLP-NN	0,70
Boost	0,72

TABLE 7 – Résultats obtenus sur le corpus d'apprentissage pour la tâche 2.

MultiLayer Perceptron (MLP-NN [5]) et un système de stacking de type XGB qui combine les résultats de plusieurs systèmes (RegLog, LSTM-NN, MLP-NN et Boost).

Syst	Macro F-Score
Minimum	0,48
Boost	0,66
Moyenne	0,69
Mediane	0,72
LSTM-NN	0,72
XGBoost	0,73
MLP-NN	0,77
Maximum	0,78

TABLE 8 – Résultats obtenus sur le corpus de test pour la tâche 2.

Parmi les systèmes soumis, seul le second système a fourni de bons résultats sur le test, les autres étant en dessous de la moyenne et de la médiane avec un F-score autour de $0,66$. Pourtant, le système de boosting soumis donnait de remarquables résultats sur le corpus d'apprentissage. De même, le système combinant les classifieurs appris permettaient d'augmenter les résultats obtenus en validation croisée sur le corpus d'apprentissage. En revanche, le second système à base de Gradient boosting (XGBoost) combinant les descripteurs détaillés en Section 2.1 et 2.2 obtient un score proche de celui obtenu par le meilleur système comme le montre la Table 8. Nous obtenons également des scores intéressants avec un réseau de neurones LSTM et un perceptron multi-couches (MLP).

5 Discussion

Les systèmes appris pour répondre aux deux premières tâches proposées dans DEFT 2017 ont obtenu des résultats au-dessus de la moyenne, ce qui souligne la qualité des descripteurs utilisés dans les deux tâches. Néanmoins, de nombreuses perspectives restent à explorer. Tout d'abord, les ressources linguistiques et outils pour le traitement automatique de la langue française sont plus rares que pour l'anglais. Il serait donc intéressant d'en faire un état de l'art détaillé de façon à dégager des axes à creuser pour améliorer la qualité des descripteurs utilisés pour l'apprentissage. Par ailleurs, nous

5. Après vérification, nous nous sommes rendu compte que les IDs des tweets avaient été utilisés comme descripteurs dans l'apprentissage du MLP-NN de la tâche 2. Sans les IDs, le système perd 6 points sur l'apprentissage et 4 points sur le test. Nous n'avons pas d'explication sur le fait qu'il semblerait que l'id donne une information importante sur la classe du tweet, peut-être un biais dans la constitution du corpus ?

n'avons pas exploré la possibilité d'utiliser des réseaux de neurone complexes, ni lors de la description du tweet via des *embeddings*, ni lors de la phase d'apprentissage, alors que ces méthodes semblent produire des modèles de qualité (Kim, 2014). D'autre part, il serait pertinent d'évaluer ces méthodes sur des corpus de plus grande taille, ou d'étudier si des corpus de plus grande taille permettent d'améliorer la qualité de ces systèmes. Cette question nous ramène à l'apprentissage *distant* (Go *et al.*, 2009).

Concernant la tâche 2 spécifiquement, nous avons remarqué que les descripteurs communs aux deux tâches sont déjà très expressifs, et qu'il est difficile d'augmenter les performances des systèmes avec des traits uniquement dépendants du contenu du tweet. Ainsi, nous avons fait appel à des ressources externes comme le corpus *FRWAC* ou les dictionnaires de synonymes. Cependant, nous pensons qu'il aurait été nécessaire d'aller plus loin et de poursuivre l'approche initiée par Karoui *et al.* (2015) qui consiste à utiliser des ressources externes pour affiner les décisions prises par le classifieur. Une partie des contenus ironiques n'est identifiable qu'à partir du contexte, et il est certain que ce genre d'apports est donc décisif pour permettre leur bonne classification.

Références

BALAHUR A. (2013). Sentiment analysis in social media texts. *4th workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, p. 120—-128.

BARBIERI F. & SAGGION H. (2014). Modelling irony in twitter. In *EACL*, p. 56–64.

BARONI M., BERNARDINI S., FERRARESI A. & ZANCHETTA E. (2009). The wacky wide web : a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, **43**(3), 209–226.

BREIMAN L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.

CHATZAKOU D. & VAKALI A. (2015). Harvesting opinions and emotions from social media textual resources. *IEEE Internet Computing*, p. 46–50.

CHEN T. & GUESTRIN C. (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 785–794, New York, NY, USA : ACM.

COLLOBERT R. & BENGIO S. (2004). Links between perceptrons, mlps and svms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, p. 23–, New York, NY, USA : ACM.

COSSU J.-V., LABATUT V. & DUGUÉ N. (2016). A review of features for the discrimination of twitter users : Application to the prediction of offline influence. *Social Network Analysis and Mining*, **6**(1), 1–23.

DERIU J., LUCCHI A., DE LUCA V., SEVERYN A., MÜLLER S., CIELIEBAK M., HOFMANN T. & JAGGI M. (2017). Leveraging large amounts of weakly supervised data for multi-language sentiment classification. *International World Wide Web Conference Committee (IW3C2)*.

FAN R.-E., CHANG K.-W., HSIEH C.-J., WANG X.-R. & LIN C.-J. (2008). Liblinear : A library for large linear classification. *J. Mach. Learn. Res.*, **9**, 1871–1874.

GAURANGI P., VARSHA G., VEDANT K. & KALPANA D. (2014). Sentiment analysis using support vector machine. *International Journal of Innovative Research in Computer and Communication Engineering*.

GEURTS P., ERNST D. & WEHENKEL L. (2006). Extremely randomized trees. *Mach. Learn.*, **63**(1), 3–42.

GO A., BHAYANI R. & HUANG L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, **1**(12).

GONZÁLEZ-IBÁNEZ R., MURESAN S. & WACHOLDER N. (2011). Identifying sarcasm in twitter : a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies : Short Papers-Volume 2*, p. 581–586 : Association for Computational Linguistics.

GRAVES A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer.

GRÜNWALD P. D. & VITÁNYI P. M. (2003). Kolmogorov complexity and information theory. with an interpretation in terms of questions and answers. *Journal of Logic, Language and Information*, **12**(4), 497–529.

HASTIE T., FRIEDMAN J. & TIBSHIRANI R. (2001). *Boosting and Additive Trees*. New York, NY : Springer New York.

HUTTO C. & GILBERT E. (2014). Vader : A parsimonious rule-based model for sentiment analysis of social media text.

KAROUI J., BENAMARA F., MORICEAU V., AUSSENAC-GILLES N. & BELGUITH L. H. (2015). Towards a contextual pragmatic model to detect irony in tweets. In *53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, p. PP–644.

KIM Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv :1408.5882*.

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

LIEBRECHT C., KUNNEMAN F. & VAN DEN BOSCH A. (2013). The perfect solution for detecting sarcasm in tweets# not.

MORLANE-HONDÈRE F. & D'HONDT E. (2015). Feature engineering for tweet polarity classification in the 2015 deft challenge. *Actes de l'atelier DEFT 2015*.

PYTHOUD C. (1998). Français-gutenberg : un nouveau dictionnaire français pour ispell. problèmes résolus et intégration de contributions extérieures. *Cahiers Gutenberg*, (28-29), 252–275.

REYES A. & ROSSO P. (2014). On the difficulty of automatically detecting irony : beyond a simple case of negation. *Knowledge and Information Systems*, **40**(3), 595–614.

RILOFF E., QADIR A., SURVE P., DE SILVA L., GILBERT N. & HUANG R. (2013). Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, volume 13, p. 704–714.

ROUVIER M., FAVRE B. & RAJENDRAN B. A. (2015). Talep @ deft'15 : Le plus cooool des systèmes d'analyse de sentiment. *DEFT-2015*, p. 97–103.

TOUTANOVA K., KLEIN D., MANNING C. D. & SINGER Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 NAACL - Volume 1*, NAACL '03, p. 173–180.

TRIPATHY A., AGRAWAL A. & RATH S. K. (2015). Classification of sentimental reviews using machine learning techniques. *3rd International Conference on Recent Trends in Computing (ICRTC-2015)*, p. 821—829.

TSUR O., DAVIDOV D. & RAPPOPORT A. (2010). Icwsm-a great catchy name : Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*, p. 162–169.

TURNEY P. D. (2002). Thumbs up or thumbs down ? semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 417–424.

WILSON T., WIEBE J. & RWA R. (2004). Just how mad are you ? finding strong and weak opinion clauses. *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, p. 761–769.

WOLPERT D. H. (1992). Stacked generalization. *Neural Networks*, **5**, 214–259.

ZAINUDDIN N. & SELAMAT A. (2014). Sentiment analysis using support vector machine. *International Conference on Computer, Communications, and Control Technology (I4CT)*.

ÅRUP NIELSEN F. (2011). A new anew : evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on Making Sense of Microposts : Big things come in small packages*, volume 718 of *CEUR Workshop Proceedings*, p. 93–98.