

## Est-ce que ce Tweet est drôle ? Détection automatique de tweets humoristiques

Florian Boudin<sup>1</sup> Adeline Granet<sup>2</sup> Alexis Linard<sup>2</sup>

(1) Laboratoire LINA, Université de Nantes, 2 rue de la Houssinière, BP 92208, 44322 Nantes, France

(2) Université de Nantes, 2 rue de la Houssinière, BP 92208, 44322 Nantes, France  
florian.boudin@univ-nantes.fr, adeline.granet@etu.univ-nantes.fr, alexis.linard@etu.univ-nantes.fr

**Résumé.** Nous présentons un outil, HTD4F (Humoristic Tweet Detection for French) qui prend en entrée une liste de tweets bruts issus de Tweeter (sans annotation linguistique) et qui fournit en sortie les tweets détectés comme humoristiques. Cet outil a donc comme fonctionnalité de détecter l'humour dans un tweet. HTD4F donne un taux de précision de 14,5

**Abstract.** We present a tool, HTD4F (Humoristic Tweet Detection for French) wich takes as input a list of raw tweets from Tweeter, and wich provides as output detected tweets as humoristical. This tool aims at detecting humor in a tweet. The precision rate for HTD4F is 14,5

**Mots-clés :** Twitter, Humour, Détection de tweets humoristiques, Weka, Classification.

**Keywords:** Twitter, Humor, Humoristic Tweet Detection, Weka, Classification.

## 1 Introduction

De nos jours, il n'y a plus de distinction nette entre la vie réelle et virtuelle. Il existe, chez les internautes, un besoin permanent de tout partager. Leurs succès, leurs échecs, leurs tracas, voir même leurs repas du midi prennent vie sur la toile, et ce sans aucune limite. Les outils les plus propices à cette déferlante d'informations sont les réseaux sociaux. Cet article s'intéresse à Twitter qui est rapidement devenu leader dans ce domaine avec plus de 500 millions d'utilisateurs.

Par son format limité à des publications de 140 caractères (appelé Tweet), Twitter demande aux utilisateurs de faire passer leurs émotions, leurs sentiments et leurs découvertes en étant le plus concis possible. C'est un fait, Twitter est une véritable mine d'informations grâce à la multitude de messages qui s'y trouvent, mais également à tout ce qui gravite autour. Car un tweet peut être retweeté(reposté) par d'autres utilisateurs, contenir des hashtags définissant parfois le thème dominant. Nous avons choisi de nous intéresser plus particulièrement aux tweets humoristiques.

Notre objectif est de développer un outil capable de détecter automatiquement si un tweet est drôle ou non. Voici un tweet que l'on souhaiterait classer : " Il court, il court le furet #Contrepeterie". De toute évidence, celui-ci est drôle car comme le hashtag le mentionne, il s'agit d'une contrepèterie.

Des approches similaires ont déjà été réalisées dans le domaine anglophone comme (Raz, 2012; Barbosa & Feng, 2010). Ces approches seront détaillées dans la section 2. La section 3 sera consacrée à la définition de ce qu'est un tweet avec tous les traits qui le caractérisent, ainsi que les méthodes que nous avons utilisées pour réaliser la classification des tweets grâce à l'outil Weka, et les traits que nous avons sélectionnés. La section 4 décrira le corpus d'entraînement qui a servi à construire le modèle ainsi que celui de test avec une présentation de l'accord inter-annotateurs utilisé. La section 5 expliquera en détail la démarche que nous avons suivie avec les résultats obtenus.

## 2 Etat de l'art

Les idées d'exploitations de tweet ne manquent pas du coté anglophone. Celui qui a largement inspiré la méthode présentée ici est (Raz, 2012). Dans cet article, Yishay Raz propose une méthode de classification de tweets humoristiques en anglais

selon le type de l'humour. Pour cela, il utilise un algorithme semi-supervisé qui prend en entrée des tweets annotés pour produire des ensembles avec des caractéristiques propres au classifieur. [...]

- Caractéristiques lexicales : les mots appartiennent à des lexiques particuliers, des entités nommées sont présentes, ou bien une ambiguïté se pose ;
- Caractéristiques morphologiques : analyse du temps des verbes, les mots existent-ils ;
- Phonologie : les mots sont-ils connus comme homophone ;
- Style : présence de smiley, ponctuation particulière, hashtag.

Cette approche est fortement intéressante. Malheureusement, une partie des caractéristiques nécessite d'avoir énormément de ressources de références. En français, il est difficile de trouver un lexique pour les mots vulgaires, du domaine gay, les entités nommées, les homophones, etc. L'évaluation de cette méthode a été réalisée en utilisant le site <http://www.funny-tweets.com> pour collecter un ensemble de tweets "drôles" ce qui a permis d'éviter un tri fastidieux à la main pour classer les tweets en drôle ou non. Depuis, ce site ne fonctionne plus. Nous avons donc cherché une alternative pour la collecte de tweets drôles francophones.

L'article de (Barbosa & Feng, 2010) sur la détection automatique de sentiment émis dans les tweets, montre qu'il y a beaucoup de travaux réalisés dans ce sens que se soit à travers des articles de recherche ou bien des sites proposant de la détection de sentiments en temps réels de tweets. Sa méthode repose sur trois caractéristiques principalement : le POS tagging, la polarité et la syntaxe spécifique du tweet comme les liens, la ponctuation, les émoticônes, ainsi que la casse des mots.

Une caractéristique commune aux deux articles est l'analyse du style qui n'est pas dépendante des bases de connaissances de la langue et donc exploitable dans notre étude.

### 3 Méthode de classification utilisée

#### 3.1 Le tweet

Comme cela a été mentionné plus tôt, Twitter permet de poster de courts messages de 140 caractères. Il y a certaines caractéristiques présentes dans le tweet qui sont intéressantes : l'utilisateur propriétaire qui est indiqué par 20 caractères commençant par le symbole "@" ; la mention ReTweeter avec le nom de l'utilisateur d'origine ; le hashtag "#ironie" est un mot clé donnant le thème du tweet ; les liens externes vers d'autres sources pour avoir la fin d'une blague par exemple "<http://bit.ly/114TdOF>". Parmi ces caractéristiques, un grand nombre seront exploités pour créer le modèle d'apprentissage comme cela est détaillé à la section suivante 3.2.

#### 3.2 Les traits

La tâche de classification consiste à séparer en deux classes distinctes les tweets : une classe "Drôle" et "Pas Drôle". Nous avons testé plusieurs algorithmes qui seront détaillés dans la section 3.3 Une méthode d'apprentissage non-supervisé est utilisée prenant en entrée un ensemble de tweets. Ce dernier va fournir différentes caractéristiques pour le classifieur multi-classe. Les caractéristiques étudiées sont de type lexicales, stylistiques et contextuelles.

##### Caractéristique lexicale

Lexique de mots : il a été construit à partir des tweets du corpus où chaque mot a été racinisé et auquel on a enlevé les mots creux et nettoyé le surplus comme les liens externes.

##### Caractéristique stylistique

- La présence de hashtag comme "#humour" : nous avons trouvé que les utilisateurs ajoutaient un "#humour" ou "#contre-pétie" pour identifier le thème sous-entendu dans leur message ;
- La présence de smiley content ou pas content au sein du tweet est observé. Par exemple "c'était pas moi ;)", la présence du smiley montre le caractère ironique de la phrase ;
- Le nombre de points d'exclamation est également pris en compte, par exemple dans une même phrase "je suis calme !" et "je suis calme !!!!!!!!!!" n'ont pas la même signification, et donne de l'intensité au message voir même de l'ironie dans notre cas.

### Caractéristique contextuel

- Le nombre de mots dans le tweet ;
- Le nombre de ReTweet ;
- La longueur total du tweet ;
- Si il s'agit d'un retweet.

## 3.3 Les méthodes utilisés

Pour les méthodes de classification que nous avons utilisées dans Weka, nous avons pris le parti de garder certains des paramètres par défaut.

**Naïvesbayes.** C'est le modèle probabiliste le plus utilisé. Il met en avant l'indépendance entre chaque caractéristique. Il utilise une hypothèse de distribution Gaussienne pour calculer la probabilité pour chaque caractéristique.

**J48** C'est une méthode d'apprentissage par arbre de décision. Un arbre va être construit de façon récursif en séparant les données suivant des tests sur les features définis.

**MultilayerPerceptron** Le Perceptron multicouche est un classifieur linéaire organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système.

**DecisionStump** Consiste en un arbre de décision à une seul niveau. C'est un arbre avec une racine immédiatement connectés à ses feuilles. En outre, donne une prédiction sur une valeur d'un seul trait. **RandomForest** L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

## 4 Le corpus

### 4.1 Corpus d'entraînement

Grâce à l'application twitter4j, deux corpus, un de 10 000 tweets et un autre de 15 000 ont été réalisés. Les deux corpus ont été constitués à partir de tweets drôles ou non : le premier est équilibré suivant les deux classes recherchées (tweets "Drôle" et "Pas Drôle") ; le second a une proportion de 2/5, où les tweets "drôle" sont les moins représentés, car nous avons pu observer que dans la réalité sur la quantité de tweets postés, très peu sont drôles. Le tableau suivant montre les statistiques des deux corpus :

	Equilibré		Deséquilibré	
	Tweets Drôle	Tweets Pas Drôles	Tweets Drôle	Tweets Pas Drôles
ReTweets	166	1019		
ReTweetés	2817	4009		
Non ReTweetés	2000	1273		
Contrepètries	200	/		
Autodérision	200	/		
Total	4817	5282	4785	10541

TABLE 1 – Composition des corpus d'entraînements

Pour extraire des tweets, il est important de commencer par choisir des comptes tweeter. Ces comptes doivent contenir soit uniquement (ou grande partie puisqu'il est difficile d'être sur) des tweets "drôles" ou des tweets "pas drôles".

Pour les comptes supposés "drôles", nous avons effectué une sélection de 35 comptes. Nous nous sommes basés essentiellement sur le nom de l'utilisateur contenant des mots clés comme "humour" et "blague". Voici des exemples de comptes que l'on a pu sélectionner "@100\_blagues, @BlaguesCarambar, @BlagueJour", il est facile de supposer que les tweets seront tous à caractères humoristiques et sûrement drôles. D'autres comptes ont été choisis par rapport à leur notoriété. Ils sont connus pour rassembler des tweets "drôles" par leur ironie ou montrant l'autodérision de leur auteur comme sur @viedemerde.

Pour les comptes répertoriant les tweets dits "pas drôles", nous en avons choisi 28. Nous avons supposé que tous les comptes politiques comme " @elysee ", journalistiques comme " @lemondefr, @LesEchos " ne contenaient pas de blagues, car ils sont supposés donner des informations sérieuses. Mais pour diversifier les domaines évoqués dans les tweets, nous avons ajouté des comptes commerciaux comme @m6, @nantesfr ou @conforama.

Ce corpus est utilisé pour entraîner le modèle d'apprentissage pour distinguer si un tweet est drôle ou non mais pas seulement. Il a été utilisé pour créer un lexique de mots caractéristiques des tweets. Pour cela, tous les mots ont été extraits puis racinisés. A cette liste, les mots creux de langue française, soit une liste de 460 mots, ont été retirés. Chaque mot restant constitue une caractéristique, un trait.

	Equilibré	Déséquilibré
Après nettoyage	14574	18973
Après racinisation	10168	12956

TABLE 2 – Constitutions des corpus d'entraînements

## 4.2 Corpus de test

Pour constituer le corpus de test, nous avons extrait 250 tweets provenant des comptes des humoristes Gad Elmaleh, Florence Foresti et Cyprien (un youtubeur). Nous les avons choisis, car ils sont représentatifs de Twitter c'est-à-dire une grande majorité de tweets sur le quotidien et quelques tweets drôles. Plus que quiconque ils sont plus susceptibles de par leur métier de poster des tweets drôles.

L'ensemble de ces tweets ont été annoté par 3 annotateurs. Chaque tweet a été annoté par deux annotateurs particulièrement, qui devaient indiquer la mention "Drôle" ou "Pas Drôle". Une fois que cela a été réalisé, un accord annotateur a été calculé. Nous avons utilisé le coefficient  $\kappa$  de (Cohen, 1960). Cet accord est utilisable dans notre cas, car nous n'avons que deux annotateurs pour chaque fichier. Et il nous permet de vérifier que le choix des classes n'est pas du au hasard. La table 3 montre les résultats que nous avons obtenus :

Document	$\kappa$
Fichier 1	0.978
Fichier 2	0.967
Fichier 3	0.974

TABLE 3 – Résultat accords inter-annotateurs

Nous sommes sur un accord inter-annotateurs presque parfait dans notre cas. Cela s'explique simplement. En effet, le calcul se base sur l'ensemble des tweets annotés or sur les 250 tweets annotés seulement moins d'une vingtaine ont été annotés comme "drôle" par au moins un annotateur. Les tweets dit "drôles" sont donc noyés dans la masse. Il est donc important de re-centrer l'accord inter-annotateur sur uniquement les tweets annotés au moins une fois comme "drôle".

## 5 Phase d'évaluation

### 5.1 Méthodologie

Nous avons commencé par créer les corpus d'entraînement, de test et la baseline.

Ensuite, nous avons construit les traits qui se basaient sur le corpus, comme le lexique de mots. Avant de créer le fichier contenant les caractéristiques de chaque tweet du corpus d'entraînement.

A l'aide de Weka, nous avons entraîné des modèles grâce aux méthodes de classifieurs cité à la section 3.3. Le but de notre démarche est de trouver des tweets " drôle " à coup sûr et non de trouver des tweets "pas drôle" comme étant "drôle". C'est pour cela que sur certaines méthodes nous avons ajouté une matrice de coût. Cette matrice de coût permet de sanctionner lourdement les erreurs de classification d'un tweet "pas drôle" en un tweet "drôle". Nous avons testé la classification avec

deux valeurs de matrice : la première avec un poids de 10 à chaque erreur et la seconde avec un poids de 100. Le détail des résultats avec cette matrice est décrite à la section 5.4.

Et pour finir, nous avons intégré directement la phase de test sur le corpus de test que nous avons préalablement annoté à la main.

## 5.2 Notre baseline

Nous avons créé un dernier corpus pour représenter notre baseline. Notre réflexion a été la suivante : si un tweet comporte un smiley, il est classé comme "drôle" et dans le cas contraire il sera classé comme "Pas drôle". Nous devons insister sur le fait que nous voulons un système très précis c'est à dire avec une forte précision. Le rappel est peu intéressant. En effet, nous voulons être sûr de ne pas classer un tweet "Pas Drôle" comme étant "Drôle" et plutôt que de classer un tweet comme "Pas drôle" alors qu'il était "Drôle", ce qui est moins grave.

## 5.3 Mesures d'évaluation

## 5.4 Les résultats

faire un joli tableau, comparer à une baseline ( mais laquelle ? par exemple 1 smiley = 1 tweet drôle mais il faut justifier cette baseline )

# 6 Conclusion et discussion

De notre expérience, nous en déduisons ceci : pour que les traits qui sont des mots soient exploitables, il faudrait y associer le contexte dans lequel il est sorti. Par exemple, dans le corpus de test, nous avons le tweet suivant "Ceux qui vous disent "j'ai dormi comme un bébé"... n'ont jamais eu de bébé :)" est selon nous "drôle". Le souci est que le mot "bébé" a la même période faisait les gros titres des journaux pour des faits graves. Sauf que sans avoir le contexte, le tweet cité sera classé comme "Pas Drôle" à cause du choix du corpus d'entraînement. L'étude du contexte de chaque tweet pourrait également aider à la recherche de passage ironique. [Même mettre une feature numérique plutôt que booléenne pour les mots ne garantirait pas d'éviter l'ambiguïté]

## Références

- BARBOSA L. & FENG J. (2010). Robust sentiment detection on twitter from biased and noisy data. p. 36–44 : Association for Computational Linguistics.
- COHEN J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, **20**(1), 37.
- RAZ Y. (2012). Automatic humor classification on twitter. p. 66–70 : The Association for Computational Linguistics.