DEFT 2015

Table des matières

Analyse des émotions, sentiments et opinions exprimés dans les tweets : présentation et résultats de l'édition 2015 du défi fouille de texte (DEFT)

Thierry Hamon[1,2] Amel Fraisse[1] Patrick Paroubek[1] Pierre Zweigenbaum[1] Cyril Grouin[1]
(1) LIMSI-CNRS, Campus universitaire d'Orsay, Rue John von Neumann, Bât 508, 91405 Orsay
(2) Université Paris 13, Villetaneuse, France
prenom.nom@limsi.fr

Résumé. L'édition 2015 du défi fouille de texte (DEFT) porte sur la fouille d'opinion et l'analyse des sentiments et des émotions dans les messages postés sur Twitter en relation avec la thématique du changement climatique. Trois tâches ont été proposées : (i) déterminer la polarité globale des tweets, (ii) identifier les classes génériques (opinion, sentiment, émotion, information) et spécifiques (parmi 18 classes) de ces tweets, et (iii) analyser la source, la cible et l'expression porteuse d'opinion, de sentiment ou d'émotion. Douze équipes ont participé. Les meilleurs résultats, en macro-précision, sont de 0,736 (polarité), 0,613 (classes génériques) et 0,347 (classes spécifiques). Aucun participant n'a soumis de données pour la dernière tâche. Les méthodes utilisées reposent majoritairement sur des approches par apprentissage statistique supervisé (SVM, Naïve Bayes, réseaux neuronaux, PPMC), et utilisent de nombreux lexiques d'opinions (ANEW, Casoar, Emotaix, Feel, Lidilem) et de polarités (Polarimots) comme traits.

Abstract.

Analysis of emotion, sentiment and opinion within tweets. Presentation and results of the 2015 DEFT text mining challenge

The 2015 DEFT text mining challenge focused on opinion mining, emotion and sentiment analysis of messages from Twitter, on the climate change thematic. Three tasks were proposed : (i) determine the general polarity of tweets, (ii) identify generic classes (opinion, sentiment, emotion, information) and specific classes (among 18 classes) from these tweets, and (iii) analyze source, target, and opinion, sentiment, emotion focus. Twelve teams participated. The best results, in terms of macro-precision, are of 0.736 (polarity), 0.613 (generic classes) and 0.347 (specific classes). No run was submitted for the last task. The methods used by the participants mainly rely on statistical machine learning approaches (SVM, Naïve Bayes, neural network, PPMC), using several opinion lexicon (ANEW, Casoar, Emotaix, Feel, Lidilem) and polarity lexicon (Polarimots) as features.

Mots-clés : Fouille d'opinion, analyse d'émotions, analyse de sentiments, réseaux sociaux, campagne d'évaluation.

Keywords: Opinion mining, Emotion analysis, Sentiment Analysis, Social network, NLP Challenge.

1 Introduction

Le défi DEFT est un atelier annuel d'évaluation francophone en fouille de textes. Les thématiques abordées relèvent du domaine expérimental et visent à vérifier la faisabilité des tâches proposées au moyen des méthodes disponibles.

La fouille d'opinion constitue une activité qui a déjà été proposée lors de deux précédentes éditions de DEFT. En 2007, nous proposions de travailler sur la retranscription de débats parlementaires pour déterminer l'opinion véhiculée par le message (Grouin *et al.*, 2007, 2009b). Il s'agissait donc pour les participants de travailler sur des textes correctement écrits, composés de phrases assez longues. L'édition 2009 a permis de comparer les opinions exprimées dans deux corpus différents, un corpus de débats parlementaires d'une part, et un corpus d'articles de journaux (éditoriaux, articles d'analyse et de débat, articles de fond) d'autre part (Grouin *et al.*, 2009a). En dehors de l'identification du caractère objectif ou subjectif d'un texte, nous avons proposé aux participants de cette édition d'identifier les expressions porteuses d'opinion. Pour cette tâche, faute de disposer de moyens pour constituer la référence, nous avions évalué les résultats des participants sur la base des annotations communes aux différentes soumissions.

Pour cette onzième édition, nous proposons de travailler sur l'analyse de l'opinion, des sentiments et des émotions dans des tweets rédigés en français. Contrairement aux précédentes éditions, nous proposons cette année un ensemble de tâches permettant de traiter la fouille d'opinion de manière complète, tant du point de vue de la polarité et de l'opinion globale d'un message que du point de vue de l'identification de la source, de la cible et de l'expression porteuse d'opinion à l'intérieur de chaque message. Le corpus proposé a fait l'objet d'une annotation complète par des annotateurs humains, en fonction de principes définis dans un guide d'annotation (Fraisse & Paroubek, 2014).

2 Corpus

2.1 Présentation

Le corpus se compose de 15 000 messages postés sur le réseau social Twitter, en relation avec la thématique du changement climatique. Ces messages, appelés « *tweets* » et d'une longueur maximale de 140 caractères, sont caractérisés par : (i) des phrases courtes, (ii) l'utilisation d'abréviations génériques ou propres à l'Internet (*MDR, STP, klk1, kan, dla*), (iii) un style littéraire plus ou moins familier selon l'émetteur du message (« *ils vous bananes au calme* »), (iv) la présence d'émoticônes (ou *smiley* : `xD`), et (v) la présence de mots-clés spécifiques au réseau Twitter : des *hashtags* ou *mot-dièses* commençant par le symbole dièse pour marquer une thématique (`#Irak`, `#NoControlDay`) ou un état d'esprit présenté de manière sarcastique (`#FeedTheTroll`), et des noms d'utilisateurs commençant par le symbole arobase (`@CNRS`).

2.2 Annotation

2.2.1 Procédure

Les tweets ont été annotés par deux annotateurs. Une première étape de double annotation portant sur 500 messages a été réalisées. Elle a permis aux annotateurs de se familiariser avec le guide d'annotation. La suite des messages a été annotée en simple annotation. Nous avons calculé les accords inter-annotateurs sur les 500 messages annotés en double.

Le tableau 1 renseigne des probabilités observées (Pa), probabilités attendues (Pe) et des valeurs de Kappa et de coefficient de Dice calculées en première approximation pour l'accord sur le cardinal (nombre d'instances trouvées par un annotateur independamment des positions respectives de ces instances dans les documents) pour chacune des catégories possibles d'annotation, sur le corpus de 500 tweets annotés en double. Le choix de calculer l'accord sur le cardinal des catégories d'annotation au lieu de le faire sur les annotations est le résultat de contraintes temporelles qui n'ont pas permis de développer des mesures de Kappa adaptées aux annotations fines de la tâche (iii) pour prendre en compte de manière raisonnable les différences de frontières d'empan de texte qui aboutissent à des valeurs artificiellement faibles de kappa si l'on considère une égalité stricte au niveau du caractère.

Si globalement, les accords inter-annotateurs témoignent d'une absence d'accord entre annotateurs avec une valeur de $\kappa < 0,8$ (Artstein & Poesio, 2008), précisons que cette valeur tient compte de l'ensemble des catégories et relations utilisées dans le schéma d'annotation. Nous observons qu'il n'est pas possible de dégager un consensus entre les deux annotateurs sur certaines catégories, en particulier la polarité *Positif* ($\kappa = 0,18$), l'identification de la cible ($\kappa = 0,39$) et plusieurs classes spécifiques (du plus consensuel au moins consensuel, pour des valeurs de $\kappa < 0,8$: *Accord, Valorisation, Mépris, Tristesse, Apaisement, Satisfaction, Dévalorisation, Colère*).

2.2.2 Guide d'annotation

Chaque message est annoté au moyen de sept groupes, dont un comprend 19 catégories fines, et de cinq relations [1].

Groupes Les sept groupes utilisés sont les suivants :

– SOURCE : groupe de mots qui référence l'auteur de l'expression d'opinion, sentiment, émotion (OSEE : *Opinion Sentiment Emotion Expression*). On annote la mention explicite de la source. Celle-ci doit être la plus large possible incluant ses modifieurs, ses circonstants, ses multiples apposés, ses relatives, mais également les conjonctions

1. Le guide utilisé est accessible à l'adresse suivante : `https://deft.limsi.fr/2015/guideAnnotation.fr.php?lang=fr` Nous invitons le lecteur à consulter le guide d'annotation pour accéder aux exemples associés à chacun des groupes, catégories et relations.

	OSEE_GLOBALE	RELATIONS					SOURCE, CIBLE, EXPRESSION					POLARITE		
		DIT	SUR	MOD	NEG	RECEPTEUR	Source	Cible	Modifieur	Négation	Destinataire	Positif	Négatif	Inconnu
Pa(==)	0,76	1,00	1,00	1,00	1,00	1,00	0,95	0,83	0,99	0,99	1,00	0,94	0,93	1,00
Pa	0,76	1,00	1,00	1,00	1,00	1,00	0,95	0,83	0,99	0,99	1,00	0,94	0,93	1,00
Pe	0,24	0,75	0,35	0,94	0,94	0,98	0,75	0,39	0,94	0,94	0,98	0,92	0,82	1,00
Kappa	0,69	1,00	1,00	1,00	1,00	1,00	0,79	0,72	0,82	0,80	1,00	0,18	0,62	1,00
Dice	0,76	1,00	1,00	1,00	1,00	1,00	0,95	0,83	0,99	0,99	1,00	0,94	0,93	1,00

	CLASSES SPECIFIQUES																		
	Amour	Plaisir	Apaisement	Surprise_positive	Satisfaction	Accord	Valorisation	Désaccord	Dévalorisation	Insatisfaction	Mépris	Colère	Tristesse	Déplaisir	Ennui	Peur	Dérangement	Surprise_négative	Instruction_Demande
Pa(==)	1,00	0,99	1,00	1,00	0,99	0,97	0,89	0,99	0,98	1,00	0,98	0,99	1,00	1,00	1,00	0,99	1,00	1,00	0,97
Pa	1,00	0,99	1,00	1,00	0,99	0,97	0,89	0,99	0,98	1,00	0,98	0,99	1,00	1,00	1,00	0,99	1,00	1,00	0,97
Pe	1,00	0,97	0,99	1,00	0,98	0,90	0,66	0,92	0,97	1,00	0,94	0,98	0,99	0,99	1,00	0,91	1,00	1,00	0,74
Kappa	0,00	0,71	0,66	1,00	0,50	0,73	0,69	0,83	0,37	1,00	0,69	0,36	0,67	0,80	1,00	0,86	0,00	1,00	0,87
Dice	1,00	0,99	1,00	1,00	0,99	0,97	0,89	0,99	0,98	1,00	0,98	0,99	1,00	1,00	1,00	0,99	1,00	1,00	0,97

TABLE 1 – Taux d'accord inter-annotateur (Kappa, Dice) calculés sur les 500 tweets annotés en double pour l'accord sur les cardinaux de catégories d'annotation

de modifieurs de toutes sortes, y compris de relatives, de manière à avoir le maximum d'information sémantique (*En tant que cuisinier amateur qui a de l' expérience , je n' aime pas vraiment les pâtes .*) ;

- CIBLE : mention explicite de la cible la plus large possible incluant ses modifieurs, ses circonstants, ses multiples apposés, ses relatives, voire aussi les conjonctions de modifieurs de toutes sortes y compris de relatives afin d'avoir le maximum d'information sémantique. Lorsque l'OSEE porte sur plusieurs cibles, chaque cible est identifiée dans un empan de texte distinct (*Les lynx , les loups , les tortues sont des éspéces protégées .*) ;
- NÉGATION : marqueurs de négation (*ne pas, ne plus,* etc.) (*Le serpent n' est pas une espèce protégée .*) ;
- MODIFIEUR : tout modifieur (*le plus*, etc.) (*La Sardine l' un des poissons les plus en danger en Méditerranée http :*) ;
- DESTINATAIRE : mention explicite du destinataire. Ce groupe sera essentiellement utilisé dans le cas ou l'expression d'opinion, sentiment, émotion est adressée à une entité *(personne, organisation, etc.)* (*Mme Ségolène Royale vous êtes priée de respecter les loups .*) ;
- EXPRESSION D'OPINION, SENTIMENT, ÉMOTION (OSEE) : empan de texte dont la valeur sémantique correspond à l'expression d'opinion, de sentiment ou d'émotion. Cette expression est annotée au moyen de l'une des 19 catégorie sémantiques fines [2] :
  - Classes sémantiques affectives fines : ces classes relèvent de trois grandes catégories :
    - Opinions (intellectif) : cette catégorie contient deux classes sémantiques positives (*Accord* et *Valorisation*) de deux classes sémantiques négatives (*Désaccord* et *Dévalorisation*) ;
    - Sentiments (affectif-intellectif) : cette catégorie contient une classe sémantique positive (*Satisfaction*) et une classe sémantique négative (*Insatisfaction*) ;
    - Émotions (affectif) : cette catégorie contient 4 classes sémantiques positives (*Plaisir, Apaisement, Amour* et *Surprise positive*) et 8 classes sémantiques négatives (*Déplaisir, Dérangement, Mépris, Tristesse, Peur, Colère, Ennui* et *Surprise négative*).

    Une dernière classe sémantique correspond aux instructions et aux informations (*instruction*).
  - Classes génériques de polarité :

2. Pour des raisons de lisibilité, les définitions et exemples de chacune de ces 19 catégories sont donnés dans l'annexe A.

  - *Négatif* (expressions d'opinion/sentiment/émotion qui ont une polarité négative et dont il est difficile d'identifier avec certitude la classe sémantique exacte),
  - *Positif* (expressions d'opinion/sentiment/émotion qui ont une polarité positive et dont il est difficile d'identifier avec certitude la classe sémantique exacte).
- OSE_GLOBALE : catégorie sémantique globale et générale du message. Dans le cas où le message contient plus d'une catégorie sémantique, il faut indiquer la catégorie sémantique dominante du message. Dans le cas contraire, l'OSE_GLOBALE aura la même valeur que le groupe d'expression d'opinion, sentiment, émotion du message.

Relations Les cinq relations considérées sont les suivantes :
- DIT : permet de mettre en rapport la SOURCE avec l'OSEE. Elle met toujours en rapport 2 groupes (*Je n' aime pas vraiment les pâtes* .) ;
- SUR : permet de mettre en rapport l'expression d'opinion, sentiment, émotion avec la CIBLE. Elle met toujours en rapport 2 groupes (*Je n' aime pas vraiment les pâtes* .) ;
- MOD permet de mettre en rapport les éventuels modifieurs de l'OSEE. Cette relation va mettre en rapport une forme (e.g., modifieur) avec le groupe d'expression d'opinion, sentiment, émotion ;
- NEG : permet de mettre en rapport les éventuels marqueurs de négation avec l'OSEE dont ils modifient la sémantique. Chaque marqueur suscite la création d'une relation NEG (notamment les deux éléments de la négation *ne* et *pas*) ;
- RECEPTEUR : permet de mettre en rapport le groupe de l'OSEE avec le groupe DESTINATAIRE.

2.2.3 Annotations de référence

En fonction de ces annotations fines utilisées comme annotations de référence pour la tâche 3 (voir section 3.1), nous avons dégagé automatiquement les valeurs de référence des tâches plus génériques (tâches 1, 2.1 et 2.2) au moyen d'une simple correspondance entre catégories fines et valeurs génériques (voir tableau 2). Les annotations de référence des premières tâches n'ont donc pas été directement établies par les annotateurs humains, mais inférées des annotations humaines de la troisième tâche.

Catégorie fine (T3)	Type (T2)	Polarité (T1)
Accord, Valorisation	opinion	+
Désaccord, Dévalorisation		-
Satisfaction	sentiment	+
Insatisfaction		-
Plaisir, Apaisement, Amour, Surprise positive	émotion	+
Déplaisir, Dérangement, Mépris, Surprise négative, Peur, Colère, Ennui, Tristesse		-
Instruction, Information	information	=

TABLE 2 – Correspondance entre catégories fines, type et polarité

Le tableau 3 renseigne de la distribution des annotations pour chacune des catégories proposées dans chaque tâche, entre corpus d'apprentissage et de test. Notons que l'équilibre entre les deux corpus est respecté pour chaque catégorie, à l'exception de la catégorie *Sentiment* de la tâche 2.1, sous-représentée dans le corpus d'apprentissage, et qui induit un déséquilibre sur les autres catégories. Ce déséquilibre provient de la difficulté de maîtriser la distribution entre catégories, à la fois entre corpus et entre tâches, lorsque le même corpus est utilisé pour plusieurs tâches.

3 Présentation du défi

3.1 Tâches proposées

Nous avons proposé trois tâches complètes autour des émotions, sentiments et opinions exprimées dans les messages postés sur Twitter. Les différentes tâches proposent un niveau d'analyse du plus global *(polarité, classes génériques)* au plus fin *(identification des classes spécifiques ; reconnaissance des expression porteuse d'opinion, cible et source dans le texte du tweet)*. L'ensemble de ces tâches permet de couvrir une large part des travaux possibles en matière d'analyse des

Tâche	Catégorie	Apprentissage	Test
T1	+	2464 (31,08%)	1057 (31,28%)
	-	1894 (23,89%)	804 (23,79%)
	=	3571 (45,04%)	1518 (44,92%)
T2.1	Emotion	826 (12,23%)	351 (10,39%)
	Information	3571 (52,87%)	1518 (44,92%)
	Opinion	2275 (33,68%)	973 (28,80%)
	Sentiment	82 (1,21%)	537 (15,89%)
T2.2	Accord	154 (4,84%)	67 (4,92%)
	Amour	8 (0,25%)	4 (0,29%)
	Apaisement	9 (0,28%)	5 (0,37%)
	Colère	210 (6,60%)	87 (6,39%)
	Déplaisir	47 (1,48%)	21 (1,54%)
T2.2 (suite)	Dérangement	13 (0,41%)	6 (0,44%)
	Désaccord	216 (6,79%)	92 (6,76%)
	Dévalorisation	401 (12,60%)	170 (12,49%)
	Ennui	4 (0,13%)	2 (0,15%)
	Insatisfaction	9 (0,28%)	5 (0,37%)
	Mépris	176 (5,53%)	75 (5,51%)
	Peur	274 (8,61%)	114 (8,38%)
	Plaisir	35 (1,10%)	15 (1,10%)
	Satisfaction	73 (2,29%)	32 (2,35%)
	Surprise_négative	10 (0,31%)	4 (0,29%)
	Surprise_positive	4 (0,13%)	2 (0,15%)
	Tristesse	36 (1,13%)	16 (1,18%)
	Valorisation	1504 (47,25%)	644 (47,32%)

TABLE 3 – Distribution des annotations par catégories et par tâches pour chaque corpus

émotions, sentiments et opinions appliquée aux messages courts postés sur les réseaux sociaux. Chaque participant était libre de choisir les tâches auxquelles il souhaitait participer..

Pour illustrer chacune de ces tâches, nous prenons appui sur le tweet suivant pour illustrer les annotations réalisées.

> Energie renouvelable pleine de promesses, la géothermie souffre pourtant d'un manque de visibilité...

FIGURE 1 – Extrait du corpus (identifiant : 519507340304084992)

3.1.1 Tâche 1 – Polarité des tweets

La première tâche vise à détecter la polarité des tweets parmi trois valeurs possible : *positif* (+), *neutre ou mixte* (=), et *négatif* (−). La catégorie *neutre ou mixte* renvoie aussi bien aux messages présentant une polarité neutre (ni positif, ni négatif), que ceux présentant les deux polarités en même temps (un sentiment positif et un sentiment négatif). Le tweet présenté en exemple est classé *négatif* pour la tâche 1.

3.1.2 Tâche 2 – Classe des tweets

Cette tâche vise une classification fine des tweets. Nous avons divisé cette tâche en deux sous-tâches.

Tâche 2.1 – Classe générique Cette première sous-tâche vise l'identification de la classe générique de l'information exprimée dans le tweet, parmi quatre classes : *opinion, sentiment, émotion, information*. Le tweet présenté en exemple est classé *émotion* pour la tâche 2.1.

Tâche 2.2 – Classe spécifique Cette deuxième sous-tâche vise l'identification de la classe spécifique de l'opinion, du sentiment, ou de l'émotion exprimée, parmi dix-huit classes : *accord, amour, apaisement, colère, déplaisir, dérangement, désaccord, dévalorisation, ennui, insatisfaction, mépris, peur, plaisir, satisfaction, surprise négative, surprise positive, tristesse, valorisation*. Le tweet présenté en exemple est classé *déplaisir* pour la tâche 2.2.

3.1.3 Tâche 3 – Source, cible et expression d'opinion

Cette dernière tâche vise à analyser plus précisément les opinions, du point de vue de l'expression porteuse de l'opinion, de la source (l'émetteur) et de la cible (le récepteur).

Sur le tweet d'exemple, les portions suivantes sont annotées comme suit :

– Entités : « *Energie renouvelable* » est marquée *cible*, « *pleine de promesses* » est marquée *valorisation*, « *souffre* » est marqué *déplaisir*, et « *manque de visibilité* » est marqué *négatif* ;
– Relations : des relations DIT entre « *pleine de promesses* » et « *Energie renouvelable* », et entre « *manque de visibilité* » et « *souffre* ».

3.2 Organisation

A l'image des campagnes d'évaluation précédentes, cette édition s'est déroulée en deux temps. La première phase permet aux participants de développer et d'entraîner leurs systèmes à partir des données annotées qui leur sont fournies. Les inscriptions (remplissage d'un simple formulaire sur internet) et l'accès aux données annotées d'entraînement ont été autorisés à partir du 16 février 2015. Nous relevons que les inscriptions se sont réparties entre le 16 février et le 1^{er} mai, avec une majorité d'inscriptions au mois de mars. Au total, 23 équipes se sont inscrites, dont deux issues d'industriels (Proxem et Synapse Développement) et une équipe académique de l'Université Technique de Moldavie (TU Moldova).

La deuxième phase permet aux participants d'appliquer les méthodes qu'ils ont développées pendant la phase d'entraînement sur le corpus de test. Cette phase de test s'est déroulée du 4 au 10 mai 2015. Chaque participant a bénéficié d'une fenêtre de trois jours (définie par chaque équipe selon ses préférences) entre l'accès aux données de test et la soumission des résultats produits par son système. Nous avons reçu les soumissions de 12 équipes, chaque équipe pouvant soumettre jusqu'à trois sorties différentes de leur système, pour chacune des tâches proposées.

Conformément aux règles d'accès à Twitter et d'utilisation des tweets, lors des phases de développement et de test, nous avons fourni aux participants les identifiants des tweets et les outils permettant de constituer le corpus par eux-mêmes.

Sur la tâche 1, nous avons reçu un total de 27 soumissions correspondant à l'ensemble des 12 équipes ayant participé au défi. Sur la tâche 2.1, nous avons reçu 24 soumissions pour 9 équipes, et 21 soumissions pour 7 équipes sur la tâche 2.2. Aucune équipe n'a participé à la tâche 3.

Les évaluations des soumissions effectuées ainsi que les annotations de référence sur le corpus de test ont été communiqués aux participants entre le 14 et 18 mai 2015. Pour chacune des tâches proposées, chaque participant a eu accès à ses résultats individuels (pour l'ensemble des soumissions effectuées) ainsi qu'à des éléments de comparaison calculés sur la meilleure soumission de chaque équipe (moyenne, médiane, écart-type, valeurs minimum et maximum), le classement final n'étant dévoilé que le jour de l'atelier de clôture du défi (voir section 5). Pendant cette période, les participants ont été invités à vérifier les résultats calculés et à se prononcer sur le cas d'équipes ayant soumis des résultats avec des noms de catégories erronées sur la tâche 2.2 (voir section 3.1, utilisation de la catégorie « *instruction* » au lieu de « *information* », ou de versions abrégées « i », « o », « e » et « s » au lieu de « *information* », « *opinion* », « *émotion* » et « *sentiment* ») dont la correction modifie les résultats et le classement final.

3.3 Évaluation

Les résultats des tâches 1, 2.1 et 2.2 ont été évalués en termes de macro-précision (formule 1) (Manning & Schütze, 2000).

$$\text{Macro-précision} = \frac{\sum_{i=1}^{n} \left(\frac{\text{vrais positifs}(i)}{\text{vrais positifs}(i)+\text{faux positifs}(i)} \right)}{n} \quad (1)$$

Tous les tweets devant se voir attribué une catégorie, nous avons choisi de pénaliser fortement les systèmes ne prenant pas de décision ou proposant une catégorie non attendue initialement.

Le fait que les participants devaient constituer les corpus par eux-mêmes a conduit à une difficulté supplémentaire lors de l'évaluation. En effet, les tweets pouvant être supprimés par leur auteurs à n'importe quel instant, il était possible qu'un participant puisse récupérer un tweet qui ne serait plus disponible plus tard, pour d'autres participants. Il fallait nous assurer que tous les participants soient évalués sur le même ensemble de tweets. A la fin de la phase de test, nous avons donc identifiés les tweets supprimés. Ainsi, deux tweets avaient été supprimés pendant la période de test. Nous avons également constaté que plusieurs participants ont eu des difficultés, probablement techniques, à accéder à deux autres tweets. Nous avons choisi de ne pas prendre en compte ces quatre tweets dans l'évaluation finale.

4 Méthodes des participants

Chaîne de traitements TextAnalyst La société Synapse (équipe 19) a traité les corpus au moyen de la chaîne de traitements TextAnalyst, composée de lexiques et de plusieurs modules qui s'enchaînent en cascade (analyse syntaxique avec Cordial, détection des expressions d'opinion, détection et application d'opérateurs (négation, intensification, modalités), et calcul de l'opinion globale du tweet fondé sur le modèle parabolique) (Chardon *et al.*, 2015).

Apprentissage statistique La majorité des systèmes utilisés par les participants repose cependant sur des approches par apprentissage statistique supervisé. Les principaux algorithmes utilisés sont : SVM (LINA/Dictanova, équipe 6 ; LIRMM, équipe 17 ; LINA–Dimeco, équipe 25), Naïve Bayes (IRISA, équipe 14 ; LIMSI, équipe 23), réseau de neurones (IRISA, équipe 14 ; Proxem, équipe 15), ou encore un algorithme de prédiction par correspondance partielle : PPMC (TU Moldova, équipe 22). Le LIF (équipe 3) a utilisé plusieurs modèles probabilistes dont les résultats ont été fusionnés au moyen d'une procédure de vote.

Parmi les traits utilisés, l'équipe LINA–Dimeco (Lejeune & Dumonceaux, 2015) a émis l'hypothèse que le style utilisé dans un tweet reflète l'émotion de l'émetteur, et a donc mobilisé des critères stylométriques sur les caractères et mots du message pour en déterminer les émotions, opinions et sentiments. Le LIRMM (Abdaoui *et al.*, 2015) a par ailleurs pris en compte les patrons syntaxiques. Plusieurs équipes n'ont pris en compte que les descripteurs les plus discriminants, notamment sur le plan sémantique. Le LIMSI (Morlane-Hondère & D'hondt, 2015) a appliqué une méthode de sélection d'attributs par évaluation du gain d'information (fonction `InfoGainAttributeEval` de Weka (Witten & Frank, 2005)) afin de restreindre l'analyse des tweets aux 450 n-grammes de mots les plus discriminants. De manière similaire, l'INaLCO (équipe 2) a dégagé des descripteurs sémantiques au moyen d'une analyse textométrique. Plusieurs équipes ont également étudié la polarité des tweets, soit par la présence de négations (IRIT/LIMSI, Synapse), soit par des listes de termes polarisés positifs et négatifs (LIMSI). Notons que l'équipe TU Moldova (Bobicev, 2015) a réalisé plusieurs expériences fondées sur le traitement des caractères uniquement ou des mots uniquement, avec et sans normalisation. Les meilleurs résultats obtenus sont ceux fondés sur les caractères uniquement, sans normalisation. L'utilisation d'un algorithme de clustering non-supervisé pour la création de vecteurs de mots (notamment `word2vec` (Mikolov *et al.*, 2013)) sur des gros volumes de données (notamment Wikipedia) a également été utilisé par l'INaLCO, Proxem (Marty *et al.*, 2015) et l'équipe IRISA (Vukotic *et al.*, 2015).

Enfin, les particularités inhérentes aux messages postés sur les réseaux sociaux en général (émoticônes, abréviations) et à Twitter en particulier (mots-dièses, noms d'utilisateur et présence de liens courts [3]), voir section 2, ont été prises en compte par plusieurs équipes (LIRMM ; LIMSI ; Synapse).

Lexiques Nous relevons que toutes les équipes ont utilisé des lexiques d'opinions, d'émotions et de sentiments, soit dans une chaîne de traitements (Synapse ; ANEX et LIDILEM par l'équipe mixte LINA/Dictanova), soit comme traits pour l'apprentissage statistique (CASOAR et EMOTAIX par l'équipe mixte IRIT/LIMSI, équipe 10 ; FEEL par le LIRMM ; Polarimots et DES – Dictionnaire Electronique des Synonymes par le LIMSI). Le LIMSI a également complété ces lexiques par des listes d'insultes tandis que l'équipe LINA/Dictanova (Hernandez *et al.*, 2015) a construit un lexique d'émoticônes composé de 40 classes.

5 Résultats

5.1 Évaluation officielle (par rapport à la référence)

5.1.1 Tâche 1

Sur cette tâche, les résultats en macro-précision (tableau 4) varient de 0 à 0,736, avec une moyenne de 0,582, une médiane de 0,693 et un écart-type de 0,238. L'analyse de la distribution des résultats montrent qu'un majorité des systèmes reconnaît la polarité des tweets avec une macro-précision comprise entre 0,54 et 0,75. Deux équipes (les équipes 4 et 25) ont cependant rencontré des problèmes techniques qui ne permettent pas l'obtention de résultats concluants.

3. La contrainte des 140 caractères maximum dans un message posté sur Twitter impose de réduire les URL. De nombreux services proposent ainsi des raccourcisseurs de liens internet tels que le service t.co propre à Twitter, dont le résultat sera de la forme `http://t.co/identifiant`.

Équipe	Soumissions			Classement
LIF (équipe 3)	**0,736**	0,722	0,688	1
INaLCO (équipe 2)	0,692	0,711	**0,734**	2
LIRMM (équipe 17)	0,732	0,725	**0,733**	3
Synapse (équipe 19)	**0,701**			4
Proxem (équipe 15)	**0,699**			5 ex-aequo
IRISA (équipe 14)	**0,699**	0,672	0,658	5 ex-aequo
LIMSI (équipe 23)	0,687	**0,688**		7
LINA / Dictanova (équipe 6)	0,655	**0,676**		8
IRIT / LIMSI (équipe 10)	0,577	0,578	**0,580**	9
TU Moldova (équipe 22)	**0,559**	0,547		10
LINA–Dimeco (équipe 25)	0,000	0,000	**0,136**	11
(équipe 4)	**0,041**			12

TABLE 4 – Résultats (macro-précision) par équipe sur la tâche 1, par résultats décroissants, la meilleure soumission de chaque équipe est en gras

5.1.2 Tâche 2.1

Sur cette tâche, les résultats en macro-précision (tableau 5) varient de 0,029 à 0,613, avec une moyenne de 0,514, une médiane de 0,217 et un écart-type de 0,029. Les résultats se répartissent selon deux grandes classes. Une première partie des systèmes catégorisent les tweets avec une macro-précision comprises entre 0,33 et 0,38, tandis qu'un deuxième ensemble de systèmes permettent de reconnaître les classes affectives génériques avec une macro-précision variant entre 0,5 et 0,62.

Équipe	Soumissions			Classement
LIRMM (équipe 17)	**0,613**	0,563	0,552	1
INaLCO (équipe 2)	0,572	0,562	**0,575**	2
IRISA (équipe 14)	**0,572**	0,478	0,502	3
LIF (équipe 3)	0,558	**0,560**	0,535	4
LINA / Dictanova (équipe 6)	0,508	**0,514**		5
TU Moldova (équipe 22)	**0,383**	0,382		6
IRIT / LIMSI (équipe 10)	0,269	**0,332**	**0,332**	7
LINA–Dimeco (équipe 25)	0,000	0,000	**0,097**	8
(équipe 4)	0,029	**0,029**		9

TABLE 5 – Résultats (macro-précision) par équipe sur la tâche 2.1, par résultats décroissants, la meilleure soumission de chaque équipe est en gras

5.1.3 Tâche 2.2

Sur cette tâche, les résultats en macro-précision (tableau 6) varient de 0 à 0,347, avec une moyenne de 0,180, une médiane de 0,200 et un écart-type de 0,152. Les résultats se répartissent selon trois grandes classes : un premier ensemble de systèmes réalisent une catégorisation fine des tweets avec une macro-précision inférieure à 0,05, la macro-précision des systèmes du deuxième groupe est comprise entre 0,17 et 0,23, tandis que le troisième ensemble de systèmes reconnaissent les catégories affectives fines avec une macro-précision variant entre 0,32 et 0,35.

5.2 Combinaison par vote pondéré (ROVER)

C'est John Fiscus (Fiscus, 1997) qui a proposé pour la première fois dans une campagne d'évaluation un algorithme de combinaison par vote pondéré des données produites par les participants. Cet algorithme, baptisé « ROVER » (*Reduced Output Voting Error Reduction*) par son auteur, a été créé pour une campagne d'évaluation sur la transcription automatique de parole organisée par le DARPA/NIST. Il permet à moindre coût d'augmenter la quantité de corpus annotés, de qualité et disponibles, en particulier réutilisables pour l'apprentissage automatique. Pour DEFT 2015 nous avons considéré une

Équipe	Soumissions			Classement
LIF (équipe 3)	**0,347**	0,327	0,327	1
INaLCO (équipe 2)	**0,337**	0,292	0,304	2
IRISA (équipe 14)	**0,325**	0,258	0,316	3
TU Moldova (équipe 22)	**0,226**	0,175		4
LIRMM (équipe 17)	0,037	**0,174**	0,007	5
LINA / Dictanova (équipe 6)	**0,028**	0,027		6
(équipe 4)	0,002	**0,002**		7
LINA–Dimeco (équipe 25)	0,000	0,000	0,000	8

TABLE 6 – Résultats (macro-précision) par équipe sur la tâche 2.2, par résultats décroissants, la meilleure soumission de chaque équipe est en gras

somme de vote pondérée par la mesure de performance obtenue par le participant qui a produit l'annotation. L'annotation retenue est celle qui obtient le score maximum. Les résultats de l'évaluation du ROVER appliqué aux tâches 1, 2.1 et 2.2 sont donnés dans le tableau 7. On observe que l'algorithme fonctionne bien s'il existe suffisamment de données pour chaque item d'annotation comme c'est le cas pour la tâche 1, alors que pour les autres tâches, il n'y a pas de gain de performance. Si l'on restreint la combinaison aux quatre systèmes les mieux placés (*ROVER4best* : INaLCO équipe 2, LIF équipe 3, IRISA équipe 14, et LIRMM équipe 17), on voit que le gain diminue un peu pour la tâche 1 mais reste positif tandis que la perte diminue pour les tâche 2.1 et tâche 2.2, confirmant l'intuition que moins l'on a de données, plus il faut sélectionner les données d'entrée en fonction de leur qualité (performance) pour ne retenir que les meilleures, l'effet de masse ne jouant plus pour éliminer le bruit.

tâche	max	ROVER	ROVER - max	ROVER4best	ROVER4best - max
tâche 1	0,736	**0,765**	**0,029**	**0,760**	**0.024**
tâche 2.1	0,613	0,589	-0,024	0,607	-0,006
tâche 2.2	0,347	0,330	-0,016	0,346	-0,001

TABLE 7 – Comparaison pour chaque tâche entre la meilleure performance des participants et celle du ROVER. max = meilleur résultat, ROVER = rover sur les meilleures soumissions de chaque système, ROVER - max = écart entre le rover et le meilleur système, ROVER4best = rover sur les meilleures soumissions des 4 meilleurs systèmes, ROVER4best - max = écart entre le rover4best et le meilleur système

6 Conclusion

Pour la troisième fois, le défi fouille de textes (DEFT) a proposé aux participants de travailler sur la fouille d'opinion. Contrairement aux éditions précédentes qui portaient sur des textes correctement rédigés (retranscriptions de débats parlementaires et articles de journaux) d'une part, et pour un nombre plutôt restreint de catégories d'autre part (favorable/défavorable, objectif/subjectif), l'édition 2015 s'est focalisée sur l'analyse complète des opinions, sentiments et émotions exprimées dans des messages postés sur le réseau social Twitter, sur la thématique du changement climatique. Cette analyse complète a été répartie en trois tâches : (i) déterminer la polarité globale des tweets, (ii) identifier les classes génériques (opinion, sentiment, émotion, information) et spécifiques (parmi 18 classes) de ces tweets, et (iii) analyser la source, la cible et l'expression porteuse d'opinion, de sentiment ou d'émotion.

Douze équipe ont participé. Les meilleurs résultats, en macro-précision, sont de 0,736 (polarité), 0,613 (classes génériques) et 0,347 (classes spécifiques). Les méthodes utilisées reposent majoritairement sur des approches par apprentissage statistique supervisé (SVM, Naïve Bayes, réseaux neuronaux, PPMC), et utilisent de nombreux lexiques d'opinions (ANEW, Casoar, Emotaix, Feel, Lidilem) et de polarités (Polarimots) comme traits. Il faut noter qu'aucun participant n'a soumis de données pour la dernière tâche, annotation fine des opinions, sentiment et émotions. Les premiers retours des participant indiquent qu'ils ont jugé la tâche trop difficile, les données d'entrainement ne leur permettant pas de construire une représentation exploitable.

Remerciements

Ce travail a été réalisé dans le cadre du projet uComp [4] financé par l'ERA Net CHIST-ERA (ANR-12-CHRI-0003).

A Définitions

Nous renseignons dans l'annexe suivante les définitions des 19 catégories fines du guide d'annotation (voir section 2.2).

A.1 Opinions

- *Accord* : opinion positive, la personne est d'accord avec au moins une autre personne sur un événement (*Tout à fait d'accord, le recyclage est devenu une nécessité*) ;
- *Désaccord* : opinion négative, la personne n'est pas d'accord (*Non aux éoliennes de la ferme du Torpt à Tourville et St - Meslin*) ;
- *Valorisation* : opinion positive, la personne désire une entité (événement, objet, personne) et a l'intention de réaliser une action en faveur de cette entité (*#tortueluth espèce menacée " Vu du ciel " #Gabon , les héros de la nature " super doc*) ;
- *Dévalorisation* : opinion négative, la personne ne désire pas une entité (événement, objet, personne) et n'a aucune intention de réaliser une action en faveur de cette entité (*Les saloperies promues par Royal : c' bruyant , laid , dévalorisant pour le foncier , et même pas efficace*) ;

A.2 Sentiments

- *Satisfaction* : sentiment positif, qui est suscité par la réalisation d'une intention résultant d'un désir (*Après un tel repas, je suis rassasié !*) ;
- *Insatisfaction* : sentiment négatif, qui est suscité par la non réalisation d'une intention résultant d'un désir (*Je n'ai pas pu partir en vélo*).

A.3 Emotions

- *Amour* : émotion positive, qui est suscité par le désir d'une autre personne ou animal (*L' amour et la fidelité sont des espéces en voie de disparition*) ;
- *Apaisement* : émotion positive, suscitée par la réalisation d'une intention suite à un événement non désiré (*Je suis soulagé, sa vie n'est plus en danger*) ;
- *Colère* : émotion négative, suscité par la réalisation d'un événement non désiré pare la personne et qui peut susciter ou pas une intention de réaction chez la personne (*Vent de colère sur nos villages . Éoliennes : l' arnaque totale*) ;
- *Déplaisir* : émotion négative qui résulte de la réalisation d'un événement non désiré par la personne (*Un lion en cage , un singe et deux serpents : une soirée cirque qui passe mal au Stamp #Waterloo*) ;
- *Dérangement* : émotion négative qui résulte de la réalisation d'un événement non désiré par la personne et qui suscite une intention d'action pour remédier à l'événement (*Les éoliennes vraiment source de nuisances*) ;
- *Ennui* : émotion négative, suscité par la connaissance de l'absence d'un événement désiré par la personne (*Je m'ennui, il y a rien d'intéressant à faire dans cette ville*) ;
- *Mépris* : émotion négative qui résulte d'une connaissance sur un objet, un événement, une personne qui est en opposition avec nos désirs (*Les losers et les saloppes , deux espèces en voie d' extinction*) ;
- *Peur* : émotion négative, suscité par la réalisation ou l'éventuelle réalisation d'un événement non désiré par la personne (*La sardine l' un des poissons les plus en danger en Méditerranée*) ;
- *Plaisir* : émotion positive, résulte de la réalisation d'un événement désiré par la personne (*je suis content que tu sois là*) ;
- *Surprise négative* : émotion négative, suscitée par la réalisation d'un événement non désiré et non attendu par la personne (*Mauvaise nouvelle : le ministre Henry a accordé le permis unique à Spe Luminus pour pour 5 éoliennes*) ;

4. http://www.ucomp.eu/

– *Surprise positive* : émotion positive, qui est suscitée par la réalisation d'un événement désirable et non attendu par la personne (*Bonne nouvelle pour Brest qui construira les jackets ! Saint - Brieuc .*) ;
– *Tristesse* : émotion négative, suscitée par la non réalisation d'un événement désiré et dont la réalisation est, soit possible dans le futur, soit impossible (*Je vois des trucs clignoter mais je sais pas si c' est des éoliennes ou des feux d' artifices c trist*).

A.4 Instruction

– *Instruction* : *Pensez à recycler vos , bouteilles vides ..*

Références

ABDAOUI A., TAPI NZALI M. D., AZÉ J., BRINGAY S., LAVERGNE C., MOLLEVI C. & PONCELET P. (2015). ADVANSE : Analyse du sentiment, de l'opinion et de l'émotion sur des tweets français. In *Actes de DEFT*, Caen, France : TALN.

ARTSTEIN R. & POESIO M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, **34**(4), 555–96.

BOBICEV V. (2015). Sentiment detection using PPM. In *Actes de DEFT*, Caen, France : TALN.

CHARDON B., MULLER S., LAURENT D., PRADEL C. & SÉGUÉLA P. (2015). Chaîne de traitement symbolique pour l'analyse d'opinion – l'analyseur d'opinions de Synapse Développement face à Twitter. In *Actes de DEFT*, Caen, France : TALN.

FISCUS J. G. (1997). A post-processing system to yield reduced word error rates : recognizer output voting error reduction (rover). In *In proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, p. 347–357, Santa Barbara, CA.

FRAISSE A. & PAROUBEK P. (2014). Toward a unifying model for opinion, sentiment and emotion information extraction. In *Proc. of LREC*, Reykjavik, Iceland.

GROUIN C., ARNULPHY B., BERTHELIN J.-B., EL AYARI S., GARCIA-FERNANDEZ A., GRAPPY A., HURAULT-PLANTET M., PAROUBEK P., ROBBA I. & ZWEIGENBAUM P. (2009a). Présentation de l'édition 2009 du defi fouille de textes (deft'09). In *Actes de DEFT*, p. 35–50, Paris, France.

GROUIN C., BERTHELIN J.-B., EL AYARI S., HEITZ T., HURAULT-PLANTET M., JARDINO M., KHALIS Z. & LASTES M. (2007). Présentation de DEFT'07. In *Actes de DEFT*, Grenoble, France : AFIA.

GROUIN C., HURAULT-PLANTET M., PAROUBEK P. & BERTHELIN J.-B. (2009b). DEFT'07 : une campagne d'évaluation en fouille d'opinion. *Revue des Nouvelles Technologies de l'Information*, **RNTI E-17**, 1–24.

HERNANDEZ N., JADI G., LARK J. & MONCEAUX L. (2015). Exploitation de lexiques pour la catégorisation fine d'émotions, de sentiments et d'opinions. In *Actes de DEFT*, Caen, France : TALN.

LEJEUNE G. & DUMONCEAUX F. (2015). Une approche stylométrique pour la fouille d'opinion. In *Actes de DEFT*, Caen, France : TALN.

MANNING C. D. & SCHÜTZE H. (2000). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts : MIT Press.

MARTY J.-M., WENZEK G., SCHMITT E. & COULMANCE J. (2015). Analyse d'opinions de tweets par réseaux de neurones convolutionnels. In *Actes de DEFT*, Caen, France : TALN.

MIKOLOV T., YIH W.-T. & ZWEIG G. (2013). 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*, p. 746–751, Atlanta, Georgia : Association for Computational Linguistics.

MORLANE-HONDÈRE F. & D'HONDT E. (2015). Feature engineering for tweet polarity classification in the 2015 DEFT challenge. In *Actes de DEFT*, Caen, France : TALN.

VUKOTIC V., CLAVEAU V. & RAYMOND C. (2015). IRISA at DeFT 2015 : supervised and unsupervised methods in sentiment analysis. In *Actes de DEFT*, Caen, France : TALN.

WITTEN I. H. & FRANK E. (2005). *Data Mining - Pratical Machine Learning Tools and Techniques*. Morgan Kaufmann - Elsevier.

Une approche stylométrique pour la fouille d'opinion

Gaël Lejeune, Frédéric Dumonceaux
(1) LINA, 2 rue de la Houssinière, 44322 Nantes, France
prenom.nom@univ-nantes.fr

Résumé. Dans cet article nous proposons une approche stylométrique pour l'édition 2015 du Défi Fouille de Textes. Cette édition du défi portait sur l'analyse d'opinions, de sentiments et d'émotions dans un corpus issu de *Twitter*. Nous avons participé dans trois tâches du défi : classification des *tweets* selon leur polarité (Tâche 1, 3 classes), identification de la classe générique de l'information exprimée dans le *tweet* (Tâche 2.1, 4 classes) et identification de la classe spécifique de l'opinion, sentiment ou émotion exprimée dans le *tweet* (Tâche 2.2, 18 classes). L'approche stylométrique que nous avons utilisée est fondée sur l'utilisation de n-grams de caractères de manière à traiter ces tâches de fouille d'opinion comme des tâches d'attribution d'auteur. Notre hypothèse était la suivante : les traits qui permettent de caractériser le style d'un auteur devraient permettre d'identifier le style inhérent à une classe d'opinion, de sentiment ou d'émotion. Finalement, cette hypothèse s'est avérée erronée, particulièrement sur la tâche 3 qui était la plus difficile. La première interprétation que l'on peut faire serait qu'il n'existe pas véritablement de traits stylistiques inhérents aux classes étudiées. Une autre explication possible est la faible longueur des messages qui rendrait les méthodes stylométriques inopérantes.

Abstract.

A stylometric approach for opinion mining

This article tries to tackle the DEFT'15 opinion mining challenge using a stylometric approach. The dataset proposed by the organizers was a set of microblog messages extracted from Twitter. We participated in three tasks : classification according to polarity (Task 1, 3 classes), classification according to information (Task 2.1, 4 classes) and classification according to specific classes (Task 3, 18 classes). The stylometric approach we used was based on recent work on Autorship Attribution using character n-grams as features. Our assumption was that the features efficient for characterizing an author style would be efficient as well for identifying the opinions or emotions expressed in tweets. We showed that this assumption was wrong, especially on task 3. It appears that the stylometric features might not be well suited for opinion mining tasks. Another hypothesis to explain this result is that the length of the microblog messages might be too small to take advantage of such a stylometric approach.

Mots-clés : stylométrie, attribution d'auteur, analyse d'opinion, analyse de sentiment, classification, chaînes de caractères, microblogs, tweets.

Keywords: stylometry, autorship attribution, opinion mining, sentiment analysis, classification, character substrings, microblogs, tweets.

1 Introduction

L'édition 2015 du Défi Fouille de Textes est consacrée à l'analyse d'opinion dans un corpus de *tweets*. Cette édition comportait trois tâches dont une découpée en deux sous-tâches : Nous n'avons pas participé à la tâche 3, nous concentrant sur les tâches de classification T1, T2.1 et T2.2 :

T1 Classification des *tweets* selon leur polarité (3 classes) ;

T2 Classification fine des *tweets* ;

- **T2.1** Identification de la classe générique de l'information exprimée dans le *tweet* (4 classes) ;
- **T2.2** Identification de la classe spécifique de l'opinion, sentiment ou émotion (18 classes) ;

T3 Détection de la source, la cible et de l'expression d'opinion.

Les *tweets* ont été annotés manuellement avec les différentes classes auxquels ils se rattachaient. Pour une explication plus précise des modalités d'annotation, nous renvoyons au guide d'annotation mis en ligne par les organisateurs du défi [1]. L'intérêt de ces tâches de classification réside par exemple dans le fait de repérer si un *tweet* a une connotation positive ou négative, s'il est purement informatif ou s'il exprime une opinion. . .Dans le cas particulier des *tweets*, cela permet d'aller au-delà de la simple description des *tweets* par les *hashtags*. Parmi les débouchés possibles de ce type d'analyse, nous pouvons citer la veille commerciale (popularité d'un produit) et la veille sociétale (viabilité d'un projet politique).

Dans une première approche de la fouille d'opinion, l'indicateur du nombre de *tweets* traitant d'un sujet pourrait suffire à déterminer la popularité de ce sujet. Ceci rappelle un aphorisme célèbre [2] : « Qu'on parle de moi en bien ou en mal, peu importe. L'essentiel, c'est qu'on parle de moi ! ». L'idée est donc que l'on ne s'intéresse qu'au fait que des émotions soient exprimées, quelles qu'elles soient. De telle sorte que l'on accorde une importance centrale à la récurrence d'un thème dans un corpus ou plus généralement dans l'« actualité ». À l'opposé, l'on serait plus indifférent vis-à-vis d'autres thèmes, moins fréquemment rencontrés dans un corpus indépendamment de la polarité de leur traitement. Au contraire, une approche plus « qualitative »s'intéresserait beaucoup plus au contenu réel de ce qui est transmis, à ce qui est dit du sujet. D'un point de vue linguistique, un accent serait donc mis sur le rhème et non plus seulement sur le thème.

Dans la section 2 nous décrivons l'approche que nous avons employée pour cette édition du défi. Dans la section 3 nous présenterons les données mises à disposition pour le défi ainsi que les résultats obtenus par notre approche. Nous proposerons nos conclusions et perspectives de recherche dans la section 4.

2 Description de l'approche

Les travaux en fouille d'opinion sont classiquement répartis entre des approches symboliques (Zhang *et al.*, 2012) et approches dites statistiques (Pak & Paroubek, 2010) avec au centre les approches dites mixtes ou hybrides (Vernier *et al.*, 2009). Notre angle d'attaque pour cette édition du défi est fondé sur la stylométrie, domaine où l'on trouve les trois types d'aproche précités. La stylométrie, parfois nommée *forensic linguistics* dans la littérature, consiste à chercher les indices qui rattachent un énoncé à un style particulier. Ceci peut par exemple permettre de rattacher un texte à un sous-genre, méthode employée dans l'édition 2014 du DEFT pour classer des nouvelles (Lecluze & Lejeune, 2014). Une des applications les plus fréquentes est la tâche d'attribution d'auteur qui consiste à identifier les indices laissés par un auteur dans les textes qu'il a produit de manière à disposer d'un modèle pouvant prédire les auteurs de textes anonymes (Brixtel *et al.*, 2015). (Daelemans, 2013) propose une tripartition des connaissances que l'on peut extraire d'un texte : objectives (informations factuelles), subjectives (opinions exprimées) et meta-connaissances (extraire ce qui n'est pas dans le texte. Dans cette classification, la stylométrie s'attaque donc plus généralement aux méta-connaissances, l'auteur et le genre textuel dans les deux exemples cités.

Notre hypothèse est que la stylométrie peut aussi permettre d'extraire de la connaissance dans le domaine subjectif. Les traits exploités sont des signes de ponctuation et des n-grams de caractères, traits qui sont les plus efficaces selon les travaux récents sur l'attribution d'auteur (Sun *et al.*, 2012).

Pour notre premier *run*, nous avons exploité l'effectif des caractères non-alphanumériques, l'effectif total de lettres et l'effectif total de chiffres dans chaque *tweet*. Pour chacun de ces traits nous avons ajouté en plus de l'effectif, la proportion de cet effectif en fonction de la taille du *tweet*. Pour les *run* 2 et 3 nous avons exploité des n-grams de caractères avec respectivement $n = 1$ et $3 <= n <= 4$. L'objectif du *run* 1 est de mesurer si l'ajout dans les traits de l'effectif de tous les caractères alpha-numériques (et non simplement de la classe lettres et de la classe chiffres) amènerait une perte de résultat. Le *run* 3 correspond à des valeurs de n pour lesquels les modèles en n-grams de caractères sont connus pour être efficaces.

La chaîne de traitement utilisée est illustrée dans la figure 1. Le classifieur utilisé est un SVM (Séparateur à Vaste Marge ou *Support Vector Machine*) à noyau linéaire, classique dans les analyses stylométriques orientées attribution d'auteur. Nous avons utilisé $C = 1$ pour paramètrer la fonction de coût. D'une part, cela permettait de nous placer dans la même configuration que les approches classiques d'attribution d'auteur et donc de faciliter les comparaisons. D'autre part, nous avons observé que faire varier ce paramètre avait une influence marginale sur les résultats obtenus avec les traits utilisés.

1. `https://deft.limsi.fr/2015/guideAnnotation.fr.php?lang=fr` accédé le 28 mai 2015
2. Attribué à Léon Zitrone

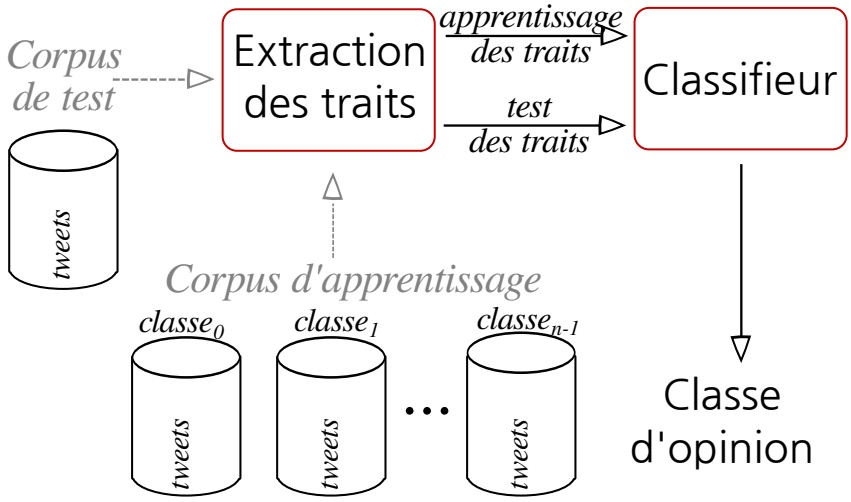

FIGURE 1 – Chaîne de traitement utilisée pour l'apprentissage des classes

3 Résultats

Pour rappel, le *run* 1 correspond à une analyse stylométrique fondée sur l'usage des caractères non alpha-numériques et du nombre de lettres et de chiffres contenus dans chaque *tweet*. La *run* 2 correspond à une classification à partir des 1-grams de caractères tandis que le *run* 3 consiste en une classification à partir de 3-grams et 4-grams de caractères. Nos résultats pendant la phase d'entraînement n'étaient pas très satisfaisants mais nous avons obtenu des résultats bien plus décevants à l'issue de la phase de test (Tableau 1).

	run 1	*run* 2	*run* 3
T1 (3 classes)	0	0,0000986	0,136
T2.1 (4 classes)	0	0	0,097
T2.2 (18 classes)	0	0	0

TABLE 1 – Résultats officiels (macro-précision)

Il s'est avéré que nous avons commis plusieurs erreurs dans la génération des fichers de résultat, aboutissant ainsi dans la moitié des cas à un score nul. Pour cet article, nous avons donc corrigé ces erreurs de manière à présenter des résultats plus fidèles à ce que nous avions constaté lors de la phase d'entraînement. Les résultats recalculés sont présentés dans le tableau 2 et sont, fort heureusement, nettement différents de ceux figurant au classement officiel du défi. Nous souhaitons préciser que les résultats présentés ici sont générés à partir de la même chaîne de traitement que celles que nous avions développé pour le défi. Pour éviter toute confusion, nous avons renommé les runs en « variantes ».

Si nos résultats restent assez décevants, ils amènent tout de même quelques observations intéressantes. La première est que l'approche stylométrique que nous avons utilisé reste inopérante pour la tâche la plus difficile (T2.2). Par contre, elle semble plus prometteuse pour les deux autres tâches bien qu'éloigné des meilleurs résultats des autres équipes participant au défi. Les classes présentes dans ces deux tâches semblent plus compatibles avec l'approche stylométrique que nous

	Résultats globaux du défi			Nos résultats corrigés		
	Moyenne	Médiane	Maximum	Variante 1 (*run 1*)	Variante 2 (*run 2*)	Variante 3 (*run* 3)
T1 (3 classes)	0,581	0,693	0,735	0,369	0,091	0,289
T2.1 (4 classes)	0,408	0,515	0,708	0,318	0,0513	0,251
T2.2 (18 classes)	0,119	0,137	0,231	0,078	0,019	0,059

TABLE 2 – Résultats (macro-précision) après correction des fichiers de sortie

avons adopté. La tâche 2.2 était la plus difficile avec 18 classes différentes mais nous pensions que la méthode stylométrique s'adapterait bien à cette profusion de classes. En effet, il est fréquent en attribution d'auteur de traiter simultanément 50 ou 60 auteurs (et donc autant de classes). Cette hypothès a été contredite par nos résultats. Nous proposons dans la section 4 quelque réfléxions sur ces résultats.

4 Conclusion

Nous avons proposé pour cette édition du Défi Fouille de Textes, une approche stylométrique fondée sur des n-grams de caractères. Notre hypothèse était la suivante : les traits qui permettent d'attribuer la paternité d'un texte devraient être en mesure de déterminer également un style en termes d'opinion ou d'émotion. Indépendamment des problèmes rencontrés dans la phase de test, il apparaît que cette approche était insatisfaisante. Le manque de connaissances proprement linguistiques dans cette méthode a été un facteur trop limitant. Il est également difficile de savoir si ces résultats faibles sont dues à la tâche de fouille d'opinion en elle-même ou bien au genre textuel encore peu formalisé que constituent les *tweets*. Leur faible longueur a pu notamment être un facteur d'échec pour l'approche stylométrique. Par exemple, (Forsyth & Holmes, 1996) considèrent qu'il faut au minimum des messages de 250 mots (1500 caractères) pour que l'analyse stylométrique soit viable. Au contraire, les travaux récents de (Bhargava *et al.*, 2013) et (Almishari *et al.*, 2014) montrent que l'on peut contourner ce problème pour les *tweets* dès lors que l'analyse se porte sur des auteurs très actifs. Malgré toutes ces réserves, il serait intéressant d'explorer de manière plus approfondie ces approches stylométriques pour d'autres tâches de détection d'émotion mais en intégrant, par exemple, des traits plus fins tels que les étiquettes morpho-syntaxiques.

Remerciements

Nous tenons à remercier une fois de plus les organisateurs pour les efforts fournis pour proposer chaque année de nouveaux défis à relever.

Références

ALMISHARI M., KAAFAR D., OGUZ E. & TSUDIK G. (2014). Stylometric linkability of tweets. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, p. 205–208, New York, NY, USA : ACM.

BHARGAVA M., MEHNDIRATTA P. & ASAWA K. (2013). Stylometric analysis for authorship attribution on twitter. In V. BHATNAGAR & S. SRINIVASA, Eds., *Big Data Analytics*, volume 8302 of *Lecture Notes in Computer Science*, p. 37–47. Springer International Publishing.

BRIXTEL R., LECLUZE C. & LEJEUNE G. (2015). Attribution d'Auteur : approche multilingue fondée sur les répétitions maximales. In *Actes de la 22e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2015)*.

DAELEMANS W. (2013). Explanation in computational stylometry. In A. GELBUKH, Ed., *Computational Linguistics and Intelligent Text Processing*, volume 7817 of *Lecture Notes in Computer Science*, p. 451–462. Springer Berlin Heidelberg.

FORSYTH R. S. & HOLMES D. I. (1996). Feature-finding for text classification. *Literary and Linguistic Computing*, **11**(4), 163–174.

LECLUZE C. & LEJEUNE G. (2014). Deft 2014, analyse automatique de textes littéraires et scientifiques en langue française. In *Actes de DEFT 2014 : 10ème DÉfi Fouille de Textes*, p. 11–19, Marseille, France.

PAK A. & PAROUBEK P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta : European Language Resources Association (ELRA).

SUN J., YANG Z., LIU S. & WANG P. (2012). Applying stylometric analysis techniques to counter anonymity in cyberspace. *Journal of Networks*, **7**(2).

VERNIER M., MONCEAUX L. & DAILLE B. (2009). DEFT'09 : détection de la subjectivité et catégorisation de textes subjectifs par une approche mixte symbolique et statistique. In *Atelier Défi Fouille de Textes (DEFT'09)*, p. 101–112, Paris, France.

ZHANG L., FERRARI S. & ENJALBERT P. (2012). Opinion analysis : The effect of negation on polarity and intensity. In J. JANCSARY, Ed., *Proceedings of KONVENS 2012*, p. 282–290 : ÖGAI. PATHOS 2012 workshop.

IRISA at DeFT 2015: Supervised and Unsupervised Methods in Sentiment Analysis

Vedran Vukotic Vincent Claveau Christian Raymond
INRIA[1] – CNRS[2] – INSA de Rennes[3] – IRISA[1,2,3],
Campus de Beaulieu, 35042 Rennes cedex
{vedran.vukotic, vincent.claveau, christian.raymond}@irisa.fr

Résumé. Cet article décrit la participation de l'équipe LinkMedia de l'IRISA à DeFT 2015. Notre équipe particpé à deux tâches : la classification en valence des tweets (tâche 1) et la classification à grain fin, elle même, décomposée en deux sous-tâches, à savoir la détection des classes génériques de l'information exprimée dans un tweet (tâche 2.1) et la classification des classes spécifiques (tâches 2.2) de l'émotion/sentiment/opinion exprimée. Pour ces trois tâches, nous adoptons une démarche d'apprentissage artificiel. Plus précisément, nous explorons l'intér de trois méthodes : i) le *boosting* d'arbres de décision, ii) l'apprentissage bayésien utilisant une technique issue de la recherche d'information, et iii) les réseaux neuronnaux convolutionnels. Nos approches n'exploitent aucune ressource externe (lexiques, corpus) et sont uniquement fondées sur le contenu textuel des tweets. Cela nous permet d'évaluer l'intérêt de chacune de ces méthodes, mais aussi des représentations qu'elles exploitent, à savoir les sacs-de-mots pour les deux premières et le plongement de mots (*word embedding*) pour les réseaux neuronaux.

Abstract.

IRISA participation at DeFT 2015: Supervised and Unsupervised Methods in Sentiment Analysis
In this work, we present the participation of IRISA Linkmedia team at DeFT 2015. The team participated in two tasks: i) valence classification of tweets and ii) fine-grained classification of tweets (which includes two sub-tasks: detection of the generic class of the information expressed in a tweet and detection of the specific class of the opinion/sentiment/emotion. For all three problems, we adopt a standard machine learning framework. More precisely, three main methods are proposed and their feasibility for the tasks is analyzed: i) decision trees with boosting (bonzaiboost), ii) Naive Bayes with Okapi and iii) Convolutional Neural Networks (CNNs). Our approaches are voluntarily knowledge free and text-based only, we do not exploit external resources (lexicons, corpora) or tweet metadata. It allows us to evaluate the interest of each method and of traditional bag-of-words representations vs. word embeddings.

Mots-clés : Fouille d'opinion, apprentissage artificiel, boosting, apprentissage bayésien, plongement de mots.

Keywords: Opinion mining, machine learning, boosting, Bayesian learning, word embedding.

1 Introduction

All the three classification problems analyzed in this paper are sentiment analysis problems. They consist of the same data samples, French tweets, whose target labels vary according to the task. After describing these classification tasks and their corresponding datasets in Section 1, we present and analyze the classifiers used in this work. In section 2, decision trees with boosting are applied directly to symbolic words. Section 3 describes a system consisting of Bayesian learning relying on information retrieval techniques. Section 4 deals with Convolutional Neural Networks (CNNs) combined with various word embedding methods. Finally, in section 5, all the three systems are compared for the three classifications tasks and a conclusion is given in section 6.

1.1 Task 1: valence classification

The provided training set consists of 7929 tweets, 102 of which were not found at the time of retrieval. Each sample belongs to one of the following three classes: positive (+), negative (-) and neutral/mixed (=). The distribution of the classes in the training set is sufficiently uniform: 31% (+), 24% (-) and 45% (=).

1.2 Task 2.1: detection of the generic class of the information

The provided training set for this task consists of 6754 tweets, 81 of which were not found at the time of retrieval. Each sample in this task belongs to one of the following four classes: information, opinion, sentiment and emotion. The classes are not very well distributed within this task, thus possibly introducing biases in the classifiers: 53% (INFORMATION), 34% (OPINION), 1% (SENTIMENT) and 12% (EMOTION).

1.3 Task 2.2: detection of the specific class of the opinion/sentiment/emotion

For this task, 3183 tweets were provided, 40 of which were not found during retrieval. There are 18 classes in this task: negative surprise / negative astonishment (SURPRISE_NEGATIVE), agreement / understanding / approbation (ACCORD), sadness / sorrow / suffering / despair / resignation (TRISTESSE), valorization / interest / appreciation (VALORISATION), dissatisfaction / unhappiness (INSATISFACTION), fear / terror / worry / anxiety (PEUR), appeasement / relief / gratefulness / forgiving / serenity (APAISEMENT), anger / rage / irritation / exasperation / nervousness / impatience (COLERE), satisfaction / contentment / pride (SATISFACTION), disagreement / disapprobation (DESACCORD), displeasure / deception (DEPLAISIR), disturbance / embarrassment (DERANGEMENT), love / sweetness / affection / devotion / passion / envy / desire (AMOUR), contempt / disdain / disgust / hate (MEPRIS), pleasure / entertainment / joy / euphoria / happiness / ecstasy (PLAISIR), positive surprise / positive astonishment (SURPRISE_POSITIVE), boredom (ENNUI), devalorization / disinterest / depreciation (DEVALORISATION).

In this task, the classes are also not well balanced and could thus introduce a bias into the classifiers: 4.8% (ACCORD), 0.3% (AMOUR), 0.3% (APAISEMENT), 6.6% (COLERE), 1.5% (DEPLAISIR), 0.4% (DERANGEMENT), 6.8% (DESACCORD), 12.6% (DEVALORISATION), 0.1% (ENNUI), 0.3% (INSATISFACTION), 5.5% (MEPRIS), 8.6% (PEUR), 1.1% (PLAISIR), 2.3% (SATISFACTION), 0.3% (SURPRISE_NEGATIVE), 0.1% (SURPRISE_POSITIVE), 1.1% (TRISTESSE), 47.3% (VALORISATION). Also, it is worth noticing that this task has the least samples and the most classes, which together with their nonuniform distribution is negatively effecting the learning of meaningful decision boundaries.

1.4 Test set

The test set of the DeFT 2015 challenge consists of 3383 tweets (2 of which were not found at the time of retrieval for some participants and were removed by the DeFT commission). The correct target labels were unknown at the time of developing the methods described in this articles, so all the experiments performed and described are done on splits of the provided training data into train, validation and test sets, as described in each of the following sections. The performance of our methods on the official test set is given at the end of this article.

2 Boosting decision trees

In order to tackle the three tasks presented in the introduction, we first adopt a standard Machine learning approach based: boosting decision Trees. This approach, using a very usual text representation of the tweets, is intended to be a baseline. In this section, we first describe the principles of this machine learning technique and then the performance obtained with the training dataset.

Task	Parameters	**Macro-precision**	**Micro-precision**
Task 1	$n = 2000d = 4$	0.702	0.684
Task 2.1	$n = 100d = 3$	0.712	0.717
Task 2.2	$n = 100d = 10$	0.427	0.625

Table 1: Performance of Bonzaiboost in a 10-fold cross-validation

2.1 Principles

Decision Trees are well-known classifiers based on a succession of test on the features of an example in order to determine its class. They can be easily inferred from training examples, they can deal with numeric or symbolic features, and they are able to handle any number of classes.

In our case, in order to achieve better performance, several Decision Trees are inferred and from the training data and combined following the popular boosting algorithm AdaBoost.MH (Schapire & Singer, 2000). This meta-learning framework is an iterative process, in which errors (tweets receiving the wrong class label) obtained with the tree inferred at the previous step are given more weight so that the next decision tree will focus more on this problematic examples.

The implementation that we used in our experiments is BONZAIBOOST [1].

2.2 Practical details and performance

The tweet are represented as sets of words (unigrams) in order to be manipulated by the decision trees. The test used in the nodes of the trees is thus just to check if a tweet contains a word or not.

Boosting was first proposed to improve the classification performance of weak learners, that is, classifiers with a low precision (though it has also be shown to improve results of strong classifiers). Thus, in the case of Decision Trees, boosting is often used with low depth trees (Antoine Laurent & Raymond, 2014), such as decision stumps (depth=1). During the training phase, we experimented with several number of iterations (n) for boosting and several depth of trees (d). These two parameters (n and d) were chosen through cross-validation in order to maximize the macro-precision, as it is the main score proposed by the organizers. The best performance is found for $d =$ and $n =$; they are summarized in Table 1. Yet, it is noteworthy that several other settings (n and d) gave very close results. This is due to the fact that macro-precision is a very unstable score, since a simple tweet, if it is correctly or incorrectly classified in a class with very few tweets, can drastically change the final score.

One interesting thing with decision trees is that they can be interpreted, to some extents. It is even easier with decision stumps since one can directly see the (only) test, that is, in our case, the presence or absence of a word in a tweet, with the label of the class and a weight. Moreover, when using boosting over decision stumps, we get a high number of decision stumps, and thus a high number of these word/label/weight associations. They may be gathered to form a lexicon suited to the set of class used in the tasks. For instance, here is a small excerpt of such a lexicon concerning the word `menace` for Task 2.2 in which one can see that `menace` is strongly associated with PEUR, and to the contrary, strongly opposed to VALORISATION.

[1] `http://bonzaiboost.gforge.inria.fr/`

word	label	vote
menace	ACCORD	-1.450
menace	AMOUR	-0.626
menace	APAISEMENT	-0.473
menace	COLERE	-1.447
menace	DEPLAISIR	-1.100
menace	DERANGEMENT	-0.411
menace	DESACCORD	-1.520
menace	DEVALORISATION	0.219
menace	ENNUI	-0.306
menace	INSATISFACTION	-0.797
menace	MEPRIS	-1.321
menace	PEUR	2.132
menace	PLAISIR	-1.046
menace	SATISFACTION	-1.023
menace	SURPRISE_NEGATIVE	-0.741
menace	SURPRISE_POSITIVE	-0.266
menace	TRISTESSE	-1.042
menace	VALORISATION	-2.195

3 Bayesian learning

In this section we present the second machine learning approach tested by IRISA for Tasks 1, 2.1 and 2.2. As for the previous one, it relies on a simple representations of text, that is bags-of-words. It is a somewhat classical framework based on Bayesian learning whose only originality is to use Information Retrieval techniques in order to estimate certain probabilities needed to predict class labels of the tweets.

3.1 Principles

Let us note ω_i the label of one the class and d_j is the description of one of the tweets. Under a probabilistic framework, the goal of the task can be expressed as finding the class that maximizes this probability $P(\omega_i|d_j)$, i.e.

$$\omega^* = \arg\max_i P(\omega_i|d_j)$$

By using the well-known Bayesian rule, this probability can be decomposed into:

$$P(\omega_i|d_j) = \frac{P(d_j|\omega_i) * P(\omega_i)}{P(d_j)}$$

Moreover, since $P(d_j)$ is constant for a given tweet, we have:

$$\omega^* = \arg\max_i P(d_j|\omega_i) * P(\omega_i)$$

Finally the two probabilities to estimate on the training data are the a priori probability of a class $P(\omega_i)$ and the likelihood $P(d_j|\omega_i)$.

In this work, the a priori probability is simply estimated by counting the number of tweets of each class, and by smoothing (Laplace smoothing) not to put a disadvantage on less populated classes:

$$P(\omega_i) = \frac{\text{nb tweets of class } omega_i + 1}{\text{nb tweets } + \text{ nb classes}}$$

The main originality of our approach is to use an information retrieval technique in order to estimate the likelihood of a tweet given a class. For a given tweet, we use a k-nearest-neighbors approach: the k closest training tweets are retrieved by using an information retrieval system. More precisely, we use a system based on Okapi-BM25 (Robertson *et al.*, 1998)

Task	Macro-precision	Micro-precision
Task 1	0.679672521722188	0.615226864906434
Task 2.1	0.792678272906324	0.641901037750038
Task 2.2	0.449395650820018	0.590676883780332

Table 2: Performance of Bayesian learning on the training set (leave-one-out)

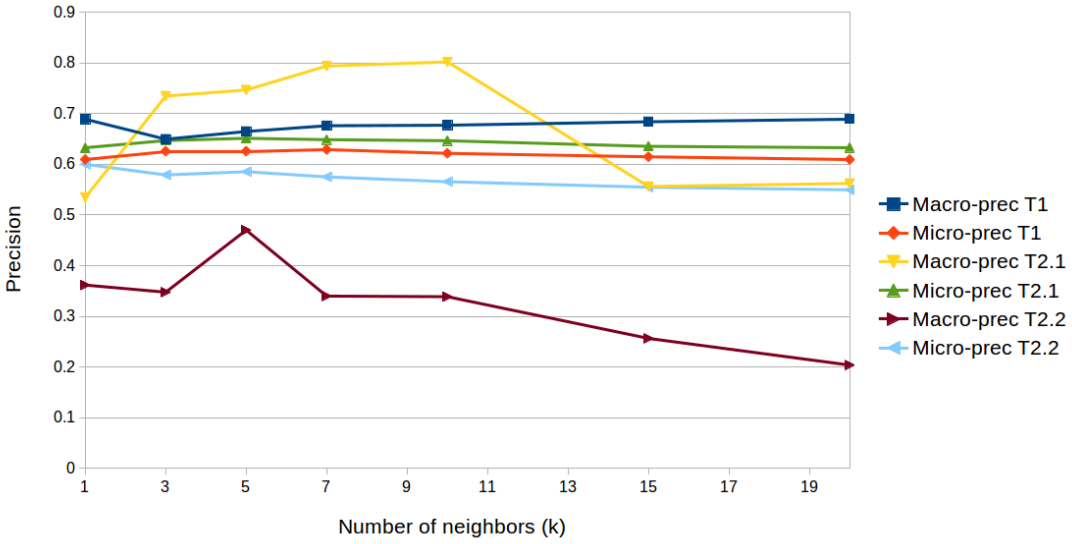

Figure 1: Macro and micro-precisions for Tasks 1, 2.1, 2.2 according to k

which has proven robust and efficient in many information retrieval or extraction tasks. The considered tweet is thus considered as a query q, containing words noted t_i. A score is computed for each training tweet d, as follows (df gives the document frequency of a word, dl gives the document length, dl_{avg} is the average document length in the collection, $k_1 = 1$, $b = 0.75$ and $k_3 = 1000$ are constants):

$$\begin{aligned} score_{BM25}(d) &= \sum_{t_i} qTF(t_i, q) * TF_{BM25}(t_i, d) * IDF_{BM25}(t_i) \\ &= \sum_{t_i} \frac{(k_3 + 1) * tf(t_i, q)}{k_3 + tf(t_i, q)} * \frac{tf(t_i, d) * (k_1 + 1)}{tf(t_i, d) + k_1 * (1 - b + b * dl(d)/dl_{avg})} * \log \frac{N - df(t_i) + 0.5}{df(t_i) + 0.5} \end{aligned} \quad (1)$$

From the k tweets with highest scores, we estimate the likelihood of the tweet for a given class as the proportion of tweets from this class in these k neighboring tweets.

3.2 Practical details and performance

For our experiments, the value for k was tuned on the training set through leave-one-out in order to maximize the macro-precision. The values found are $k = 10$ for task 1, $k = 10$ for task 2.1 and $k = 3$ for task 2.2.

The performance of our Bayesian learning approach is shown in table 2. It is computed with a 10-fold cross-validation over the training dataset.

In order to illustrate the influence of the number of neighbors considered when computing the likelihood, Figure 1 presents the macro and micro-precisions for each task, according to different values of k. One can notice that this value has only a limited influence on micro-precision (about 0.05 between maximum and minimum micro-precision values. Concerning macro-precisions, the situation is contrasted: for Task 1, k has a small influence. It is easily explained by the fact that there many tweets for each class, and thus it does not influe on the kNN process. Conversely, for Task 2.1 and 2.2, macro-precisions varies a lot. This is due to the fact that for some classes with very few training data, if k gets higher

than the population of the class, the kNN will necessarily includes tweets from other classes. This fact is made worse by the instability of the macro-precision score (see Section 5.3), since tweets from these less populated classes will tend to be misclassified, even if the number of misclassified tweets do not vary so much globally.

4 Convolutional Neural Networks

Deep learning methods have been shown to produce state-of-the-art results in various tasks and different modalities (vision, speech, etc.). Convolutional Neural Networks (LeCun & Bengio, 1995) represent one of the most used deep learning method in visual recognition. Recent works have shown that CNNs are also well suited for sentence classification problems and can produce state-of-the-art results (Kim, 2014; Johnson & Zhang, 2014). These studies have analyzed the performance of CNNs in datasets like movie reviews, the Stanford sentiment treebank, customer reviews and similar datasets that usually consists of positive/negative labels (either binary or with fine-grained intensities). In this section, we explore the feasibility and measure the performance of similar CNN models in multi-class emotion classification of short sentences (tweets).

CNNs consist of a series of interchanged convolutional and pooling layers, followed by a simple classifier (usually a MLP or an RBF). CNNs have two interesting properties that make them valuable and robust classifiers in computer vision: i) autonomous learning of meaningful features and ii) robustness to spatial variations (translational invariance).

Convolutional layers scan the inputs of the previous layer with learned kernels. The kernels are learned by backpropagation and become filters that respond to features relevant for the classification task. By applying multiply convolutional layers, the network learns meaningful features at different levels.

Pooling layers reduce the dimensionalites of their respective previous layers either by different downsampling methods or by more complex pooling methods that use different metrics. The most common pooling method in CNNs is max-pooling. Max-pooling have been shown to provide good results while also being very computationally inexpensive (Boureau *et al.*, 2010).

A typical approach in machine learning for improving generalization consists of combining different architectures. However, that can be computationally expensive. The dropout method suggest randomly disabling some hidden units during the training part, thus generating a big set of combined of virtual classifiers without the computational overhead (Hinton *et al.*, 2012). For a simple multilayer perceptron with N neurons in one hidden layer, an equivalent of 2^N virtual architectures would be generated by applying dropout. Dropout has been shown to significantly improve generalization of CNNs.

CNNs have recently been shown to be useful and perform well in NLP classification problems (Kim, 2014). The difference between CNNs applied to computer vision and their equivalent in NLP lies in the input dimensionality and format. In computer vision, inputs are usually single-channel (eg. grayscale) or multi-channel (eg. BGR) matrices (2D), usually of constant dimensions. In sentence classification, each input consist of a sequence of words of non-constant length. Each word w is represented with a numerical vector (embedding) $\vec{e_w}$ of a constant size. Word embeddings can be learned jointly, with the classifier, in a supervised manner or separately, usually in a unsupervised manner (eg. word2vec (Mikolov *et al.*, 2013)). All the word representations are then concatenated (in their respective order) and padded with zero-vectors to a fixed length (maximum possible length of the sentence). Eg. the phrase "Que la force soit avec toi #4mai" would be represented to the CNN input as $\vec{x} = \vec{e}_{que} \oplus \vec{e}_{la} \oplus \vec{e}_{force} \oplus \vec{e}_{soit} \oplus \vec{e}_{avec} \oplus \vec{e}_{toi} \oplus \vec{e}_{TAG} \oplus \vec{e}_{UNKNOWN} \oplus \vec{0} \oplus \ldots \vec{0}$ with $|\vec{x}_i| = const., \forall i$, where the $\oplus$ operator represents concatenation. Although this representation may look counter-intuitive from a more classical NLP viewpoint, it is a perfectly good representation for a CNN, due to their translational invariance property that enables them to be robust to changes in position within sentences, while still being able to grasp relevant features from the words and their context.

4.1 Model description

The model used in this work shares the same architecture as the simple model in (Kim, 2014): a single convolutional layer, followed by a max-over-time (Collobert *et al.*, 2011) pooling layer and a standard soft-max fully connected layer at the end. A standard dropout of 0.5 (50% of the neurons are disabled in each iteration) is used.

The cross-validation of this method was done with 5-folds. For each epoch of the CNN the training set (the current

training set of a 5-folds validation) is first randomly split into a smaller training set (90%) and a validation set (10%). Figure 2 shows the errors of the validation sets for different variations that will be described in the following subsections.

Prior to converting words to their numerical representation (word embedding), all the sentences were preprocessed in the following manner: all the words were converted to their lowercase equivalent; punctuation marks were isolated into separate elements; URLs and references were converted to respective tags ("<url>" and "<ref>"); finally, hashtags were converted to two separate elements: a tag and the original hashtag string (eg. "#nowplaying" → "<tag> nowplaying").

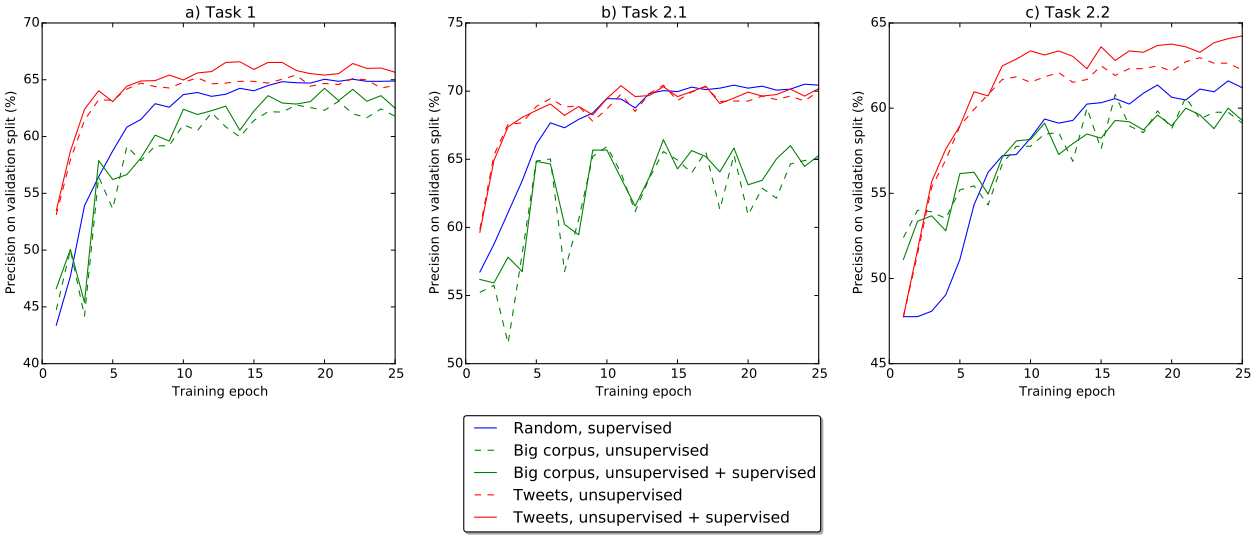

Figure 2: Performance comparison of various text embedding methods with CNNs

4.2 Randomly initialized word embeddings

In these experiments, word embeddings were initialized randomly from a uniform distribution $\mathcal{U}(-0.25, 0.25)$. Afterwards, the embeddings were trained in a supervised manner together with the CNN. This works by simply by extending backpropagation to the layer before the first convolutional layer (the lookup table). As seen is in figure 2, this method takes a significant number of epochs more to converge to a stable precision, but does perform quite well. It can be noticed that in Task 2.2 (figure 2 c.), due to the biggest number of classes and the least number of examples, it converged slower and with less good precision on the validation sets.

4.3 Word embeddings learned on a big corpus

These six experiments used external big corpuses to train embeddings for the French language in an unsupervised manner. It is important to note that obviously these embeddings had no special knowledge (an appropriate representation) of Twitter specifics (hashtags, references, URLs, etc.). Embeddings of unknown words are initialized randomly and are later either fine-tuned in a supervised manner or not. For each task, two variations were analyzed: one in which the embeddings learned in an unsupervised manner were kept static and one in which they were updated (fine-tuned) while learning the CNNs. The latter has been shown to perform slightly better but with no significant differences.

Compared to random embeddings, these embeddings trained on large corpora have a slight initial advantage but do not perform as well after more training epochs. Our explanation for this phenomena is that in such tasks, having a notion of the twitter specifics within the embeddings is advantageous and the method lacks that advantage.

4.4 Word embeddings learned on DeFT tweets

Here, we describe six experiments with embeddings learned directly from the tweets (the training set of the current fold) in an unsupervised manner, with word2vec (Mikolov *et al.*, 2013): three with static embeddings and three with furtherly

Emb. initialization	Fine-tuning	Macro-precision	Micro-precision
Task 1			
Random	Supervised (CNN backpropagation)	0.651662928327043	0.653123802223074
Big corpus (unsupervised)	Supervised (CNN backpropagation)	0.624707901585991	0.626038073335889
Big corpus (unsupervised)	-	0.615873905343281	0.614283889101827
Tweets (unsupervised)	Supervised (CNN backpropagation)	0.657463685838549	0.66002299731698
Tweets (unsupervised)	-	0.651445850997478	0.652484987862527
Task 2.1			
Random	Supervised (CNN backpropagation)	0.663228030898926	0.676007792597033
Big corpus (unsupervised)	Supervised (CNN backpropagation)	0.594594050894346	0.639892102502623
Big corpus (unsupervised)	-	0.633633717113728	0.647984414805934
Tweets (unsupervised)	Supervised (CNN backpropagation)	0.650912059769033	0.675108646785554
Tweets (unsupervised)	-	0.679203233664306	0.679304660572456
Task 2.2			
Random	Supervised (CNN backpropagation)	0.255742475472588	0.608017817371938
Big corpus (unsupervised)	Supervised (CNN backpropagation)	0.324818272474731	0.599427298759147
Big corpus (unsupervised)	-	0.315797307860250	0.596881959910913
Tweets (unsupervised)	Supervised (CNN backpropagation)	0.340052284654461	0.618835507476933
Tweets (unsupervised)	-	0.360776096220910	0.620108176901050

Table 3: Performance analysis of various embedding methods and CNNs in a 5-fold cross-validation

fine-tuned embeddings (in a supervised manner, while performing backpropagation on the CNNs).

These methods seem to perform the best. The unsupervised representation learning brought initial advantage (faster convergence). Fine-tuning the embeddings contributed to a very small advantage, except for task 2.2 where the supervised fine-tuning brought a significant advantage.

Combining embeddings learned in an unsupervised manner directly on the tweets (thus enabling them to have notions of Tweeter specifics) and fine tuning them in a supervised manner while training the CNNs is the best choice for this kind of tasks, especially if there is a significantly small number of training samples and a bigger number of classes, like in task 2.2.

4.5 Performance

Table 3 shows the results of the various methods with CNNs described in this section. Macro and micro precisions are computed with the official DeFT 2015 evaluation tool. Only macro-precision is evaluated during for the challenge, but here, we report also micro-precision as it might provide a better performance measure for this type of multiclass classification problems.

All the methods were evaluated with a 5-folds cross-validation (80% training and 20% test for each fold). It is clear that the best methods consists of training word embeddings in an unsupervised manner directly on tweets. Additionally supervised fine-tuning improves the performance, but just slightly.

Training embeddings in generic big corpus for specific classification tasks (like short tweets) does not seem to be a feasible method.

Random initialization of embeddings and supervised learning seems to also be feasible. One could think that this method is faster than training unsupervised embeddings first, but unsupervised word representation methods like word2vec (Mikolov *et al.*, 2013) are very fast (especially in small datasets like this one) and significantly improves the speed of convergence and precision.

Method	Macro-precision	Micro-precision
Task 1		
Bayesian learning	0.6985095279	0.6898490678
Bonzaiboost	0.6723841209	0.5995856762
CNN, word2vec on tweets + supervised fine tuning	0.6580527369	0.6531518201
Task 2.1		
Bayesian learning	0.5722246948	0.60165729506
Bonzaiboost	0.4779050332	0.52352767091
CNN, word2vec on tweets + supervised fine tuning	0.5020312287	0.57147084937
Task 2.2		
Bayesian learning	0.1738004542	0.23024563481
Bonzaiboost	0.1725898801	0.20775377331
CNN, word2vec on tweets + supervised fine tuning	0.1814082223	0.22255105061

Table 4: Performance of the three proposed methods on the official DeFT 2015 test set

5 Performance comparison and discussion

5.1 Official results

The DeFT 2015 accepted three submissions (runs) per team. We opted to submit one run from Bayesian learning, one with Bonzaiboost and a last one from the CNN method with embeddings learned on tweets and fine tuned. It is important to notice that cross-validations differ in the experiments described in this paper prior to the final official evaluation (eg. CNNs use a 5-fold cross-validation while Naive Bayes was evaluated with a 10-fold cross-validation). The most indicative performance evaluation is thus given on the official DeFT 2015 test set and is shown in table 4.

Bayesian learning performed best in tasks 1 and 2.1. CNNs performed best in Task 2.2. This improvement of CNNs performances in multiclass data with less samples is due to gain brought by the directly learned embeddings. It is important to note that there should be room for improvement in CNNs to close the gap in simpler datasets (task 1 and 2.1) and probably improve upon the other results. We think that it would be valuable to research applying the same embeddings to a deeper CNN (more than one convolutional and one pooling layer) thus enabling the network to grasp concepts at greater distances within a sentence.

The difference observed between these values and the one estimated on the training set, especially for task 2.2, is left unexplained. A technical problem or the choice of wrong parameters may explain that, as well as a difference when building the training and the test sets. But we also give additional thoughts on the fact that the macro-precision was not a reliable choice in order to chose the best parameters (see below).

5.2 Data inconsistencies and its consequences

The training data contains many tweets whose provided label, whatever the tagset (ie. task 1, 2.1 or 2.2), is hard to explain. For instance, consider the following tweets:

INFORMATION	`@ONPCofficiel Sortir du nucléaire oui, à condition d'avoir des nouvelles énergies plus écologiques, Cécile Duflot!`
POSITIVE_SURPRISE	`@LoireBretagne On s'étonne qu'elle soit ouverte, cette pêche à l'anguille. Sauf si pêcher une espèce menacée est une forme de protection...`

This even more problematic with less-populated classes; here are three of the eight tweets labeled as AMOUR (task 2.2) in the training data.

AMOUR	. @RoyalSegolene "Les Français souhaitent à 90% le développement des énergies renouvelables" http://t.co/TYyr2jFR81
AMOUR	j'ai envie d'autres paysages que le vide et les cinq pauvres éoliennes de l'aube
AMOUR	Les sénateurs écologistes fans de @RoyalSegolene » http://t.co/kSP3l2cv5s

Most striking, very similar tweets may receive different annotations (here, labels for tasks 1 and 2.1):

+/NONE	@fanph94 3/4: Lorsque la combustion est optimum, les particules fines émises sont composées de sels alcalins peu nocifs
=/INFORMATION	@fanph94 4/4: Combustion de qualité + combustible sec et propre = quantités de particules fines émises réduites et moins nocives

This inconsistencies are even more obvious with quasi-identical tweets:

=/INFORMATION	#TribunedeGenève . L'Etat s'est trouvé face à un Diogène industriel: Trois cents tonnes de déchets toxiques à ... http://t.co/b4lJ9UsBrS
+/NONE	L'Etat s'est trouvé face à un Diogène industriel: Trois cents tonnes de déchets toxiques à assainir. ... http://t.co/m8MeDGAvRg #Genève
=/INFORMATION	Développement durable : 20 ans après le Caire, l'ONU réaffirme le rôle central des gens: A l'ouver... http://t.co/q9gD1sMXOQ RT @Mediaterre
+/OPINION	Développement durable : 20 ans après le Caire, l'ONU réaffirme le rôle central des gens http://t.co/XCaVGb16F6
+/OPINION	Développement durable : 20 ans après le Caire, l'ONU réaffirme le rôle central des gens http://t.co/LUelHU4flR #ecologie Mediaterre

Of course, such inconsistencies question the validity of the evaluation since the test set certainly contains identical flaws. They are also harmful for any machine learning technique during the training step. Unfortunately, this is more specifically the case for our Bayesian learning, to some extent, and for boosting. Concerning The former, one can easily understand that the kNN used to compute the likelihood will make its decision based on very close or similar examples. A similar example with a wrong label will chiefly biase the decision, especially when k is small. Thus, these inconsistencies are more particularly harmful for task 2.2, where the small number of examples for certain class requires us to set k to such small values. The boosting is also known to sensible to low quality training data. Since more weight is given to misclassified examples at each iteration, the inconsistent examples presented above will receive more and more weight and thus biased the final combination of weak learners (Philipp M. Long, 2010)

5.3 Discussion about the stability of macro-precision

The choice of macro-precision as the main evaluation score raise several issues. As it was explained before (cf. Sec. 3.2), this choice, combined with the fact that classes are strongly unbalanced (especially for task 2.2), is not well taken into

System	macro-precision	micro-precision
S1 (whole dataset)	0.3696	0.5885
S2 (whole dataset)	0.4485	0.6265
S1 (average)	0.3563	0.5885
S2 (average)	0.3744	0.6265

Table 5: Macro and Micro-precisions computed over the dataset as a whole or divided into 30 subsets

account by our simple machine learning frameworks.

The other problem we want to stress here is the instability of this score. As said earlier the correct or incorrect classification of a single tweet, in a less-populated class, may have a strong impact on the final result. It means that the difference between two systems' macro-precisions, is not necessarily statistically significant, even if the difference itself may appear as important. In order to illustrate that point, we consider two systems S1 and S2 based on on our boosting approach with sligthly different settings. S1 and S2 uses $n = 100$ iterations, but S1 builds decision stumps ($d = 1$) while the maximal depths for S2 is $d = 10$. These two systems are trained and evaluated on the training dataset (through 10-fold cross-validation) on task 2.2. The micro and macro-precisions for the whole dataset (concatenation of the 10 test folds) are given in Table 5. It is worth noting S2 performs better than S1; the difference between the two systems' macro-precision is high, and also important in a smaller extent in terms of micro-precisions. In order to study the stability of these scores, we also divide the dataset into 30 equal parts and compute for each of them and for each system the two scores. Thus, we have 30 macro-precisions and 30 micro-precisions for S1 and the same for S2. We indicate the average precisions over these 30 subsets in the Table 5. It appears that the average macro-precision for S2 is in fact much lower than when considering the dataset as a whole. In order confirm that, we compute a paired Student t-test between these two systems with the 30 pairs of scores. The p-value for the macro-precision is $p = 0.2582$, and the micro-precision's one is $p = 0.00045$. It means that the difference of macro-precision between S1 and S2 is not statistically significant, while the micro-precision difference can be considered as statistically significant.

This tends to demonstrate that macro-precision computed over a whole dataset of tweets is not a reliable score to evaluate the systems, especially when some classes only contains a few tweets, as it is the case for several classes of Task 2.2. Another side-effect is that using macro-precision to chose the final parameters of our machine learning techniques has certainly led us to pick up sub-optimal solutions.

6 Conclusion

For this participation of IRISA to Tasks 1, 2.1 and 2.2 of DeFT2015, we focused on Machine learning approaches exploiting only the texts of the tweets, with no external knowledge sources or metadata. Our goal was mainly to examine the interest of embeddings over traditional bag-of-words approach. We showed that traditional methods outperform simple deep methods (CNNs) in tasks with less classes and a sufficient number data samples for each class. For more complex tasks, deep learning has been shown to be a slightly better approach, especially when combined with embeddings previously learned in an unsupervised manner and fine tuned during classification. Yet, the added-value is very small and simple bags-of-words still appear as competitive.

The question of whether a deeper CNN architecture would be able match standard methods or even outperform them on all datasets is open for further future experimentation. Another point to investigate is the difference of results between train set (with cross-validation) and the test-set.

From the point of view of the tasks itself, we have pointed some inconsistencies in the ground-truth and explained how they challenged our machine learning techniques. More importantly, we have shown that the macro-precision is a too unstable score to evaluate the systems directly, at least for Task 2.2 in which some classes contains only a few tweets. As a result, the rankings of the participants based on this sole score, is unfortunately not reliable. On the contrary, micro-precision appears much more stable score and is more naturally optimized by usual machine learning frameworks.

7 Bibliography

ANTOINE LAURENT N. C. & RAYMOND C. (2014). Boosting bonsai trees for efficient features combination : application to speaker role identification. In *Proc. of InterSpeech*.

BOUREAU Y., PONCE J. & LECUN Y. (2010). A theoretical analysis of feature pooling in vision algorithms. In *Proc. International Conference on Machine learning (ICML'10)*.

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, **12**, 2493–2537.

HINTON G. E., SRIVASTAVA N., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

JOHNSON R. & ZHANG T. (2014). Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

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

LECUN Y. & BENGIO Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, **3361**.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations*.

PHILIPP M. LONG R. A. S. (2010). Random classification noise defeats all convex potential boosters. *Machine Learning Journal*, **78**(3).

ROBERTSON S. E., WALKER S. & HANCOCK-BEAULIEU M. (1998). Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive. In *Proc. of the 7th Text Retrieval Conference, TREC-7*, p. 199–210.

SCHAPIRE R. E. & SINGER Y. (2000). Boostexter: A boosting-based system for text categorization. *Mach. Learn.*, **39**(2-3), 135–168.

Chaîne de traitement symbolique pour l'analyse d'opinion - l'analyseur d'opinions de Synapse Développement face à Twitter

Baptiste Chardon, Sophie Muller, Dominique Laurent, Camille Pradel, Patrick Séguéla[1]
(1) Synapse Développement, 5 rue du Moulin Bayard, 31000 TOULOUSE
{baptiste.chardon, sophie.muller, dlaurent, camille.pradel, patrick.seguela}@synapse-fr.com

Résumé. Cet article présente la chaîne de traitement d'analyse d'opinion de Synapse Développement qui a été utilisée dans le cadre de la campagne DEFT 2015. Cette chaîne repose sur une analyse syntaxique et sémantique généraliste du français et sur des lexiques généralistes parfois associés à des lexiques spécialisés selon des besoins spécifiques de clients. Nous présenterons en première partie le contexte de cette campagne d'évaluation. La deuxième partie donnera une vue d'ensemble de la chaîne de détection de l'opinion proposée par Synapse. La troisième et la quatrième partie seront consacrées aux adaptations des ressources syntaxiques et lexicales au contexte du tweet et au lexique de spécialité relatif au développement durable. La partie 5 permettra de décrire la gestion des opérateurs et l'agrégation des opinions au niveau du tweet. Nous conclurons sur les résultats obtenus en termes de classification selon la polarité sur les corpus d'apprentissage et de test de DEFT 2015.

Abstract.

Symbolic pipeline for opinion mining – Synapse opinion miner vs. Twitter.
This paper presents the Synapse Développement processing system analyzing opinions in discourse which has been used during DEFT 2015 campaign. This system is based on syntactic and semantic analysis and general lexicons sometimes associated with specialized lexicons according to specific needs of clients. In first part, we will present the context of this evaluation campaign. The second part will give an overview of the opinion detection system. The third and the fourth parts will be dedicated to adaptations of syntactic and lexical resources in the context of tweets and the lexicon of specialty related to sustainable development. Part 5 will describe the management of the operators and the opinion's aggregation at the tweet level. We conclude on the results obtained in terms of classification according to the polarity on the learning and test corpus of DEFT 2015.

Mots-clés : Analyse d'opinion, extraction d'opinion, analyse de sentiment, analyse de subjectivité, détection d'émotion
Keywords: Opinion extraction, opinion mining, sentiment analysis, subjectivity analysis, emotion detection

1 Introduction et contexte

Avec le développement des réseaux sociaux, spécialement de la plate-forme de microblogging Twitter, donner son avis sur Internet est aisé et fréquent. Par conséquent, tout organisme souhaitant obtenir un retour concernant son e-réputation, l'accueil reçu par un produit, le sentiment général sur un sujet de société, etc. a à sa disposition un flux d'informations colossal : Twitter propulse en effet actuellement plus de 500 millions de tweets chaque jour, et un million de sites intègrent des tweets, comme fil d'actualité ou comme lien avec une communauté.

L'analyse d'opinion est un domaine très actif dans la recherche en traitement des langues naturelles. Une partie importante des efforts de recherche s'intéresse à l'analyse de l'opinion dans les tweets : citons notamment les travaux de (Pak & Paroubek, 2010), (Koloumpsis et al., 2011), (Jiang et al., 2011), (Prinyanthan et al., 2012), ou pour le français de (Brun et Roux, 2014).

Dans ce contexte, de nombreuses sociétés sont également intéressées par les outils d'analyse automatique de l'opinion. Synapse Développement propose une chaîne de traitement professionnelle de l'opinion basée sur une analyse syntaxique et sémantique de pointe, et l'évaluation DEFT 2015 nous permet de confronter nos technologies avec un corpus annoté de tweets.

Afin de répondre au mieux aux besoins spécifiques de clients professionnels, notre chaîne de traitement repose sur des mécanismes symboliques. Ceci nous permet de réagir rapidement sur des demandes clients, et de maîtriser l'impact d'une modification liée à un besoin spécifique. La section suivante décrit plus en détail cette chaîne de traitement.

2 Vue d'ensemble de la chaîne d'analyse d'opinion TextAnalyst *by Synapse*

L'objet du produit TextAnalyst *by Synapse* est d'extraire les signaux faibles du bruit ambiant, d'identifier les informations pertinentes pour une activité, de produire des vues synthétiques et facilement appréhendables du domaine d'étude. Ce produit permet, entre autres applications de cette extraction d'informations, d'annoter les opinions et de les caractériser.

La chaîne d'analyse d'opinion est composée de plusieurs modules en cascade / pipeline. Le schéma suivant récapitule cette architecture :

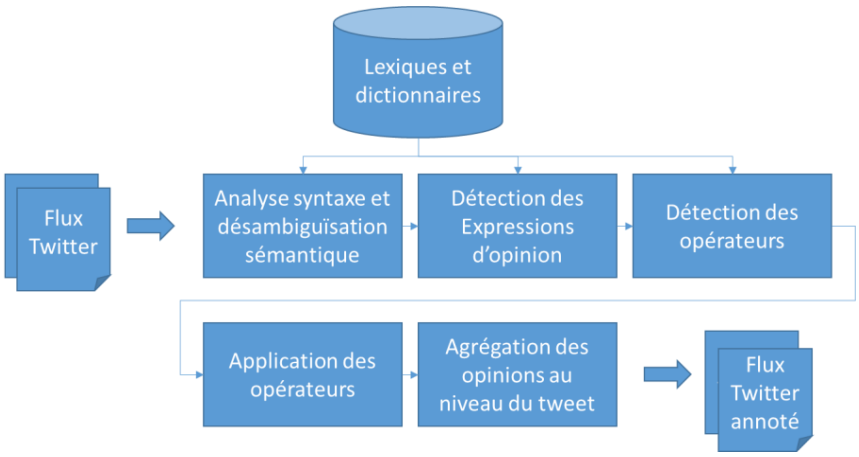

FIGURE 1. Aperçu de la chaîne de traitement.

Les modules utilisés sont les suivants :

- Connecteur d'entrée : permet la mise au format du texte d'entrée pour traitement par la chaîne d'analyse
- Analyse syntaxique et sémantique : il s'agit de l'analyseur syntaxique de Synapse Développement (cf. section 3)
- Reconnaissance des opinions unitaires : il s'agit d'un module de reconnaissance des expressions d'opinions en contexte. Ce module s'appuie sur l'analyse syntaxique et la désambiguïsation sémantique du module précédent, ainsi que sur des ressources lexicales, décrites en section 4.
- Reconnaissance et application des opérateurs : ce module détecte les opérateurs venant s'appliquer sur l'opinion (intensifieurs, négation, modalités) et applique leur effet sur les expressions d'opinion précédemment repérées. La section 5.1 décrit comment ces effets sont appliqués.
- Agrégateur d'opinion : ce module permet, à partir de la collection des expressions d'opinion repérées et d'éléments de contexte discursif, de calculer une opinion globale pour le document (ici le tweet). La section 5.2 décrit plus en détail le fonctionnement de ce module.
- Connecteur de sortie : permet l'écriture des sorties au format attendu pour l'évaluation DEFT

Les principales originalités de cette chaîne sont :

- le calcul de la modification (négation, modaux) de la polarité du segment via le modèle parabolique présenté dans (Chardon, 2013).
- la très large couverture du lexique généraliste (plus de 3000 entrées) créé avec le CNRS. Chaque entrée du lexique est associée à une catégorie d'opinion.
- l'aspect opérationnel de la chaîne de traitement, actuellement en production.

3 L'analyseur syntaxique et sémantique de Synapse

3.1 L'analyseur dans un contexte généraliste

Cordial (acronyme de CORrecteur D'Imprécisions et Analyseur Lexico-sémantique) est un analyseur syntaxique conçu à l'origine pour la correction orthographique et grammaticale. Développé au début des années 90, constamment maintenu et enrichi depuis, Cordial est le fondement de nombreux développements : composants linguistiques de nettoyage

automatique de texte, extracteur de mots-clés et de phrases-clés, extracteur de thèmes et de concepts, extracteur de terminologie et d'entités nommées, moteur de question-réponse.

Afin d'analyser le mieux possible un texte comportant éventuellement des fautes d'accord ou d'autres fautes, il est difficilement envisageable d'utiliser une grammaire formelle, qui fait le plus souvent appel à des règles d'appariement de groupes privilégiant le genre et le nombre. Nous avons donc associé à des règles générales et très peu liées aux attributs de genre et de nombre (analyseur à relâchement de contraintes) un ensemble d'outils statistiques, en particulier pour effectuer la désambiguïsation grammaticale.

Cordial est un analyseur à approche essentiellement statistique et probabiliste, même s'il utilise quelques règles pour la désambiguïsation grammaticale et surtout de très nombreuses règles pour la correction grammaticale.

Tous les mots sont alors "désambiguïsés" grammaticalement. En fait, la liste des formes grammaticales possibles pour chaque mot est conservée jusqu'à la fin de l'analyse, de même que la probabilité de chaque forme grammaticale, permettant des changements jusqu'à la phase de détection des relations entre les groupes. Un module spécifique traite alors les ambiguïtés "lourdes", c'est-à-dire celles constituées d'au moins deux mots ambigus, pour lesquelles les tables sont parfois insuffisantes et qui nécessitent la prise en compte d'un contexte supérieur à quatre mots. C'est le cas des couples adjectif/nom, nom/adjectif ("bonne alerte"), déterminant/nom ou personnel/verbe ("l'aide", "les avions").

Une première phase de désambiguïsation sémantique est alors effectuée. Elle se base essentiellement sur le contexte gauche et droit pour affecter une probabilité à chacun des sens possibles des mots polysémiques. Nous prenons en compte environ 25 000 sens pour 9 000 mots polysémiques, ce qui est un peu inférieur à un dictionnaire papier, le but étant de séparer des sens correspondant à des usages syntaxiques hétérogènes et surtout à des concepts nettement différenciés. Ainsi "abdomen" est monosémique pour Cordial alors que de nombreux dictionnaires distinguent la région inférieure du corps des mammifères et la partie postérieure du corps des arthropodes. Pour chaque mot, Synapse possède un nombre fini de sens (<=8). Ces sens, les informations qui les caractérisent ainsi que la façon de les différencier dans les textes, sont stockés dans le Lexique grammaire. Il référence 154 884 lemmes (noms, adjectifs, verbes, adverbes), 85 938 expressions nominales (“pomme de terre”, “foie gras”, etc.) et 12 059 expressions verbales (“avoir faim”).

3.2 L'analyse syntaxique dans le contexte des tweets

Twitter est une plate-forme de microblogging qui permet de poster des messages courts, éventuellement liés à un contenu extérieur. Chaque message ne peut contenir que 140 caractères maximum, ce qui correspond généralement à une seule phrase. De ce format contraint découlent plusieurs spécificités :

- la texte est trop court pour donner un contexte au tweet. Pour pallier cela, les utilisateurs emploient fréquemment des « hashtags » ou « mots-dièse » permettant de spécifier de quoi il est question dans le tweet (e.g. : #Suisse, #ChangementClimatique, cf. figure N). Il peut également arriver qu'un hashtag ne soit pas neutre en termes d'opinion véhiculée (e.g. #PasCool). Ces hashtags, identifiés par le caractère '#', sont également utilisés par Twitter pour l'indexation des tweets dans leur moteur de recherche, et peuvent par la suite permettre à d'autres utilisateurs de retrouver des tweets sur un sujet donné.

- les tweets comportent fréquemment des liens vers des ressources externes sur le web, soit parce que le tweet porte sur le contenu d'un site web, soit parce que le site web contient un article détaillant le point de vue exprimé succinctement dans le tweet,

- les utilisateurs de Twitter sont identifiés par un pseudonyme unique. Ce pseudonyme ou nom d'utilisateur peut ensuite être utilisé dans un tweet pour référer à l'utilisateur en question. Ces noms d’utilisateurs sont précédés d'un '@'.

FIGURE 2. Deux tweets de @IGNFrance – captures d'écran de l'interface Twitter

La figure précédente montre deux exemples de tweets, publiés par le même compte @IGNFrance (compte officiel de l'Institut National de l'Information Géographique et Forestière – https://twitter.com/IGNFrance). On peut remarquer sur ces tweets qu'il y a deux manières d'insérer un de ces éléments dans le texte :

- le tweet du dessous, en date du 8 Mai, comporte plusieurs hashtags, une référence à un utilisateur, et une URL en fin de texte, non intégrés à la phrase.

- le tweet du dessus intègre directement le hashtag #ChangementClimatique dans la phrase, en remplacement du syntagme nominal « Changement climatique ».

Afin de préserver au mieux l'analyse syntaxique des tweets, l'analyseur syntaxique de Synapse Développement traite ces éléments de la façon suivante :

- les URL sont détectées et traitées comme dans n'importe quel texte

- les '#' et '@' sont ignorés, et les hashtags et nom d'utilisateurs sont traités comme des mots simples : si ceux-ci sont présents dans nos dictionnaires (e.g. : #changement #climatique), ils sont correctement reconnus et désambiguïsés. Si ceux-ci sont absents de nos dictionnaires, qu'ils représentent un syntagme (e.g. #ChangementClimatique), ou autre chose (e.g. #cop21), ils sont reconnus comme noms (commun ou propre) inconnu, et n'interfèrent donc pas avec la désambiguïsation sémantique du reste de la phrase.

4 Ressources lexicales pour l'analyse d'opinion

4.1 Ressources génériques issues de collaboration

Le lexique originel sur lequel se base notre chaîne de traitement est le lexique issu du projet CASOAR[1]. Ce lexique de termes subjectifs se compose de 270 verbes, 632 adjectifs, 296 noms, 594 adverbes, 51 interjections et 178 expressions. Il a été construit manuellement, à partir de l'étude de corpus variés (articles de presse, commentaires web, et courrier des lecteurs). Chaque entrée possède une polarité et une intensité, ainsi qu'une catégorie sémantique, telle que définie par (Asher et al., 2008). Ces catégories sémantiques sont indépendantes d'une langue donnée.

Ce lexique gère à la fois l'ambiguïté de polarité et l'ambiguïté de sens : pour chaque entrée, seuls les sens subjectifs ont été associés à une polarité et une intensité. Ceci permet donc d'exploiter la désambiguïsation sémantique

Ainsi, lors de l'analyse syntaxique, si le sens détecté par l'analyseur n'est pas encodé dans le lexique, on peut en conclure que l'entrée lexicale a un sens objectif sinon elle est subjective. La figure suivante présente l'entrée "cher" : les balises "sense" définissent la catégorie sémantique, la polarité et la force de l'entrée lexicale, les balises "dicoSense" définissent un sens reconnu par la ressource dictionnairique associée. Une balise "sense" peut contenir une ou plusieurs balises "dicoSense".

```
<entry id="ADJ_cher">
        <lemma>cher</lemma>
        <pos>ADJ</pos>
        <sense type="jugement" category="évaluation" strength="1" polarity="pos">
                <dicoSense id="$1" description="aimé"/>
        </sense>
        <sense type="jugement" category="évaluation" strength="1" polarity="neg">
                <dicoSense id="$2" description="coûteux"/>
        </sense>
</entry>
```

Cette ressource est encore en cours de finalisation concernant le lien entre sens subjectif et sens des dictionnaires de Synapse Développement. À l'heure actuelle, seuls les liens pour les adjectifs sont complètement renseignés ; nous nous sommes donc limités à cette catégorie syntaxique pour les travaux présentés ici.

De plus, la chaîne dispose d'un « stop-lexique ». Ce lexique permet, dans des contextes spécialisés de supprimer l'annotation sur un sens, un lemme ou une expression. Par exemple, l'adjectif « écologique », porte généralement une connotation positive, mais est tout à fait contre-productif dans un corpus de tweets consacré au développement durable. Son annotation est donc désactivée par ce biais. De même, ce lexique permet de gérer des cas d'expressions figées qui ne correspondent pas vraiment à l'utilisation classique d'un terme. En effet, certaines expressions portent un sens si minoritaire qu'elles ne sont pas gérées dans le mécanisme de désambiguïsation sémantique. Ainsi le nom « top » est globalement positif dans son sens de supériorité, mais ne porte pas la même subjectivité dans l'expression « top départ ».

[1] projetcasoar.wordpress.com

De même, le terme « souci » est généralement associé à une polarité négative et à la catégorie Desinteret_Devalorisation_Depreciation mais ne sera pas porteur de cette notion lorsqu'il est employé dans l'expression « dans un souci de ».

Le lexique spécialisé contient donc des entrées du type :

```
<lex_entry amb_pol="false" amb_sense="false" id="SADV_dans_un_souci">
        <lemma> dans un souci
                <lemma_compose >
                        <element lemma="dans" pos="1" rank="1"/>
                        <element lemma="un" pos="2" rank="2"/>
                        <element lemma="souci" pos="3" rank="3"/>
                </lemma_compose>
        </lemma>
        <pos>EXP</pos>
        <sense type="filter" category="Desinteret_Devalorisation_Depreciation" polarity="neutre"
strenght="0">
                <synapseSens id=" " description=" "/>
                <example/>
        </sense>
</lex_entry>
```

Enfin, nous avons mis en place un système de priorités sur les lexiques pris en compte dans le système. En effet, les termes absents de nos lexiques d'opinion génériques ainsi que les termes présents dans les bases Casoar et redéfinis en fonction du corpus de spécialité devaient permettre une annotation plus fine. De même, le lexique d'expressions complexes, en prenant plus en compte le contexte, permet un plus grand degré de certitude sur la qualification subjective qu'un terme isolé.

Ces constatations nous ont naturellement amenés à définir un ordre d'intervention de chaque lexique. Celui de spécialité arrivant naturellement en premier, suivi de celui des expressions complexes. Cette approche permet non seulement une meilleure précision mais évite aussi la multiplication de tags subjectifs sur une même expression, générant, par là même, des incohérences.

4.2 Extensions de ces ressources

4.2.1 Ressources lexicales à vocation généraliste

L'extension des ressources lexicales à vocation généraliste a été réalisée selon plusieurs méthodes. La confrontation à des textes de natures très différentes de ceux traités habituellement par notre système a nécessité l'élaboration d'une méthode pour augmenter le lexique spécialisé à partir du corpus d'apprentissage.

*4.2.1.1 **Méthode 1 : Méthode basée sur la coordination d'un adjectif avec un adjectif issu du lexique.***

Les lemmes non porteurs d'une opinion d'après notre première analyse mais utilisés en coordination avec un terme porteur d'opinion sont des candidats à l'entrée dans le lexique.

- Les termes rencontrés avec des catégories différentes sont exclus.
- Les termes supprimés manuellement du premier lexique généré sont exclus.

L'intensité a été automatiquement fixée à 2, ce qui correspond à une intensité moyenne.

Cette méthode permet d'enrichir assez facilement le lexique mais n'apporte que peu d'amélioration en termes de rappel et de précision. En effet, l'ensemble des tweets dans lesquels ils entrent en coordination sont déjà bien classifiés. Ainsi dans le tweet numéro 48973538002144768, on relève la coordination entre « calme » et « reposant » qui propose « reposant » comme candidat à l'entrée dans le lexique avec les mêmes attributs de polarité et de catégorisation sémantique que « calme ».

Un endroit calme et reposant "les #étangs d'or". #beaune #cotedor #biodiversite #nature # vacances http://t.co/c4sucgr2H1

Sur certains tweets, la coordination fait remonter des candidats plus difficiles à catégoriser, par exemple la coordination entre « saine » et « musclée » dans le tweet numéro 520630800509718528 :

Nouvelle recette saine et musclée sur le blog! </phr><phr id="phr_1_1g_38">Pas de gluten,100% naturelle, 100% énergie de qualité! Régalez-vous! http://t.co/XqHb5aBuSs

Cette méthode n'a été appliquée que sur les adjectifs du fait de la plus grande variation de polarité observée entre deux noms ou verbes.

Pourtant certains cas peuvent se révéler intéressants, par exemple en ce qui concerne la coordination de « conseiller » et « assister » dans le tweet suivant, sachant que « conseiller » figurait déjà dans nos lexiques.

En effet, si cette méthode a validé l'intégration de « assister » suite à sa coordination avec « conseiller » dans des tweets tels que :

Steria conseille et assiste l'ESMA dans la gestion de ses projets IT et ses interactions avec son écosystème IT http://t.co/mgwXUzfTzB

*4.2.1.2 **Méthode 2 : Méthode basée sur l'occurrence de certains lemmes dans des tweets portant la même polarité.***

Cette méthode a impliqué différentes sous-tâches afin de produire un lexique de qualité. L'idée était de pouvoir tester les ajouts selon diverses combinaisons de caractéristiques.

La première phase a consisté à analyser syntaxiquement et sémantiquement l'ensemble du corpus de tweets d'apprentissage. L'analyse syntaxique et sémantique a permis de relever les termes candidats les plus occurrents selon chacune de leur catégorie morphosyntaxique et selon chacun de leur sens. Dans un deuxième temps, le système d'apprentissage a attribué une polarité et une catégorie sémantique à chaque sens de chaque lemme.

A l'issue de cette phase, nous disposions d'un lexique de 20 976 termes candidats lemmatisés associés :

- à leur fréquence brute dans le corpus,
- à la fréquence d'apparition dans chaque groupe de tweets d'une même polarité
- et à la fréquence d'apparition dans chaque groupe de tweets d'une même classe sémantique.

Un premier filtrage des termes n'apparaissant que dans des tweets à polarité étiquetée neutre dans le corpus d'apprentissage a permis d'éliminer 7 987 lemmes candidats.

Objectif précision

Dans un premier temps, une analyse visant l'amélioration de la précision a été menée. Pour ce faire, nous avons conservé uniquement les lemmes qui n'apparaissaient que dans des tweets portant la polarité positive ou négative. Ceci a permis de sélectionner 8 481 lemmes dont 7 737 n'apparaissant qu'une fois et 744 plus d'une fois dans le corpus selon les conditions suivantes :

- Un candidat retrouvé (hors négation) dans des passages à la fois étiquetés positivement et négativement est exclu (ex : ambiance) dans un premier temps

- Un candidat retrouvé (hors négation) dans des passages liés à des catégories sémantiques éloignées est exclu.

Un filtrage manuel de ces 744 lemmes a ramené cette liste à 239 lemmes.

Notons, que nous avons volontairement choisi de ne pas pousser plus avant l'apprentissage de termes et d'expressions et d'exercer un filtrage très strict afin de ne pas être victime d'un sur-apprentissage. Celui-ci aurait donné d'excellents résultats, tant sur le corpus d'apprentissage que sur celui de test, mais n'aurait revêtu que peu d'intérêt quant à l'évolution du système, voire l'aurait dégradé sur la plupart des corpus sur lesquels notre système pourrait intervenir. De fait, les lemmes intervenant moins de 3 fois dans le corpus d'apprentissage n'ont pas du tout été étudiés. Une analyse rapide des expressions relevées a donné les mêmes conclusions : peu d'entre-elles étaient utilisables sur corpus générique. La piste des expressions figées a donc été mise de côté.

À l'issue de ces apprentissages et après validation manuelle de ces candidats issus de la coordination et de la méthode d'analyse d'occurrence avec objectif précision, nous disposions d'un lexique comportant 51 adjectifs, 66 noms, 42 verbes et 6 adverbes. Soit, 167 lemmes désambiguïsés syntaxiquement et sémantiquement et étiquetés selon leur polarité, leur catégorie sémantique et leur intensité. Ils se répartissent comme suit : 68 lemmes/sens portant une polarité positive et 99 lemmes/sens portant une polarité négative

Objectif rappel

L'idée de cet objectif est de parvenir à repérer l'opinion portée par des termes qui apparaissent parfois en contexte positif, parfois en contexte négatif et parfois en neutre. Cette méthode permet de pallier les difficultés d'analyse des négations et de certains modaux liés à la structure même des tweets ainsi que les erreurs potentielles d'annotation dans le corpus d'apprentissage.

Le seuil d'analyse manuelle de ces candidats a été fixé à 80 % d'utilisation dans des tweets de même polarité afin de conserver la cohérence et de ne pas perdre des termes peu fréquents mais tout de même intéressants tels que « triste » qui apparaît dans 1 tweet étiqueté positif, 6 tweets étiquetés négatif et 1 tweet étiqueté neutre. Par contre, il sélectionne aussi quelques cas plus inattendus qui ne seront pas retenus ici. Par exemple, on ne s'attendrait pas à relever « transcanada » comme marqueur de subjectivité. Pourtant il apparaît dans 34 tweets négatifs contre 4 tweets positifs et 3 neutres.

De même on pouvait s'attendre à ce que les smileys, régulièrement considérés comme marqueurs objectifs de l'opinion, et même parfois comme éléments permettant l'évaluation d'un système, soient systématiquement dans des tweets portant la même polarité. Or il n'en est rien : ils sont régulièrement utilisés dans des tweets étiquetées par des polarités différentes, voire comme neutre.

Ainsi on relève les occurrences suivantes :

	tweets positifs	tweets négatifs	tweets neutres
: (	1		
:-(	1		1
:-))	1		
:)	7	4	11
:-)	4	1	6
:-/	1		
;-)	2	4	5
;-p		1	

TABLE 3 : Occurrences de smileys et polarité des tweets dans le corpus d'apprentissage

Ce filtrage a conduit à extraire 136 lemmes-candidats, revus manuellement.

À l'issue de cet apprentissage et après validation manuelle de ces candidats issus de la méthode d'analyse d'occurrence avec objectif rappel, nous disposions d'un lexique comportant 5 adjectifs, 17 noms, 20 verbes et 1 interjection. Soit, 43 lemmes désambiguïsés syntaxiquement et sémantiquement et étiquetés selon leur polarité, leur catégorie sémantique et leur intensité. Ils se répartissent comme suit : 15 lemmes/sens portant une polarité positive et 28 lemmes/sens portant une polarité négative

Les intensités ont été revues manuellement afin de conserver une cohérence avec les lexiques de la chaîne de traitement et de permettre une bonne gestion lors de l'agrégation des opinions au niveau du tweet.

Enfin, les termes ont été entrés un à un dans la chaîne de traitement afin de s'assurer une dernière fois de l'efficacité de ceux-ci.

4.2.2 Ressources spécifiques à Twitter : lexique de hashtags

Comme décrit dans la section 2, les tweets peuvent contenir un ou plusieurs hashtags, s'intégrant ou non dans la phrase. Dans le cas général, nos choix d'analyse syntaxique nous permettent de repérer les hashtags composés du dièse et d'un terme dans nos lexiques d'opinion.

Nous n'avons par contre pas souhaité étendre le lexique d'opinion à des termes n'ayant pas d'existence autre en français qu'en tant que hashtag (e.g., #Onsenfou). Pour cela, nous avons donc développé un lexique complémentaire permettant d'étiqueter comme positif certains hashtags. Ce lexique a été peuplé à partir d'une extraction des hashtags dans le corpus d'entraînement.

5 Agrégation des opinions : de l'expression d'opinion à la polarité d'un tweet

Une fois réalisée la résolution de la polarité au niveau du segment, un ensemble de règles permet de fusionner les opinions au niveau de la phrase ou d'un tag XML identifié dans le flux d'entrée.

Par défaut, 3 niveaux sont proposés : le niveau de la proposition, de la phrase et du document global. Dans le contexte de cette évaluation, le niveau choisi est donc le document global, c'est à dire le tweet, comme spécifié dans le guide d'annotation.

Afin de remonter les informations de polarité et d'intensité aux niveaux demandés, nous nous sommes appuyés sur le modèle de calcul d'opinion parabolique, développé dans le cadre de la thèse de Baptiste Chardon. Cette section décrit succinctement le modèle parabolique : pour plus de détails sur ce modèle, cf. (Chardon et al., 2013), (Chardon, 2013).

Le principe général derrière le modèle parabolique est une projection d'une opinion sur une parabole suivant sa polarité et son intensité. La figure suivante représente la parabole de projection.

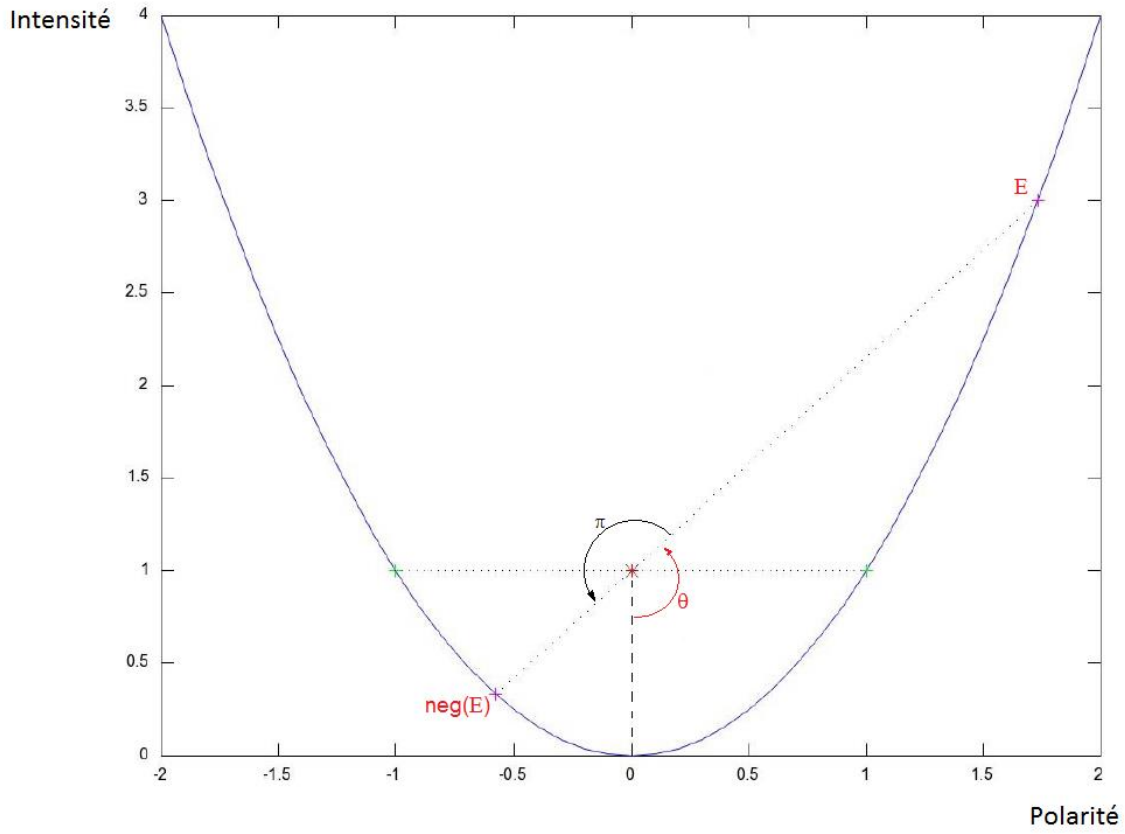

FIGURE 4. Exemple de parabole de projection

Le point E correspondant à une expression d'opinion est déterminé par son angle θ ; le signe de ce dernier est déterminé par la polarité de l'opinion, sa valeur absolue correspond à l'intensité de l'expression.

Par exemple, l'expression d'opinion "excellent" serait projetée au point E illustré sur la figure précédente.

Cette parabole associe un angle de π/2 aux opinions d'intensité standard, ce qui permet l'établissement de règles simples pour le calcul de l'effet des opérateurs.

5.1 Gestion des opérateurs

Notre lexique permet de reconnaître en contexte divers opérateurs s'appliquant sur l'opinion (négations, intensifications, opérateurs de modalité). Notre système prend en compte les opérateurs de négations et d'intensité, ainsi que l'effet de certaines modalités sur l'intensité de l'opinion.

5.1.1 Opérateurs de négation

L'opération de négation du modèle parabolique est basée sur deux hypothèses linguistiques, validées empiriquement dans les travaux (Chardon et al., 2013a).

- La négation renverse la polarité d'une expression d'opinion.
- La négation d'une expression d'intensité forte résulte en une expression d'intensité faible (par exemple, "excellent" étant d'intensité plutôt forte, "pas excellent" sera d'intensité plutôt faible).

Concrètement, l'opération consiste en un ajout de π à l'angle, comme illustré sur la FIGURE 4 par le point neg(E).

Dans la littérature, le modèle de négation par décalage de [Taboada et al., 2010] ne satisfait pas à la première hypothèse proposée, tandis que le modèle de renversement simple (entre autres, [Choi et Cardie, 2008]) ne satisfait pas à la seconde.

5.1.2 Opérateurs d'intensification

L'opération d'intensification du modèle parabolique est implémentée comme un décalage à atténuation progressive : l'effet des opérateurs est maximal aux abords de l'intensité standard ($\pi/2$), et t lorsque l'on se rapproche des extrema, afin de maintenir une cohérence dans les valeurs retournées. Ceci revient à dire qu'intensifier des opinions déjà très intenses apporte moins de sens qu'intensifier une opinion d'intensité standard.

Concrètement, deux fonctions remplissant ces conditions sont :

$$\begin{cases} \text{int}_+(\theta) = \begin{cases} |\theta|/\theta * 2 * \mu * |\theta| & \text{si } |\theta| <= \pi/3 \\ |\theta|/\theta * (\pi/2 + \tfrac{1}{2} * \mu * |\theta|) & \text{si } |\theta| >= \pi/3 \end{cases} \\ \text{int}_-(\theta) = \pi - \text{int}_+(\pi - \theta) \end{cases}$$

avec μ un coefficient correspondant à la force de décalage de l'opérateur (donné par lexique, égal à 1 en l'absence d'information).

Cette opération se rapproche de l'implémentation simple de la littérature (décalage de +1/-1), avec une gestion plus fine des cas extrémaux.

5.2 Agrégation des opinions au niveau du tweet

Enfin, l'étape finale consiste en la fusion de l'ensemble des expressions d'opinions modifiées par les opérateurs en une information pertinente au niveau demandé. Pour cela, nous nous reposons sur deux mécanismes principaux : un ensemble de règles se basant sur une analyse surfacique du discours, et une heuristique de fusion.

Les règles de fusion discursive se basent sur les travaux de (Chardon et al., 2013b) , qui décrivent un ensemble de règles permettant de fusionner deux opinions portées par deux segments de discours reliés par une relation SDRT (e.g. : conditionnelle, contraste). Dans le cas de notre chaîne de traitement, nous nous basons sur le repérage d'un nombre restreint d'opérateurs pour appliquer certaines de ces règles.

Si plusieurs opinions sont toujours dans le document une fois cette phase de traitement par règles effectuée, nous appliquons une heuristique numérique de fusion pour obtenir une polarité et une intensité au niveau du document. Cette heuristique favorise les opinions d'intensité tranchée dans le cas d'opinions portant la même polarité.

Dans le cas d'un document de la taille d'un tweet, repérer plusieurs expressions d'opinion dans le document est moins fréquent que sur un document non contraint. Sur l'ensemble des tweets récupérés en début de phase d'entraînement (7816), seuls 914 comportaient plusieurs expressions d'opinions repérées par la chaîne de traitement.

6 Résultats

Synapse Développement a participé à la campagne d'évaluation du 05 au 07 mai 2015.

6.1 Corpus d'apprentissage

La campagne d'évaluation DEFT 2015 propose de confronter les différents systèmes selon 3 tâches très distinctes :

- T1 : La classification des tweets selon leur polarité (positive, négative et neutre),
- T2 : La classification fine des tweets selon une classification de premier niveau comportant 4 classes génériques (information, opinion, sentiment et émotion) et 18 sous-classes liées au projet uComp (colère, peur, tristesse, dégoût, ennui, dérangement, déplaisir, surprise négative, apaisement, amour, plaisir, surprise positive, insatisfaction, satisfaction, accord, valorisation, désaccord, et dévalorisation).
- T3 : La détection de la source, de la cible et de l'expression d'opinion.

Pour notre part, nous avons choisi de n'évaluer notre système d'analyse de l'opinion que sur la première tâche de cette évaluation. En effet, ne disposant pas de la même classification sémantique des opinions, une correspondance n'aurait fait que dégrader les sorties du système. De plus, nous avons noté que l'intérêt principal des clients sur cette technologie réside essentiellement sur les notions de polarité. La catégorisation selon les classes sémantiques d'opinion est un résultat de recherche qui ne motive nos clients.

Notons qu'une analyse manuelle d'une sous-partie de ces tweets nous amène à penser que 15 à 20% de ces attributions de polarité sont, à tout le moins, discutables.

Prenons l'exemple du tweet identifié par le numéro 488730478061432832 :

Pluie de permis accordés par Ph. Henry co/ à Tinlot : 5 éoliennes entre Soheit et Terwagne http://t.co/FGMyrPzGud Pourquoi?

Ce tweet est classifié comme positif dans le corpus d'apprentissage, ce qui semble très douteux. De fait, les erreurs de ce type ont tendance à dégrader légèrement les résultats de la phase d'apprentissage. Les candidats à l'entrée au lexique de subjectivité de la chaîne de traitement étant revus manuellement, ils n'impactent pas la précision mais peuvent occasionner une légère baisse du rappel.

De même, certains tweets se répètent à l'identique dans le corpus et ne portent pas la même polarité. Une preuve de plus, s'il en fallait, de la grande difficulté d'attribution de l'opinion sur certains contenus, même par un humain. Il pourrait être intéressant de prendre en compte le taux d'accord inter-annotateurs lors de l'évaluation. Les erreurs sur les tweets à faible taux d'accord inter-annotateurs pourraient moins pénaliser un système que les tweets à polarité marquée et indiscutable.

Prenons l'exemple de la chaîne « Ségolène Royal: "Je ne veux pas que l'écologie soit punitive" », relevée pratiquement à l'identique dans 7 tweets, 6 annotés positivement, et 1 annoté négativement.

La chaîne initiale d'analyse d'opinion de Synapse, c'est-à-dire sans aucune adaptation au sujet abordé et au style de Twitter donnait un taux d'accuracy de 0,5658 et les résultats détaillés suivants :

Rappel *positif*	Précision *positif*	F-score *positif*	Rappel *négatif*	Précision *négatif*	F-score *négatif*	Rappel *neutre*	Précision *neutre*	F-score *neutre*
0,3111	0,5742	0,4036	0,3475	0,6666	0,4569	0,8565	0,5462	0,6671

TABLE 5 : Rappel, Précision et F-score avant adaptation au format tweet

Micro-précision	Macro-précision
0,564602661	0,595666667

TABLE 6 : Micro et Macro-précision avant adaptation au format tweet

Ces résultats, relativement décevants en baseline, s'expliquent du fait de la grande dépendance des annotations d'opinion, d'une part au contexte et aux relations de discours et d'autre part au sujet abordé.

L'intégration des principes d'adaptation à la tâche d'un point de vue de l'analyse syntaxique et de la prise en compte des spécificités du tweet (hashtag, smileys, patrons syntaxiques propres) ont permis d'atteindre un taux d'accuracy de 0,6632 et les résultats détaillés suivants :

Rappel *positif*	Précision *positif*	F-score *positif*	Rappel *négatif*	Précision *négatif*	F-score *négatif*	Rappel *neutre*	Précision *neutre*	F-score *neutre*
0,4823	0,6752	0,5627	0,5407	0,7816	0,6392	0,8526	0,6272	0,7227

TABLE 7 : Rappel, Précision et F-score après adaptation au format tweet

Micro-précision	Macro-précision
0,657098567	0,694666667

TABLE 8 : Micro et Macro-précision après adaptation au format tweet

L'ajout du lexique d'adjectifs coordonnés et de termes issus de l'analyse d'occurrence en objectif précision a encore amélioré les retours du système pour atteindre un taux d'accuracy de 0,6725 et les résultats détaillés suivants :

Rappel *positif*	Précision *positif*	F-score *positif*	Rappel *négatif*	Précision *négatif*	F-score *négatif*	Rappel *neutre*	Précision *neutre*	F-score *neutre*
0,4914	0,6817	0,5711	0,5683	0,7846	0,6592	0,8523	0,6371	0,7292

TABLE 9 : Rappel, Précision et F-score après ajout du lexique à visée précision

Micro-précision	Macro-précision
0,663961067	0,701133333

TABLE 10 : Micro et Macro-précision après ajout du lexique à visée précision

Enfin, l'ajout des termes issus de l'analyse d'occurrence en objectif rappel on amené le système à une accuracy de 0,6796 en fin d'apprentissage.

Rappel *positif*	Précision *positif*	F-score *positif*	Rappel *négatif*	Précision *négatif*	F-score *négatif*	Rappel *neutre*	Précision *neutre*	F-score *neutre*
0,4901	0,6867	0,572	0,6066	0,7838	0,6839	0,8489	0,6448	0,733

TABLE 11 : Rappel, Précision et F-score après ajout du lexique à visée rappel

Micro-précision	Macro-précision
0,668538408	0,7051

TABLE 12 : Micro et Macro-précision après ajout du lexique à visée rappel

6.2 Corpus de test

Le corpus de test a été relevé le 6 mai 2015 et analysé et soumis le 7 mai 2015. Les résultats ont été reçus le 14 mai 2015 et une phase d'adjudication a suivi, jusqu'au 18 mai 2015. Les résultats suite à cette adjudication ne sont pas en notre possession lors de la rédaction de cet article. Soulignons toutefois, que sur les 1088 différences entre l'évaluation et les sorties de l'analyse d'opinion de la chaîne TextAnalyst *by Synapse*, nous relevons 448 attributions de polarité contestables (accord inter-annotateurs 100% pour 2 annotateurs internes). Notons aussi que certains tweets sont redondants, tout comme dans le corpus d'apprentissage, ce qui impacte d'autant plus lorsque ceux-ci sont d'une polarité qui peut prêter à débat.

La synthèse des résultats de l'ensemble des participants (12 équipes ont proposé des soumissions, pour 22 équipes inscrites) est présentée dans le tableau suivant :

Moyenne	Médiane	Ecart-type	Min	Max
0.5819492417	0.6933297846	0.238071241	0.0408344543	0.7359865456

TABLE 13 : Présentation de résultats de l'ensemble des participants

Les résultats par classe de polarité de la chaîne d'extraction de Synapse :

- Polarité négative : 0.767255216693419

Vrai positifs : 478 Faux positifs : 145

- Polarité neutre : 0.647663071391885
  Vrai positifs : 1 261 Faux positifs : 686

- Polarité positive: 0.687268232385661
  Vrai positifs : 556 Faux positifs : 253

Micro-précision	Macro-précision
0.679195028114827	0.700728840156988

TABLE 14 : Micro et Macro-précision lors de la phase de test

Références

BENAMARA, F., CHARDON, B., MATHIEU, Y., POPESCU, V., ASHER, N. (2012). How do negation and modality impact on opinions?. Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics- Association for Computational Linguistics, 10-18.

BRUN, C., ROUX, C. (2014). Décomposition des «hash tags» pour l'amélioration de la classification en polarité des «tweets». Actes de TALN. 146.

CHARDON, B. (2013). Chaîne de traitement pour une approche discursive de l'analyse d'opinion (Doctoral dissertation, Université de Toulouse, Université Toulouse III-Paul Sabatier)

CHARDON, B., BENAMARA, F., MATHIEU, Y., POPESCU, V., ASHER, N. (2013). Measuring the effect of discourse structure on sentiment analysis. Actes de *Computational Linguistics and Intelligent Text Processing*, 25-37. Springer Berlin Heidelberg.

CHARDON, B., BENAMARA, F., MATHIEU, Y., POPESCU, V., ASHER, N. (2013). Sentiment composition using a parabolic model. Proceedings of the *10th International Conference on Computational Semantics (IWCS 2013)*. 47-58.

CHARDON, B., BENAMARA, F., POPESCU, V.(2012). Projet CASOAR : le discours pour l'analyse de l'opinion. Actes des *Journée ATALA Discours et TAL : des modèles linguistiques aux application*s.

CHOI, Y., CARDIE, C. (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. Actes de *Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics. 793-801.

GODARD, D. (2013). Les négateurs. La grande Grammaire du français. Éditions Actes Sud

JIANG, L., YU, M., ZHOU, M., LIU, X., ZHAO, T. (2011). Target-dependent twitter sentiment classification. Proceedings of the *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 151-160).

KOULOUMPIS, E., WILSON, T., MOORE, J. (2011). Twitter sentiment analysis: The good the bad and the omg!. *ICWSM*, 11, 538-541.

LAURENT, D., NÈGRE, S., SÉGUÉLA, P. (2009). L'analyseur syntaxique Cordial dans Passage. Actes de *TALN, 9*.

PAK, A., PAROUBEK, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *LREC*. Vol. 10, 1320-1326.

PRIYANTHAN, P., GOKULAKRISHNAN, B., RAGAVAN, T., PRASATH, N.,PERERA, A. S. (2012). Opinion mining and sentiment analysis on a twitter data stream. *ICTer 2012*.

TABOADA, M., VOLL, K., BROOKE, J. (2008). Extracting sentiment as a function of discourse structure and topicality. Simon Fraser Univeristy School of Computing Science Technical Report.

ZHOU, L., LI, B., GAO, W., WEI, Z., WONG, K. (2011). Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. Actes de *Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*.

Sentiment Detection Using PPM

Victoria Bobicev
Technical University of Moldova
victoria.bobicev@gmail.com

Résumé. **Détection de sentiments utilisant PPM**
Cet article rend compte de notre travail dans le DEFT 2015, Défi Fouille de Texte. Le sujet de ce défi était la fouille d'opinion, l'analyse des sentiments et la détection de l'émotion dans les tweets écrits en français. La tâche a été résolue par un système qui utilise le PPM (Prédiction par correspondance partielle), algorithme de compression basé sur un modèle n-gram statistique. Nous avons présenté deux points: l'algorithme de PPMC basé sur des caractères avec et sans normalisation. Les résultats des expériences avec l'algorithme PPMC basé sur les caractères étaient meilleurs que pour les expériences avec l'algorithme basé sur les mots. La méthode de normalisation appliquée dans le processus de classification afin de surmonter l'imbalanceness des données n'était pas appropriée dans ces conditions et n'a pas aidé à améliorer les résultats.

Abstract.
Sentiment Detection Using PPM.
This paper reports on our work in the DEFT 2015 French Text Mining Challenge. The topic of this challenge was opinion mining, sentiment analysis and emotion detection in tweets written in French. The task was solved by a system that used the PPM (Prediction by Partial Matching) compression algorithm based on an n-gram statistical model. We submitted two runs; character-based PPMC algorithm with normalization and without. The results in the experiments on character based PPMC algorithm were better than word-based. The normalisation method applied in the process of classification in order to overcome the imbalanceness of the data was not appropriate in this case and did not help in improving the results.

Mots-clés : fouille d'opinion, analyse de sentiments, détection d'émotion, Prédiction par correspondance partielle.
Keywords: Opinion Mining, Sentiment Analysis, Emotion Detection, Prediction by Partial Matching.

1 Introduction

The exponential growth of social media and publicly available user-generated content appealed for diversity of data mining tasks. The initial data mining task was the automatic or semi-automatic analysis of large quantities of data to extract some information about a topic and usually to populate a database with the extracted information for the further use. Lately the extraction and aggregation of factual information was supplemented with sentiment and opinion analysis.

Social web data has become the rich source of information about people's sentiments, opinions, preferences and moods and NLP community make every effort to obtain and analyse this information. The NLP challenges are the demonstrative indicator of these efforts. One of the first sentiment analysis task appeared in SemEval 2007, task 14 "Affective text"[1]. The objective of this task was to annotate the short text (news headlines) with the emotion label using a predefined list of emotions (e.g. joy, fear, surprise), and/or for polarity orientation (positive/negative). The next SemEval contained the task of disambiguation of sentiment ambiguous adjectives[2]. The sentiment ambiguous words are pervasive in many languages, the authors of the task wrote in their task description. They concentrated on Chinese, but suggested to use language-independent disambiguation techniques. Twitter was the source of sentiments in SemEval 2013 in the task 2[3]. The task's organisers aimed to promote the research which would lead to a better understanding of

[1] http://nlp.cs.swarthmore.edu/semeval/tasks/task14/summary.shtml
[2] http://semeval2.fbk.eu/semeval2.php?location=tasks#T3
[3] https://www.cs.york.ac.uk/semeval-2013/task2/

how sentiment is conveyed in tweets and texts. They proposed two sub-tasks: an expression-level task and a message-level task. SemEval 2015 continued with the similar task[4] "Sentiment Analysis in Twitter". This time the organisers proposed five sub-tasks: an expression-level task, a message-level task, a topic-related task, a trend task, and a task on prior polarity of terms. In addition to SemEval tasks various other sentiment analysis challenges were announced online, as for example GESTALT: GErman SenTiment AnaLysis shared Task[5]. It included two different tracks: task 1 on mining political debates, and task 2 on product reviews.

This paper reports participation in DEFT 2015[6] French Text Mining Challenge. We participated in two subtasks, Task 1 "Valence Classification of tweets" and task 2 "Fine-grained classification of the tweets". The paper is organised as follows: the next section introduces some related work; section 3 describes the used methodology; the experiments and their results are presented in section 4 which is followed by the discussion and conclusions.

2 Related Work

Text Data Mining intensively analysed sentiments and opinions that appear in consumer-written product reviews (Bisio et al., 2013), financial blogs and political discussions (Kim, Hovy, 2007). Text analysis of user-written online messages has been demanded by the need for such studies from the one hand and an easy access to the online data from the other (Chmiel et al., 2011), (Dodds et al., 2011).

Twitter is one of the most dynamic social nets with very fast reaction to various events. Recently, sentiment and opinion analysis in Twitter become the hot topic. Overcoming the difficulties of classical NLP analysis of tweets (Bontcheva et al., 2013) various applications of tweet sentiment analysis appear consta ntly (Chmiel et al., 2011), (Derczynski et al., 2013), (Hassan et al., 2013).

Sentiment lexicons are the largely used resources for sentiment analysis. Although there are already numerous lexicons with sentiment information such as SentiWordNet (Baccianella et al., 2010), MPQA (Wilson, 2008), SenticNet (Cambria, Hussain, 2012), DepecheMood (Staiano, Guerini, 2014) and others most of them are English; there is lack of similar resources for other languages.

However, the scarcest sources in the sentiment analysis domain are the annotated corpora. Although there were made some efforts to annotate various types of texts with various types of affective information this is still definitely not enough (Sabou et al., 2014). The early works in this domain were performed manually by the skilled linguists (Wiebe et al., 2005), (Boldrini et al., 2010), but this type of annotation was time and effort consuming. (Balahur and Steinberger 2009) discussed the problem of multiple annotators and inter-annotator agreement. They demonstrated that the elaborated annotation guidelines were necessary to obtain good inter-annotation agreement. They had to go through two iterations of annotation and to re-write the annotation guidelines on the base of the annotation errors made during the first iteration.

The later experiments used pre-annotated by the users corpora, as, for example, customers reviews marked with zero to five stars, or simply "thumbs up – thumbs down" (Turney 2002) or tweets with hashtags and emoticons indicating author's sentiment (Pak, Paroubek, 2010), (Kouloumpis et al., 2011). The other methods of sentiment annotation are Amazon Mechanical Turk (Narr et al., 2012) and games with purpose (Hong et al., 2013). The latter paper emphasized the fact that most sentiment resources have been created for English and describes creation of the language independent platform in the form of a game similar with tetris for online sentiment annotation of Korean words.

Although (Narr et al., 2012) created resources for four European languages: English, German, French and Portuguese using Amazon Mechanical Turk, (Hong et al., 2013) pointed out that such method is not socially attractive and well designed online game integrated with social networks such as Facebook and adapted to mobile devices are more appropriate tools for obtaining sentiment related lexical resources such as annotated corpora and lexicons. They also discussed in the conclusion that sometimes three classes of sentiment (positive, neutral, or negative) are not adequate to accurately capture the sentiment perceived by human judges. A partial solution of the problem is addition of granularity to sentiment classes such as sentiment scores in real numbers or even better, introduction of extra dimensions of sentiments as for example, 'anxiousness', 'anger', and 'inhibition'.

[4] http://alt.qcri.org/semeval2015/task10/
[5] https://sites.google.com/site/iggsasharedtask/
[6] https://deft.limsi.fr/2015/

3 Methodology Description

Detection sentiment and opinions in text can be viewed as a type of classification task. As it was discussed in the previous section such tasks are solved using machine learning methods. We used PPM in the sentiment classification experiments.

Class	Number of extracted tweets	Per cent		Number of lost tweets
positive	2435	31.2%		29
negative	1853	23.7%		41
neutral	3523	45.1%		48
total	7811	100%		118

TABLE 1 : The number and percent of tweets annotated as positive, negative and neutral for the task 1.

Class	Number of extracted tweets	Per cent		Number of lost tweets
information	3523	52.9%		48
opinion	2243	33.7%		32
sentiment	82	1.2%		0
emotion	809	12.2%		17
total	6657	100%		97

TABLE 2 : The number and percent of tweets annotated as : information, opinion, sentiment and emotion for the task 2.1.

3.1 Tasks Description

There are different types of classifications in the sentiment analysis domain. The paper describes the experiments for three classification tasks:

– Valence Classification of tweets. The aim of the task was to classify automatically the tweets depending on the opinion, sentiment or emotion expressed in the text: positive, negative, neutral or mixed, when the message held both positive and negative opinions, sentiments or emotions.
– Fine-grained classification of the tweets. The aim of this task was to assess the performance of textual opinion, sentiment and emotion detection system. It was divided into two sub-tasks:
  – Detection of one of the four proposed generic classes of the information expressed in the tweet. The generic classes proposed in this context were: INFORMATION, OPINION, SENTIMENT and EMOTION.
  – Detection of the specific class of the opinion/sentiment/emotion among 18 classes, as proposed in the uComp[7] project: COLÈRE (anger), PEUR (fear), TRISTESSE (sadness), DÉGOÛT (disgust), ENNUI (boredom), DÉRANGEMENT (disturbance), DÉPLAISIR (displeasure), SURPRISE NÉGATIVE (negative surprise),

[7] http://www.ucomp.eu/

APAISEMENT (appeasement), AMOUR (love), PLAISIR (pleasure), SURPRISE POSITIVE (positive surprise), INSATISFACTION (dissatisfaction), SATISFACTION (satisfaction), ACCORD (agreement), VALORISATION (valorization), DÉSACCORD (disagreement) and DÉVALORISATION (devalorization).

Class	Number of extracted tweets	Per cent		Number of lost tweets
valorisation	1487	47.4%		17
devalorisation	393	12.5%		8
peur	269	8.6%		5
desaccord	212	6.8%		4
colere	205	6.5%		5
mepris	173	5.5%		3
accord	151	4.8%		3
satisfaction	73	2.3%		0
deplaisir	47	1.5%		0
tristesse	34	1.1%		2
plaisir	34	1.1%		1
derangement	12	0.4%		1
surprise_negative	10	0.3%		0
apaisement	9	0.3%		0
insatisfaction	9	0.3%		0
amour	8	0.3%		0
ennui	4	0.1%		0
surprise_positive	4	0.1%		0
total	3134	100%		49

TABLE 3 : The number and percent of tweets annotated with 18 classes of sentiments for the task 2.2.

3.2 The Data Description

A set of annotated French tweets provided by DEFT 2015 organisers was used in the experiments. In agreement with Twitter access and usage policy, the organisers only provided tweet identifiers and a toolkit to collect the data from Twitter. Total of 7929 tweets id was provided by the organisers but only 7811 tweets were extracted due to the fact that some authors deleted their tweets after their extraction for the annotation. The class distribution for the first task for the extracted tweets is presented in the table 1.

While for the task 1 annotation was provided for all 7929 tweet id, only 6754 id were annotated for the task 2.1. After extraction 6657 tweets were obtained for the task 2.1. Table 2 presents the statistics for the tweet annotation for this task.

The class information from the task 2.1 was not included in the task 2.2. Thus for the task 2.2 only 3183 tweet id with annotation was provided. We collected 3134 annotated tweets fir this task. The distribution of sentiment annotation for these tweets is reflected in the table 3.

3.3 The Algorithm Description

In this paper, the application of the PPM (Prediction by Partial Matching) model for automatic text classification is explored. Prediction by partial matching (PPM) is an adaptive finite-context method for text compression that is a back-off smoothing technique for finite-order Markov models (Bratko et al., 2006). It obtains all information from the original data, without feature engineering, it is easy to implement and relatively fast. PPM produces a language model and can be used in a probabilistic text classifier.

PPM is based on conditional probabilities of the upcoming symbol given several previous symbols (Cleary and Witten, 1984). The PPM technique uses character context models to build an overall probability distribution for predicting upcoming characters in the text. A blending strategy for combining context predictions is to assign a weight to each context model, and then calculate the weighted sum of the probabilities:

$$P(x) = \sum_{i=1}^{m} \lambda_i \, p_i(x), \qquad (1)$$

where

λ_i and p_i are weights and probabilities assigned to each order i $(i=1...m)$.

For example, the probability of character '***m***' in context of the word '***algorithm***' is calculated as a sum of conditional probabilities dependent on different context lengths up to the limited maximal length:

$$P_{PPM}(\text{'}\boldsymbol{m}\text{'}) = \lambda_5 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{orith}\text{'}) + \lambda_4 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{rith}\text{'}) + \lambda_3 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{ith}\text{'}) +$$
$$+ \lambda_2 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{th}\text{'}) + \lambda_1 \cdot P(\text{'}\boldsymbol{m}\text{'} \mid \text{'}\boldsymbol{h}\text{'}) + + \lambda_0 \cdot P(\text{'}\boldsymbol{m}\text{'}) + \lambda_{-1} \cdot P(\text{'esc'}), \qquad (2)$$

where

λ_i (i = 1…5) is the normalization weight;

5 is the maximal length of the context;
P('esc') is so called 'escape' probability, the probability of an unknown character.

PPM is a special case of the general blending strategy. The *PPM* models use an escape mechanism to combine the predictions of all character contexts of length *m*, where *m* is the maximum model order; the order 0 model predicts symbols based on their unconditioned probabilities, the default order -1 model ensures that a finite probability (however small) is assigned to all possible symbols. The *PPM* escape mechanism is more practical to implement than weighted blending. There are several versions of the *PPM* algorithm depending on the way the escape probability is estimated. In our implementation, we used the escape method C (Bell et al., 1989), named PPMC. Treating a text as a string of characters, a character-based *PPM* avoids defining word boundaries; it deals with different types of documents in a

uniform way. It can work with texts in any language and be applied to diverse types of classification; more details can be found in (Bobicev, 2007). Our utility function for text classification was cross-entropy of the test document:

$$H_d{}^m = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i), \qquad (3)$$

where
n is the number of symbols in a text d,
$H_d{}^m$ – entropy of the text d obtained by model m,
$p^m(x_i)$ is a probability of a symbol x_i in the text d.
$H_d{}^m$ was estimated by the modelling part of the compression algorithm.

Usually, the cross-entropy is greater than the entropy, because the probabilities of symbols in diverse texts are different. The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. Hence, if several statistical models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the basis of each model, the lowest value of cross-entropy will indicate the class of the unknown text. In this way cross-entropy is used for text classification.

On the training step, we created *PPM* models for each class of documents; on the testing step, we evaluated cross-entropy of previously unseen texts using models for each class. Thus, cross-entropy was used as similarity metrics; the lowest value of cross-entropy indicated the class of the unknown text.

The maximal length of a context equal to 5 in PPM model was proven to be optimal for text compression (Teahan, 1998). In all our experiments with character-based PPM model we used maximal length of a context equal to 5; thus our method is PPMC5.

The character-based *PPM* models were used for spam detection, source-based text classification and classification of multi-modal data streams that included texts. In (Bratko et al., 2006), the character-based PPM models were used for spam detection. In (Bobicev, 2007), the PPM algorithm was applied to text categorization in two ways: on the basis of characters and on the basis of words.

In (Teahan et al., 2000), a PPM-based text model and minimum cross-entropy as a text classifier were used for various tasks; one of them was an author detection task for the well known Federalist Papers[8]. In (Bobicev, Sokolova, 2008), the PPM algorithm was applied to the short text categorization. Character-based model performed almost as well as SVM, the best method among several machine learning methods compared in (Debole, Sebastiani 2004) for the Reuters-21578[9] corpus.

Usually, PPM models are character-based. However, word-based models were also used for various purposes. For example, if texts are classified by the contents, they are better characterized by words and word combinations than by fragments consisting of five letters. For some tasks words can be more indicative text features than character sequences. That's why we decided to use both character-based and word-based models for PPM text classification. In the case of word-based PPM, the context is only one word and an example for formula (1) looks like the following:

$$P_{PPM}(\text{word}_i) = \lambda_1 \cdot P(\text{word}_i \mid \text{word}_{i-1}) + \lambda_0 \cdot P(\text{word}_i) + \lambda_{-1} \cdot P(\text{'esc'}), \qquad (4)$$

where
word_i is the current word;
word_{i-1} is the previous word.

This model is coded as PPMC1 because of the same C escape method and one length context used for probability estimation.

[8] The Federalist Papers by Alexander Hamilton, James Madison, John Jay, Digireads Publishing, Neeland Media LLC, 2006.
[9] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Training and testing data is distributed quite unevenly in many tasks, for example, in Reuters-21578 corpus. This imbalance drastically affected the results of the classification experiments; the classification was biased towards classes with a larger volume of data for training. Such imbalance class distribution problems were mentioned in (Bobicev, Sokolova, 2008), (Stamatatos, 2009), (Narayanan et al., 2012). Considering the fact that imbalanced data affected classification results in such a substantial way we used a normalization procedure for balancing entropies of the statistical data models.

The first step of our algorithm was training. In the process of training, statistical models for each class of texts were created. This meant that probabilities of text elements were estimated. The next step after training was calculation of entropies of test documents on the basis of each class model. We obtained a matrix of entropies of class statistical models x test documents'. The columns were entropies for the class statistical models and rows were entropies for a given test documents. After this step the normalization procedure was applied. The procedure consisted of several steps:
(1) Mean entropy for each class of texts was calculated on the base of the matrix;
(2) Each value in the matrix was divided by the mean entropy for this class. Thereby we obtained more balanced values and classification improved considerably.

Although the application of PPM model to the document classification is not new, PPM was never applied to the task of sentiment analysis.

In order to evaluate the PPM classification method for sentiment analysis in French tweets a number of experiments were performed. The aim of the experiments was twofold:
- to evaluate the quality of PPM-based sentiment classification;
- to compare letter-based and word-based PPM classification.

4 The Experiments

The experiments were carried out during the DEFT 2015 shared task event. The first set of the experiments was performed on the base of training data released by the organisers in February. The second set consisted of evaluation runs on test data released in May and the results for these experiments were provided by the organizers.

4.1 The First Set of the Experiments

The first set of the experiments consisted in solving task 1, task 2.1 and task 2.2 of the DEFT challenge using PPM classification algorithm. We used two modification of the algorithm: on the base of characters and on the base of words. Taking into consideration the imbalanced class distribution we used normalization procedure. 10-fold cross-validation was used in order to evaluate the performance of the method in case of the task 1 and task 2.1. We used 4 fold cross-validation for the task 2.2 as some of the 18 classes were presented only with 4 tweets (see table 3). Thus, for these classes 3 files were used for training and 1 for test in each run. The results for the task 1 are reflected in the table 4.

Method	**Precision**	**Recall**	**Macroaverage F-score**
Character-based PPMC5 method without normalization	0.58	0.56	0.57
Character-based PPMC5 method with normalization	0.56	0.58	0.56
Word-based PPMC1 method without normalization	0.50	0.52	0.51
Word-based PPMC1 method with normalization	0.50	0.52	0.51

TABLE 4 : The results obtained on character-based and letter-based PPM models with and without normalization for the task 1.

Method	**Precision**	**Recall**	**Macroaverage F-score**
Character-based PPMC5 method without normalization	0.47	0.40	0.43
Character-based PPMC5 method with normalization	0.42	0.42	0.42
Word-based PPMC1 method without normalization	0.47	0.36	0.41
Word-based PPMC1 method with normalization	0.39	0.43	0.41

TABLE 5: The results obtained on character-based and letter-based PPM models with and without normalization for the task 2.1.

Method	**Precision**	**Recall**	**Macroaverage F-score**
Character-based PPMC5 method without normalization	0.23	0.16	0.18
Character-based PPMC5 method with normalization	0.16	0.20	0.18
Word-based PPMC1 method without normalization	0.26	0.14	0.17
Word-based PPMC1 method with normalization	0.13	0.16	0.14

TABLE 6: The results obtained on character-based and letter-based PPM models with and without normalization for the task 2.2.

As it is seen from the tables, the overall results are not very high which indicate that PPM method is not suitable for the sentiment analysis task. We expected word-based method to perform better as it works with words, the units which sentiments were represented in. However this presupposition was also wrong. The character-based method gave better results in all experiments. The possible reason could be that word-based method was loosing all special characters (such as emoticons) which were registered and used by character-based method. It should be noted that normalization did not improve the results as it was expected. It even made them worse for word-based method. In previous cases it helped to improve the results (Bobicev et al., 2013).

4.2 The Second Set of the Experiments

The second set consisted of evaluation runs on test data released in May and the results for these experiments were provided by the organisers. Taking into consideration that word-based method was worse for all tasks and we were allowed to submit no more than 3 experiment runs for each task we decided to submit only two runs of character based method (with normalization and without it) for each task. Thus, we submitted six runs, two runs for task1, two runs for task 2.1 and two runs for task 2.2. The organisers were interested in Precision, thus only this metric was reported. Tables 7, 8 and 9 contain the results reported by the organisers for the task 1, 2.1 and 2.2.

Method	**Micro precision**	**Macro precision**
Character-based PPMC5 method without normalization	0.568	0.558
Character-based PPMC5 method with normalization	0.542	0.547

TABLE 7: The results obtained on character-based PPM model with and without normalization for the task 1.

Method	**Micro precision**	**Macro precision**
Character-based PPMC5 method without normalization	0.495	0.383
Character-based PPMC5 method with normalization	0.376	0.382

TABLE 8: The results obtained on character-based PPM model with and without normalization for the task 2.1.

Method	**Micro precision**	**Macro precision**
Character-based PPMC5 method without normalization	0.478	0.226
Character-based PPMC5 method with normalization	0.289	0.175

TABLE 9: The results obtained on character-based PPM model with and without normalization for the task 2.2.

It is seen from the tables that the results are similar with the results for the first set of the experiments. The normalisation did not help although the data was imbalanced, especially in the task 2.2 where the results were the worse.

5 Discussion and Conclusions

The paper reports on our work in the DEFT 2015 French Text Mining Challenge. Three tasks of tweet sentiment analysis were proposed in the framework of this challenge. All three tasks analysed French tweets about politics and elections in France. We participated in task1: "Valence Classification of tweets" in which the tweets were classified in three classes: (1) positive, (2) negative, (3) neutral and mixed. We also participated in task 2: "Fine-grained classification of the tweets" which consisted of two subtasks. Task 2.1: "Detection of the generic class of the information expressed in the tweet" classified tweets in four classes: information, opinion, sentiment and emotion. Task 2.2: "Detection of the specific class of the opinion/sentiment/emotion" aimed at detecting the class of the opinion, sentiment or emotion among 18 classes.

We used the system that used the PPM (Prediction by Partial Matching) compression algorithm based on character n-gram statistical model for all tasks. We submitted two runs; character-based PPMC algorithm with normalization and without for each of the subtask. We supposed that word-based algorithm would be better in sentiment detection but the experiments demonstrated that this presupposition was wrong. The results of the experiments on character based PPMC algorithm were better than the results of the experiments on word-based PPMC algorithm for all experiments.

The data released for the tasks was very imbalanced as it is seen in the tables 1, 2, and 3. Such situation is quite common in real classification tasks. Working with imbalanced data we developed a normalisation procedure described in the paper but our normalisation method applied in the process of classification in order to overcome the imbalanceness of the data was not appropriate in this case and did not help in improving the results.

References

BACCIANELLA S., ESULI A., SEBASTIANI F. (2010). SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 7th Conference on International Language Resources and Evaluation*, 2200-2204.

BALAHUR, A. STEINBERGER R (009) Rethinking Sentiment Analysis in the News: from Theory to Practice and back. *Proceedings of the 1st Workshop on Opionion Mining and Sentiment Analysis*.

BELL, T.C., WITTEN, I.H., CLEARY, J. G. (1989). Modeling for text compression. *Computing Surveys*, 21(4), pp. 557-591.

BISIO F., GASTALDO P., PERETTI C., ZUNINO R., CAMBRIA E.(2013) Data intensive review mining for sentiment classification across heterogeneous domains. *ASONAM 2013*.

BOBICEV V. (2007). Comparison of Word-based and Letter-based Text Classification. *Recent Advances in Natural Language Processing V*, pp. 76–80.

BOBICEV V., SOKOLOVA M. (2008). An effective and robust method for short text classification. *Proceedings of the 23rd national conference on Artificial intelligence* - Volume 3, pp. 1444–1445.

BOBICEV V., SOKOLOVA M., EMAM K., JAFER Y., DEWAR B., JONKER E., MATWIN S. (2013). Can Anonymous Posters on Medical Forums be Reidentified? *Journal of Medical Internet Research,* 15(10):e215.

BOLDRINI E., BALAHUR A., MARTÍNEZ-BARCO P., MONTOYO A. (2010). EmotiBlog: a fine-grained annotation schema for labelling subjectivity in the new-textual genres born with the Web 2.0. *Procesamiento del Lenguaje Natural* 45.

BONTCHEVA K., DERCZYNSKI L., FUNK A., GREENWOOD M.A., MAYNARD D., ASWANI N. (2013). TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*.

BRATKO A., CORMACK G. V., FILIPIC B., LYNAM T. R., ZUPAN B. (2006). Spam filtering using statistical data compression models, *Journal of Machine Learning Research* 7:2673–2698.

CAMBRIA, E., HUSSAIN A. (2012). *Sentic Computing: Techniques, Tools, and Applications*. Springer.

CHMIEL A., SIENKIEWICZ J., THELWALL M., PALTOGLOU G., BUCKLEY K., KAPPAS A., HOLYST J. (2011). Collective Emotions Online and Their Influence on Community. *Life PLoS one*.

CLEARY J., WITTEN I. (1984). Data compression using adaptive coding and partial string matching, *IEEE Trans. Commun.* 32(4):396–402.

DEBOLE F., SEBASTIANI F. (2004). An Analysis of the Relative Hardness of Reuters-21578 Subsets. *Journal of the American Society For Information Science And Technology* 56(6):971–974. DOI: 10.1002/asi.v56:6.

DERCZYNSKI L., YANG B., JENSEN C.S. (2013). Towards Context-Aware Search and Analysis on Social Media Data. *In Proceedings of the Extending Database Technology conference (EDBT 2013)*.

DODDS, P., HARRIS K., KLOUMANN I., BLISS C., DANFORTH C. (2011). Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter. *PLoS ONE,* 6, e26752.

HASSAN S., FERNANDEZ M., HE Y., ALANI H. (2013). Evaluation Datasets for Twitter Sentiment Analysis. A survey and a new dataset, *the STS-Gold. first ESSEM workshop*.

HONG Y., KWAK H., BAEK Y., MOON S. (2013). Tower of Babel: A Crowdsourcing Game Building Sentiment Lexicons for Resource-scarce Languages. *WWW '13 Companion Proceedings of the 22nd international conference on World Wide Web companion*. Pages 549-556.

KIM, S.-M., HOVY E. (2007). Crystal: Analyzing predictive opinions on the web. *EMNLP-CoNLL*.

KOULOUMPIS E., WILSON T., MOORE J. D. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG!. *Proceedings of the Fifth International Conference on Weblogs and Social Media*. AAAI Press, p. 538-541.

NARAYANAN A., PASKOV H., GONG N. Z., BETHENCOURT J., STEFANOV E., SHIN E. C. R., AND SONG D. (2012). On the Feasibility of Internet-Scale Author Identification, *in 2012 IEEE Symposium on Security and Privacy (SP)*, pp. 300 – 314.

NARR S., HULFENHAUS M., ALBAYRAK S. (2012). Language-Independent Twitter Sentiment Analysis. *Knowledge Discovery and Machine Learning (KDML)*, LWA, 12-14.

PAK A., PAROUBEK P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *LREC 2010.*

SABOU M., BONTCHEVA K., DERCZYNSKI L., SCHARL A. (2014). Corpus Annotation through Crowdsourcing: Towards Best Practice Guidelines. *9th Language Resources and Evaluation Conference (LREC-2014)*.

STAIANO J., GUERINI M. (2014). DepecheMood: a Lexicon for Emotion Analysis from Crowd-Annotated News. *Proceedings of ACL-2014*.

STAMATATOS E. (2009). A survey of modern authorship attribution methods, *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556.

TEAHAN W. (1998). Modelling English text. *PhD Thesis,* University of Waikato, New Zealand.

TEAHAN W. J., MCNAB R., WEN Y., WITTEN I. H. (2000). A compression-based algorithm for Chinese word segmentation, *Comput. Linguist.*, vol. 26, no. 3, pp. 375–393.

TURNEY P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Pages 417-424.

WIEBE J., WILSON T., CARDIE C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, volume 39, issue 2-3, pp. 165-210.

WILSON T. (2008). Fine-Grained Subjectivity Analysis. PhD Dissertation, *Intelligent Systems Program, University of Pittsburgh*.

Exploitation de lexiques pour la catégorisation fine d'émotions, de sentiments et d'opinions

Nicolas Hernandez[1] Grégoire Jadi[1] Joseph Lark[1,2] Laura Monceaux[1]
(1) LINA, UMR CNRS 4261, 2 chemin de la Houssinière, Nantes, France
(2) Dictanova, 2 chemin de la Houssinière, Nantes, France
{Prénom.Nom}@univ-nantes.fr

Résumé. Nous présentons dans cet article notre proposition pour la 11ème édition du Défi Fouille de Textes (DEFT). Nous participons à trois tâches proposées dans le cadre de cet atelier en fouille d'opinion. Les objectifs de ces tâches sont de classer des tweets en français sur le sujet des énergies renouvelables, respectivement du point de vue de la polarité, du type général d'information énoncé, et enfin de la classe fine du sentiment, de l'émotion ou de l'opinion exprimée. Pour réaliser cette catégorisation, nous proposons d'explorer et d'évaluer différentes méthodes de construction de lexiques typés sémantiquement : outre des lexiques affectifs construits manuellement, nous expérimentons des lexiques typés construits semi-automatiquement sur le corpus d'évaluation et d'autres sur un corpus tiers.

Abstract.
Using affective lexicons for fine-grained sentiment, opinion and emotion analysis
In this article, we present our contribution to the 11th DEFT workshop (Défi Fouille de Textes). We take part in three tasks proposed in this opinion mining challenge. The goal of these tasks is to analyse a corpus of french tweets about renewable energy, through the inference of their polarity, general semantic class, and fine-grained sentiment class respectively. Our proposition makes use of a machine learning process that combines various ways of building semantically classified lexicons. We explore the use of external lexicons, semi-supervisely extracted lexicons from the training corpus, and semi-supervisely extracted lexicons from a third-party corpus.

Mots-clés : Fouille d'opinion, expression d'émotions, analyse de sentiments, construction de lexique, classification fine.

Keywords: Sentiment analysis, opinion mining, fine-grained emotion classification, lexicon acquisition.

1 Introduction

Avec l'expansion du web social, les internautes sont de plus en plus enclins à partager leurs avis sur les réseaux sociaux ou les sites spécialisés. Le domaine de la fouille d'opinion vise à désambiguïser automatiquement ces informations en ce qui concerne le sentiment exprimé, en le traduisant par une valence affective (polarité "positive" ou "négative", score de subjectivité...) ou par une catégorie de sentiment. C'est dans cette optique que notre travail s'inscrit, répondant ainsi à la tâche de classification fine de tweets (tâche 2.2) proposée pour la compétition DEFT 2015. Nous avons inféré les informations plus générales (tâches 1 et 2.1) depuis nos résultats en catégorisation fine.

La classification demandée compte 18 catégories sémantiques, à laquelle s'ajoute une classe "neutre" correspondant à l'énoncé d'une information objective. Ces catégories détaillées impliquent des variations relativement sensibles entre les classes pouvant entraîner des ambiguïtés, ce qui constitue la difficulté majeure de cette tâche. En particulier, la résolution de ces ambiguïtés peut être complexe dans le cas où deux catégories sémantiquement proches ne sont pas représentées de façon équilibrée, car il peut exister un biais en faveur de la classe la plus présente. Afin de catégoriser les tweets, nous avons, tout d'abord, utilisé une représentation en sac de bigrammes de mots. Nous avons par la suite amélioré ces premiers résultats au moyen (1) de lexiques construits manuellement, (2) de l'acquisition semi-automatique de mots sémantiquement liés aux catégories définies au sein d'un corpus externe ou (3) au sein du corpus d'entraînement.

Dans la suite de cet article, nous dressons un bref état de l'art des méthodes liées aux problématiques soulevées par ce défi (section 2), puis nous exposons notre démarche (section 3) ainsi que les détails des méthodes utilisées (section 4). Ensuite, nos présentons les résultats observés sur le corpus fourni (section 5). Enfin, nous commentons ces résultats, et le travail effectué lors de cette participation en général (section 6).

2 Travaux connexes

Ce travail est selon nous fortement lié à la détection de subjectivité : il s'agit en effet de déterminer dans quelle mesure l'information présente dans un tweet renvoie à un sentiment ou une émotion. Nous considérons dans ce contexte qu'une première différenciation peut être faite entre la classe neutre du point de vue de la subjectivité ("Information") et toutes les autres. Ce type de différenciation fait l'objet de plusieurs travaux dans la littérature (Wiebe & Mihalcea, 2006; Murray & Carenini, 2011). Cependant la notion de subjectivité peut recouvrir plusieurs concepts. Liu (2012) distingue ainsi les expressions d'un désir, d'une opinion, d'une croyance, d'une spéculation... et montre que l'on peut être amené à confondre la subjectivité d'une phrase et le fait qu'elle exprime un sentiment ou une opinion. C'est sur ce plan que l'on peut distinguer par exemple un jugement rationnel d'une opinion passionnée. À ces modalités de l'expression d'un sentiment s'ajoutent les différentes émotions humaines. Parrott (2001) identifie six émotions primaires que sont la joie, l'amour, la surprise, la colère, la tristesse et la peur. L'ensemble de ces catégories d'expression sont représentées par les classes que nous cherchons à identifier ici. Les travaux réalisant une identification similaire reposent pour la plupart sur des lexiques affectifs. Dans cette optique, Staiano & Guerini (2014) exploitent un corpus journalistique annoté par les internautes selon l'émotion suscitée par chaque nouvelle afin d'en extraire un lexique d'émotion de près de 37 000 mots. Yang *et al.* (2014) utilisent une version de l'allocation de Dirichlet latente (LDA) pour rechercher des mots sémantiquement proches de graines définies manuellement exprimant une émotion. En français, Vernier *et al.* (2009) proposent une méthode d'apprentissage automatique des structures caractéristiques d'une évaluation dans le texte.

3 Approche globale

Nous décrivons ici notre position sur les différentes tâches et notre approche pour intégrer les différents traits observés.

3.1 Appréhension des différentes tâches

Nous avons considéré la tâche 2.2 (identification de la classe spécifique de l'opinion, sentiment ou émotion d'un tweet donné) comme une spécialisation de la tâche 2.1 (identification de la classe générique de l'information exprimée dans le tweet), et celle-ci comme une spécialisation de la tâche 1 (classification des tweets selon leur polarité). En consacrant nos efforts sur la tâche 2.2, nous avions ainsi par différents degrés de généralisation la possibilité d'obtenir des résultats pour les tâches moins spécifiques.

3.2 Intégration des différents traits par apprentissage

Comme indiqué dans (Pustejovsky & Stubbs, 2012), les modèles génératifs de type bayésien naïf (NB) ou discriminatifs de type machines à vecteurs de support (SVM) ou d'entropie maximale sont connus pour être mieux adaptés que d'autres sur des tâches de classification d'énoncés selon un jeu de catégories.

La disponibilité d'un corpus d'entrainement, la taille des données de test et la multiplicité des classes à reconnaître nous a conduit vers la voie de l'apprentissage supervisé. Nous n'étions néanmoins pas réfractaires à la mise en place de post-traitements à base de règles pour corriger des biais identifiés.

Du fait de l'hétérogénité des données (déséquilibre des classes, faible représentativité de certaines), et au regard du nombre d'instances et de traits (de quelques centaines à plusieurs milliers) que nous souhaitions considérer (possiblement quelques milliers chacun), nous avons opté pour l'utilisation d'un classifieur linéaire de type SVM comme conseillé dans la littérature (Hsu *et al.*, 2003). Ce type de classifieur offre la possibilité de manipuler plusieurs milliers de traits et d'obtenir des modèles parmi les plus performants en seulement quelques secondes [1]. Nous avons utilisé en particulier l'implémentation offerte par (Yu *et al.*, 2013) qui exploite sur la bibliothèque `LIBLINEAR` [2] (Fan *et al.*, 2008). Nous avons opté pour une représentation binaire des traits, une normalisation des instances et un classifieur multi-classes (Crammer & Singer, 2000). Ce paramétrage correspond au mode par défaut ; celui-ci produisait les meilleurs résultats sur nos données. Alors qu'une représentation des traits en nombre d'occurrences était légèrement moins bon la représentation en fonction de la

1. En comparaison, un NB prend quelques minutes et un arbre de décision type C4.5 plusieurs heures avec un corpus comptant approximativemet 6 000 instances décrites chacunes par environ 80 000 traits.

2. `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`

fréquence des termes ou de la fréquence des termes plus l'inverse de la fréquence documentaire, donnaient de moins bons résultats sur les données d'entraînement. Le peu de matériel lexical des énoncés à annoter et la taille modeste du corpus (voire très petite pour certaines classes) expliquent l'inadéquation de ces représentations à notre tâche.

Pour estimer les paramètres optimaux (notamment le coût de la pénalité C), nous avons procédé par une recherche "par quadrillage" qui consiste à tester différentes valeurs (incrémentées exponentiellement) et à sélectionner celles qui donnent les meilleurs résultats par validation croisée sur cinq strates. Un autre argument pour motiver notre choix pour un SVM venait du fait que n'ayant aucune information sur la distribution des classes dans le corpus de test, nous souhaitions un classifieur robuste à une distribution différente. Certains classifieurs comme le NB prend en compte les probabilités d'occurrence observées dans le corpus d'entraînement. Nous craignions qu'avec un tel classifieur les classes prédites soient seulement les dominantes de notre corpus d'entraînement.

4 Descriptions des différents traits considérés

Dans cette section, nous décrivons notre approche de base et les différentes approches reposant sur des lexiques existants, des lexiques construits sur le corpus d'entraînement (endogènes)et d'autres sur un corpus tiers (exogènes).

4.1 Un modèle de bigrammes de tokens mots

Nous choisissons une modélisation en bigrammes de tokens mots avec une représentation binaire comme approche de base. Sur le corpus d'entraînement, cela correspond à 86 018 traits distincts pour 21 295 tokens uniques. Ces bigrammes de tokens mots ont été calculés sans normalisation du texte (surface brute, aucune tokenization ni racinisation).

4.2 Construction et utilisation de lexiques exogènes

Nous avons projeté au corpus d'entraînement trois lexiques construits manuellement à partir de ressources externes : le lexique des affects *LIDILEM* de (Augustyn *et al.*, 2006), une traduction du lexique *ANEW* de (Bradley & Lang, 1999) et un lexique d'émoticônes.

Le lexique *LIDILEM* modélise « le ressenti ou l'attitude du narrateur et/ou de ses personnages [qui] sont caractérisés lexicalement, comme relevant de la joie, de la tristesse, etc. » (Augustyn *et al.*, 2006). Ce lexique est décomposé en trois parties : les verbes, les adjectifs, les noms. Le lexique *ANEW* est constitué de 2476 mots évalué selon trois critères : plaisir, excitation et dominance. Pour chaque critère de toutes les entrées du lexique, les auteurs ont associé un score correspondant à la moyenne et à l'écart type. Le lexique des émoticônes a été construit manuellement à partir de la liste des émoticônes de Wikipedia [3]. De cette liste, nous n'avons gardé que 40 classes pour un total de 218 émoticônes. Nous avons ignoré les classes dont les émoticônes n'apparaissaient pas dans un corpus de plus de 7,5 millions de tweets français [4]. Cette approche compile 25 traits.

4.3 Construction semi-automatique de lexiques spécialisés à partir du corpus fourni

Dans la mesure où l'expression d'une opinion ou d'un sentiment dépend fortement de son contexte, nous nous sommes intéressés à l'acquisition d'éléments lexicaux caractéristiques de ces expressions au sein même du corpus fourni.

Cette extraction est réalisée en 3 étapes. Tout d'abord, nous entraînons un modèle de classification SVM sur le voisinage des mots du corpus afin de reconnaître les marqueurs d'opinion des autres mots. Dans le modèle appris, les exemples positifs sont les contextes entourant un marqueur connu, dans un lexique de marqueurs d'opinion peu dépendants du domaine (*horrible*, *catastrophique*, *joyeux*, etc.). Cette liste initiale contient environ 200 mots de ce type. Les traits de classification caractérisant un contexte appris sont :

- la forme lemmatisée et le rôle grammatical des mots entourant le candidat dans une fenêtre de 7 mots
- un marqueur indiquant si le contexte contient une négation (*ne, pas, jamais*, etc.)
- un marqueur indiquant si le contexte contient un déictique (*je, me, franchement*, etc.)

3. `https://en.wikipedia.org/wiki/List_of_emoticons`
4. Ce corpus représente une semaine de collecte à partir du 26 mars 2015

Le lexique extrait est ensuite filtré manuellement, afin de réduire le bruit dû aux erreurs de prétraitement ou aux ambiguïtés inhérentes à l'aspect subjectif de cette classification. Enfin nous recherchons les formes dérivées des mots acquis (flexions, variations grammaticales, ajout ou suppression d'affixes) pour les ajouter à notre ressource.

Nous avons réalisé cette extraction pour chaque catégorie sémantique, produisant ainsi 18 lexiques dont la taille varie fortement en fonction de la taille en nombre de tweets de la catégorie (de 4 à 86 mots). Cette approche compile 185 traits.

4.4 Des lexiques typés construits automatiquement à partir d'amorces et sur un corpus tiers

Approche Une de nos contributions réside dans la proposition d'une approche pour construire un lexique conséquent représentatif de classes sémantiques à reconnaître à partir d'amorces lexicales manuellement définies. Le principe de cette approche consiste dans un premier temps à définir des *graines*, instances des classes à reconnaître, puis dans un second temps à rechercher itérativement des variantes de celles-ci dans un corpus d'apprentissage et à les rajouter aux graines déjà récoltées avant de rechercher de nouvelles variantes de celles-ci. Le processus itératif peut se terminer après convergence (quand il n'y a plus de variantes à découvrir) ou au bout d'un nombre d'itérations prédéfinis (nous avons arbitrairement fixé cette limite à 20).

Pour définir nos graines nous sommes partis des classes spécifiques d'une classe telles que décrites dans le projet `ucomp` [5] et le document de présentation de DEFT'2015 [6]. Nous avons décliné les classes sémantiques avec une étiquette grammaticale (nom, adjectif ou participe passé, verbe infinitif, autres formes de verbes conjugués, adverbe). Le genre et le nombre des instances étaient masculin et singulier. Nous avons dérivé à la main les instances dans les catégories autres que nominale et autres formes de verbes conjuguées. Cette dernière forme a été automatiquement construite en sélectionnant dans les variantes récoltées les formes verbales qui n'étaient pas des autres étiquettes grammaticales retenues. La classe `MÉPRIS-NOM` était par exemple représentée par les graines *mépris, dédain, dégoût* et *haine*.

Pour rechercher les variantes, nous avons exploité la technique de construction de vecteurs de mots offerte par `word2vec` (Mikolov *et al.*, 2013). Cette technique présente l'avantage de rapprocher des formes avérées dans le corpus, variantes orthographiques, morphologiques et lexicales. Suivant cette approche, nous avons constitué différents lexiques à partir de différents jeux de classes : un jeu de classes décrivait la polarité, un autre les classes émotionnelles fines et un autre les classes émotionnelles fines (en fusionnant les classes antonymes). Le lexique classé en polarité a bénéficié d'un post-traitement qui consistait à changer la classe d'une occurrence en fonction de sa distribution sur les tweets annotés dans le corpus d'entraînement de DEFT. Pour un terme donné, si la différence entre son nombre d'occurrences dans un tweet positif et dans un tweet négatif était inférieure au nombre d'occurrences de neutre, le terme était classé neutre. Sinon il était classé en positif ou négatif selon le nombre d'occurrence dans la classe majoritaire.

Réalisation Pour construire ce lexique, nous avons exploité le corpus de tweets français utilisé pour filtrer le lexique d'émôticones décrit en section 4.2. Nous l'avons prétraité linguistiquement (uniformisation de la casse, tokenization et suppression des tokens non alphabétiques et d'un seul caractère) et nettoyé (retrait des tweets doublons).

Quel que soit le jeu de classes sémantiques initiales, nos graines sont au nombre de 273. La taille des lexiques construits diffèrent selon le jeu de classes initial : le lexique en polarité compte 2 650 termes, celui en classes émotionnelles fines 9 631 et celui en classes émotionnelles fines avec fusion des antonymes 4 804. Un oeil critique sur le contenu des classes obtenues nous conduit à relever quelques erreurs de classement, la présence de termes ambigüs et bien sûr un problème de complétude. Mais l'essentiel des regroupements reste cohérent. De part notre précédé de recherche de variantes, le filtrage grammatical joue un rôle primordial dans la qualité des lexiques extraits. Cette approche compile 128 traits.

4.5 Divers traits surfaciques et linguistiques

Nous avons défini des traits pour caractériser la forme des tweets. Parmi ces traits, nous comptons : le hashtag, le cashtag, la mention, l'url, le token entièrement (ou partiellement, ou débutant) en majuscules, le token constitué entièrement ou contenant au moins un symbole, le token constitué entièrement ou contenant au moins un chiffre, la répétition de caractères quelconques ou alphabétique ou numérique, de marques de ponctuation, le nombre de tokens. Nous comptons également un trait pour représenter chacun des lexiques fermés manuellement constitués suivants : déterminants et pronoms d'emphase, négation, comparaisons, pronoms selon leur personne. Cette approche compile 30 traits.

5. `http://www.ucomp.eu/`
6. `https://deft.limsi.fr/2015/descriptionTaches.fr.php?lang=fr`

5 Expériences et résultats

Après avoir brièvement présenté les données et le protocole expérimental, nous rapportons les scores de différentes approches sur le corpus d'entraînement et sur le corpus de test, à savoir l'approche de base, puis l'utilisation de lexiques exogènes et endogènes (`RUN 1`) à partir de cette approche de base, et enfin cette même approche mais en utilisant uniquement les lexiques endogènes (`RUN 2`). Nous discutons ensuite ces résultats.

5.1 Protocole expérimental et données

Par la suite, nous utilisons les mesures suivantes pour discuter de nos résultats. Un *vrai positif* est un test jugé correctement positif, un *faux positif* est un test incorrectement jugé positif, un *faux négatif* est un test incorrectement jugé négatif. La *précision* d'un système correspond au nombre de tests positifs corrects sur le nombre de tests estimés positifs (somme des corrects et des incorrects). Le *rappel* est le nombre de tests positifs corrects sur la nombre total de tests positifs réels (somme des vrais positifs et des faux négatifs). La *F-mesure* est une moyenne harmonique de la précision et du rappel. La *micro-précision* est le nombre de tests positifs corrects toute classe confondue (somme des vrais positifs quelle que soit la classe) sur le nombre de tests estimés positifs toute classe confondue. La *macro-précision* est la moyenne des précisions obtenues sur chaque classe. Le *macro-rappel* est la moyenne des rappels obtenus pour chaque classe.

En pratique, les mesures de précision utilisées dans DEFT sont légèrement différentes puisqu'elles comptabilisent un faux négatif à chaque classe dès qu'une instance n'a pas été classée dans une des classes disponibles. Cette situation est rencontrée à chaque fois qu'un système a assigné une instance à la classe `INFORMATION`. Dans la mesure classique, seuls les faux positifs de cette classe sont incrémentés. Pour les organisateurs, cette approche vise à "fortement pénaliser les systèmes qui ne répondent pas (ou qui fournissent des classes non prévues) alors que c'était une information disponible". De cette manière, un système qui attribue une classe prévue pour un tweet mais se trompe aura un meilleur score qu'un système qui ne choisit pas ou qui "invente" une classe.

Les scores sur le corpus d'entraînement ont été obtenus par validation croisée sur 10 partitions ; les moyennes des scores sont obtenues par 10 systèmes entraînés sur 9 partitions et testés sur la dixième. Le contenu des partitions a initialement été tiré au hasard. Le corpus d'entraînement compte 6 672 instances dont 3 531 classées `INFORMATION` et 3 142 `NON-INFORMATION`. D'après ce corpus, un système qui classerait toutes les instances dans cette classe majoritaire aurait, pour la tâche 2.2, 0.5292 de micro-précision et 0.0294 de macro-précision. Le corpus de test compte lui 3 379 instances pour les tâches 1 et 2.1 avec 1 861 `INFORMATION` et 1 518 `NON-INFORMATION`. Le corpus de test pour la tâche 2.2 compte 1 361 instances `NON-INFORMATION`. Pour la tâche 1, nous n'avons pas considéré la classe mixte. Pour la tâche 2.2, nous avons considéré la classe `INFORMATION` comme étant une des classes fines à reconnaître en plus des 18 autres définies.

Classe	**Approche de base**			**RUN 1**			**RUN 2**		
	Mic-P	**Mac-P**	**Mac-R**	**Mic-P**	**Mac-P**	**Mac-R**	**Mic-P**	**Mac-P**	**Mac-R**
Polarité (t1)	0.5969	0.6647	0.5224	**0.714**	**0.7582**	**0.619**	0.6736	0.6783	0.6021
Méta-classe (t2.1)	0.5634	0.4876	0.3930	**0.71**	**0.702**	0.438	0.6649	0.6231	**0.4865**
Classe fine (t2.2)	0.2792	0.0217	0.0224	**0.681**	**0.518**	0.241	0.6349	0.4709	**0.2604**

Tableau 1 – Performance de l'approche de base, `RUN 1` et `RUN 2` sur le corpus d'*entraînement* avec Micro-Précision, Macro-Précision et Macro-Rappel.

Classe	**Général**					**Approche de base**		**RUN 1**		**RUN 2**	
	Moy	Méd	E-T	Min	Max	**Mic-P**	**Mac-P**	**Mic-P**	**Mac-P**	**Mic-P**	**Mac-P**
Polarité (t1)	0.581	0.693	0.238	0.04	0.735	0.5969	0.6647	0.6087	0.6552	**0.6232**	**0.6769**
Méta-classe (t2.1)	0.408	0.514	0.217	0.029	0.612	0.5634	0.4876	0.5711	0.5081	**0.5750**	**0.5143**
Classe fine (t2.2)	0.179	0.199	0.152	0	0.346	0.2792	0.0217	**0.3343**	**0.0281**	0.3159	0.0273

Tableau 2 – Performance de l'approche de base, `RUN 1` et `RUN 2` sur le corpus de *test* avec Moyenne, Médiane, Ecart-type, Min, Max, Micro-Précision et Macro-Précision.

5.2 Discussion des résultats

Le tableau 1 rapporte les résultats globaux obtenus sur le corpus d'entraînement et le tableau 2 sur le corpus de test ; avec dans les deux cas les mesures d'évaluation modifiées dans le cadre de DEFT. Dans le tableau 1, les résultats de l'approche `RUN 1` ont été calculés sur un échantillonnage en 10 parties et ceux de l'approche `RUN 2` sur un échantillonnage en 2 parties. Pour la suite de l'analyse, nous supposons que cela n'affecte pas les résultats.

Les tableaux 3, 4 et 5 présentent le détail des scores obtenus par l'approche de base sur les différentes classes avec les mesures d'évaluation originales sur le corpus d'entraînement. Les tableaux 6, 7 et 8 présentent le détail des scores obtenus par l'approche `RUN 1` sur les différentes classes avec les mesures d'évaluation originales sur le corpus d'entraînement. Les tableaux 9, 10 et 11 présentent le détail des scores obtenus par l'approche `RUN 2` sur les différentes classes avec les mesures d'évaluation originales sur le corpus d'entraînement.

Les résultats présentés dans le tableau 1 indiquent que les deux approches proposées sont plus efficaces que l'approche de base sur le corpus d'entraînement. L'approche `RUN 1` obtient une meilleure micro et macro précisions que l'approche `RUN 2`, mais cette dernière obtient un meilleur rappel.

Sur le corpus de test, les deux approches proposées sont également meilleures que l'approche de base (tableau 2). Cependant, l'approche `RUN 1` n'est meilleure que sur la tâche 2.2. L'approche `RUN 2` obtient les meilleures performances sur les tâches 2.1 et 1.

Après analyse des résultats, nous pensons que l'approche `RUN 2` est meilleure sur les tâches 2.1 et 1 car, même si l'approche `RUN 1` obtient de meilleurs résultats sur les classes les plus représentées. Cette différence tend à s'estomper quand les classes sont généralisées. Nous remarquons que les scores de rappel de l'approche `RUN 2` sont globalement meilleurs sur la classe générique `ÉMOTION`. Or cette classe générique regroupe 12 classes ce qui représente un fort déséquilibre par rapport aux classes génériques `OPINION` et `SENTIMENT` qui ne regroupent que 6 classes à elles deux. Les scores de rappel important de l'approche `RUN 2` sur les classes de l'`ÉMOTION` se traduisent par une précision accrue lorsque les classes sont regroupées au sein des classes génériques.

À la lumière de ces résultats, nous constatons que les lexiques endogènes (approche `RUN 2`) permettent de faire remonter des tweets de la classe générique `ÉMOTION` au détriment de la précision sur l'ensemble des classes. À la différence de l'approche `RUN 2` dont la combinaison de lexiques endogènes et exogènes permet au contraire de désambiguïser un grand nombre de classe au détriment des moins représentées.

6 Conclusion et perspectives

Dans ce défi en fouille de texte, il s'agit de retrouver les classes sémantiques de tweets, à différents niveaux de granularité. Nous choisissons de réaliser une classification des tweets au niveau le plus fin à l'aide d'une représentation en bigrammes de mots et de plusieurs lexiques affectifs, puis d'inférer les classes sémantiques plus générales selon cette première classification. Les résultats de cette inférence sont globalement meilleurs que ceux de la classification fine, ce qui peut s'expliquer par la faible ambiguïté entre les classes sémantiques générales comparativement aux plus fines. Toutefois, les erreurs de classification au niveau le plus fin ont nécessairement des répercussions sur les classes générales. En effet, quelques classes fines sémantiquement proches (les classes `DÉSACCORD` et `DÉPLAISIR` par exemple) mais ne partagent pas la même classe d'information (respectivement, `OPINION` ou `ÉMOTION`). En ce qui concerne la classification fine des tweets, nous constatons que l'approche de base utilisant une représentation en sac de bigrammes de mots fournit des résultats satisfaisants sur les classes les plus présentes, mais ne parvient pas à désambiguïser avec justesse les classes moins fréquentes. Les lexiques affectifs que nous avons utilisés permettent d'affiner dans une certaine mesure cette classification. Nous observons cependant deux tendances selon le type de lexique employé. Dans le cas des lexiques affectifs construits indépendamment du corpus fourni, la plupart des classes bénéficient d'un gain modéré tandis qu'en utilisant les lexiques issus du corpus de ce défi, quelques classes sémantiques sont particulièrement bien identifiées, au détriment des autres.

Ce travail a été l'occasion de nous intéresser à la catégorisation d'émotions pour la fouille d'opinion, et nous envisageons de poursuivre certaines pistes dans ce domaine. Parmi celles-ci, nous prévoyons de comparer l'intérêt du point de vue de la désambiguïsation de l'opinion des expressions ou multi-mots aux unigrammes de mots et de mesurer l'apport de l'analyse syntaxique pour une telle tâche.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
+	1772	1262	846	0.6703	0.4774	0.5576
=	3531	4727	3224	0.6820	0.9130	0.7808
-	1370	684	561	0.8201	0.4094	0.5462
Total	6673		4631	0.6939 (Micro-P) 0.7241 (Macro-P)	0.5999 (Macro-R)	

Tableau 3 – Détail de la performance de l'*approche de base* pour la *tâche 1* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
EMOTION	776	354	264	0.7457	0.3402	0.4672
INFORMATION	3531	4727	3224	0.6820	0.9130	0.7808
OPINION	2250	1551	1107	0.7137	0.492	0.5824
SENTIMENT	82	40	26	0.65	0.3170	0.4262
Total	6673		4621	0.6924 (Micro-P) 0.5583 (Macro-P)	0.4124 (Macro-R)	

Tableau 4 – Détail de la performance de l'*approche de base* pour la *tâche 2.1* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
ACCORD	153	47	31	0,6596	0,2026	0,3100
AMOUR	8	0	0	0,0000	0,0000	0,0000
APAISEMENT	9	2	0	0,0000	0,0000	0,0000
COLERE	206	106	67	0,6321	0,3252	0,4295
DEPLAISIR	47	1	0	0,0000	0,0000	0,0000
DERANGEMENT	12	5	5	1,0000	0,4167	0,5882
DESACCORD	212	132	97	0,7348	0,4575	0,5640
DEVALORISATION	394	202	106	0,5248	0,2690	0,3557
ENNUI	4	0	0	0,0000	0,0000	0,0000
INFORMATION	3531	4727	3224	0,6820	0,9131	0,7808
INSATISFACTION	9	3	2	0,6667	0,2222	0,3333
MEPRIS	173	42	8	0,1905	0,0462	0,0744
PEUR	269	191	155	0,8115	0,5762	0,6739
PLAISIR	34	6	3	0,5000	0,0882	0,1500
SATISFACTION	73	37	24	0,6486	0,3288	0,4364
SURPRISE_NEGATIVE	10	1	1	1,0000	0,1000	0,1818
SURPRISE_POSITIVE	4	0	0	0,0000	0,0000	0,0000
TRISTESSE	34	1	0	0,0000	0,0000	0,0000
VALORISATION	1491	1170	708	0,6051	0,4748	0,5321
Total	6673		4431	0,6640 (Micro-P) 0,4556 (Macro-P)	0,2327 (Macro-R)	

Tableau 5 – Détail de la performance de l'*approche de base* pour la *tâche 2.2* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
+	1772	1156	836	0,7232	0,4718	0,5710
=	3531	4808	3319	0,6903	0,9400	0,7960
-	1370	709	611	0,8618	0,4460	0,5878
Total	6673		4766	0,7142 (Micro-P)	0,6192 (Macro-R)	
				0,75841 (Macro-P)		

Tableau 6 – Détail de la performance du `RUN 1` pour la *tâche 1* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
EMOTION	776	395	318	0,8051	0,4098	0,5431
INFORMATION	3531	4808	3319	0,6903	0,9400	0,7960
OPINION	2250	1434	1083	0,7552	0,4813	0,5879
SENTIMENT	82	29	20	0,6897	0,2439	0,3604
Total	6673		4744	0,7109 (Micro-P)	0,4385 (Macro-R)	
				0,7023 (Macro-P)		

Tableau 7 – Détail de la performance du `RUN 1` pour la *tâche 2.1* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
ACCORD	153	65	40	0,6154	0,2614	0,3670
AMOUR	8	0	0	0,0000	0,0000	0,0000
APAISEMENT	9	1	0	0,0000	0,0000	0,0000
COLERE	206	107	72	0,6729	0,3495	0,4601
DEPLAISIR	47	1	0	0,0000	0,0000	0,0000
DERANGEMENT	12	4	4	1,0000	0,3333	0,5000
DESACCORD	212	134	96	0,7164	0,4528	0,5549
DEVALORISATION	394	176	103	0,5852	0,2614	0,3614
ENNUI	4	0	0	0,0000	0,0000	0,0000
INFORMATION	3531	4808	3319	0,6903	0,9400	0,7960
INSATISFACTION	9	2	2	1,0000	0,2222	0,3636
MEPRIS	173	47	18	0,3830	0,1040	0,1636
PEUR	269	230	182	0,7913	0,6766	0,7295
PLAISIR	34	4	2	0,5000	0,0588	0,1053
SATISFACTION	73	27	18	0,6667	0,2466	0,3600
SURPRISE_NEGATIVE	10	1	1	1,0000	0,1000	0,1818
SURPRISE_POSITIVE	4	0	0	0,0000	0,0000	0,0000
TRISTESSE	34	7	4	0,5714	0,1176	0,1951
VALORISATION	1491	1059	688	0,6497	0,4614	0,5396
Total	6673		4549	0,6817 (Micro-P)	0,2414 (Macro-R)	
				0,5180 (Macro-P)		

Tableau 8 – Détail de la performance du `RUN 1` pour la *tâche 2.2* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
+	1771	1261	765	0.6066	0.4319	0.5046
=	3530	4463	3016	0.6757	0.8543	0.7546
-	1369	946	712	0.7526	0.5200	0.6151
Total	6670		4493	0.6736 (Micro-P) 0.6783 (Macro-P)	0.6021 (Macro-R)	

Tableau 9 – Détail de la performance du `RUN 2` pour la *tâche 1* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
EMOTION	809	463	342	0.7386	0.4227	0.5377
INFORMATION	3530	4463	3016	0.6757	0.8543	0.7546
OPINION	2250	1709	1061	0.6208	0.4715	0.5359
SENTIMENT	81	35	16	0.4571	0.1975	0.2758
Total	6670		4435	0.6649 (Micro-P) 0.6231 (Macro-P)	0.4865 (Macro-R)	

Tableau 10 – Détail de la performance du `RUN 2` pour la *tâche 2.1* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Classe	ref.	hyp.	correct	Précision	Rappel	F-Mesure
ACCORD	153	70	38	0.5428	0.2483	0.3408
AMOUR	8	1	0	0.0	0.0	0
APAISEMENT	9	3	2	0.6666	0.2222	0.3333
COLERE	206	72	52	0.7222	0.2524	0.3741
DEPLAISIR	47	6	0	0.0	0.0	0
DERANGEMENT	12	3	3	1.0	0.25	0.4
DESACCORD	212	163	105	0.6441	0.4952	0.56
DEVALORISATION	394	331	156	0.4712	0.3959	0.4303
ENNUI	4	0	0	0	0.0	0
INFORMATION	3530	4463	3016	0.6757	0.8543	0.7546
INSATISFACTION	9	9	0	0.0	0.0	0
MEPRIS	172	89	36	0.4044	0.2093	0.2758
PEUR	269	258	191	0.7403	0.7100	0.7248
PLAISIR	34	16	7	0.4375	0.2058	0.28
SATISFACTION	72	26	16	0.6153	0.2222	0.3265
SURPRISE_NEGATIVE	10	3	3	1.0	0.3	0.4615
SURPRISE_POSITIVE	4	0	0	0	0.0	0
TRISTESSE	34	12	6	0.5	0.1764	0.2608
VALORISATION	1491	1146	605	0.5279	0.4057	0.4588
Total	6670		4236	0.6349 (Micro-P) 0.4709 (Macro-P)	0.2604 (Macro-R)	

Tableau 11 – Détail de la performance du `RUN 2` pour la *tâche 2.2* sur le corpus d'*entraînement* avec nombre absolu en référence, dans l'hypothèse et dans l'ensemble correct ainsi que les scores de Précision, Rappel F-mesure, Micro-Précision et Macro-Précision originale.

Remerciement

Ce travail a bénéficié du soutien du fond unique interministériel (FUI) 17 au travers du projet ODISAE [7].

Références

AUGUSTYN M., BEN HAMOU S., BLOQUET G., GOOSSENS V., LOISEAU M. & RINCK F. (2006). Lexique des affects : constitution de ressources pédagogiques numériques. In *Colloque International des étudiants-chercheurs en didactique des langues et linguistique.*, Grenoble, France.

BRADLEY M. M. & LANG P. J. (1999). Affective norms for english words (ANEW) : Instruction manual and affective ratings.

CRAMMER K. & SINGER Y. (2000). On the learnability and design of output codes for multiclass problems. In *Proceedings of COLT '00*, p. 35–46, Palo Alto, CA, USA.

FAN R.-E., CHANG K.-W., HSIEH C.-J., WANG X.-R. & LIN C.-J. (2008). LIBLINEAR : a library for large linear classification. *Journal of Machine Learning Research*, **9**, 1871–1874.

HSU C.-W., CHANG C.-C. & LIN C.-J. (2003). *A practical guide to support vector classification*. Rapport interne, Department of Computer Science, National Taiwan University.

LIU B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, **5**(1).

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

MURRAY G. & CARENINI G. (2011). Subjectivity detection in spoken and written conversations. *Natural Language Engineering*, **1**(1).

PARROTT W. (2001). *Emotions in Social Psychology*. Philadelphia, PA, USA : Psychology Press.

PUSTEJOVSKY J. & STUBBS A. (2012). *Natural Language Annotation for Machine Learning*. O'Reilly Publishers.

STAIANO J. & GUERINI M. (2014). DepecheMood : a Lexicon for Emotion Analysis from Crowd-Annotated News. *CoRR*, p. 427–433.

VERNIER M., MONCEAUX L., DAILLE B. & DUBREIL E. (2009). Catégorisation des évaluations dans un corpus de blogs multi-domaine. *Revue des Nouvelles Technologies de l'Information (RNTI)*.

WIEBE J. & MIHALCEA R. (2006). Word sense and subjectivity. In *Proceedings of ACL'06*, Sydney, Australia.

YANG M., PENG B., CHEN Z., ZHU D. & CHOW K. (2014). A Topic Model for Building Fine-grained Domain-specific Emotion Lexicon. *anthology.aclweb.org*, p. 421–426.

YU H.-F., HO C.-H., JUAN Y.-C. & LIN C.-J. (2013). *LibShortText : a library for short-text classification and analysis*. Rapport interne, Department of Computer Science, National Taiwan University. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libshorttext`.

7. `www.odisae.com`

Feature engineering for tweet polarity classification in the 2015 DEFT challenge

François Morlane-Hondère Eva D'hondt
LIMSI, CNRS, Rue John Von Neumann, 91400 Orsay, France
francois.morlane-hondere@limsi.fr, eva.dhondt@limsi.fr

Résumé. Dans cet article, nous présentons notre participation à la tâche 1 du Défi Fouille de Textes (DEFT) 2015. Cette dernière consiste à identifier la polarité de tweets en français. Notre système de classification s'appuie sur des traits de nature variée tels la présence des mots du tweet dans les lexiques, leurs propriétés typographiques, la façon dont sont utilisés les éléments de la syntaxe de Twitter (hashtags, mentions) ou encore le fait qu'un tweet ait été généré automatiquement ou produit par un humain. Nos deux soumissions ont respectivment obtenu une macro-précision de 0.687 and 0.688. Elles se situent au-dessus de la moyenne de l'ensemble des participants (0.582) mais légèrement en dessous de la médiane (0.693).

Abstract.

Feature engineering for tweet polarity classification in the 2015 DEFT challenge
In this paper we present our contribution to the first task of the 2015 DEFT challenge which dealt with polarity classification of French tweets. We explored the impact of a large number of different types of features, such as lexicon-based features, typography-based features, Twitter-specific features and features that incorporate external (world) knowledge. We submitted two runs and achieved macro-averaged precision scores of 0.687 and 0.688 respectively, which is above the average of all submitted runs (0.582) and slightly below the median (0.693).

Mots-clés : détection de polarité, analyse de sentiments, DEFT, Twitter, réseaux sociaux.

Keywords: polarity classification, sentiment analysis, DEFT, Twitter, social media.

1 Introduction

Over the last eight years microblogging sites such as Twitter have become a powerful and inlfuential means of communication on a global scale. As Twitter is increasingly regarded as the digital voice of public opinion, there is a high demand for automated tools that can analyze the content of tweets for the purposes of sentiment classification or knowledge extraction. This is a challenging task, however : Twitter's 140-character limit forces the user to express their message in a terse, compact manner. Moreover, the language use in tweets is very informal, with creative spelling and punctuation, emoticons, typos, slang, new words, abbreviations, and the inclusion of URLs and #hashtags. Twitter language changes fairly quickly as well : hashtags used to be added to the end of a message as a sort of label independent of the main 'tweet text', but are now increasingly used within sentences, e.g. *On passe à la #chasse aux #loups sans le dire*. Over the last few years, twitter (language) analysis has gathered a lot of attention from the Natural Language Processing and Machine Learning research communities (Kong *et al.*, 2014), but the vast majority of existing resources and systems are limited to English. In the 2015 DEFT challenge, three different twitter analysis tasks were proposed for a given corpus of French tweets on the theme of ecology and climate change. This article presents our contribution to the first task : *polarity classification of French tweets* into three different categories (positive, negative and neutral). We explore the impact of a large number of different types of features, such as lexicon-based features, typography-based features, Twitter-specific features and features based on world knowledge.

2 DEFT 2015 task and corpus

The first task of the 2015 DEFT challenge constituted *polarity classification of French tweets*. In the task description, *polarity* was defined as expressing either an opinion, sentiment or emotion [1]. A given tweet had to be assigned one label, either *positive*, *negative*, *neutral*, and there was no *mixed* or other residual category. Please note that the polarity was assigned to the tweet as a whole, i.e. on message level (cf. task B of the SemEval 2013 challenge [2]).

The DEFT 2015 corpus consisted of about 15,000 tweets on the topic of climat change, which were collected in the context of the uComp project [3]. The training and test set were released separately and contained 7,929 and 3,383 tweet IDs, respectively. We used the downloader program made available by the track organizers to download the individual tweets. For some IDs, however, the corresponding tweet was no longer available. As a result we trained and evaluated our system on a training and test set of 7,830 and 3,381 tweets respectively.

Please note that the features described in section 3.2 are only generated over the actual text of the tweet and that we did not use any twitter meta-data such as timestamps, username, ... as additional features.

While the tweet corpus was clean (i.e. no formatting errors), the annotations in the training set were rather inconsistent at times. For example, the following tweet text appeared 9 times in the training corpus (each time retweeted to another user but containing the exact same message) but it was classified five times as positive, three times as negative, and one time as having a neutral polarity.

@lalibrebe 10° à Werchter. 35° à Werchter. Le réchauffement climatique accélère. `https://t.co/x1JS7Tox3C` *Merci de RT.*

3 System description

3.1 Classifier

Like Pak & Paroubek (2010), we build a polarity classifier using the multinomial Naïve Bayes classification algorithm (as implemented in the Weka 3.6 toolkit). This algorithm yielded similar performance to SVM models but was much faster.

3.2 Features

3.2.1 Unigrams

Word n-grams are sequences of n words extracted from a given text. Baseline classification systems generally use n-gram features, as they generally yield good performance and are computationally cheap to compute. A classic approach consists in combining them with other features to achieve higher accuracy.

In our system, we used (the presence of) the 450 most discriminating unigrams – 1-grams – in the training corpus, as calculated by the InfoGainAttributeEval function implemented in Weka 3.6.12 (Hall *et al.*, 2009), as the basis classification features. As its name suggests, this function computes the information gain of each feature in respect to the polarity of the tweets. Following Pang *et al.* (2002), we defined these features as binary, that is, the feature capture the presence of a unigram term in a given tweet, irrespective of its frequency.

3.2.2 Lexicons

A traditional approach in polarity (or sentiment) classification is that of dictionary look-up methods using lists of *positive* and *negative* words, usually nouns, verbs and adjectives. Such lexicons can be used in many ways. We chose to compile several lexicons and generate features that denote the presence of a term in each lexicon as a binary feature.

1. `https://deft.limsi.fr/2015/guideAnnotation.fr.php`
2. `http://www.cs.york.ac.uk/semeval-2013/task2/`
3. `http://www.ucomp.eu/`

For this task we chose to work with relatively small lexicons so that we would be able to manually check their content. By doing so, we could identify and remove polysemous words, thus eliminating potential sources of noise (see examples below). Table 1 indicates the number of words in each lexicon. Overall, we have a higher coverage of negative words.

Lexicon name		# of positive words	# of negative words
Polarimots		54	363
Dictionnaire électronique des Synonymes	nouns	242	355
	verbs	210	384
	adjectives	377	489
Swear words		-	102
Complementary		10	17
total		893	1,710

TABLE 1 – Number of words included in each lexicon.

Polarimots Polarimots [4] is a lexicon containing 7,483 French nouns, verbs, adjectives and adverbs whose polarity – positive, negative or neutral – has been semi-automatically annotated (Gala & Brun, 2012). There are three degrees of annotation confidence. We built a positive and a negative lexicon from the positive and negative words whose annotation confidence is the highest, i.e. when all the annotators agreed (Gala & Brun (2012) showed that including annotations that have a lower agreement score slightly degraded the performance of a polarity classification system).

Dictionnaire électronique des Synonymes A second series of six lexicons was built using the *Dictionnaire électronique des Synonymes* [5] – DES –, a French thesaurus containing 203,311 synonyms (Manguin *et al.*, 2004). We manually built sets of ten seed words for nouns, adjectives and verbs, in the positive and negative polarity, respectively. Two of the six resulting seed sets can be seen below :

— Positive adjectives : *beau*, *gentil*, *intelligent*, *utile*, *agréable*, *sympathique*, *honnête*, *prudent*, *propre*, *bon*
— Negative nouns : *scandale*, *désastre*, *violence*, *mensonge*, *douleur*, *agression*, *tristesse*, *peur*, *haine*, *mort*

Then, assuming that the polarity of a word is propagated through its synonyms (Rao & Ravichandran, 2009), we extracted all the synonyms of the seeds. Below are some of the synonyms of the seed words listed above :

— Synonyms of positive adjectives : *consciencieux*, *euphorique*, *digne*, *peinard*, *humanitaire*, *moral*, *reposant*, ...
— Synonyms of negative nouns : *colère*, *terreur*, *mystification*, *agitation*, *lâcheté*, *rancune*, *inquiétude*, ...

The automatic expansion was followed by a manual filtering of the extracted synonyms in which we removed polysemous terms, like *salade* ('salad'), which can be used as a synonym of *mensonge* ('lie').

Although we took care to choose relatively monosemous seeds, some synonyms tend to have multiple meanings. Thus, the extraction of second degree synonyms – synonyms of the synonyms of the seeds – was found to be too noisy and was therefore abandoned.

One of the limitations of the use of the DES is that it was built from traditional dictionaries and thesauri written between 1864 and 1992 (François *et al.*, 2003). Thus, the DES reveals some discrepancies with today's French usage – especially on Twitter – that we had to rectify. For example, the word *bath* – a synonym of *beau* ('beautiful') – is not used in French since the 1970's.

Swear words and insults A list of swear words and insults was compiled from the Web [6] with the assumption that these words tend to be associated uniquely with a negative mood.

Interestingly, polysemy is also a problem here. For example, the words *fumier* ('manure') and *ordure* ('garbage') can both be figuratively used to refer to a despicable person. But in the corpus, they occur in their proper meaning, in positive or neutral tweets :

4. http://polarimots.lif.univ-mrs.fr/
5. http://www.crisco.unicaen.fr/des/
6. http://francais-oral.wikispaces.com/Lexique+des+insultes

— *Quand le fumier de cheval sert à se chauffer, un beau projet de #methanisation à @Caenlamer* `http://t.co/rRZiKQLePQ` (+)
— *@DrDree_ non non je parlais en fait de l'ecologie. Puis je suis arrive aux problemes de la gestion des ordures* (=)
We manually went through the list to discard such words.

Complementary lexicon These are two small lists of additional positive and negative words we found in the training corpus and that were not already included in other lexicons.

3.2.3 Handling negation

The assumption that positive words occur in positive tweets and negative words in negative ones is not as straightforward as it may sound : Many contextual phenomenons or stylistic factors can affect the meaning – and, thus, the polarity – of words. Benamara *et al.* (2012) showed that different types of negation can affect polarity in different ways.

We handled this highly complex problem with a simple – simplistic – polarity shifting system consisting in regular expressions checking for the following words : *pas* ('not'), *aucun* ('none'), *jamais* ('never'), *non* ('no'), *peu* ('few'), *ni* ('nor') and *rien* ('nothing'), in a window of two words before and after occurrences in tweets of words found in our lexicons. Like the previous features, the polarity shifter feature is binary : If a polarity shifter is found in the context of a word included in a positive (resp. negative) lexicon, then the value of its negative (resp. positive) counterpart is set to 1.

3.2.4 Term extraction on the training corpus

Using the Alchemy [7] keyword and entity extraction software, we processed the positive, negative and neutral subsets of the training corpus to obtain the most discriminating (multiword) features for each subset. Alchemy uses deep learning to find dependencies between words over large corpora. We used the extracted words and phrases as additional weighted binary features in our second submitted run : While the presence (resp. absence) of an extracted term in a given tweet resulted in a 1- (resp. 0-) value, the feature weight was a normalized version of the extraction score that Alchemy returned for that term. This way the presence of an extracted term for which Alchemy had a low confidence score had less impact on the classification process than that of a term with a high confidence score.

3.2.5 Twitter-specific features

Following Arakawa *et al.* (2014), we call *Twitter-specific features* the commands and conventions used by Twitter users in their posts. Two recurring commands are the hashtag (#), used to turn words they are added to into clickable tags, and the *at* sign (@), used to mention or to reply to another user. We used the presence of hashtags and mentions and their location in tweets (i.e. in the beginning of the tweet or in the end end) as binary features. The number of hashtags or mentions was not found to be relevant, as well as the presence of the mention *RT* (*retweet*), which is used to share a tweet with a users followers.

3.2.6 Extracting information from tinyURLs

Presence of tinyURL We observed that the majority (5849 out of 7830) of tweets in the training set contain at least one tinyURL. While not a very strong feature, a tweet without a tinyURL has a relative higher probability of belonging to the positive or negative category than the neutral one. Taking the presence of tinyURLs into account lead to small but significant improvements.

Generation history of tweet For a secondary feature based on the tinyURLs, we explored how the actual text in the tweet is generated. We found that for a substantial number of tweets in the training set, the tweet text was either the title or introductory sentence of the online article or post it referred to. These tweets are the result of (semi-)automatic sharing of online content with minimal human interaction. We hypothesized that such tweets are more likely to belong to the neutral

7. `http://www.alchemyapi.com/`

category, and that tweets which express a positive or negative opinion on a subject would contain more information written by the user (either in the form of adding hashtag to specify the information, or by an accompanying sentence that comments on the content of the article or post). We therefore added a feature that categorized a tweet as either "human" (written by a human and containing novel information), "automatic" (the result of automatic sharing of existing content with minimal manual editing) or "unknown" (a surplus category of tweets for which we lacked information to determine the level of human interaction). The features were created as follows : For each tweet, we extracted the tinyURL (if present) and downloaded the title and introductory sentence(s) from the corresponding webpage. If the text in the tweet matched with (part of) the title and introductory sentences, the tweet was categorized as "automatic". If not the case, for example, because extra information was added in the form of extra hashtags, or the tweet text was an own comment or summary of the referenced article, the tweet was categorized as "human" [8]. Please note that we used fuzzy matching as implemented in the `FuzzyWuzzy` Python package [9] to account for small edits in the original texts. For example in the following tweet

"On passe à la #chasse aux #loups sans le dire" via Pescalune `http://t.co/KiD1N1Oq7v`,

certain words in the tweet have been converted into hashtags by the user while the phrase still corresponds to the title of the referenced article. By allowing fuzzy matching with a moderately high threshold (>70%) we can still identify these "reposted" tweets. For those tweets for which we were not able to extract information on the article or post (either because of time-out errors, or difficulties in parsing the returned html), the label "unknown" was given. This category is fairly small : 525 out of 7830 tweets.

Content classification of the referenced webpages We also experimented with a third feature which was based on the content of the referenced webpage. We manually categorized a subset of the URLs in the training set into the following 6 categories : *combo* (websites such as `www.scoop.it` where users can share and publish content from other sources), *ecoBlog* (websites dedicated to ecology), *news* (news sites), *polBlog* (political blogs and websites from political parties), *science* (webpages from universities or research facilities), *other*. For each website we extracted the domain name as well as the website description and keywords from its main page. This data was then used to train a separate classifier that would classify an unseen url (and extracted information from the associated website) into the relevant category. The lack of coherence in website description and overall quality of the meta description of the websites lead to a very sparsely trained and unreliable classifier. We therefore opted not to use this feature in the submitted runs.

3.2.7 Smileys

We observed three kinds of smileys in the training corpus :

- typographic smileys. They are *compositional* smileys built with letters, numbers and ponctuation marks used to mimic eyes, noses and mouths. We found two different types of typographic smileys :
  - Western-style smileys : `:-)` `:D` `:p`
  - Japanese-style smileys (or *kaomojis*) : `O_o` `O__O` `-_-'`
- graphic smileys. They are Unicode characters : 😁😈⚽

We built two separate sets of regular expressions to check for the presence of positive and negative typographic smileys in the tweet text. Likewise, two lists of graphic smileys were built using web sources [10]. For a given tweet, the value of the features containsPosSmiley and containsNegSmiley is 1 if one or several positive – resp. negative – smiley(s) is (are) found in the message.

Please note that we disregarded *neutral* smileys. Smileys are by their very nature means of expressing emotion, so the existence – and actual usage – of neutral ones seems unlikely : This assumption was confirmed by an analysis of the corpus as we did not find any occurrence of what might be considered a neutral smiley, i.e. `:-|`, in either the train and test corpora.

8. If the tweet did not contain an tinyURL, i.e. was written from scratch, is was classified as "human" as well.
9. `https://github.com/seatgeek/fuzzywuzzy`
10. `http://unicode-table.com/fr/search/?q=emoticons`

3.2.8 Punctuation marks

Like smileys, punctuation is a common means of expressing emotion or intention in textual content. For the task, we considered five types of punctuation marks : exclamation marks, question marks, ellipsis, comma and quotation marks. The presence (or absence) of each punctuation mark resulted in a binary feature. Although exclamation marks are somewhat ambiguous – they can be associated to both joy and anger –, they can be useful to discriminate between positive/negative tweets and neutral ones. Question marks are relevant for polarity classification in that they can be used in rhetorical questions. Such questions often carry a negative polarity, as in the two examples below :

— *Bravo @RoyalSegolene ;Encore merdé, encore cédé :-(Après tout, l'écologie, c'est un truc de bobo, n'est-ce pas ?* `http://t.co/7hqIzMjYKf` (-)
— *Comment imaginer une pareille chose ? ? ? ? ? ? ? La SQ démantèle un réseau de voleurs d'huile de friture* `http://t.co/7IOQVrCO9o` (-)

Ellipsis is also interesting in that it can denote sarcasm :

— *L'écologie, cette valeur de gauche... – Ecotaxe : la carte des projets locaux menacés* `http://t.co/HBARJ3AyyT` *via @lemondefr* (-)
— *Abandonner l'écotaxe le jeudi et recevoir Schwarzenegger le lendemain pour parler de lutte contre le réchauffement climatique... Logique.* (+) [11]

The detection of the presence of quotation marks is more stringent than that of question or exclamation marks. This binary feature is only set to 1 if one or two – consecutive – words are quoted. By adding this constraint, we wanted to focus on sarcastic usage of quotations marks :

— *Chaleur et électricité "propre" ?,quelle idée saugrenue,la géothermie .....*`http://t.co/cLpnsFLqLX` (-)
— *On se fait maintenant expliquer pourquoi on se fait ROULER pour "notre bien" avec les #éoliennes #HydroQuébec à #rdieconomie #RadioCanada* (-)

3.2.9 Miscellaneous features

Interrogative markers This binary feature indicates the presence of an interrogative marker out of a manually compiled list, i.e. *quel* ('which'), *quoi* ('what'), *comment* ('how'), *combien* ('how much'), *pourquoi* ('why'), in the tweet text. The aim of this feature is to improve the identification of rhetoric questions (cf. 3.2.8).

Case This feature indicates the presence of a series of 50 characters in uppercase :

— *REFUS DE S'ATTAQUER AUX CAUSES DU PROBLÈME, INTRINSÈQUES AU CAPITALISME..* `http://t.co/NVfSeu9TL9` (-)

We assume that this is a mark of emotion and that it will not be found in neutral tweets.

Repetition This feature is set to 1 if the tweet includes a sequence of 3 identical characters :

— *@Lorenzo75019 Ah ouiiiiii c'est vrai mdddddrr c'est développement durable qui me fait rire :) mais bon je me moque pas* (-)

As for the Case feature, we assume that repetitions are emotional markers.

Separators These three binary features indicate the presence of a vertical bar (|), a square (■) or a right-pointing triangle (▶) in the tweet. These characters are exclusively used as separators in automatically-generated tweets. However, they were not found discriminating and, therefore, removed from the final set of features.

11. Although being positively annotated, the polarity of this tweet is definitely negative.

4 Submitted runs

We submitted two runs to the official evaluation. An overview of the features used in each run can be found in Table 2.

Feature Group	Feature Name	Run 1	Run 2
Unigrams	-	✓	✓
Lexical Features	inLexiquePosPolarimots	✓	✓
	inLexiqueNegPolarimots	✓	✓
	inLexiqueNegDESadj	✓	✓
	inLexiqueNegDESnom	✓	✓
	inLexiqueNegDESver	✓	✓
	inLexiquePosDESadj	✓	✓
	inLexiquePosDESnom	✓	✓
	inLexiquePosDESver	✓	✓
	inLexique d'injures	✓	✓
	InLexiquePosManuel	✓	✓
	inLexiqueNegManuel	✓	✓
	inLexiquePosSite	✓	✓
Negation	-	✓	✓
TermExtraction	inTermsPosTrainingSet		✓
	inTermsNegTrainingSet		✓
	inTermsNeutralTrainingSet		✓
Twitter-specific Features	@inTweet	✓	✓
	@inBeginTweet	✓	✓
	@atEndTweet	✓	✓
	#inTweet	✓	✓
	#inBeginTweet	✓	✓
	#atEndTweet	✓	✓
	containsRT		
	numberOf@		
	numberOf#		
tinyURL	containsTinyURL	✓	✓
	writtenByHuman	✓	✓
	catOfTinyURL		
Smileys	containsPosSmiley	✓	✓
	containsNegSmiley	✓	✓
Punctuation	containsExcl	✓	✓
	containsMultiExcl		
	containsQuestionMark	✓	✓
	containsQuotation	✓	✓
	containsElipsis	✓	✓
	containsSemiColon	✓	✓
Miscellaneous Features	containsInterrogativeMarker	✓	✓
	containsUpperCase	✓	✓
	containsRepitition (>3)	✓	✓
	containsSeparators		

TABLE 2 – Overview of features used in two official submissions

5 Results

The classification scores of the two submitted runs can be found in Table 3. We find that adding the terms extracted by the third-party Alchemy software has positive effect on classification, particularly in identifying negative tweets, which is nevertheless so small to be insignificant. Compared to the other submitted runs our system performed slightly below

average and is bulked with the main group of participants. The three top-scoring systems achieved macro-averaged scores of near 73%.

Precision	Run1	Run2	average (of all submissions for 12 groups)	median (of all submissions for 12 groups)
micro	0.676	0.672	-	-
macro	0.687	0.688	0.582	0.693

TABLE 3 – Results for submitted runs (expressed in micro- and macro-averaged precision

6 Discussion

We performed a subtractive analysis to investigate the (relative) influence of each set of features used in Run1. Table 4 shows the result of the removal of each feature set from the entire set of features. We see that the removal of the unigrams has the biggest influence on the general macro-precision. This is not surprising as the unigrams are by far the biggest set of features – 450 – and that they have been selected according to their discriminative potential. Some of the most discriminative words according to Weka's InfoGainAttributeEval function are *contre* ('against'), *menacée* ('endangered'), *espèce* ('species'), *solaire* ('solar'), *fromage* ('cheese') and *banque* ('bank'). Although *contre* ('against') and *menacée* ('endangered') – in the phrase *espèce menacée* ('endangered species') – seem to be negatively-valenced words, the presence of a word like *fromage* ('cheese') is quite surprising. This is actually due to the fact that there are, in the corpus, more than 35 retweets of a website article entitled *Le fromage, une espèce menacée ?*. These tweets being negatively annotated, the presence of the word *fromage* in a tweet has been identified as a good indicator for negative polarity. Thus, the repetition of tweets in the corpus is a bias that may lead to overfitting : Performing a simple information gain computation – as we did – does not seem to be a robust strategy.

The lexicon-based features are the second most influential features. Although most of the words in these lexicons have been chosen according to their polarity regardless of the corpus theme, and despite the many contextual phenomena – like irony – which can shift a words polarity, the simple assumption that positive and negative words are used in positive and negative tweets seems to hold.

The twitter features and the use of tinyURLs have a smaller influence, but a positive one. On the other hand, we can see that the last three features have a slightly negative influence. The fact that the presence of smileys is not discriminative is quite surprising in that they are explicit indicators of the writer's mood. We interpret the negative influence of the last two features as being due to the ambiguity of the punctuation marks, repetitions and case shifting : The hypothesis that these properties would help to discriminate positive/negative tweets and neutral ones does not seem to hold. We found that removing the last three sets of features, i.e. smileys, punctuation marks and miscellaneous features, results in a macro-averaged precision score of 0.690% over the test set.

feature	macro-precision	difference with the entire set
all features	0.687	-
- unigrams	0.596	-0.091
- lexicons	0.661	-0.026
- twitter feats.	0.682	-0.005
- tinyurl	0.679	-0.008
- smileys	0.688	+0.001
- punctuation	0.689	+0.002
- misc. feats.	0.691	+0.004

TABLE 4 – Subtractive analysis of the features used in Run1.

Analysis of the results on the test set for the highest-scoring run (Run2) shows that the main error of our classifier is overgeneration of neutral labels, which is not surprising as this category constituted the majority of the training corpus, and is therefore the best trained classifier. Of all three classifiers the positive classifier has the worst performance, particularly in distinguishing between the neutral and positive labels. We remarked a similar trend when evaluating on the training corpus with cross-validation.

Run2 \ Reference	=	-	+
=	597	121	196
-	36	204	37
+	67	51	274

TABLE 5 – Confusion Matrix of submitted results in Run2

7 Conclusion

This paper describes our participation to the tweet polarity classification task that was organized as part of the DEFT 2015 competition. In our approach we explored a variety of features, ranging from traditional dictionary look-up methods to twitter-specific features such as the presence and location of hashtags, as well as some features that were based on more external knowledge such as the source of the tinyURL in the tweet. We found that the traditional features such as unigrams and presence in lexicon had the most impact. Interestingly, the features that focused on Twitter-specific characteristics and on micro-blogging language (smileys, repetition of characters, ...) had little to no impact. Our systems achieved scores of 0.687 and 0.688 macro-averaged accuracy.

Références

ARAKAWA Y., KAMEDA A., AIZAWA A. & SUZUKI T. (2014). Adding twitter-specific features to stylistic features for classifying tweets by user type and number of retweets. *Journal of the Association for Information Science and Technology*, **65**(7), 1416–1423.

BENAMARA F., CHARDON B., MATHIEU Y. Y., POPESCU V. & ASHER N. (2012). How do negation and modality impact on opinions ? Jeju Island, Korea.

FRANÇOIS J., MANGUIN J. L. & VICTORRI B. (2003). La réduction de la polysémie adjectivale en contexte nominal : une méthode de sémantique calculatoire. In *Cahiers du CRISCO*, volume 14. Université de Caen : CRISCO.

GALA N. & BRUN C. (2012). Propagation de polarités dans des familles de mots : impact de la morphologie dans la construction d'un lexique pour l'analyse de sentiments (spreading polarities among word families : Impact of morphology on building a lexicon for sentiment analysis) [in french]. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012, volume 2 : TALN*, p. 495–502, Grenoble, France : ATALA/AFCP.

HALL M., FRANK E., HOLMES G., PFAHRINGER B., REUTEMANN P. & WITTEN I. H. (2009). The weka data mining software : An update. *SIGKDD Explor. Newsl.*, **11**(1), 10–18.

KONG L., SCHNEIDER N., SWAYAMDIPTA S., BHATIA A., DYER C. & SMITH N. A. (2014). A dependency parser for tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, to appear*.

MANGUIN J. L., FRANÇOIS J., EUFE R., FESENMEIER L., OZOUF C. & SÉNÉCHAL M. (2004). Le dictionnaire électronique des synonymes du CRISCO : un mode d'emploi à trois niveaux. In *Cahiers du CRISCO*, volume 34. Université de Caen : CRISCO.

PAK A. & PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In N. C. C. CHAIR), K. CHOUKRI, B. MAEGAARD, J. MARIANI, J. ODIJK, S. PIPERIDIS, M. ROSNER & D. TAPIAS, Eds., *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta : European Language Resources Association (ELRA).

PANG B., LEE L. & VAITHYANATHAN S. (2002). Thumbs up ? : sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, p. 79–86 : Association for Computational Linguistics.

RAO D. & RAVICHANDRAN D. (2009). Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, p. 675–682, Stroudsburg, PA, USA : Association for Computational Linguistics.

Analyse d'opinions de tweets par réseaux de neurones convolutionnels

Jean-Marc Marty[1], Guillaume Wenzek[1], Eglantine Schmitt[1,2], Jocelyn Coulmance[1]
(1) Proxem, 105 rue La Fayette, 75010 PARIS
{jmm, guw, egs, joc}@proxem.com
(2) UTC, Compiègne
eglantine.schmitt@utc.fr

Résumé. La tâche d'analyse d'opinions consiste à détecter la polarité d'un texte (du plus négatif au plus positif). Nous présentons dans cet article un réseau de neurones permettant de trier de manière faiblement supervisée un ensemble de tweets en trois catégories : négatif, neutre ou positif. L'architecture du modèle est celle d'un réseau convolutionnel à trois couches mises en parallèles où chaque couche détecte des caractéristiques différentes. Le réseau est alimenté par des vecteurs-mots appris sur un ensemble de corpus dont la Wikipédia française, sans nécessiter d'informations linguistiques. En comparant cette approche avec un ensemble de techniques classiques alimentées par des sacs de mots, nous obtenons des résultats en moyenne 25% supérieurs en macro-précision.

Abstract.
Convolutional Neural Network for Twitter Sentiment Analysis
Sentiment Analysis is a common task in natural language processing that aims to detect polarity of a text document (from the most negative to the most positive). We introduce in this article a neural network that classifies in a weakly supervised fashion a set of tweets in three classes : negative, neutral or positive. The architecture of the model is that of a convolutional neural network with three parallel layers where each layer detects distinct features. The network is fed with word embeddings learned on a set of corpus among which the French Wikipedia with few linguistic informations. This model achieves a macro-precision in average 25% higher than classical methods based on bag of words.

Mots-clés : Analyse d'opinions, réseaux de neurones convolutionnels, Twitter.

Keywords: Sentiment Analysis, Convolutional Neural Network, Twitter.

1 Introduction

1.1 Analyse d'opinions pour Twitter

En quelques années, Twitter est devenu la plateforme de microblogage la plus étudiée dans la recherche. L'intérêt pour cet objet est dû à une combinaison de facteurs. D'une part, contrairement à Facebook, les contenus publiés par les utilisateurs y sont majoritairement publics et accessibles, bien que de façon restreinte, par une interface de programmation (API). Ainsi Facebook compte 1,441 milliards d'utilisateurs actifs mensuellements contre 302 millions sur la même période pour Twitter (Statista 2015a ; Statista 2015b) mais les travaux fondés sur l'analyse de grands volumes de données sont principalement dus aux équipes de Facebook elles-mêmes (Kramer *et al.*, 2014). D'autre part, Twitter bénéficie d'une forte notoriété (91% en France en mai 2014 d'après Ipsos) et d'une certaine mythologie selon laquelle il constituerait le pouls ou le miroir des sociétés. Malgré les difficultés posées, notamment en traitement automatique du langage, par la brièveté des messages qui y sont publiés, Twitter fait ainsi aujourd'hui l'objet de centaines de publications chaque année (Fausto & Aventurier, 2015) et serait ainsi devenu l'équivalent d'un organisme modèle en biologie, polarisant l'analyse de données issues de réseaux sociaux en ligne vers un exemple unique (Tufekci, 2014).

Outre des tâches plus typiques des sciences humaines et sociales telles que l'analyse de discours après échantillonnage, les données de Twitter mobilisent un large spectre de techniques de fouille de données et de texte appliqués à des volumes massifs d'information (big data). Parmi les tâches les plus fréquentes on peut citer l'analyse de réseaux sociaux, l'analyse de séries temporelles et l'analyse de sentiment ou d'opinions (Schmitt, 2015). Cette dernière est particulièrement exigeante

du fait de la brièveté des messages postés sur Twitter (les tweets) qui conduit elle-même à de nombreuses abréviations et à un jargon propre, distinct de la langue générale. L'ironie semble également particulièrement présente sur Twitter. Il est donc difficile de mettre en œuvre tels quels les outils conçus pour des corpus journalistiques ou scientifiques tels que SentiWordNet (Baccianella *et al.*, 2010) ou le LIWC (Pennebaker *et al.*, 2001). Pour Twitter comme pour d'autres données textuelles, la classification supervisée à partir d'un corpus annoté permet de placer la question de la modélisation du sentiment ou de l'opinion en amont de l'étape de classification ; le sentiment ou l'opinion n'est pas défini(e) formellement comme le ferait un psychologue mais empiriquement et intuitivement à partir d'exemples constitués par des humains.

1.2 Méthodes classiques

L'analyse d'opinion est un problème ouvert en traitement automatique du langage, qui présente des difficultés en termes de modélisation. Les méthodes classiques (arbres de décisions, réseaux bayésiens ou SVM chez (Pak & Paroubek, 2010; Baucom *et al.*, 2013; Dodds *et al.*, 2011; Psomakelis *et al.*, 2015)), utilisant des sacs de mots, ont des taux de précision au mieux de l'ordre de 70% sur une tâche de classification ternaire (positif/neutre/négatif).

L'approche classique consistant à repérer la connotation positive ou négative d'un terme et d'en déduire l'opinion d'une phrase ou d'un texte n'est pas suffisante ; il est également nécessaire a minima d'analyser l'ordre des mots et les relations qu'ils entretiennent. Ainsi la phrase «Le plat était bon mais pas très sucré» n'a pas le même sens que «Le plat était sucré mais pas très bon». Pourtant ces deux phrases auraient la même représentation dans un sac de mot.

Au delà de ce premier problème, les structures plus complexes qu'une construction sujet-verbe-complément sont très courantes et difficiles à capturer par des modèles semi-supervisés. Parmi elles la négation : «Ce film n'était pas terrible» et la double négation : «Ce film était pas mal». Ces phénomènes peuvent être capturés par des règles linguistiques. Mais ces règles sont longues à écrire, nécessitent une bonne connaissance de la langue et sont difficilement exhaustives.

Par ailleurs, Twitter possède sa syntaxe et ses règles linguistiques propres, qui doivent être prises en compte. Certains bons résultats récemment obtenus prennent généralement en entrée, en plus des sacs de mots, des *features* (caractéristiques) fabriquées à la main, comme celles introduites par (Mohammad *et al.*, 2013) qui ont été conçues spécialement pour Twitter.

Notre approche dans ce papier est d'effectuer l'analyse d'opinions de la façon la plus faiblement supervisée possible, en injectant le moins possible d'information linguistique, de sorte que notre modèle soit facilement adaptable à plusieurs langues et à d'autres types de syntaxe.

2 Introduction aux réseaux de neurones pour le TAL

2.1 Représentation vectorielle des mots

Avant de construire un modèle capable de capturer le sens d'une phrase, il paraît important de capturer dans un premier temps le sens d'un mot. Dans l'approche de type sac de mots, on considère que chaque mot ou stemme (quand on utilise une racinisation) a un sens unique. Deux mots sont alors soit identiques soit différents. Cependant, il paraît important que notre système soit capable de comprendre que certains mots ont un sens proche, comme «bleu» et «cyan», tandis que d'autres ont un sens éloigné, comme «bleu» et «chaise». L'idéal serait de pouvoir utiliser une distance mathématique sur les représentations de nos vecteurs-mots qui nous donne une idée de la distance sémantique entre les mots.

Considérer que le sens d'un terme se déduit de son contexte d'utilisation est une idée courante en sciences du langages. Par exemple, la signification d'une proposition chez le second Wittgenstein vient de sa valeur d'usage ; chez François Rastier, la sémantique d'un texte est construite de manière interprétative au moment de la lecture et ne peut être déduite comme un calcul à partir de sa syntaxe et de dictionnaire. En 1990, (Church & Hanks, 1990) proposait une métrique pour trouver des mots avec un sens proche : la “Pointwise Mutual Information” (PMI). Le but était de trouver les actions possible avec un téléphone. Pour cela, il calcula la PMI du mot "phone" avec chaque verbe de langue anglaise. La PMI des mots x et y se définit à partir du nombre d'occurrences de chacun ($\#(x)$ et $\#(y)$), du nombre de co-occurrences $\#(x, y)$ au sein d'une fenêtre de mots, et de la taille du corpus N suivant la formule :

$$\mathrm{PMI}(x, y) = \log \frac{N\#(x, y)}{\#(x)\#(y)}$$

sit by	11.78
disconnect	9.48
answer	8.80
hang up	7.87
tap	7.69
pick up	5.63
return	5.01
be by	4.93
spot	4.43
repeat	4.39

TABLE 1 – PMI entre le mot «phone» et les verbes qui lui sont le plus souvent associés.

bleu	1,00
rouge	0,91
jaune	0,89
violet	0,87
gris	0,87
blanc	0,85
mauve	0,85
couleur bleue	0,85
bleu ciel	0,84
marron	0,84

TABLE 2 – Les mots les plus proches du mot «bleu» avec leur score de similarité cosinus.

Cette formule permet de trouver les verbes anglais les plus sémantiquement proches du mot «phone» avec les scores indiqués en table 2.1.

Il est important de noter que nous nous ne voulons sûrement pas traiter les mots "phone" et "pick up" de façon similaire, mais en revanche nous voulons traiter "answer" de la même façon que "pick up". Nous avons donc une métrique qui nous permet de dire que deux mots sont «proches» dans un certain contexte (ici «phone» joue ce rôle de contexte). Maintenant que nous disposons d'une métrique formellement définie, nous pouvons construire des vecteurs qui nous permettent de capturer cette information. On voudrait être capable de reconstruire à partir de la représentation d'un mot et de la représentation d'un contexte la PMI de ce mot par rapport à ce contexte. Nous nous imposons les contraintes suivantes :

1. Chaque mot x se verra associer deux vecteurs : $\vec{x}$ pour x en tant que mot et $\overleftarrow{x}$ pour x en tant que contexte.

2. Pour un mot x et un contexte y on veut $\mathrm{PMI}(x, y) = \vec{x} \cdot \overleftarrow{y}$

Cela revient à factoriser la matrice de PMI de taille $V \times V$, où V est la taille du vocabulaire, en une matrice de vecteurs-mots de taille $d \times V$ et une matrice de vecteurs-contextes $V \times d$. Ici d est un paramètre que l'on choisit arbitrairement comme taille de nos vecteurs. Plus d est grand plus on peut reconstruire correctement la matrice de PMI. En revanche un d trop grand rend la manipulation des vecteurs peu aisée. En pratique on prend d entre 100 et 500.

Le problème de factorisation de matrice est connu de longue date, et cette technique de fabrication de mot a déjà été utilisée. Toutefois cette méthode est longue et coûteuse car la matrice est de grande taille. De plus les algorithmes classiques de factorisation traitent indifféremment chaque ligne de la matrice alors que nous voudrions favoriser les lignes correspondant à des mots fréquents.

(Mikolov *et al.*, 2013) introduit l'algorithme Word2Vec permettant de calculer des vecteurs-mots sur des gros corpus en des temps raisonnables. Mikolov a rendu public des vecteurs-mots à 300 dimensions pour 3 millions de mots et bigrammes anglais obtenus à partir d'un corpus de 100 milliards de mots extrait de Google News. (Levy & Goldberg, 2014) montre que cette méthode revient justement à faire une factorisation de la matrice de PMI qui favorise une bonne reconstruction pour les mots fréquents. Word2Vec a l'avantage d'être très simple à implémenter et d'offrir un temps de calcul raisonnable ; il faut en effet environ une heure pour entraîner des vecteurs-mots à partir de la Wikipédia française, sur un serveur disposant de 16 cœurs.

De plus, Word2Vec calcule de bonnes représentations même pour les termes peu fréquents ; ceci nous permet d'entraîner un vecteur pour chaque forme fléchie, sans nécessiter de racinisation.

Nous avons entraîné de tels vecteurs sur la Wikipédia française en utilisant cet algorithme. Les proximités des vecteurs obtenus - au sens de la distance euclidienne - reflètent bien leurs proximités sémantiques intuitives. Ainsi les dix termes les plus proches du mot «bleu» sont indiqués en table 2. Les scores reflètent le fait que «bleu», «rouge» et «jaune» apparaissent dans des contextes similaires en français. On voit que les vecteurs-mots ainsi appris permettront d'injecter dans notre modèle une certaine connaissance du monde.

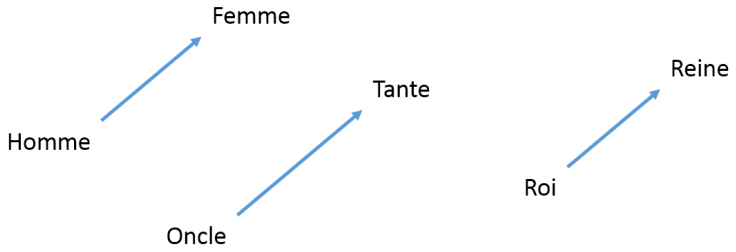

FIGURE 1 – Différences de vecteurs capturant la relation «passer au féminin». Figure extraite de (Mikolov *et al.*, 2013)

2.2 Composition des vecteurs-mots

Une autre propriété intéressante de ces vecteurs est l'additivité. On remarque que l'espace des vecteurs-mots s'est doté lors de l'apprentissage d'une structure additive. Ceci est particulièrement intéressant pour pouvoir transférer des relations. On constate ainsi que le vecteur le plus proche de $\vec{roi}+\vec{homme}-\vec{femme}$ est $\vec{reine}$. Ceci nous permet d'écrire la relation : $\vec{roi} - \vec{reine} \simeq \vec{homme} - \vec{femme}$. Ainsi le vecteur $\vec{homme} - \vec{femme}$ capture le changement de genre grammatical et sémantique. La figure 1 illustre ce phénomène.

Les vecteurs-mots capturent donc une information hiérarchisée. Ils sont localement regroupés par proximité sémantique, et ces groupes sont organisés selon des relations de type "passer au féminin". On a ainsi à la fois une information sémantique et syntaxique incluse dans les vecteurs-mots.

Une autre façon intéressante de composer les vecteurs-mots est la projection. Si on regarde par exemple le vecteur du mot «orange», on se rend compte qu'il est proche de $\vec{bleu}$ et des autres couleurs. En enlevant la composante du mot $\vec{orange}$ selon l'axe $\vec{bleu}$ on trouve un vecteur dont le plus proche voisin est $\vec{olive}$ et d'autres fruits et légumes. Ceci revient mathématiquement à projeter $\vec{orange}$ sous le sous-espace orthogonal à $\vec{bleu}$. L'opérateur de projection $\%$ est défini ainsi :

$$\vec{x} \,\%\, \vec{y} = \vec{x} - \frac{\vec{x} \cdot \vec{y}}{\vec{y} \cdot \vec{y}} \vec{y}$$

On peut donc écrire : $\vec{orange} \,\%\, \vec{bleu} \simeq \vec{pomme}$. Ceci montre que le vecteur $\vec{orange}$ porte l'ambiguïté du mot «orange», l'opérateur $\%$ permet de retirer un des sens de «orange», ici celui de couleur, et on trouve un autre sens du mot «orange» celui de fruit. Les vecteurs-mots arrivent donc à capturer beaucoup d'information même quand les mots sont ambigus.

Pour passer d'une représentation vectorielle du mot à une représentation vectorielle de la phrase, une idée naturelle est de combiner les vecteurs correspondants aux mots de la phrase. On peut se contenter d'additionner les vecteurs-mots contenus dans la phrase ou, mieux, effectuer une moyenne des vecteurs-mots de la phrase pondérée par les TF-IDF de chaque mot. Cependant ces deux méthodes ne capturent pas l'ordre des mots dans la phrase. On peut aussi s'appuyer sur des méthodes conservant une partie de la structure de la phrase. Par exemple, (Wu *et al.*, 2014) distingue d'une part les vecteurs des mots correspondant à des agents et ceux des mots correspondants à des patients, et somme les produits des couples (agent, patient). Une fois les vecteurs-mots combinés en un vecteur pour la phrase, celui-ci peut être fourni à un système de classification tel un SVM ou un réseau de neurones. Cette méthode combinée à un SVM permet à Wu de distinguer les discours de George W. Bush de ceux de Barack Obama avec une précision de 85%.

Rappelons que nous souhaitons injecter un minimum de connaissances linguistiques dans notre modèle. Nous souhaitons donc que notre système apprenne le plus automatiquement possible la bonne façon de combiner les vecteurs-mots pour la tâche d'analyse d'opinions. Nous avons vu que les vecteurs-mots ont des propriétés additives et projectives. Il parait donc logique de choisir un modèle qui puisse effectuer des opérations linéaires sur les vecteurs-mots, comme un réseau de neurones. Nous décrivons dans la section suivante pourquoi nous nous tournons vers un réseau de neurones convolutionnel.

3 Description du modèle utilisé

3.1 Introduction aux réseaux convolutionnels

Un réseau convolutionnel est un type particulier de réseau de neurones.

Chaque couche - dite de *convolution* - balaye l'ensemble de la couche précédente en appliquant à chaque petite région un même traitement local. Dans le cas d'un texte, les régions sont les n-grammes de la phrase. Pour une image, les régions sont classiquement des carrés de pixels ; la convolution permet alors par exemple d'y détecter des contours ou de flouter l'image. Ce filtre balaye ainsi l'ensemble de la sortie de la couche précédente et a pour rôle de détecter des propriétés locales. En superposant de telles couches de convolution les unes au-dessus des autres, on espère détecter des propriétés de plus en plus globales.

Cette méthode s'inspire de l'organisation des neurones du cortex visuel. Ces cellules sont sensibles à des petites régions du champ visuel et en recouvrent l'ensemble afin d'exploiter les propriétés locales de l'image reçue par les yeux. Ces réseaux, introduits par (LeCun & Bengio, 1995), semblent donc adaptés pour la vision par ordinateur et continuent à obtenir des résultats remarquables dans ce domaine (Krizhevsky *et al.*, 2012).

Entre chaque couche de convolution, on rajoute une couche - dite de *pooling* - qui ne conserve que les k plus grandes valeurs, où k est un hyperparamètre à sélectionner (c'est-à-dire un paramètre choisi arbitrairement). Cela a deux intérêts principaux : d'une part, cela diminue le nombre de calculs à effectuer. D'autre part, cela permet de calculer une forme d'invariant pour la translation, ce qui est effectivement ce qu'on recherche lors de la détection d'images par exemple.

L'intérêt de cette méthode dans le cas du TAL est principalement de détecter des motifs récurrents dans une phrase. On pourrait par exemple détecter l'intensification : «très bien», «vraiment mauvais», etc. On pourrait également détecter la négation : «j'aime pas», «pas mal», etc. Grâce à la couche de pooling, on peut également détecter des n-grams à distance : «autant ..., autant ...» ou encore «j'aime bien ..., mais». (Blunsom *et al.*, 2014) cumule des couches de convolution et de pooling pour faire de l'analyse sémantique et (Kalchbrenner *et al.*, 2014) utilise la même approche pour la modélisation de phrases.

3.2 Architecture du réseau utilisé

Dans le travail suivant, nous nous inspirons du modèle proposé par (Kim, 2014). Il utilise un réseau peu profond comprenant 3 couches de convolution, non pas successives mais mises en parallèle, avec des filtres de taille 3, 4 et 5 sur la longueur de la phrase et de taille 300 sur la taille du vecteur-mot (ce qui correspond en fait à la longueur des vecteurs). Une couche de pooling est placée après chaque couche de convolution. [1]

Enfin, pour que l'algorithme prenne une décision finale sur le sentiment de la phrase, on lui ajoute une couche de neurones avec la fonction d'activation softmax (notée σ), définie comme tel :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{k=1}^{|C|} e^{z_k}}$$

où $|C|$ est le nombre de classes. Le softmax permet de transformer la sortie du réseau en une distribution de probabilité sur les différentes catégories possible, en les normalisant.

Avant la couche de softmax, on rajoute également une couche de dropout, technique popularisée par (Srivastava *et al.*, 2014), qui permet d'améliorer la généralisation du modèle, mais rend l'apprentissage un peu plus long. Le réseau prend en entrée la matrice représentant la phrase avec pour chaque colonne la représentation vectorielle du mot correspondant (voir figure 2).

Kim a effectué un certain nombre d'expérimentations avec cette architecture :
- Il tente d'abord d'apprendre les vecteurs-mots en partant de zéro ;
- Il utilise ensuite directement les vecteurs appris par Mikolov en les laissant inchangés ;
- Enfin, il utilise les vecteurs de Mikolov et les ré-apprend au cours de l'apprentissage.

1. On peut se poser la question de l'utilité du calcul de filtres de taille 3 et 4 vu qu'on pourrait théoriquement détecter les mêmes caractéristiques avec un filtre de taille 5 et une couche de pooling. Cependant, on constate empiriquement que le temps d'apprentissage est alors plus court. Le réseau de neurones «sait» déjà quoi détecter, ce qui accélère les choses.

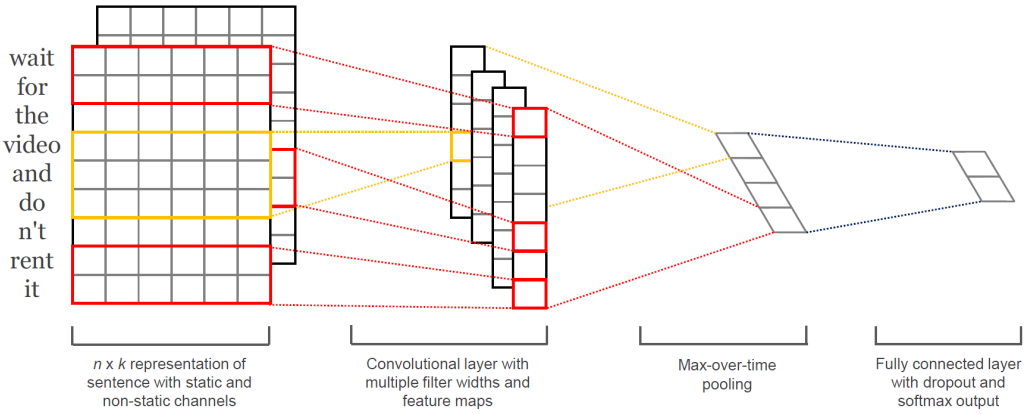

FIGURE 2 – Le réseau utilisé par (Kim, 2014)

C'est avec cette dernière approche qu'il obtient ses meilleurs résultats. En effet, les vecteurs de Mikolov sont de bonne qualité mais ne sont pas appris dans le but particulier de l'analyse d'opinions et possèdent donc naturellement du bruit. En s'inspirant de ce résultat, nous voulons conserver la possibilité pour le réseau de re-modifier les vecteurs de Mikolov. En revanche, nous ne voulons pas modifier les vecteurs précédemment appris car nous voulons pouvoir les réutiliser pour une autre tâche. De plus, nous souhaiterions diminuer la taille des vecteurs pour minimiser le temps d'apprentissage. La solution que nous avons mise en œvre est de rajouter une couche de neurones en entrée qui va faire passer la taille des vecteurs de 300 à 30, tout en allant récupérer l'information pertinente pour l'analyse d'opinions.

4 Jeu de données et protocole expérimental

4.1 Présentation et pré-processing du jeu de données

Le jeu de données du Défi Fouille de Texte DEFT 2015 est un ensemble de 15 000 tweets en français portant sur le thème de l'écologie. Ces tweets sont triés en trois classes (positif, neutre, négatif).

Toujours dans l'optique d'injecter le moins d'information linguistique *à la main*, nous n'effectuons que très peu de pré-traitements. Nous retirons les URLs ainsi que les mentions Twitter (de type `@proxem`). Nous appliquons enfin un tokenizer standard pour le français. Nous laissons le modèle apprendre les caractéristiques utiles à l'analyse d'opinions avec une approche non-supervisée.

4.2 Hyperparamètres du modèle

Nous utilisons les *rectified linear units* introduites dans (Nair & Hinton, 2010) comme fonctions d'activation des neurones. Une fois passée la première couche faisant passer la taille des vecteurs-mots de 300 à 30, on effectue l'opération de convolution avec 100 filtres linéaires de taille 3, 100 filtres linéaires de taille 4, 100 filtres linéaires de taille 5.

Le résultat du passage de chacun de ces filtres est ce qu'on appelle un *feature map* qui sera un vecteur de taille égale à celle de la phrase en entrée. Comme cette taille varie selon les tweets et que nous avons besoin d'une taille fixe pour appliquer la couche de softmax, l'opération de pooling va sélectionner la composante maximale de ce vecteur. Grâce à cette opération, nous forçons le réseau à sélectionner la caractéristique la plus importante de la phrase et nous obtenons un vecteur de taille fixe (taille que nous avons choisie égale à 1) pour chaque filtre. Une fois passée la couche de pooling notre réseau n'a donc plus connaissance des positions relatives des différentes expressions capturées par chaque filtre.

A l'issue de ces couches de convolution, nous obtenons finalement un vecteur concaténé de taille 300 (correspondant au nombre total de filtres choisi), que nous donnons en entrée d'une couche de softmax, à laquelle on rajoute une couche de dropout avec un coefficient de dropout de 0.5. Nous rajoutons également une contrainte sur la norme L^2 des poids du

modèle [2] pour qu'elle ne dépasse pas 3 lors de la descente de gradient.

L'apprentissage se fait par descente de gradient stochastique, avec des paquets de taille 50 et la règle de descente Adadelta (Zeiler, 2012). Enfin, nous arrêtons l'apprentissage en suivant l'évolution des performances par cross-validation.

5 Résultats

5.1 Comparaison avec un SVM utilisant des sacs de mots

Nous avons comparé les résultats obtenus dans notre expérience avec un certain nombre de modèles utilisant des sacs de mots et la pondération TF-IDF. Certains de ces modèles sont linéaires (régression logistique, SVM à noyau linéaire), d'autres non (forêts aléatoires). La métrique utilisée est la macro-précision, définie comme tel :

$$P_{macro} = \frac{1}{|C|} \sum_{i=1}^{|C|} \frac{VP_i}{VP_i + FP_i}$$

où $|C|$ est le nombre de classes, VP_i est le nombre de vrais positifs pour la classe i et FP_i est le nombre de faux positifs pour la classe i.

Modèles	Macro-Précision
Réseau bayésien	44.7
SVM	41.8
Forêt aléatoire	46.0
Régression logistique	41.4
Réseau Convolutionnel	**69.9**

TABLE 3 – Résultats obtenus

Bien que peu profond et alimenté de peu d'informations linguistiques, notre réseau de neurones surpasse les techniques classiques sur cette tâche comme le montre la table 3.

5.2 Durée d'apprentissage

Un autre point positif est que grâce à notre première couche qui réduit la taille de nos vecteurs-mots et du fait que le réseau soit peu profond, notre algorithme converge assez rapidement, puisqu'en 30 minutes de calcul sur CPU (serveur avec 16 cœrs) nous arrivons presque au score obtenu plus haut.

6 Conclusion

Nous avons présenté dans cet article un modèle d'analyse d'opinions faiblement supervisé, se fondant sur un simple réseau convolutionnel de neurones à une couche de convolution et prenant en entrée des vecteurs-mots appris sur la Wikipedia française ainsi que d'autres sources selon la méthode introduite par Mikolov. Nous avons obtenu des résultats encourageants sur un jeu de données de tweets en français sur le thème de l'écologie. Notre approche pourrait être évaluée sur d'autres thèmes ou en monde ouvert. Ces résultats nous conduisent à poursuivre notre travail sur les vecteurs-mots et les réseaux convolutionnels. Nous sommes par ailleurs confiants quant à l'apport des réseaux de neurones récurrents sur lesquels nous travaillons également.

2. $\forall \vec{x} \in R^n,\ \ L^2(\vec{x}) = \sum_{0 \leq i < n} x_i^2$

Références

BACCIANELLA S., ESULI A. & SEBASTIANI F. (2010). Sentiwordnet 3.0 : An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, p. 2200–2204.

BAUCOM E., SANJARI A., LIU X. & CHEN M. (2013). Mirroring the real world in social media : twitter, geolocation, and sentiment analysis.

BLUNSOM P., DE FREITAS N., GREFENSTETTE E., HERMANN K. M. *et al.* (2014). A deep architecture for semantic parsing. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing* : Proceedings of the ACL 2014 Workshop on Semantic Parsing.

CHURCH K. W. & HANKS P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, **16**(1), 22–29.

G. DIAS, Ed. (2015). *Actes de TALN 2015 (Traitement automatique des langues naturelles)*, Caen. ATALA, HULTECH.

DODDS P. S., HARRIS K. D., KLOUMANN I. M., BLISS C. A. & DANFORTH C. M. (2011). Temporal patterns of happiness and information in a global social network : Hedonometrics and twitter. *PloS one*, **6**(12), e26752.

FAUSTO S. & AVENTURIER P. (2015). Scientific literature on twitter as subject research : preliminary findings based on bibliometric analysis. In *Twitter for Research 2015*, p.Ĩp.

KALCHBRENNER N., GREFENSTETTE E. & BLUNSOM P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv :1404.2188*.

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

KRAMER A. D., GUILLORY J. E. & HANCOCK J. T. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, **111**(24), 8788–8790.

KRIZHEVSKY A., SUTSKEVER I. & HINTON G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, p. 1097–1105.

LECUN Y. & BENGIO Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, **3361**, 310.

LEVY O. & GOLDBERG Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, p. 2177–2185.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

MOHAMMAD S. M., KIRITCHENKO S. & ZHU X. (2013). Nrc-canada : Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR'13)*.

NAIR V. & HINTON G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, p. 807–814.

PAK A. & PAROUBEK P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, volume 10, p. 1320–1326.

PENNEBAKER J. W., FRANCIS M. E. & BOOTH R. J. (2001). Linguistic inquiry and word count : Liwc 2001. *Mahway : Lawrence Erlbaum Associates*, **71**, 2001.

PSOMAKELIS E., TSERPES K., ANAGNOSTOPOULOS D. & VARVARIGOU T. (2015). Comparing methods for twitter sentiment analysis. *arXiv preprint arXiv :1505.02973*.

SCHMITT E. (2015). Elements for an epistemology of instrumentation and collaboration in Twitter data research. 1st International Conference on Twitter for Research.

SRIVASTAVA N., HINTON G., KRIZHEVSKY A., SUTSKEVER I. & SALAKHUTDINOV R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**(1), 1929–1958.

TUFEKCI Z. (2014). Big questions for social media big data : Representativeness, validity and other methodological pitfalls. *arXiv preprint arXiv :1403.7400*.

WU C., SKOWRON M. & PETTA P. (2014). Reading between the lines.

ZEILER M. D. (2012). Adadelta : an adaptive learning rate method. *arXiv preprint arXiv :1212.5701*.

ADVANSE : Analyse du sentiment, de l'opinion et de l'émotion sur des Tweets Français

Amine Abdaoui[1] Mike Donald Tapi Nzali[1, 2] Jérôme Azé[1] Sandra Bringay[1] Christian Lavergne[2] Caroline Mollevi[3] Pascal Poncelet[1]
(1) LIRMM UM CNRS, UMR 5506, 161 Rue Ada, 34095 Montpellier, France
(2) Institut de Mathématiques et de Modélisation de Montpellier, Université Montpellier, France
(3) Unité de biostatistique, Institut de Cancérologie de Montpellier, France
amin.abdaoui@lirmm.fr, mike-donald.tapi-nzali@lirmm.fr

Résumé. Ce papier décrit les systèmes que nous avons soumis au défi DEFT 2015 (Défi Fouille de Texte). Cette onzième édition a porté sur l'analyse de l'opinion, du sentiment et de l'émotion dans des tweets rédigés en Français. Le défi propose trois tâches, nous avons participé à la tâche 1 qui concerne la classification des tweets selon leur polarité, à la tâche 2.1 qui concerne l'identification de la classe générique de l'information exprimée dans les tweets et enfin à la tâche 2.2 qui concerne l'identification de la classe spécifique de l'opinion, du sentiment ou de l'émotion présente dans les tweets. Nous avons proposé des méthodes supervisées basées sur les machines à vecteurs de support (SVM) utilisant plusieurs types d'attributs comme les n-grammes de mots, les n-grammes de caractères, les patrons syntaxiques les plus fréquents, etc. Nous avons également construit et utilisé des lexiques de sentiments et d'émotions spécifiques pour le français.

Abstract.

ADVANSE : Sentiment, Opinion and Emotion Analysis in French Tweets
This paper describes the methods we submitted to the DEFT 2015 Challenge (Text Mining Challenge). This eleventh edition concerned the analysis of opinions, sentiments and emotions expressed in French tweets. Three tasks have been proposed, we participated to task 1 which concerned the classification of tweets according to their polarities, to task 2.1 concerning the identification of the generic class of information expressed in the tweet, and finally to task 2.2 that concerned the identification of the specific class of opinion, sentiment or emotion. We proposed supervised methods based on support vector machines (SVM) using several types of attributes such as word n-grams, character n-grams, most common syntactic patterns, etc. Moreover, we constructed and used two French lexicons of sentiments and emotions.

Mots-clés : Analyse de sentiments, analyse d'opinions, analyse d'émotions, analyse de subjectivité, fouille de textes.

Keywords: Sentiment analysis, opinion analysis, emotion analysis, subjectivity analysis, text mining.

1 Introduction

L'analyse automatique de textes pour y détecter la présence d'états affectifs, leur polarité, les émotions associées et les opinions exprimées a suscité de nombreux travaux ces dernières années. Cet intérêt toujours croissant peut s'expliquer, en partie, par les perspectives d'applications qu'elle ouvre dans de nombreux domaines tels que : les systèmes de recommandation en ligne (Tatemura, 2000), l'intelligence économique (Melnik & Alm, 2002), la veille politique et gouvernementale (Laver *et al.*, 2003; Mullen & Malouf, 2006), etc. Les méthodes appliquées sont généralement spécifiques aux types de textes traités : aux tweets (Roberts *et al.*, 2012), aux titres de presse (Strapparava & Mihalcea, 2008), aux commentaires des internautes (Vincent & Winterstein, 2013), etc. Si la majorité des méthodes proposées ont été créées pour l'Anglais et pour la polarité, quelques travaux plus récents existent aussi pour le Français (Grouin *et al.*, 2009) et pour les émotions (Mohammad, 2012). Dans cet article, nous nous intéressons à la classification de tweets en langue française. Nous présentons des systèmes que nous avons soumis à la onzième édition du défi de fouille de textes (DEFT 2015). La première tâche (tâche 1) à laquelle nous avons participé consiste à classifier les tweets selon leur polarité (*positif, négatif* ou *neutre*). Dans la seconde tâche (tâche 2.1), il s'agit de détecter la classe générique exprimée dans un tweet (*information, opinion,*

sentiment ou *émotion*). La dernière tâche (tâche 2.2) à laquelle nous avons participé consiste à détecter la classe spécifique de l'opinion, du sentiment ou de l'émotion exprimée dans le tweet (18 classes ont été considérées telle que : *colère, peur, tristesse, amour, plaisir surprise, accord, etc.*). Nous avons obtenu 73,33% de macro-précision pour la tâche 1 à 0,27% du meilleur système. Pour la tâche 2.1 nous avons obtenu le meilleur système avec 61,29% de macro-précision. Cependant, nous n'avons pas considéré les bonnes classes lors de notre soumission pour la tâche 2.2, nous présentons ici nos résultats avec les bonnes classes. Les données d'apprentissage et de test ont été fournies par les organisateurs de la compétition ; les tableaux 1 et 2 montrent la distribution des classes sur le corpus d'apprentissage et de test.

Tâche 1					**Tâche 2.1**				
Classes	**Apprentissage**		**Test**		**Classes**	**Apprentissage**		**Test**	
	#	%	#	%		#	%	#	%
Positif	2448	31	1057	31	Émotion	820	12	351	10
Négatif	1875	24	804	24	Information	3571	53	1518	45
Neutre	3544	45	1518	45	Sentiment	2275	34	537	16
					Opinion	82	1	973	29
Total	**7867**	**100**	**3379**	**100**	**Total**	**6754**	**100**	**3379**	**100**

TABLE 1 – Distribution des classes pour la tâche 1 et la tâche 2.1 sur le corpus d'apprentissage et de test. Les données ont été accessibles par les identifiants des tweets et un script de téléchargement (fourni par les organisateurs)

Tâche 2.2				
Classes	**Apprentissage**		**Test**	
	#	%	#	%
déplaisir	47	67,72	21	1,54
dérangement	13	0,4	6	0,44
mépris	176	5,53	75	5,51
surprise négative	10	0,31	4	0,29
peur	274	8,60	114	8,37
colère	210	15,16	87	6,39
ennui	4	0,12	2	0,14
tristesse	36	1,13	16	1,17
plaisir	35	1,10	15	1,10
apaisement	9	0,28	5	0,36
amour	8	0,25	4	0,29
surprise positive	4	0,12	2	0,14
satisfaction	73	2,29	32	2,35
insatisfaction	9	0,28	5	0,36
accord	154	4,83	67	4,92
valorisation	1504	47,25	644	20,23
désaccord	216	6,77	92	6,76
dévalorisation	401	12,60	170	12,49
Total	**3183**	**100**	**1361**	**100**

TABLE 2 – Distribution des classes pour la tâche 2.2 sur le corpus d'apprentissage et de test. Les données ont été accessibles par les identifiants des tweets et un script de téléchargement (fourni par les organisateurs)

Les méthodes d'analyse de sentiments, d'opinions et d'émotions sont généralement basées sur des techniques statistiques, de traitement automatique du langage et d'apprentissage supervisé. Les plus performantes se basent souvent sur des systèmes supervisés qui utilisent des lexiques adaptés (Nakov *et al.*, 2013; Rosenthal *et al.*, 2014). Dans ce travail, nous avons choisi d'utiliser les machines à vecteurs de support (SVM) tout en exploitant deux lexiques de sentiments (polarités) et d'émotions que nous avons construit au préalable. Le premier est un lexique de sentiments et d'émotions qui a été construit en traduisant le lexique anglais NRC (Mohammad & Turney, 2010) d'une manière semi-automatique, supervisée par un traducteur humain expérimenté. Ce lexique a été étendu en anglais (avant traduction) et en français (après traduction) via l'étude des synonymes et des antonymes. Le deuxième est un lexique de sentiments (polarités seulement) construit d'une

manière automatique en utilisant le corpus d'apprentissage fourni par l'équipe d'organisation de DEFT 2015. Nous avons ensuite construit des attributs spécifiques pour prendre en considération ces deux lexiques dans l'apprentissage de nos modèles de classification. En plus des attributs exploitant les lexiques, nous en avons testé d'autres : les unigrammes de mots, les n-grammes de caractères (de longueur 5 et 6), le nombre d'émoticônes, le nombre de mots en majuscules, le nombre de lettres répétées, le nombre de hashtags, la présence de négateurs et les patrons syntaxiques les plus fréquents. Enfin, une étape de sélection d'attributs a été appliquée pour sélectionner les attributs les plus pertinents pour chaque tâche en ne conservant que les attributs pour lesquels le gain d'information est positif.

Le reste de l'article sera organisé comme suit : la section 2 décrit la création des ressources de sentiments et d'émotions utilisées. La section 3 présente les méthodes proposées : prétraitements, constructions et sélection d'attributs, classification. La section 4 présente les configurations choisies pour chaque tâche et les résultats obtenus. Enfin, la section 5 conclut et donne nos principales perspectives.

2 Création des ressources

Beaucoup de travaux liés à l'analyse de sentiments se basent sur des lexiques d'expressions de sentiments (liste de mots, phrases, idiomes, etc.). À ce titre, (Mohammad *et al.*, 2013) ont souligné l'importance des lexiques dans la classification des tweets suivant leur polarité. En effet, d'auteurs ont observé une augmentation qui dépasse les 8% de leurs macro F-scores en utilisant des lexiques qu'ils ont construit de manière manuelle et automatique. Cependant, la majorité de ces ressources a été construite pour l'anglais et la polarité. Très peu de lexiques existent pour le français et quand ils existent, ils ne contiennent pas beaucoup de termes. Pour cela, nous avons créé nous-mêmes deux ressources pour le français : le lexique de sentiments et d'émotions FEEL (Abdaoui *et al.*, 2014) et le lexique de sentiments basé sur l'information mutuelle (Church & Hanks, 1990).

2.1 Lexique de sentiments et d'émotions FEEL

Ce lexique a été construit de manière semi-automatique en traduisant et en étendant le lexique de sentiments et d'émotions anglais NRC (Mohammad & Turney, 2010). D'autres lexiques anglais existent à l'image de WordNet Affect (Strapparava *et al.*, 2004) mais à notre connaissance, NRC-2010 est le plus complet. Il inclut à la fois les polarités et les émotions ; surtout donne de bons résultats pour des tâches similaires aux nôtres (Kiritchenko *et al.*, 2014a). NRC-2010 associe à chaque terme une polarité (*positive* ou *négative*) et une ou plusieurs émotions parmi les six émotions proposées par (Ekman, 1992), à savoir : *joie, colère, tristesse, dégoût, surprise* et *peur*. Il a été construit manuellement en utilisant le service Amazon Mechanical Turk [1]. Afin de construire une ressource similaire pour le français, nous avons adopté une approche de traduction et d'extension semi-automatique. D'abord, pour chaque entrée de la ressource initiale, nous avons interrogé six traducteurs automatiques (Google, Bing, Collinsdictionary, Reverso, Babla et Wordreference). Les traductions ont été validées ou pas par un traducteur humain expérimenté. Le traducteur avait la possibilité de changer les polarités et les émotions associées via une interface graphique. Ensuite, nous avons émis l'hypothèse que la polarité était conservée par la synonymie. Nous avons donc étendu notre lexique aux synonymes. Huit outils en ligne ont été utilisés pour la recherche des synonymes (Babla, le dico de l'Institut des Sciences Cognitives, reverso, sensagent, cnrtl, synonym, thefreedictionary et thesaurus). Enfin, nous avons étendu notre lexique aux antonymes (en inversant les polarités) et en utilisant encore une fois deux outils en ligne (antonyme et cnrtl). Finalement, nous avons obtenu une ressource avec plus de 14 000 termes distincts (lemmatisés), chacun associé à une polarité et à une ou plusieurs émotions [2].

2.2 Lexique de sentiments basé sur l'information mutuelle

Étant donné que les résultats des systèmes basés sur les lexiques dépendent du type de données et du domaine d'application (Pang & Lee, 2008), nous avons décidé de construire automatiquement un lexique de sentiments en utilisant le corpus d'apprentissage fourni pour ce défi. Nous avons implémenté et adapté l'approche de (Kiritchenko *et al.*, 2014b). Au lieu de calculer un seul score pour chaque mot *w*, nous avons calculé trois scores (un par polarité) pour permettre la prise en compte de la classe neutre. Les trois scores sont présentés dans les formules (1), (2) et (3). Chaque score est basé sur la

1. https://www.mturk.com/mturk/welcome
2. Ce lexique est en téléchargement libre sur le lien : http://www.lirmm.fr/ abdaoui/FEEL.html

PMI (Pointwise Mutual Information (Bouma, 2009)) du mot dans un ensemble de tweets comme présenté dans la formule (4).

$$ScorePositif(w) = PMI(w, \{positif\}) - PMI(w, \{negatif\} \cup \{neutre\}) \quad (1)$$

$$ScoreNegatif(w) = PMI(w, \{negatif\}) - PMI(w, \{positif\} \cup \{neutre\}) \quad (2)$$

$$ScoreNeutre(w) = PMI(w, \{neutre\}) - PMI(w, \{positif\} \cup \{negatif\}) \quad (3)$$

$$PMI(w, \{NomDeLaClasse\}) = \log_2 \frac{freq(w, \{NomDeLaClasse\}) * N}{freq(w) * freq(\{NomDeLaClasse)\}} \quad (4)$$

où *freq(w, {NomDeLaClasse)}* est le nombre de fois où le mot *w* apparaît dans un tweet appartenant à *{NomDeLaClasse}*, *freq(w)* est le nombre d'apparitions du mot *w* dans le corpus, *freq ({NomDeLaClasse)}* est le nombre de tweets appartenant à la classe *NomDeLaClasse* et *N* est le nombre total de termes dans le corpus.

3 Méthodes

Pour chacune de nos trois tâches, nous appliquons d'abord des prétraitements. Ensuite, nous construisons des attributs pertinents pour la tâche en question. Une fois les attributs construits, nous ne sélectionnons que les plus pertinents. Enfin, nous apprenons un modèle de classification sur le corpus d'apprentissage et nous l'appliquons sur le corpus de test. Chacune de ces étapes est détaillée dans cette section.

3.1 Prétraitements

Comme indiqué par (Balahur, 2013), les textes issus des réseaux sociaux ont des particularités linguistiques qui peuvent influencer les performances de la classification. Pour cette raison, nous avons appliqué les prétraitements suivants : 1) le remplacement de tous les liens hypertextes qui figurent dans les tweets par 'lienHTPP' ; 2) le remplacement de toutes les adresses mails présentes dans les tweets par 'mail' ; 3) le remplacement de tous les tags utilisateur par '@tag' ; 4) la lemmatisation de tous les mots en utilisant l'outil TreeTagger (Schmid, 1994).

En outre, nous avons remarqué que certains prétraitements tels que le remplacement des mots d'argots, la mise en minuscules et le remplacement des mots allongés font chuter les macro-précisions (voir section 4). De ce fait, ces derniers n'ont pas été considéré dans nos soumissions.

3.2 Attributs

Pour la construction de nos attributs, nous avons d'abord testé ceux utilisés par (Mohammad *et al.*, 2013) lors du défi SemEval-2013 sur des tweets en langue anglaise en validations croisées sur le corpus d'apprentissage. Ensuite, nous les avons adaptés pour le français et nous en avons rajouté d'autres. Chaque tweet a donc été représenté par un sous ensemble des attributs suivants :

— **Unigrammes de mots** : présence ou absence des mots lemmatisés dans le tweet.
— **Caractères n-grammes** : présence ou absence des séquences continues de 5 et de 6 caractères. Nous n'avons sélectionné que les séquences les plus fréquentes par classe (celles qui apparaissent au moins 550 fois dans une seule classe). Ce seuil a été fixé par plusieurs validations croisées sur le corpus d'apprentissage.
— **Les patrons syntaxiques les plus fréquents** : pour construire ces attributs, nous avons d'abord utilisé l'outil TreeTagger pour remplacer chaque mot par son étiquette morphosyntaxique. Ensuite, nous avons utilisé l'algorithme de (Fournier-Viger *et al.*, 2008) pour l'extraction de motifs séquentiels fréquents. Nous en avons extrait les 1% les plus fréquents pour chaque classe. Nous n'avons gardé que les patrons discriminants, fréquents dans une classe et pas dans les autres. Ces derniers ont été rajoutés comme attributs de type « booléen ». Chaque attribut prendra la valeur *vraie* si l'étiquetage morphosyntaxique du tweet correspond au patron syntaxique en question.

— **Lexique de sentiments FEEL** : les attributs suivants ont été considérés : 1) le nombre de mots positifs ; 2) le nombre de mots négatifs ; 3) la somme des scores de tous les mots ; 4) le score le plus élevé ; 5) le score du dernier mot comme cela a été fait par (Mohammad *et al.*, 2013).
— **Lexique d'émotions FEEL** : les attributs suivants ont été considérés : 1) le nombre de mots exprimant la confiance ; 2) le nombre de mots exprimant la joie ; 3) le nombre de mots exprimant la colère ; 4) le nombre de mots exprimant la tristesse ; 5) le nombre de mots exprimant le dégoût ; 6) le nombre de mots exprimant la surprise ; 7) le nombre de mots exprimant la peur.
— **Lexique basé sur l'information mutuelle** : les attributs suivants ont été considérés pour ce lexique : 1) le nombre de mots positifs ; 2) le nombre de mots négatifs ; 3) le nombre de mots neutres ; 4) la somme de tous les scores positifs ; 5) la somme de tous les scores négatifs ; 6) la somme de tous les scores neutres ; 7) le score positif le plus élevé ; 8) le score négatif le plus élevé ; 9) le score neutre le plus élevé ; 10) le score positif du dernier mot ; 11) le score négatif du dernier mot ; 12) le score neutre du dernier mot.
— **Ponctuation** : deux attributs ont été considérés : 1) le nombre de séquences continues de points d'exclamation, de points d'interrogation et des deux ; 2) la présence ou l'absence d'un point d'exclamation ou d'un point d'interrogation dans le dernier terme dans n'importe quel position.
— **Émoticônes** : présence ou non d'un émoticône dans le tweet.
— **Les mots allongés** : nombre de mots avec au moins 3 caractères répétés séquentiellement (par exemple : *mdrrrr*).
— **Négation** : présence ou l'absence d'un négateur dans le tweet.

Concernant la négation, nous avons aussi essayé d'implémenter la méthode proposée dans (Kiritchenko *et al.*, 2014a) en rajoutant un suffixe aux mots qui se trouvent sous la portée d'un négateur. Les résultats de cette approche sont décrits dans la section 4.

3.3 Sélection d'attributs

Afin de sélectionner les attributs les plus discriminants pour chaque tâche, nous avons appliqué une étape de sélection d'attributs en mesurant le gain d'information de chaque attribut par rapport à la classe (Mitchell, 1997). L'équation 5 présente la formule utilisée :

$$GainInformation(attribut, classe) = Entropie(classe) - Entropie(classe|attribut) \quad (5)$$

Après avoir calculé le gain d'information pour chaque attribut, nous ne gardons que ceux pour lesquels ce gain est supérieur à 0. Pour la tâche 1, 762 attributs ont été sélectionnés dont les plus discriminants sont ceux obtenus à partir du lexique de la PMI suivis par ceux obtenus à partir du lexique FEEL. Pour la tâche 2.1, 460 attributs ont été sélectionnés dont les plus discriminants sont des mots comme : *menacer, contre, lien,* etc. Pour la tâche 2.2, 75 attributs ont été sélectionnés dont les plus discriminants sont ceux issus du lexique d'émotions FEEL et des mots comme : *menacer, espèce, lien,* etc.

3.4 Classification

Pour nos trois tâches, nous employons des méthodes d'apprentissage supervisé. Nous avons choisi d'utiliser les machines à vecteur de support SVM (Support Vector Machine) avec la méthode SMO (Sequential Minimal Optimization) (Platt, 1999) implémenté dans Weka (Hall *et al.*, 2009). D'après l'état de l'art, cet algorithme d'apprentissage s'est avéré efficace sur des tâches de catégorisation de textes et spécifiquement d'analyse de sentiments et d'émotions. Il est robuste sur les grands espaces de caractéristiques. En effet, notre modèle de classification exploite une variété de surface de forme, sémantique et des sentiments caractéristiques issus des lexiques.

Afin de choisir le meilleur paramètre de complexité « C » de SVM, nous avons effectué 20 validations croisées à 10 plis sur le corpus d'apprentissage. Pour chaque tâche, nous avons testé toutes les valeurs comprises entre 0,1 et 2,0 avec un pas de 0,1. Pour les trois tâches, la meilleure macro précision en validation croisée a été obtenue avec la valeur 0,4. Nous avons donc choisi cette valeur pour le paramètre de complexité « C » dans les expérimentations décrites ci-dessous.

4 Expérimentations

Dans cette section, nous présentons les configurations choisies pour chaque tâche, les résultats obtenus et leurs discussions.

4.1 Tâche 1 : Détection de la polarité d'un tweet

L'objectif de cette tâche est de déterminer la polarité d'un tweet. Il s'agit de prédire si un tweet donné est *positif, négatif* ou *neutre*. Les attributs utilisés pour cette tâche sont les suivants : les n-grammes de mots, les n-grammes de caractères, les patrons syntaxiques, le lexique de sentiments FEEL, le lexique de sentiments de la PMI, la ponctuation, les émoticônes, les mots allongés et la négation (attribut booléen). Le tableau 3 présente les macro-précisions de plusieurs expérimentations que nous avons effectué pour déterminer l'effet des attributs que nous avons construit et des étapes que nous avons effectué. Nous testons plusieurs configurations en rajoutant ou en enlevant des attributs ou des étapes.

Expérimentations	**Macro-précision**
Baseline (prédire la classe majoritaire)	33,33%
Système proposé	73,33%
- Unigrammes de mots lemmatisés	42,39% (-30,94%)
- N-grammes de caractères de longueur 5 et 6	73,46% (+0,13%)
- Lexique de polarités FEEL	72,45% (-0,88%)
- Lexique de polarités PMI	73,32% (-0,01%)
- Patrons syntaxiques les plus fréquents	72,32%(-0,01%)
- Négation (attribut booléen)	73,33% (-0,00%)
- Ponctuation, émoticônes, hashtags et mots allongés	73,45% (+0,12%)
- Sélection d'attributs	68,92% (-4,41%)
- Prétraitements {Lemmatisation, lien, mail, tag}	70,71% (-2,62%)
+ Prétraitements {Argots, minuscules, mots allongés}	73,00% (-0,33%)
+ Négation (rajouter un suffixe)	73,15% (-0,18%)

TABLE 3 – Macro-Précisions obtenues pour la tâche 1

Nous avons entraîné des classifieurs de types SVM sur un ensemble de 7 867 tweets, puis nous avons appliqué les modèles appris sur les 3 379 tweets qui nous ont été fournis pour la phase de test. La mesure choisie par les organisateurs du défi est la macro- précision. Comme baseline, nous considérons un système qui prédit toujours la classe majoritaire (ici la classe neutre). La macro-précision obtenue par un tel système est de 33,33%. Concernant le système que nous avons soumis au défi, la macro-précision obtenue est de 73,33%. Nous constatons que les attributs donnant le plus de gain sont les *unigrammes de mots lemmatisés*, ils fournissent un gain de plus de 30,94%. Par contre, le fait d'enlever les n-grammes caractères fait augmenter la macro-précision de 0,13%. Ces derniers ne permettent donc pas d'améliorer les résultats, voir ils les font même chuter un peu. Le lexique de polarités FEEL permet d'obtenir un gain de 0,88%. Concernant la négation en utilisant un attribut booléen, les patrons syntaxiques les plus fréquents et le lexique de polarités de la PMI, ces attributs n'ont pas vraiment eu d'impact sur les performances de notre système. La ponctuation, les émoticônes, le nombre de hashtags et de mots allongés font chuter la macro-précision du système de 0,12%, alors que la sélection d'attributs parait une étape cruciale puisqu'elle permet d'obtenir un gain de 4,41%. Les prétraitements que nous avons appliqués nous ont permis d'obtenir un gain non négligeable de 2,62%. Finalement, le fait de rajouter les prétraitements que nous avons décidé d'écarter et la méthode de traitement de la négation par ajout de suffixe fait chuter la macro-précision de 0,33% et de 0,18% respectivement.

4.2 Tâche 2.1 : Identification de la classe générique

Il s'agit ici d'identifier la classe générique de l'information exprimée dans un tweet. Les 4 classes génériques proposées dans le cadre de cette tâche sont : *information, opinion, sentiment* et *émotion*. Les attributs utilisés pour cette tâche sont les suivants : les n-grammes de mots, les n-grammes de caractères, la ponctuation, les émoticônes, les mots allongés et la négation. Le tableau 4 présente les macro-précisions de plusieurs expérimentations que nous avons effectué pour

déterminer l'effet des attributs que nous avons construit et des étapes que nous avons proposé pour la tâche 2.1. Nous testons plusieurs configurations en rajoutant ou en enlevant des attributs ou des étapes.

Expérimentations	**Macro-précision**
Baseline (prédire la classe majoritaire)	25,00%
Système proposé	61,29%
- Unigrammes de mots lemmatisés	11,23% (-50,06%)
- N-grammes de caractères de longueur 5 et 6	61,45% (-0,16%)
- Négation (attribut booléen)	61,32% (+0,03%)
- Ponctuation, émoticônes, hashtags et mots allongés	61,17% (-0,12%)
- Sélection d'attributs	55,67% (-5,62%)
- Prétraitements {Lemmatisation, lien, mail, tag}	62,27% (+0,98%)
+ Prétraitements {Argots, minuscules, mots allongés}	62,16% (+0,87%)
+ Négation (rajouter un suffixe)	61,26% (-0,03%)

TABLE 4 – Macro-Précisions obtenues pour la tâche 2.1

Nous avons entraîné des classifieurs de types SVM sur un ensemble de 6 784 tweets, puis nous avons appliqué nos modèles sur les 3 379 tweets qui nous ont été fournis après pour la phase de test. Nous utilisons la même baseline qui consiste à prédire la classe majoritaire ce qui donnera cette fois 25% de macro-précision (puisqu'il y a quatre classes). La macro-précision obtenue par le système soumis au défi est de 61,29% (meilleur résultat pour cette tâche). Cependant, le tableau 4 montre qu'il est possible d'améliorer encore ce résultat pour dépasser les 62%. Les unigrammes de mots lemmatisés donnent le meilleur gain (plus de 50%). Ensuite et comme pour la première tâche, en faisant une sélection d'attributs on augmente les résultats de plus de 5%, ce qui rend cette étape aussi cruciale pour la tâche 1 que pour la tâche 2.1. Les n-grammes de caractères donnent de leur côté un faible de gain de 0,16%. Par ailleurs, la négation par attribut booléen et par ajout de suffixe n'a pas vraiment d'impact sur la macro-précision. La ponctuation, les émoticônes, le hashtags et les mots allongés font chuter la macro-précision de 0,12%. Les prétraitements que nous avons choisi d'appliquer (à savoir : la lemmatisation, le remplacement des liens, des mails, et des tags) semblent être inadaptés pour cette tâche puisqu'en les appliquant nous perdons 0,92%, alors qu'en appliquant les autres prétraitements (remplacement des mots d'argots, la mise en minuscules et le traitement des mots allongés) qui permettent d'obtenir un gain de 0,87%.

4.3 Tâche 2.2 : Identification de la classe spécifique de l'opinion, du sentiment ou de l'émotion

Il s'agit d'identifier la classe de l'opinion, sentiment ou émotion. Étant donné un tweet, il faudrait reconnaître l'opinion/sentiment/émotion principal(e) exprimé(e) explicitement dans ce tweet. Pour cela, 18 classes sont proposées : *colère, peur, tristesse, dégoût, ennui, dérangement, déplaisir, surprise négative, apaisement, amour, plaisir, surprise positive, insatisfaction, satisfaction, accord, valorisation, désaccord et dévalorisation*. Lors de notre soumission nous n'avons pas considérés ces classes-là, nous avons donc obtenu des macro-précisions très faibles. Nous présentons ici les résultats du même système soumis mais en considérant les bonnes classes. Les attributs utilisés pour cette tâche sont les suivants : les n-grammes de mots, les n-grammes de caractères, le lexique d'émotions FEEL, la ponctuation, les émoticônes, les mots allongés et la négation. Le tableau 5 présente les macro-précisions de plusieurs expérimentations que nous avons effectué pour déterminer l'effet des attributs que nous avons construit et des étapes que nous avons effectué pour la tâche 2.2. Nous testons plusieurs configurations en rajoutant ou en enlevant des attributs ou des étapes au système proposé.

Nous avons entraîné des classifieurs de types SVM sur un ensemble de 3 162 tweets, puis nous avons appliqué nos modèles sur les 3 379 tweets qui nous ont été fournis après pour la phase de test. Nous utilisons la même baseline qui consiste à prédire la classe majoritaire ce qui donnera cette fois 5,56% de macro-précision (puisqu'il y a 18 classes). La macro-précision obtenue par le système soumis au défi et appris sur les bonnes classes est de 31,72%. Cependant, le tableau 5 montre qu'il est possible d'améliorer encore ce résultat pour dépasser les 37%. Encore une fois, les unigrammes de mots lemmatisés donnent le meilleur gain (25,72%). Ensuite, le lexique de polarités FEEL, la ponctuation, les émoticônes, le hashtags et les mots allongés n'ont pas d'impact sur les résultats. Par contre, lexique d'émotions FEEL (qui contient 7 émotions seulement) et la négation par attributs booléen ont considérablement diminué la macro-précision. Le lexique a causé une perte de 2,66%, alors que la négation par attribut booléen a causé une perte de 5,28%. À l'opposé des deux tâches précédentes, la sélection d'attributs a causé une perte 0,66% pour la tâche 2.2. Par ailleurs, les prétraitements que nous avons choisis d'appliquer (à savoir : la lemmatisation, le remplacement des liens, des mails, et des tags) semblent être

Expérimentations	**Macro-précision**
Baseline (prédire la classe majoritaire)	5,56%
Système proposé	31,72%
- Unigrammes de mots lemmatisés	6,00% (-25,72%)
- Lexique de polarités FEEL	31,72% (0,00%)
- Lexique d'émotions FEEL	34,38% (+2,66%)
- Négation (attribut booléen)	37,00% (+5,28%)
- Ponctuation, émoticônes, hashtags et mots allongés	31,72% (0,00%)
- Sélection d'attributs	32,38% (+0,66%)
- Prétraitements {Lemmatisation, lien, mail, tag}	29,94% (-1,78%)
+ Prétraitements {Argots, minuscules, mots allongés}	35,93% (+4,21%)
+ Négation (rajouter un suffixe)	31,65% (-0,07%)

TABLE 5 – Macro-Précisions obtenues pour la tâche 2.2

adaptés pour cette tâche puisqu'en les appliquant nous obtenons un gain 1,78%. Les autres prétraitements (remplacement des mots d'argots, la mise en minuscules et le traitement des mots allongés) permettent d'obtenir un gain de 4,21%. Finalement et comme pour les deux premières tâches, la méthode de négation en rajoutant un suffixe à tous les mots qui se trouvent sous la portée d'un négateur fait chuter légèrement la macro-précision du système (0,07%).

5 Conclusion

Nous avons présenté les systèmes soumis à la onzième édition du défi de fouille de texte (DEFT 2015). Les systèmes proposés sont basés sur des classifieurs de types SVM exploitant des traits d'ordre morphologique, syntaxique et sémantique. De plus, nous avons construit et exploité deux lexiques de sentiments et d'émotions. Le premier est un lexique de sentiments et d'émotions obtenu en traduisant semi-automatiquement le lexique NRC alors que le deuxième est un lexique de sentiments obtenu d'une manière automatique en utilisant le corpus d'apprentissage. Le système soumis pour la tâche 1 a obtenu une macro-précision de 73,33% à 0,27% du meilleur système. Le système que nous avons proposé pour la tâche 2.1 a obtenu une macro-précision de 61,29% soit la meilleure macro-précision pour cette tâche. Le système soumis pour la tâche 2.2 et appris sur les bonnes classes a obtenu une macro-précision de 31,72%. À noter que pour cette tâche, nous obtenons une macro-précision de 37% pour une de nos expérimentations (en enlevant la négation), alors que le meilleur système pour cette tâche soumis au défi a obtenu 34.68%. À travers les résultats obtenus, nous préconisons l'utilisation de SVM comme classifieur et des unigrammes comme attributs pour les trois tâches proposées dans ce défi. De plus, nous constatons que la méthode de traitement de la *négation* en rajoutant un suffixe à tous les mots qui se trouvent sous la portée d'un négateur ne donne pas de meilleures macro-précisions pour les trois tâches. On pourrait donc rejoindre la conclusion de (Vincent & Winterstein, 2013) qui montre que cette approche qui a été créée pour l'anglais tend à ne pas fonctionner aussi bien pour le français. Enfin, nous soulignons l'importance de la sélection d'attributs pour les tâches 1 et 2.1 qui améliorent les macro-précisions de plus de 4%.

Comme perspectives, nous prévoyons de tester plusieurs autres configurations que nous n'avons pas eu le temps de tester pour ce défi. Une perspective à court terme est de rajouter comme attributs les identifiants des clusters de Brown obtenus sur le corpus d'apprentissage (Brown *et al.*, 1992; Blitzer & Zhu, 2008). Le Clustering de Brown permet de représenter des textes sous un format moins sparse, et cela en regroupant les mots selon leurs contextes. Nous avons déjà implémenté cette méthode mais la construction des attributs prenant beaucoup de temps, nous n'avons donc pas pu les intégrer durant la période de test qui a duré 3 jours seulement. Une deuxième perspective est d'estimer le seuil du gain d'information de l'algorithme de sélection d'attributs en validation croisée comme cela a été fait pour le paramètre de complexité de SVM. En effet, ces estimations peuvent se faire en même temps comme décrits dans (Li *et al.*, 2015). Pour pallier au problème des classes non équilibrées dans la tâche 2.2, nous comptons aussi tester une technique de sur-échantillonnage implémentée dans Weka et appelée SMOTE (Synthetic Minority Oversampling TEchnique) (et. al., 2002). Cette technique crée de nouvelles instances pour les classes minoritaires en se basant sur les instances existantes. Finalement, il serait intéressant de tester nos méthodes sur d'autres domaines que l'environnement et d'autres types de données que les tweets et d'essayer d'en faire des méthodes génériques pour le français.

Références

ABDAOUI A., JÉRÔME A., BRINGAY S. & PONCELET P. (2014). Feel : French extended emotional lexicon. volume ISLRN : 041-639-484-224-2. ELRA Catalogue of Language Resources.

AUGUSTYN M., BEN HAMOU S., BLOQUET G., GOOSSENS V., LOISEAU M. & RINCK F. (2006). Lexique des affects : constitution de ressources pédagogiques numériques. In *Colloque International des étudiants-chercheurs en didactique des langues et linguistique.*, p. 407–414, Grenoble, France.

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

BLITZER J. & ZHU X. J. (2008). Semi-supervised learning for natural language processing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies : Tutorial Abstracts*, p. 3–3 : Association for Computational Linguistics.

BOUMA G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, p. 31–40.

BROWN P. F., DESOUZA P. V., MERCER R. L., PIETRA V. J. D. & LAI J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, **18**(4), 467–479.

CHURCH K. W. & HANKS P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, **16**(1), 22–29.

EKMAN P. (1992). An argument for basic emotions. *Cognition & emotion*, **6**(3-4), 169–200.

ET. AL. N. V. C. (2002). Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321–357.

FOURNIER-VIGER P., NKAMBOU R. & NGUIFO E. M. (2008). A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems. In *MICAI 2008 : Advances in Artificial Intelligence*, p. 765–778. Springer.

GROUIN C., HURAULT-PLANTET M., PAROUBEK P. & BERTHELIN J.-B. (2009). Deft'07 : une campagne d'évaluation en fouille d'opinion. *Fouille de données d'opinion*, **17**, 1–24.

HALL M., FRANK E., HOLMES G., PFAHRINGER B., REUTEMANN P. & WITTEN I. H. (2009). The weka data mining software : an update. *ACM SIGKDD explorations newsletter*, **11**(1), 10–18.

KIRITCHENKO S., ZHU X., CHERRY C. & MOHAMMAD S. (2014a). Nrc-canada-2014 : Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 437–442, Dublin, Ireland : Association for Computational Linguistics and Dublin City University.

KIRITCHENKO S., ZHU X. & MOHAMMAD S. M. (2014b). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, p. 723–762.

LAVER M., BENOIT K. & GARRY J. (2003). Extracting policy positions from political texts using words as data. *American Political Science Review*, **97**(02), 311–331.

LI X., LI J. & WU Y. (2015). A global optimization approach to multi-polarity sentiment analysis. *PLoS ONE 10(4)*.

MELNIK M. I. & ALM J. (2002). Does a seller's ecommerce reputation matter ? evidence from ebay auctions. *The journal of industrial economics*, **50**(3), 337–349.

MITCHELL T. M. (1997). *Machine learning.*, volume 45. Burr Ridge, IL : McGraw Hill.

MOHAMMAD S. (2012). Portable features for classifying emotional text. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 587–591, Montréal, Canada : Association for Computational Linguistics.

MOHAMMAD S. M., KIRITCHENKO S. & ZHU X. (2013). Nrc-canada : Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR13)*, p. 321.

MOHAMMAD S. M. & TURNEY P. D. (2010). Emotions evoked by common words and phrases : Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, p. 26–34, Stroudsburg, PA, USA : Association for Computational Linguistics.

MULLEN T. & MALOUF R. (2006). A preliminary investigation into sentiment analysis of informal political discourse. In *AAAI Spring Symposium : Computational Approaches to Analyzing Weblogs*, p. 159–162.

NAKOV P., KOZAREVA Z., RITTER A., ROSENTHAL S., STOYANOV V. & WILSON T. (2013). Semeval-2013 task 2 : Sentiment analysis in twitter.

PANG B. & LEE L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, **2**(1-2), 1–135.

PLATT J. C. (1999). Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, p. 185–208. Cambridge, MA, USA : MIT Press.

ROBERTS K., ROACH M. A., JOHNSON J., GUTHRIE J. & HARABAGIU S. M. (2012). Empatweet : Annotating and detecting emotions on twitter. In *LREC*, p. 3806–3813.

ROSENTHAL S., NAKOV P., KIRITCHENKO S., MOHAMMAD S., RITTER A. & STOYANOV V. (2015). Semeval-2015 task 10 : Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, p. 451–463, Denver, Colorado : Association for Computational Linguistics.

ROSENTHAL S., RITTER A., NAKOV P. & STOYANOV V. (2014). Semeval-2014 task 9 : Sentiment analysis in twitter. p. 73–80.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, p. 44–49 : Citeseer.

STRAPPARAVA C. & MIHALCEA R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, p. 1556–1560 : ACM.

STRAPPARAVA C., VALITUTTI A. *et al.* (2004). Wordnet affect : an affective extension of wordnet. In *LREC*, volume 4, p. 1083–1086.

TATEMURA J. (2000). Virtual reviewers for collaborative exploration of movie reviews. In *Proceedings of the 5th international conference on Intelligent user interfaces*, p. 272–275 : ACM.

VINCENT M. & WINTERSTEIN G. (2013). Construction et exploitation d'un corpus français pour l'analyse de sentiment. *TALN-RÉCITAL 2013*, p. 764.

Combiner analyses textométriques, apprentissage supervisé et représentation vectorielle pour l'analyse de la subjectivité

Egle Eensoo[1] Damien Nouvel[1] Amélie Martin[1,2] Mathieu Valette[1]
(1) ERTIM, INALCO, 2 rue de Lille 75007 Paris
(2) SNCF Innovation et Recherche, 40 avenue des Terroirs de France, 75012 Paris
egle.eensoo@inalco.fr, damien.nouvel@inalco.fr, amelie.martin2@sncf.fr,
mathieu.valette@inalco.fr

Résumé. Cet article présente le bilan de notre participation au Défi Fouille de Textes (DEFT 2015) pour les tâches 1 et 2. Il s'agit de classer un corpus de tweets selon leur polarité (tâche 1) et détecter les classes génériques (tâche 2.1) et spécifiques (tâche 2.2) de ces derniers. Nous avons implémenté deux systèmes pour ce défi. La première méthode repose sur la sélection dans le corpus d'entraînement d'un ensemble de descripteurs sémantiquement motivés pour chaque tâche à partir d'une analyse textométrique, qui sont ensuite injectés dans un algorithme d'apprentissage automatique supervisé, permettant le calcul de modèles sur ce même corpus. La seconde méthode s'appuie sur une représentation vectorielle des mots apprise par utilisation de l'outil word2vec sur un corpus hétérogène et volumineux, cette représentation étant ensuite utilisée pour réaliser un apprentissage automatique supervisé, pour chaque tâche, sur les corpus de développement. Un troisième système a été réalisé par combinaison des deux précédents à l'aide d'heuristiques simples. Les résultats obtenus sur les corpus de tests montrent que chaque méthodologie a ses avantages et que leur combinaison peut donner de très bonnes performances.

Abstract.

Combining Textometric Analysis, Machine Learning and Vector Space Representation for Subjectivity Analysis. This paper reports the results of our participation in Evaluation Campaign of Text Mining (DEFT 2015) for tasks 1 and 2. The aim is to classify tweets according to their polarity (Task 1) and detect the generic (task 2.1) and specific classes (task 2.2) thereof. We implemented two systems for this challenge. The first method is based on the selection in the training corpus of a set of semantically motivated descriptors for each task from a textometric analysis, which are then injected into a supervised machine learning algorithm, allowing the development of models on the same corpus. The second method is based on a vector representation of words learned by using the tool of word2vec leveraging heterogeneous and large corpora. This representation is then used to perform automatic supervised learning, for each task, on the development corpus. A third system was designed by combination of both, using simple heuristics. The results obtained on the test corpora show that each methodology has its advantages and that their combination can achieve very high performance.

Mots-clés : analyse de la subjectivité, textométrie, word2vec, classification automatique, linguistique de corpus.

Keywords : subjectivity analysis, textometry, word2vec, machine learning, corpus linguistics.

1 Introduction

1.1 Campagne DEFT 2015

La fouille de données subjectives (sentiments, opinions, émotions) est depuis plusieurs années maintenant un domaine très dynamique de la fouille de textes, aussi bien dans le domaine académique que dans l'industrie. Sommairement, on observe quatre tendances en termes de positionnement épistémologique : méthodes par apprentissage (Pang *et al.*, 2002), méthodes symboliques d'inspiration cognitiviste (vocabulaire des émotions, etc. (Ghorbel & Jacot, 2011 ; Maurel & Dini, 2009), méthodes symboliques d'inspiration pragmatique ou analyse du discours (Vernier *et al.*, 2009a,b), méthodes hybrides combinant certaines de ces approches (Turney, 2002 ; Yi *et al.*, 2003 ; Yu & Hatzivassiloglou, 2003).

La campagne d'évaluation DEFT 2015 propose des tâches de détection de subjectivité (opinions, sentiments et émotions)

sur les tweets en français portant sur la thématique de changement climatique. Nous avons participé aux trois tâches suivantes :

— **Tâche 1** : La première tâche vise à classer les tweets selon une grille macroscopique de polarité : positif, négatif, neutre (ou mixte).
— **Tâche 2.1** : Cette tâche consiste à identifier le type de subjectivité (ou d'objectivité). Les classes proposées sont les suivantes : information (tweet objectif), opinion (l'expression intellective et réfléchie), sentiment (l'expression intellective-affective) et émotion (l'expression purement affective).
— **Tâche 2.2** : Dans cette tâche, l'objectif est d'identifier une classe fine correspondant à trois catégories subjectives (opinion, sentiment, émotion). DEFT nous propose 18 classes fines.

1.2 Travaux précédents et positionnement méthodologique

Notre participation au DEFT 2015 a été motivée par nos travaux antérieurs (Eensoo & Valette, 2012, 2014b,a) portant sur la détection d'opinions et l'analyse des sentiments sur divers corpus issus essentiellement du Web 2 (forums de discussions, commentaires d'internautes des articles de presse). Ainsi, nous avons pu élaborer une méthodologie qui s'inspire de la sémantique textuelle (Rastier, 2001) pour identifier des critères linguistiques pertinents pour une classification sémantique des textes subjectifs. Cette méthodologie s'appuie sur une analyse différentielle du corpus par des méthodes de textométrie comme le calcul de spécificités (Lafon, 1980), de collocations (n-grammes) et des cooccurrences (Lafon, 2981). Ces travaux se démarquent des approches traditionnelles fondées sur la recherche de marqueurs axiologiques explicites par l'utilisation de critères qui ne sont pas considérés d'ordinaire comme prioritaires pour la détection de l'information subjective. Ils relèvent des représentations des acteurs (composante dialogique), des structures argumentatives et narratives des textes (composante dialectique) et des thèmes instanciés (composante thématique). Nous avons pour objectif de proposer une méthodologie mixte alliant l'analyse du linguiste qui, en expertisant le corpus en extrait les éléments linguistiquement pertinents pour l'expression de la subjectivité et les méthodes statistiques qui automatisent l'analyse du corpus et rendent les résultats reproductibles. Les deux verrous scientifiques auxquels nous confrontons notre méthodologie en participant au défi DEFT 2015 ont trait au genre textuel du tweet d'une part, et à l'annotation fine d'autre part.

1. *Textualité et forme brève.* Notre méthodologie repose en effet sur une analyse sémantique de la textualité (cohésion textuelle, marqueurs structuraux etc.). Le tweet, forme brève réputée parataxique, est intrinsèquement pauvre en marqueurs de textualité et peut s'apparenter à un ensemble de mots-clés faiblement articulés textuellement. Conséquence probable de cette pauvreté textuelle, le tweet est hyperlexicalisé, comme en atteste l'innovation du mot-dièse (hashtag) qui promeut les mots du texte et parfois même des syntagmes complexes, voir des phrases au rang de mots-clés ou de candidats mots-clés. Notre méthodologie est conçue pour évaluer la capacité classificatoire des différents marqueurs sémantiques en particulier non thématiques et non axiologiques en privilégiant les éléments de structuration des textes et de positionnements énonciatifs (Eensoo & Valette, 2015). Elle serait donc peu adaptée à un genre textuel court donnant *a priori* le primat aux lexèmes porteurs de signification référentielle.
2. *L'annotation fine.* Il apparaît que l'annotation fine du corpus est en fait une annotation lexicale. C'est peut-être un corollaire du premier verrou scientifique : le guide d'annotation avec lequel le corpus semble avoir été produit [1] apparaît orienté vers une catégorisation très lexicale des tweets. Autrement dit, c'est davantage les significations des unités lexicales de chaque tweet, prises isolément, qui font l'objet d'annotation que le sens du tweet pris dans son ensemble. Au fond, on est ici confronté à une imprécision méthodologique. L'annotation fine ne signifie pas que les émotions vont être annotées avec finesse mais en fonction des seuls mots du texte, considérés comme des mots-clés indexant des émotions. En définitive, annotation fine signifie à grain fin (le grain étant celui du mot). À titre d'exemple, on peut reprendre celui du guide d'annotation « L'amour et la fidelité sont des espéces en voie de disparition » (orthographe respectée). Ce tweet est annoté AMOUR mais il est manifeste que l'émotion exprimée ici n'est pas l'amour, elle est vraisemblablement déceptive (pessimisme, consternation, résignation) mais pourrait également être, dans une perspective nihiliste, l'espoir, la joie (« enfin, nous voilà débarrassés de l'amour et de la fidélité »). En bref, on ne peut guère statuer sur l'émotion exprimée. De la même façon, un tweet tel que « Moi aussi j'aime l'entreprise, celle sans patron, sans actionnaire et qui produit des biens et services durables socialement et écologiques ! » (extrait du corpus) peut-il sérieusement être annoté comme porteur de l'émotion AMOUR ? la seule présence du verbe aimer ne permet pas, selon nous, d'en juger. On pourrait même argumenter que l'amour de l'entreprise exprimé ici l'est par contraste avec un désamour tout aussi explicite et même peut-être plus saillant

1. https ://deft.limsi.fr/2015/guideAnnotation.fr.php ?lang=fr

envers d'autres acteurs du tweets : le patron, les actionnaires. Nous développerons cette analyse critique dans le paragraphe 2.1.1.

2 Méthodologie de la détection de subjectivité

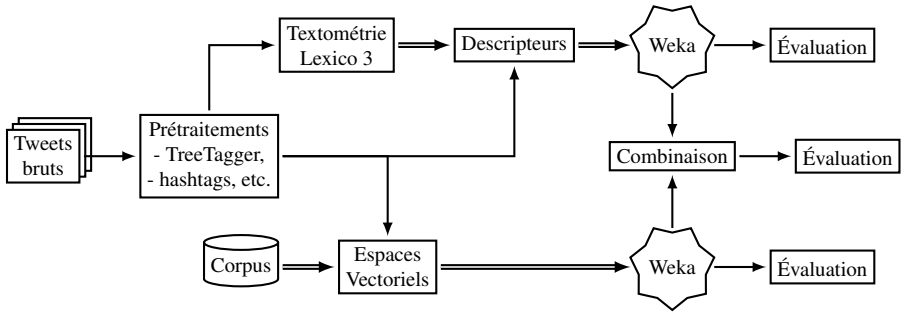

FIGURE 1 – Fonctionnement général

Pour aborder la problématique de la détection automatique de la subjectivité, nous exploitons deux méthodes qui ont fait leurs preuves ces dernières années. Le système que nous avons développé est décrit en figure 1.

La première consiste à utiliser la textométrie afin didentifier des descripteurs qui serviront de critères pour classifier les tweets (voir supra, 2.1). La seconde repose sur un apprentissage non-supervisé d'espaces vectoriels, à partir d'un autre corpus, dans lequel projeter les tweets avant de les classifier. Nous réalisons également un troisième système dont la sortie est déterminée par les deux premiers. Dans les trois cas, quelles que soient les premières étapes de traitement, la classification finale est supervisée, par appel à un logiciel d'apprentissage automatique.

2.1 Linguistique de corpus et textométrie

2.1.1 Le corpus DEFT et son annotation

Le corpus DEFT et son annotation sont caractérisés par quelques particularités qui conditionnent les résultats éventuels d'un système de classification automatique de tweets. Tout d'abord, il nous a semblé qu'une proportion non-négligeable du corpus est constituée de tweets générés automatiquement (boutons de partage situés sur les sites d'actualité). Ces tweets sont reconnaissables : à la mention d'un *via x* où *x* est le nom d'un site d'actualité ; à des énoncés tronqués ; à la présence d'URLs dans le corps du texte (par exemple : « *Forte hausse des raccordements éolien et photovoltaïque : Les raccordements d'éoliennes et de panneaux solaires... [URL]* »). Ils reprennent ainsi, dans un grand nombre de cas, les titres ou des chapeaux d'articles de presse et peuvent difficilement être classés en terme d'opinion ou de sentiments. En voici trois exemples tirés du corpus d'apprentissage :

— « *#Euthanasie : Un chien survit à une tentative d'euthanasie - 7sur7 [URL]* » : polarité positive, pas de classe générique (NULL) ;
— « *Un #chien survit miraculeusement à une tentative d'#euthanasie ! [URL]* » : polarité neutre, classe générique INFORMATION ;
— « *Un chien survit à une tentative d'euthanasie [URL] via @7sur7* » : polarité positive, classe générique OPINION, sous-classe VALORISATION.

De surcroît, comme nous le voyons dans l'exemple ci-dessus, les tweets peuvent être très similaires, et les divergences d'annotation importantes. L'annotation de certains tweets semble avoir été réalisée à partir de la simple présence d'un terme porteur d'une émotion ou d'une opinion spécifique (par exemple, le tweet « *Elles font **fureur**... Leur toucher doux, leur couvercle cristal, leur respect d'environnement ! URL* », classé dans la sous-classe COLERE). Enfin, il semble aussi que certains annotateurs font le choix de classer tous les tweets comportant un terme connoté positivement dans le domaine de l'écologie (« *renouvelable* », « *durable* », « *solaire* », etc.) dans la sous-classe VALORISATION, sans que ce choix soit unanime, comme nous le voyons dans ces tweets :

— « *BFM Business : Transition énergétique : le Syndicat des énergies renouvelables confiant [URL]* » polarité positive, pas de classe générique (NULL) ;
— « *Transition énergétique : le Syndicat des énergies renouvelables confiant [URL]* » : polarité positive, classe générique OPINION, sous-classe VALORISATION.

Sous-classe	**Corpus d'origine**	**Proportion (%)**	**Corpus réannoté**	**Proportion (%)**
ACCORD	**14**	3,11	**5**	1,11
AMOUR	0	-	0	-
APAISEMENT	1	0,22	3	0,67
COLERE	19	4,22	15	3,33
DEPLAISIR	3	0,67	6	1,33
DERANGEMENT	1	0,22	0	-
DESACCORD	5	1,11	8	1,78
DEVALORISATION	19	4,22	19	4,22
ENNUI	0	-	0	-
INSATISFACTION	1	0,22	0	-
MEPRIS	**8**	1,78	**19**	4,22
PEUR	**17**	3,78	**10**	2,22
PLAISIR	**0**	-	**5**	1,11
SATISFACTION	8	1,78	3	0,67
SURPRISE_NEGATIVE	1	0,22	2	0,44
SURPRISE_POSITIVE	1	0,22	0	-
TRISTESSE	1	0,22	2	0,44
VALORISATION	**78**	17,33	**30**	6,67
INFORMATION	**220**	48,89	**301**	66,89
NULL [pas de classe]	53	11,78	22	4,89
Total	450	-	450	-

TABLE 1 – Répartition du nombre de tweets par sous-classe au sein de notre échantillon réannoté

Ainsi, dans le cadre de cette campagne d'évaluation, nous avons voulu comparer l'annotation d'origine fournie par l'organisation de DEFT avec une annotation du même corpus réalisée par nos soins, en lançant une mini campagne d'étiquetage des tweets. A l'issue de la réannotation (en suivant le guide d'annotation de DEFT [2]) d'un échantillon de 450 tweets extraits au hasard du corpus d'entraînement, le taux de recoupement est d'environ 70%. Les disparités, surtout présentes pour les classes majoritaires INFORMATION et VALORISATION, s'expliquent par l'application de règles plus strictes dans notre annotation. Par exemple, nous avons choisi de ne classer dans VALORISATION que les tweets qui comportent un commentaire valorisant ou qui portent une marque d'engagement du rédacteur :

— « *Amateurs vegan y'a un super livre de @100vegetal qui va paraitre sur les fromages (j'en ai gouté 1, c'est trop bon) [URL]* » (dans le corpus d'origine : polarité neutre, classe générique INFORMATION) ;
— « *Je salue le courage des écologistes japonais qui luttent contra la chasse aux dauphins dans leur propre pays.* » (dans le corpus d'origine : polarité positive, classe générique SENTIMENT, sous-classe SATISFACTION).

Nous avons également classifié davantage de tweets sarcastiques dans la sous-classe MEPRIS : « *#Écologie #findumonde Cet homme, que dis-je ce héros, va nous sauver. #ohwait [URL]* ».

Ce processus de réannotation et de comparaison nous a permis d'évaluer l'homogénéité de certaines classes de manière qualitative. Le tableau 1 montre le nombre de tweets par sous-classe au sein de notre échantillon. En gras apparaissent les classes ou sous-classes qui présentent les plus fortes divergences : INFORMATION, VALORISATION et MEPRIS, mais aussi ACCORD, PEUR, PLAISIR et SATISFACTION. Ces deux dernières sous-classes se confondent facilement (avec APAISEMENT également). Par exemple, nous avons classifié le tweet *@Actuenviro : Transition énergétique : le maintien de Ségolène Royal rassure écologistes et industriels [URL]* dans la sous-classe APAISEMENT, alors qu'il apparaissait dans la sous-classe SATISFACTION : les deux annotations semblent correctes. Quant aux divergences pour la sous-classe PEUR, elles résultent de la frontière très ténue entre inquiétude et information (*Éolien - La possible extension des zones d'exclusion militaires provoque la crainte des professionnels : [URL]*).

2. https://deft.limsi.fr/2015/guideAnnotation.fr.php?lang=fr

2.1.2 Élaboration textométrique de critères de classification

L'élaboration textométrique des critères consiste à trouver des critères de classification linguistiquement explicables et suffisamment robustes pour servir de descripteurs aux méthodes d'apprentissage supervisé. L'analyse du corpus et le repérage des critères linguistiques ont été effectués avec deux logiciels textométriques : Lexico 3 (Salem *et al.*, 2003) et TXM (Heiden *et al.*, 2010) qui implémentent notamment les algorithmes de spécificités (Lafon, 1980) et de collocations (« Segments répétés » de Lexico 3) ainsi que les concordances qui nous ont permis le retour au texte et donc la vérification de la pertinence linguistique des critères.

An amont de l'analyse du corpus, nous avons effectué quelques prétraitements :
- les URLs ont été remplacés par la chaîne de caractère *URL*,
- les émoticônes ont été supprimés,
- les hashtags ont été considérés comme des mots simples (séparés du marqueur #),
- enfin, le corpus a été lemmatisé avec TreeTagger (Schmid, 1994). La lemmatisation, bien qu'elle fasse l'objet de débat en textométrie (Brunet, 2000) comme en analyse d'opinion (Pang *et al.*, 2002) nous a semblé un choix judicieux à cause de la particularité du corpus (textes courts avec peu de redondance de mots dans un texte) et de sa taille (en effet, nous avons constaté auparavant que les lemmes étaient plus performants sur de grands corpus).

Pour l'expérience nous avons utilisé trois types de critères :
- critères unitaires : choix des lemmes pertinents
- critères composites adjacents : choix des n-grammes de longueur variable de 2 à 6 unités
- cooccurrences textuels (dans la fenêtre d'un tweet) de 2 lemmes

Tous les critères sont sélectionnés selon trois principes : (i) leur caractère spécifique à une catégorie (ii) leur fréquence et (iii) leur pertinence linguistique.

Nous avons choisi les deux premiers types de critères selon le procédé suivant :

1. calcul des spécificités des lemmes isolés et de leur n-grammes (fonction « Segments Répétés » de Lexico 3) pour chaque catégorie ;
2. analyse des contextes d'apparition des lemmes spécifiques (au moyen de concordances textuelles) afin de s'assurer de leur pertinence textuelle et de l'unicité de leur fonction (les critères ayant une seule fonction et signification ont été privilégiés) ;

La sélection des cooccurrences a été réalisée comme suit :

1. calcul des paires de lemmes cooccurrents pour chaque tweet
2. calcul de spécificités de chaque cooccurrence pour toutes les catégories (avec le logiciel TXM)
3. sélection des cooccurrents sémantiquement interprétables (élimination des cooccurrents avec des mots-outils fréquents, choix des cooccurrents qui soit précisent un item déjà présent dans parmi les lemmes isolés soit apporte un nouveau critère sémantique).

2.1.3 Descripteurs linguistiques extraits

Nous présentons ici succinctement les principales catégories de critères qui ont servi à la classification des tweets. Nous exposons les critères obtenus avec la première méthode (méthode textométrique).

Nous distinguons quatre catégories de critères linguistiques : thymiques, dialogiques, dialectiques et thématiques.
- **Les critères thymiques** sont réputés intrinsèquement axiologiques et relèvent d'une vision classique de l'expression de la subjectivité. Pour catégoriser les tweets positifs, on trouve des marqueurs comme *bon, beau, intéressant, mieux, bien, positif, super, bravo, aimer*. Dans les tweets de polarité négative on recense les mots comme *mauvais, suspect, polémique, inquiéter, pire, mal, colère, foutre, con, merde, gueule*. Néanmoins, la proportion des marqueurs axiologiques reste relativement faible par rapport aux autres catégories, ce qui nous amène à penser que l'expression de la subjectivité est un phénomène complexe que l'on ne peut réduire à l'identification des marqueurs thymiques.
- **Les critères dialogiques** concernent la représentation des acteurs, le positionnement énonciatif et la distribution des rôles actanciels. Ils actualisent essentiellement les pronoms personnels, les pronoms possessifs et certaines entités nommées. On le trouve essentiellement dans les tweets de polarité négative ce qui peut s'interpréter comme un ancrage plus prononcé dans la présence du locuteur et dans l'interaction. Il s'agit essentiellement des pronoms comme *elle, lui, tu, te, on, me* et quelques entités nommées du domaine : *Ecologistes, Ségolène Royal, communiste*.

— **Les critères dialectiques** sont dédiés à la représentation du temps et du déroulement aspectuel, des structures argumentatives et de certaines modalités. Le vocabulaire la caractérisant est plus varié. Il peut s'agir de marqueurs de structuration, des verbes modaux, et des indicateurs rhétoriques (emphases, points d'interrogation, mots interrogatifs, etc.). Dans notre corpus, les critères dialectiques se trouvent principalement dans les tweets négatifs (*comment, pourquoi, ?, pourtant, ah*). Les tweets neutres se caractérisent par des ponctuations de phrase qui structurent le texte ; par conséquent on peut également les considérer comme dialectiques : *%, (,), ;, ...*
— **Les critères thématiques** sont les plus nombreux dans ce corpus. Ils caractérisent les différents thèmes abordés qui sont dans notre cas porteur d'une polarité. Les critères positifs sont liés à la sauvegarde de la nature et au développement des solutions alternatives pour l'énergie. Voici quelques exemples : *investir, réduire, soutenir, crowdfunding, géothermie, construire, développer, protection, cellule solaire, photovoltaïque, financement participatif, énergie positif, réduire CO2, réduire aéroport, développer renouvelable, créer écosystème*. Les critères négatifs expriment les problèmes écologiques : *en danger, disparition, crise, réchauffement climatique, espèce menacer, impasse climatique, écologie punitif, oiseau, neige, assassiner, mort, tuer, indifférence*. Les critères des tweets neutres (informationnelles) comportent quasi exclusivement les critères thématiques qui relatent les actualités. Les exemples sont les suivants : *publication, emploi, job, programme, panorama, consultation, rencontre, étudier, conférence, test, observatoire*.

2.1.4 Apprentissage automatique

Pour classer les tweets nous avons utilisé les algorithmes d'apprentissage supervisé. Nous en avons testé plusieurs, nous ne présentons ici que les résultats obtenus avec l'algorithmes de Machines à Vecteurs de Support (SMO) (Platt, 1998) intégré dans Weka (Hall *et al.*, 2009) qui a donné les meilleurs résultats. En amont des résultats sur le corpus de test, nous présentons les résultats obtenus sur le corpus d'apprentissage avec la validation croisée à dix plis en table 2.

Tâche	Macro-précision	Micro-précision
1	71,73	69,5
2.1	70,35	70,10
2.2	52,00	63,70

TABLE 2 – Résultats obtenus sur le corpus dapprentissage par validation croisée

2.2 Utilisation de l'apprentissage non supervisé

2.2.1 Calcul de l'espace vectoriel et projection des tweets

Dans le contexte de cette campagne d'évaluation, nous nous sommes aperçu que la taille des corpus d'entraînement est limitée et peu de ressources sont facilement disponibles pour le français. Afin de tester des approches qui limitent la dépendance au corpus en terme de vocabulaire, nous nous sommes tournés vers des algorithmes non supervisés. Les travaux récents de Mikolov *et al.* (2013) ont montré l'efficacité que l'on peut obtenir lors de l'utilisation de représentations de mots (ou d'expressions) dans des espaces vectoriels qui sont calculés selon leurs contextes. C'est l'objectif atteint par l'outil word2vec [3] qui a fait ses preuves dans d'autres domaines et que nous avons entraîné sur les corpus suivants :

Corpus	Mots (K)	Description
AFP	500 558	Dépêches AFP sur les années 2007-2013
Deft (train)	116	Corpus d'entraînement de DEFT
CoMeRe	568	Tweets de personnalités politiques (Longhi *et al.*, 2014)
Feelings	1 686	Extraction de tweets avec l'outil twitter-feelings
Hashtags	593	Extraction de tweets avec twython à partir de hashtags

TABLE 3 – Volumétrie et description du corpus d'entraînement de word2vec

3. https://code.google.com/p/word2vec/

Notre corpus dépasse les 500 millions de mots, dont la très large majorité est constituée de dépêches AFP. Une partie à été collectée à l'aide de l'outils twitter-feelings [4] ou par recherche de hashtags liés à l'écologie [5]. Le corpus a ensuite été lemmatisé avec TreeTagger (Schmid, 1994). De plus, les hashtag et les mentions sont introduits sous trois formes : tels quels ; sans leurs préfixes (@ ou #) ; segmentés selon la présence de majuscules. Le corpus est ensuite traité par word2vec afin d'apprendre des vecteurs de 500 composantes, sur une fenêtre contextuelle de 10 mots, en 20 itération (autres paramètres laissés par défaut).

Les tweets provenant du corpus d'entraînement ou du corpus de test sont prétraités avec les mêmes procédures, puis projetés dans l'espace vectoriel des lemmes créé par word2vec. Comme à chaque mot un vecteur est associé dans cet espace, la projection est une somme normalisée des mots présents dans chaque tweet (notre hypothèse étant que la longueur d'un message n'impacte pas les opinions / sentiments / émotions qui y sont présents). Nous ajoutons par ailleurs pour cette méthode la distance cosinus des mots du tweet avec chaque descripteur déterminé dans la partie 2.1.3.

2.2.2 Apprentissage automatique

Notre premier objectif est d'évaluer les performances obtenues par diverses approches. Pour ce faire, nous avons utilisé les algorithmes fournis par Weka (Hall *et al.*, 2009), ainsi que le filtre utilisé par défaut (StringToWordVector) permettant de convertir des textes sous formes de vecteurs de mots. Les tweets pouvaient être fournis : dans leur forme brute ; après une lemmatisation ; après projection dans l'espace vectoriel ; après projection dans l'espace vectoriel avec ajout des distances aux descripteurs. Nous avons ensuite évalué nos résultats sur le corpus d'entraînement grâce à l'outil proposé dans le cadre de la campagne, en réalisant nous-même une validation croisée à 10 plis.

Prétraitements	Algorithme	Macro-précision	Micro-précision
Tokens	ZeroR	15,04	45,12
	NaiveBayes	46,05	47,20
	NaiveBayesMultinomial	37,14	35,29
	SMO	58,34	59,46
Lemmes	NaiveBayes	47,41	48,19
	NaiveBayesMultinomial	38,39	35,71
	SMO	58,77	60,37
Vecs	NaiveBayes	49,71	51,15
	SMO	66,41	67,32
Vecs+Desc	NaiveBayes	50,09	51,42
	NaiveBayesMultinomial	62,67	60,89
	SMO	65,72	66,70

TABLE 4 – Comparaison des prétraitements et algorithmes pour T1

Les résultats obtenus par les algorithmes présentés en table 4 montrent la supériorité de l'algorithme SMO (Platt, 1998). Nous voyons également que, si la lemmatisation apporte assez peu, le passage aux représentations vectorielles apporte des gains très significatifs en performance. Effectivement, les représentations vectorielles permettent de limiter la dépendance aux corpus d'entraînement et donc de couvrir un vocabulaire qui n'est pas compris dans ce dernier. Par contre, l'ajout des distances aux descripteurs n'apportent pas plus (et font même légèrement baisser les performances).

2.3 Combinaison des méthodologies

Les méthodes sont combinées en regardant quelles catégories ont été bien annotées par chaque système sur une sous-partie du corpus de test que nous avons réannoté, comme cela a été décrit dans la partie 2.1.1. Ainsi, selon les résultats de chaque système, nous avons déterminé des heuristiques déterministes simples qui, à partir de la sortie des deux systèmes, prend une décision. Pour la tâche 1, les règles sont les suivantes :

4. https ://github.com/cblavier/twitter-feelings

5. Liste des hashtags : #ecologie #Ecologie #écologie #Environnement #Biodiversité #DD #biodiversité #énergie #solaire #Energie #environnement #énergies #EELV #Animaux #Durable #climat

— choisir neutre si telle est la sortie du système word2vec,
— choisir positif ou négatif si telle est la sortie de la méthode textométrique,
— sinon, mettre neutre.

Pour la tâche 2.1, nous donnons la priorité par classe, quel que soit le système considéré, dans l'ordre suivant : sentiment, information, émotion, opinion. Nous adoptons le même principe pour la tâche 2.2 avec l'ordre suivant : DÉPLAISIR, TRISTESSE, INSATISFACTION, PLAISIR, DÉSACCORD, DÉRANGEMENT, AMOUR, SATISFACTION, MÉPRIS. Notons que cette combinaison n'a été réalisée qu'à titre expérimental, ce qui explique son peu de sophistication (utiliser un apprentissage automatique à ce niveau aurait probablement été plus efficace).

3 Résultats

Les résultats obtenus sur le corpus de test sont présentés en table 5. Pour la tâche 1, nous constatons que la méthode textométrique obtient de meilleurs résultats que word2vec, et que la combinaison de ces deux systèmes nous permet d'obtenir des résultats très proches du meilleur système de la campagne. Pour les tâches 2.1 et 2.2, word2vec obtient des résultats meilleurs que la méthode textométrique, approchant une nouvelle fois le meilleur système pour la tâche 2.2, la combinaison n'obtenant de bons résultats que pour la tâche 2.1.

Tâche	Textométrie	word2vec	Combinaison	Max. DEFT
1	71,09	69,17	73,44	73,60
2.1	56,22	57,19	57,53	61,29
2.2	29,23	33,72	30,42	34,68

TABLE 5 – Résultats DEFT 2015

Ces résultats sont satisfaisants au regard des meilleurs systèmes de la campagne et montrent bien les avantages et inconvénients de chaque méthode utilisée. La méthode textométrie permet effectivement, pour une classification binaire, de s'appuyer sur des descripteurs plutôt que sur les mots du corpus afin de construire des représentations pertinentes des messages pour l'apprentissage automatique. Pour des tâches demandant une classification plus fine, la méthode textométrique a montré ici des limites. Les représentations vectorielles donnent de meilleurs résultats. Nos expériences et évaluations sur le corpus d'entraînement nous font suspecter une forte spécificité des descripteurs pour ces tâches, et donc un sur-apprentissage, ce que pallie word2vec en évitant de s'appuyer sur les mots eux-mêmes, mais sur leur projection dans un espace continu.

Conclusion

La campagne d'évaluation DEFT 2015 sur l'annotation subjective de tweets nous permet de mener des travaux dans deux directions. La première porte sur l'annotation elle-même et vise à déterminer comment il est possible d'extraire des indices permettant (pour un humain ou un algorithme) de déterminer quelle classe attribuer à un texte court. Si nos conclusions à cet égard sont encore parcellaires, nous nous apercevons de la difficulté de la tâche et de ses variabilités. La seconde direction vise à fonder expérimentalement les méthodes adéquates pour construire des systèmes qui classent automatiquement ces tweets. Nous expérimentons une méthode textométrique et mettons ici en avant les bonnes performances qu'elle obtient en combinaison avec un apprentissage automatique. Par ailleurs, nous la confrontons également aux représentations vectorielles, qui montrent également leur intérêt, en particulier lorsque les catégories sont nombreuses et le corpus d'entraînement de taille limité. Comme les résultats obtenus le montrent, combiner le deux permet d'obtenir des résultats très compétitifs, une perspective que nous envisageons d'approfondir dans nos travaux futurs.

Références

BRUNET E. (2000). Qui lemmatise dilemme attise. *Lexicometrica*, **2**.

EENSOO E. & VALETTE M. (2012). Sur l'application de méthodes textométriques à la construction de critères de classification en analyse des sentiments. In *Actes de la conférence conjointe JEP-TALN-RECITAL 2012*, volume 2, p. 357–374, Grenoble.

EENSOO E. & VALETTE M. (2014a). Approche textuelle pour le traitement automatique du discours évaluatif. *A. Jackiewicz, (éd.), Études sur l'évaluation axiologique, Langue française*, (184), 107–122.

EENSOO E. & VALETTE M. (2014b). Sémantique textuelle et tal : un exemple d'application à l'analyse des sentiments. *D. Ablali, S. Badir, D. Ducard, Eds., Documents, textes, œuvres, Presses Universitaires de Rouen, Collection Rivages linguistiques*.

EENSOO E. & VALETTE M. (2015). Une méthodologie de sémantique de corpus appliquée à des tâches de fouille d'opinion et d'analyse des sentiments : étude sur l'impact de marqueurs dialogiques et dialectiques dans l'expression de la subjectivité. In *Actes de la conférence TALN 2015*.

GHORBEL H. & JACOT D. (2011). Further experiments in sentiment analysis of french movie reviews. In E. MUGELLINI, P. SZCZEPANIAK, M. PETTENATI & M. SOKHN, Eds., *Advances in Intelligent Web Mastering 3*, volume 86 of *Advances in Intelligent and Soft Computing*, p. 19–28. Springer Berlin / Heidelberg. 10.1007/978-3-642-18029-3_3.

HALL M., EIBE F., HOLMES G., PFAHRINGER B., REUTEMANN P. & WITTEN I. H. (2009). The weka data mining software : An update. *SIGKDD Explorations*, **11**(1).

HEIDEN S., MAGUÉ J. P. & PINCEMIN B. (2010). Txm : Une plateforme logicielle open-source pour la textométrie conception et développement. In S. BOLASCO, Ed., *Actes de la conférence JADT 2010*, volume 2, p. 1021–1032.

LAFON P. (1980). Sur la variabilité de la fréquence des formes dans un corpus. *Mots*, **1**, 127–165.

LAFON P. (2981). Analyse lexicométrique et recherche des cooccurrences. *Mots*, (3), 95–148.

LONGHI J., MARINICA C., BORZIC B. & ALKHOULI A. (2014). Polititweets, corpus de tweets provenant de comptes politiques influents. *corpus]. In Chanier T.(ed) Banque de corpus CoMeRe. Ortolang. fr : Nancy. http ://hdl. handle. net/11403/comere/cmr-polititweets*.

MAUREL S. & DINI L. (2009). Exploration de corpus pour l'analyse des sentiments. In *Actes de DEFT'09 ń DÉfi Fouille de Textes ż, Atelier de clôture*.

MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

PANG P., LEE L. & VAITHYANATHAN S. (2002). Thumbs up ? sentiment classification using machine learning techniques. In *In Proceedings of EMNLP*, p. 79–86.

PLATT J. (1998). Machines using sequential minimal optimization. In B. SCHOELKOPF, C. BURGES & A. SMOLA, Eds., *Advances in Kernel Methods - Support Vector Learning*.

RASTIER F. (2001). *Arts et sciences du texte*. Presses Universitaires de France.

SALEM A., LAMALLE C., MARTINEZ W., FLEURY S., FRACCHIOLLA B., KUNCOVA A. & MAISONDIEU A. (2003). *Lexico3 Outils de statistique textuelle. Manuel d'utilisation*. Syled-CLA2T, Université Sorbonne Nouvelle.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

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

VERNIER M., MONCEAUX L. & DAILLE B. (2009a). Deft'09 : détection de la subjectivité et catégorisation de textes subjectifs par une approche mixte symbolique et statistique. In *Actes de l'atelier de clôture de la 5ème édition du Défi Fouille de Textes*.

VERNIER M., MONCEAUX L., DAILLE B. & DUBREIL E. (2009b). Catégorisation des évaluations dans un corpus de blogs multi-domaine. *Revue des nouvelles technologies de l'information (RNTI)*, p. 45–70.

YI J., NASUKAWA T., BUNESCU R. & NIBLACK W. (2003). Sentiment analyzer : Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, p. 427–, Washington, DC, USA : IEEE Computer Society.

YU H. & HATZIVASSILOGLOU V. (2003). Towards answering opinion questions : separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, p. 129–136, Stroudsburg, PA, USA : Association for Computational Linguistics.

TALEP @ DEFT'15 : Le plus cooool des systèmes d'analyse de sentiment

Mickael Rouvier Benoit Favre Balamurali Andiyakkal Rajendran
Aix-Marseille University, CNRS, LIF UMR 7279, 13000, Marseille, France
prenom.nom@lif.univ-mrs.fr

Résumé. Nous présentons dans cet article les systèmes développés par l'équipe TALEP au LIF pour la campagne d'évaluation DEFT'15. La campagne comporte deux tâches : classification des tweets selon leur polarité et classification fine des tweets. Plusieurs systèmes basés sur des modèles probabilistes ont été développés pour chacune des tâches. Puis un système de fusion a été développé combinant les scores des précédents systèmes. La bonne robustess des sytèmes individuels et le système de fusion entre le corpus d'apprentissage et de test nous a permis d'obtenir de bons résultats, bien que très contrastés selon les tâches.

Abstract.

TALEP @ DEFT'15 : The coooolest sentiment analysis systems

This paper describes the systems developed by TALEP team LIF for the DEFT'15 evaluation campaign. This campaign includes two different tasks : valence classification of tweets and fine-grained classification of the tweets. Several systems, all based on probabilistic models, were developed. A final fusion step was developed combining the scores of previous steps. The good robustness of the individual systems and the fusion system between the training and testing corpora allowed us to obtain good results, although well contrasted over the various task.

Mots-clés : analyse de sentiment, réseaux de neurones profonds, word embeddings.

Keywords: sentiment analysis, deep neural network, word embeddings.

1 Introduction

Cette onzième édition du Défi Fouille de Texte (DEFT) était consacrée à l'analyse des sentiments des tweets en français. L'équipe Traitement Automatique du Langage Ecrit et Parlé (TALEP) du Laboratoire d'Informatique Fondamentale (LIF) a participé à cette édition. C'est la première participation dans DEFT de l'équipe TALEP du LIF. L'objectif de notre participation s'inscrit dans le cadre du projet européen SENSEI [1] fondé sur l'étude des conversations humaines ; nous sommes amenés à analyser les sentiments, opinions, émotions des corpus, tels que des transcriptions de conversations téléphoniques ou des commentaires web.

Cette année trois tâches ont été proposées. La première tâche (T1) consiste à classifier les tweets en fonction de l'expression qu'ils expriment (positive, négative ou neutre). La seconde permet de classifier les tweets selon une polarité fine, cette tâche est divisée en deux sous-tâches : la première (T2.1) identifie la classe générique exprimée dans le tweet (opinion, information, sentiment ou émotion) et la seconde (T2.2) identifie le tweet en fonction de l'une des 18 classes proposées [2]. La dernière et troisième tâche (T3) consiste à détecter : la source (l'empan du texte qui désigne explicitement la personne qui exprime l'opinion/sentiment/émotion), la cible (l'empan du texte qui désigne explicitement l'objet de l'opinion/sentiment/émotion) et l'expression du tweet (l'empan de texte dont la valeur sémantique correspond à l'une des 18 classes). L'équipe TALEP a participé seulement aux deux premières tâches (T1, T2.1 et T2.2).

Nous décrivons dans cet article les techniques et les méthodes automatiques utilisées pour ce défi. La section 2 présente le corpus ainsi que les métriques utilisées lors du défi DEFT'2015. Nous présentons ensuite l'architecture du système dans la section 3 et les étapes de pré-traitement de ce système dans la section 4. Dans la section 5, nous présentons les systèmes dont les résultats apparaissent dans la section 6.

1. http://www.sensei-conversation.eu
2. déplaisir, dérangement, mépris, surprise négative, peur, colère, ennui, tristesse, plaisir, apaisement, amour, surprise positive, satisfaction, insatisfaction, accord, valorisation, désaccord, dévalorisation

2 Description de la tâche

2.1 Corpus

Les organisateurs ont mis à la disposition des participants un corpus d'apprentissage composé de 7929 tweets et un corpus de test composé de 3379 tweets. Ce corpus n'a pas été entièrement annoté sur toutes les tâches. Le Tableau 1 donne le nombre de tweets annotés sur le corpus d'apprentissage et de test pour les tâches 1, 2.1 et 2.2 :

	T1	T2.1	T2.2
Apprentissage	7929	6754	3183
Test	3379	3379	1361

TABLE 1 – Nombre de tweets annotés sur le corpus d'apprentissage et de test pour les tâches 1, 2.1 et 2.2.

Afin de tester nos méthodes, de régler leurs paramètres et de palier au phénomène de sur-apprentissage, nous avons décidé de scinder l'ensemble d'apprentissage en 3 sous-ensembles approximativement de la même taille. La procédure d'apprentissage a été la suivante : 2 des 3 sous-ensembles sont concaténés pour produire un corpus d'entraînement et le troisième est utilisé pour le test. La procédure est effectuée trois fois afin que chacun des sous-ensembles du corpus d'apprentissage soit utilisé une fois pour le test. Les ensembles ainsi concaténés seront appelés dorénavant ensembles de développement et le restant ensemble de validation.

2.2 Métriques

Les différents systèmes sont évalués en terme de document correctement classifié (*Accuracy*) et de macro-précision. A noter que dans le cadre du défi DEFT'15, la métrique officielle est la macro-précision. L'*accuracy* et la macro-précision sont calculés comme suit :

$$Accuracy = \frac{\sum_i \text{Nb de documents correctement attribués à la classe i}}{\text{Nb de documents}} \quad (1)$$

$$Macro\_Precision = \frac{\sum_i N_i \cdot P_i}{\sum_i N_i} \quad (2)$$

où N_i est le nombre de documents appartenant à la classe i et P_i est la précision de la classe i.

3 Architecture du système

Le système proposé repose sur une architecture à 2 niveaux (Figure 1). Le premier niveau consiste à obtenir différents points de vue d'un tweet en faisant tourner différents systèmes d'analyse de sentiment. Nous proposons d'utiliser 5 systèmes qui sont décrits plus en détail dans les prochains paragraphes :

— **SVM** : ce système ré-implémente l'approche état-de-l'art de (Mohammad *et al.*, 2013) qui consiste à utiliser un classifieur de type SVM et des uni-grammes, un lexique d'émotion et des paramètres morphologiques.
— **DNN** : ce système est très proche du précédent (SVM) et consiste à utiliser un classifieur de type réseau de neurones profonds, des bi-grammes et un lexique d'émotion.
— **CNN** : ce système ré-implémente l'approche de (Collobert *et al.*, 2011; Kim, 2014) qui consiste à utiliser un classifieur de type réseau de neurones convolutionnels (CNN) et des *Word embeddings*.
— **Doc2Vec** : ce système ré-implémente l'approche de (Le & Mikolov, 2014), qui consiste à utiliser un classifieur de type SVM et comme paramètre des *Doc2Vec*.
— **SuperVector** : ce système est une nouvelle approche qui consiste à utiliser un classifieur de type réseau de neurones profonds sur les statistiques de premier ordre obtenu à partir d'un modèle de mises-von fisher et des *Word embeddings*.

Le second niveau permet de combiner les systèmes du niveau-1. Ainsi, les scores donnés par les différents systèmes du niveau-1 sont groupés dans un vecteur. Un classifieur de type SVM est utilisé sur ce vecteur pour détecter la classe du tweet.

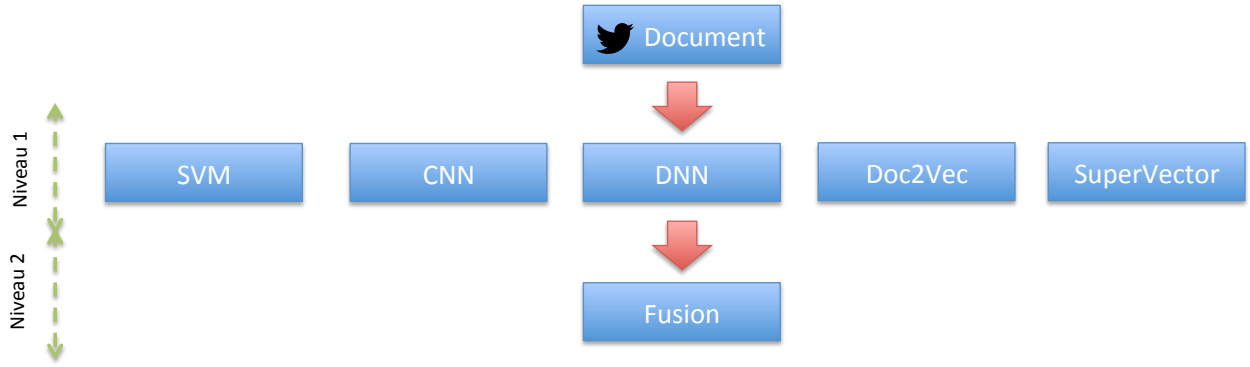

FIGURE 1 – Architecture générale du système TALEP @ DEFT'15

4 Pré-traitements

4.1 Corpus embeddings

La tâche de classification d'analyse de sentiment nécessite de disposer de larges bases de données annotées, afin de capturer l'ensemble des variabilités. De récentes approches ont montré l'intérêt d'intégrer des informations sémantiques dans les systèmes d'analyse de sentiment. Ces approches ont notamment permis de réduire la taille des corpus d'apprentissages. Une approche intéressante est la sémantique distributionnelle qui est une théorie empirique de la sémantique fondée sur l'hypothèse que les mots similaires apparaissent dans les mêmes contextes.

Les *Word embeddings* est une approche de la sémantique distributionelle qui permet de représenter des mots sous la forme de vecteurs de nombres réels. Ces *embeddings* présentent d'intéressantes propriétés de regroupement, puisqu'elle permet de regrouper ensemble les mots qui sont sémantiquement et syntaxiquement proche (Mikolov *et al.*, 2013b). Par exemple, les mots "café" et "thé" vont être très proches dans cet espace. Le but est d'utiliser ces traits dans les classifieurs.

Pour apprendre les *Word embeddings*, nous avons créé un corpus non-annoté de tweets de sentiment en français. Ces tweets ont été récupérés sur la plateforme Twitter [3] en effectuant des recherches avec des mots-clefs porteurs d'émotion, sentiment ou d'opinion. Ces mots-clefs peuvent être des termes (comme par exemple : vexé, annihilé, ridicule...), des hashtags (#good, #like, #mauvais...) ou des smileys (;), :-), :-D). Ce corpus est composé d'environ 16 millions de tweets en français, téléchargés entre le 27 février et le 14 avril 2015. Le corpus collecté ainsi que tous les mots-clefs servant à collecter celui-ci sont disponibles ici : `https://github.com/mrouvier/tweet_corpus_fr`.

Dans nos expériences, nous utilisons le toolkit Word2Vec (Mikolov *et al.*, 2013a) pour extraire les *Word embeddings*. Cette approche consiste à entraîner un réseau de neurones linéaires, où la matrice des poids de la couche linéaire peut ainsi être interprétée comme une projection linéaire permettant de passer de l'espace des mots à une représentation vectorielle. Nous utilisons l'approche Continuous Bag of Words (CBOW) qui consiste à entraîner le réseau à prédire un mot à partir de son contexte.

4.2 Pré-traitements

Une étape de pré-traitement est appliquée aux tweets :

— Encodage des caractères : tous les tweets sont encodées au format UTF-8

3. `http://www.twitter.com/`

— Encodage des balises HTML : certains caractères ont des significations spéciales en HTML, et doivent être remplacés par des entités HTML (comme par exemple : &lt ;, &gt ;...)
— Minuscule : tous les caractères sont convertis en minuscule
— Rallongement : le rallongement des caractères qui consiste à répéter plusieurs fois un caractère dans un mot. C'est une méthode souvent utilisée sur le web pour insister sur un fait. Ce rallongement est souvent corrélé à un sentiment. Si un caractère est répété plus de trois fois, il sera réduit à trois caractères.
— Tokenization : la tokenization réalise le découpage d'une phrase en unités pré-lexicales. Cette tokenization est basée sur le toolkit macaon (maca_tokenize) (Nasr *et al.*, 2010). Il repose sur une grammaire régulière qui définit un ensemble de types d'atomes. Un analyseur lexical détecte les séquences de caractères (en fonction de la grammaire) et leur associe un type. Nous avons rajouté les atomes pour la détection des smileys, hashtags et noms d'utilisateurs (atome spécifique aux tweets).
— Ponctuation : nous supprimons ici tous les caractères de ponctuation.
— Stemming : le stemming consiste à supprimer le suffixe et le préfixe des mots, laissant ainsi son radical. L'algorithme utilisé est le *Porter stemming algorithm*.

L'ensemble des outils est disponible ici : `https://github.com/mrouvier/tweet_tokenizer_fr`.

5 Systèmes

Nous allons dans cette section présenter les cinq systèmes du niveau-1 puis le système de fusion du niveau-2.

5.1 SVM

Notre premier système appelé *SVM* consiste à ré-implémenter l'approche état-de-l'art pour l'analyse de sentiments sur les tweets de (Mohammad *et al.*, 2013). Ce système utilise comme classifieur un *Support Vector Machine* (SVM) et comme paramètre : un sac-de-mots (uni-gramme), un lexique d'émotion ainsi que des paramètres morphologiques.

Concernant le lexique d'émotion, de nombreux travaux ont montré l'importance d'utiliser des dictionnaires de polarités (Hatzivassiloglou & McKeown, 1997; Taboada *et al.*, 2011; Kanayama & Nasukawa, 2006; Wilson *et al.*, 2005). Ces lexiques d'émotion permettent d'augmenter l'espace des traits ou bien d'affiner la sélection des traits pertinents.

Nous proposons de créer un lexique d'émotion en français en traduisant de manière automatique les lexiques d'émotions disponibles en anglais (Bing Liu's Opinion Lexicon, MPQA Subjectivity Lexicon, SentiWordNet et Harvard General Inquirer). L'approche classique consiste à utiliser un système de traduction automatique. Malheureusement ces systèmes montrent leur limite lorsque l'on veut traduire un corpus de spécialité (comme ici les tweets).

Nous proposons d'utiliser l'approche *Bilingual word embeddings* (Zou *et al.*, 2013). Cette approche consiste à estimer une matrice de projection (mapping) d'un jeu d'embedding à un autre, tout en préservant (à des degrés différents) les structures syntaxique et sémantique. Plus concrètement, nous estimons un jeu de *Word embeddings* en anglais et en français. A l'aide d'un dictionnaire, nous apprenons une matrice de projection qui consiste à réduire la distance entre l'ensemble des paires du dictionnaire, d'un jeu d'embeddings, d'une langue à une autre. Ainsi, la traduction d'un mot en une autre langue se fait à l'aide des embeddings et de cette matrice de projection.

Cette approche a permis, par exemple, de traduire correctement l'expression "loool" par "mdrrr" ; ce qui n'aurait pas été le cas à l'aide d'un système de traduction classique.

Nous utilisons les paramètres morphologiques suivants :

— *All-caps* : Le nombre de mots en majuscule.
— *Emoticons* : Est-ce que le dernier token du tweet est un émoticon ?
— *Elongated units* : Le nombre de mots dont les caractères se répètent plus de deux fois (par exemple : looooool)
— *Punctuation* : Le nombre de séquences contigües de points, points d'exclamation et points d'interrogation

Dans nos expériences, les *Word embeddings* sont appris sur des corpus de tweets. Le dictionnaire utilisé est celui obtenu après alignement des bibtex du corpus Europarl. La taille des *Words embeddings* est de 300 et nous utilisons le classifieur libLINEAR [4].

4. http ://www.csie.ntu.edu.tw/ cjlin/liblinear/

5.2 DNN

Le deuxième système appelé *DNN* utilise comme classifieur un réseau de neurones profonds (Deep Neural Network - DNN) et deux paramètres : un sac-de-mots (bi-gramme) et un lexique de polarité (présenté dans la section précédente). Le DNN est composé de deux couches cachées, contenant chacune 2048 neurones. Les fonctions d'activation utilisées sont des *tanh* et nous utilisons comme DNN le toolkit Kaldi (Povey *et al.*, 2011).

5.3 CNN

Le troisième système appelé *CNN* consiste à ré-implémenter l'approche proposée dans (Collobert *et al.*, 2011; Kim, 2014). Cette approche est basée sur l'utilisation d'un réseau de neurones convolutionnels (Convolutionnal Neural Network - CNN), composé de trois couches cachées : la première consiste à extraire un vecteur pour chacun des mots ; la seconde est une couche convolutionnelle (qui partage les poids entre tous les mots) ; et la dernière est une couche de max-pooling.

Dans nos expériences, nous initialisons la première couche cachée du CNN avec la matrice des *Word embeddings* obtenue sur le corpus embeddings. La taille des *Word embeddings* est de 300, la taille du vecteur convolutionnel est de 400 et nous utilisons un dropout à 0.4.

Tout le matériel nécessaire (code source et données) pour reproduire les résultats sont accessibles ici : `https://github.com/mrouvier/tweet_cnn_fr`

5.4 Doc2vec

Ce quatrième système appelé *Doc2Vec* consiste à extraire un vecteur continu d'une phrase. Cette approche consiste à ré-implémenter l'approche proposée dans (Le & Mikolov, 2014). C'est une approche non-supervisée similaire à l'approche *Word2Vec*. L'avantage de cette approche est que le vecteur est extrait sur des phrases de tailles variables. Dans nos expériences, la taille du vecteur du système est de dimension 600 et le classifieur utilisé est un SVM.

5.5 Supervector

Ce cinquième et dernier système appelé *SuperVector* est basé sur l'idée des Speakers Embeddings proposé dans (). L'idée est de structurer l'espace des *Word embeddings* avec un modèle mises-von fisher, puis d'extraire les statistiques de premier ordre de ce modèle. Le SuperVector obtenu est utilisé comme paramètre d'entrée d'un DNN qui contient deux couches cachées de 2048 neurones chacune. La fonction d'activation utilisée est une *tanh*. Le toolkit utilise pour les DNN est celui de Kaldi.

5.6 Fusion

Dans l'optique d'améliorer les résultats, nous proposons de fusionner des systèmes, ce qui permet d'augmenter facilement la robustesse des règles de classification en multipliant les points de vue sur le même phénomène. Cette approche a été utilisée régulièrement dans les différents défis DEFT (Oger *et al.*, 2010; Torres-Moreno *et al.*, 2007, 2009; Grouin, 2014) et a permis d'améliorer les gains de classification. Nous proposons de réaliser la fusion au niveau des scores fournis par chaque système en ajoutant chacun d'eux dans un vecteur utilisé avec un classifieur de type SVM. L'idée est d'apprendre au SVM les différentes régularités existantes entre les systèmes.

6 Résultats

Nous reportons dans cette section les résultats pour les tâches 1, 2.1 et 2.2.

6.1 Tâche 1

Le Tableau 2 reporte les résultats obtenus par le système de fusion et les systèmes du niveau-1 (les systèmes *SVM*, *CNN*, *Doc2Vec*, *DNN* et *SuperVector*) pour la tâche 1. Le meilleur système du niveau-1 est le système *CNN* qui permet d'obtenir une macro-précision de 71.74%. Le système de fusion permet d'obtenir un gain de 1.86 points.

Système	Accuracy	Macro-precision
Fusion	**0.7257**	**0.7360**
SVM	0.6893	0.6882
CNN	0.7100	0.7174
Doc2Vec	0.6055	0.5970
DNN	0.6632	0.6600
SuperVector	0.5857	0.574

TABLE 2 – Résultat en terme d'*accuracy* et de macro-précision pour le système de fusion et les systèmes de niveau 1 sur la tâche 1.

6.2 Tâche 2.1

Le Tableau 3 reporte les résultats du système de fusion et les systèmes de niveau-1 pour la tâche 2.1. On constate que le meilleur système du niveau-1 est le système *CNN*. Il permet d'obtenir une macro-précision de 57.26%. Malheureusement dans cette tâche, le système de fusion n'a pas permis d'améliorer la macro-précision et obtient une macro-précision de 55.82%.

Système	Accuracy	Macro-precision
Fusion	**0.6188**	0.5582
SVM	0.5963	0.5355
CNN	0.6011	**0.5624**
Doc2Vec	0.5407	0.4350
DNN	0.5750	0.5000
SuperVector	0.5439	0.4702

TABLE 3 – Résultat en termes d'*accuracy* et de macro-précision pour le système de fusion et les systèmes de niveau 1 sur la tâche 2.1.

6.3 Tâche 2.2

Le Tableau 4 reporte les résultats du système de fusion et des systèmes de niveau 1 pour la tâche 2.2. Le meilleur système de niveau 1 est le système *SVM* qui permet d'obtenir 31.9% de macro-précision. Le système de fusion a permis d'améliorer les résultats et d'obtenir 32.69% de macro-précision (soit un gain de 0.79 point).

Système	Accuracy	Macro-precision
Fusion	**0.612**	**0.3269**
SVM	0.5981	0.319
CNN	0.6113	0.3065
Doc2Vec	0.5672	0.2805
DNN	0.5893	0.3062
SuperVector	0.5511	0.2922

TABLE 4 – Résultat en termes d'*accuracy* et de macro-précision pour le système de fusion et les systèmes de niveau 1 sur la tâche 2.2.

7 Conclusion et perspectives

La classification de tweets est une tâche qui peut être très difficile en fonction du type de tweets. Comme cela avait été constaté lors des défis précédents, "La lecture directe ne suffit pas toujours pour se forger un avis et privilégier une classification par rapport à une autre" (Torres-Moreno *et al.*, 2007). Nous avons utilisé des approches de représentation numérique et probabiliste afin de rester aussi indépendant que possible des sujets traités. Concernant les systèmes de base, le CNN obtient de bonnes performances sur l'ensemble des trois tâches, et la fusion des systèmes ont permis d'améliorer les résultats.

Remerciements

Ces travaux de recherche ont été financés en partie par l'Union Européenne à travers le projet SENSEI [5] (FP7/2007-2013 - nº 610916 – SENSEI).

Références

COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch.

GROUIN C. (2014). Les 10 ans du défi fouille de texte deft.

HATZIVASSILOGLOU V. & MCKEOWN K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the Computational linguistics*.

KANAYAMA H. & NASUKAWA T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

KIM Y. (2014). Convolutional neural networks for sentence classification.

LE Q. V. & MIKOLOV T. (2014). Distributed representations of sentences and documents.

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

MIKOLOV T., YIH W.-T. & ZWEIG G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the conference on Human Language Technology (HLT)*.

MOHAMMAD S. M., KIRITCHENKO S. & ZHU X. (2013). Nrc-canada : Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR)*.

NASR A., BÉCHET F. & REY J.-F. (2010). Macaon : Une chaîne linguistique pour le traitement de graphes de mots. *Actes de Traitement Automatique du Langage Naturel (TALN)*.

OGER S., ROUVIER M., CAMELIN N., KESSLER R., LEFÈVRE F. & TORRES-MORENO J.-M. (2010). Système du lia pour la campagne deft'10 : datation et localisation d'articles de presse francophones. *Actes du sixième Défi Fouille de Textes (DEFT)*.

POVEY D., GHOSHAL A., BOULIANNE G., BURGET L., GLEMBEK O., GOEL N., HANNEMANN M., MOTLÍČEK P., QIAN Y., SCHWARZ P. *et al.* (2011). The kaldi speech recognition toolkit.

TABOADA M., BROOKE J., TOFILOSKI M., VOLL K. & STEDE M. (2011). Lexicon-based methods for sentiment analysis. *Proceedings of the Computational linguistics*.

TORRES-MORENO J. M., EL-BÈZE M., BÉCHET F. & CAMELIN N. (2007). Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? application au défi deft 2007. *Actes du troisième Défi Fouille de Textes (DEFT)*.

TORRES-MORENO J.-M., EL-BÈZE M., BÉCHET F. & CAMELIN N. (2009). Fusion probabiliste appliquée à la détection et classification d'opinions. *Actes du cinquième Défi Fouille de Textes (DEFT)*.

WILSON T., WIEBE J. & HOFFMANN P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology (HLT)*.

ZOU W. Y., SOCHER R., CER D. M. & MANNING C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, p. 1393–1398.

5. http://www.sensei-conversation.eu