

## Décomposition des « hash tags » pour l'amélioration de la classification en polarité des « tweets »

Caroline Brun<sup>1</sup>, Claude Roux<sup>1</sup>

(1) XRCE, 6, chemin de Maupertuis, 38240 Meylan  
caroline.brun@xerox.xrce.com, claude.roux@xerox.xrce.com,

**Résumé.** Les « mots dièses » ou « hash tags » sont le moyen naturel de lier entre eux différents tweets. Certains « hash tags » sont en fait de petites phrases dont la décomposition peut se révéler particulièrement utile lors d'une analyse d'opinion des tweets. Nous allons montrer dans cet article comment l'on peut automatiser cette décomposition et cette analyse de façon à améliorer la détection de la polarité des tweets.

**Abstract.** Hash tags are the natural way through which tweets are linked to each other. Some of these hash tags are actually little sentences, whose decompositions can prove quite useful when mining opinions from tweets. We will show in this article how we can automatically detect the inner polarity of these hash tags, through their decomposition and analysis.

**Mots-clés :** hash tag, tweet, analyse d'opinion, TAL

**Keywords:** hash tag, tweet, opinion mining, NLP.

### 1 Introduction

Twitter est devenu en quelques années le service de micro-blogging le plus populaire sur Internet. Les utilisateurs, appelés aussi « Tweetos » peuvent envoyer sur une plate-forme de partage de petits messages d'au plus 140 caractères, les tweets. De nombreuses expériences ont montré que ces « tweets » étaient souvent un très bon indicateur des choses à venir. Ainsi, [Asur et al. 2010] ont découvert que l'on pouvait prévoir le succès d'un film uniquement sur la base des tweets envoyés à son sujet. Les tweets jouent aussi un rôle dans les plans marketing des entreprises pour détecter comment leurs produits sont perçus par leurs clients. Les hash tags (ou hash tags) en particulier, jouent un rôle spécifique pour assurer un lien entre tous ces messages. Ces « hash tags » sont des métadonnées, des annotations libres définies par les utilisateurs qui servent à marquer l'appartenance d'un message à un domaine particulier, de construire un canal implicite de communication. [Wang et al. 2011] définit en particulier trois familles de hash tags :

- Sujet : hash tags utilisés pour définir grossièrement un sujet particulier : #Sarkozy, #LeDebat... ;
- Sentiment : #Idiot, #Deception, qui indiquent essentiellement une tonalité ;
- Sentiment-Sujet : hash tags, qui recoupent à la fois les sentiments et un sujet particulier : #ViveHollande, #SarkoOnTaime ;

De fait, comme les hash tags sont un élément essentiel des tweets, la plupart des systèmes d'analyse d'opinion cherchent à les incorporer dans leur calcul. [Davidov et al 2010] montre, par exemple, comment l'on peut améliorer les techniques standards de classification supervisée en intégrant la polarité des hash tags les plus fréquents comme paramètre, polarité qui est assignée ici manuellement. [Kouloumpis et al. 2011] ont aussi employé une méthode similaire, mais ont rajouté les émoticônes dans la détection de la polarité des tweets.

Cependant, définir à la main la polarité des hash tags n'a rien d'une aventure exaltante et peut se révéler plutôt coûteuse comme opération. [Dave et al. 2012] n'utilise la polarité des hash tags qu'à la condition qu'ils appartiennent à une liste connue à l'avance de mots tels que : #efficace, #nul, #incapable, #visionnaire etc. Mais, malheureusement, ils perdent là la majorité des expressions à mots multiples, dont pourtant raffolent les Tweetos. Pourtant, toutes ces expériences prouvent combien l'utilisation des hash tags peut se révéler précieuse dans l'évaluation de messages dont la taille s'avère souvent trop courte pour que les méthodes traditionnelles fonctionnent de manière optimale. Les hash tags sont par ailleurs souvent la clef pour déterminer l'ironie ou l'humour dans un message donné. Or, ils se révèlent particulièrement difficiles à analyser. Ils peuvent être aussi bien des noms propres, des noms de lieux ou des phrases complètes sans souvent le moindre indice sur leur construction interne.

Dans ce papier, nous proposons une méthode qui permet de détecter automatiquement la polarité d'un hash tag, en appliquant tout d'abord des méthodes de décomposition pour les segmenter, suivie d'une analyse avec une grammaire des sentiments pour en détecter la polarité.

## 2 Le contexte

L'une des tâches de base dans l'extraction d'opinion ou de sentiment est de classer la polarité d'un document selon plusieurs axes tels que: neutre, positif ou négatif. Il existe de nombreuses approches pour résoudre ce problème. Certains chercheurs classent les documents selon une échelle numérique, associant aux différents mots une valeur négative ou positive selon leur polarité propre. Ils en déduisent ainsi une polarité globale en fonction du nombre de mots positifs ou négatifs présents dans le document. Mais ces approches ont l'inconvénient de mal refléter les opinions contrastées d'un individu donné. Ainsi dans le cas d'un restaurant, il/elle peut apprécier l'ambiance mais critiquer le service ou la cuisine. L'approche à laquelle nous nous intéressons ici consiste donc à détecter ces différents « aspects » au sein d'un document, aspects pour lesquels nous voulons calculer une polarité adaptée, (voir [Hu & Liu 2004], [Liu 2012]). Ce travail nécessite une approche plus fine, le mot « rouge » par exemple peut être positif pour une viande, mais plutôt négatif s'il est appliqué à l'humeur d'un serveur.

### 2.1 Imagiweb

Nous avons réalisé l'ensemble de nos expériences avec le corpus de tweets du projet Imagiweb. Ce projet financé par l'ANR<sup>1</sup> consiste à étudier l'image de personnalités politiques ou de compagnies telle qu'elle apparaît sur la Toile, à travers des blogs ou des tweets. En particulier, l'une des tâches consiste à analyser les images de N. Sarkozy et de F. Hollande lors de la dernière élection présidentielle en Mai 2012, pour effectuer une analyse d'opinion sur certains aspects de la perception de ces hommes politiques au sein de la population des Tweetos.

Les données ont été extraites automatiquement de Twitter avec des requêtes appropriées par l'un des partenaires du projet. Le corpus ainsi construit correspond à environ 1500 utilisateurs, sur la base de 10 personnalités politiques différentes. De ce corpus ont été isolés environ 20.000 tweets, distribués de façon uniforme entre N. Sarkozy et F. Hollande, qui ont été ensuite annotés manuellement en polarité selon les aspects suivants : apparence physique, projet politique, sens éthique, communication etc. Soit environ 10 catégories, décomposées en 17 sous-cibles au total.

## 3 Décomposition

Les données dont nous nous servons au sein du projet ImagiWeb comprennent un grand nombre de « hash tags » correspondant à de petites phrases. Le terme « hash » réfère à une méthode d'indexation bien connue en informatique, le « hachage » qui consiste à calculer un index numérique pour une valeur donnée afin de la ranger dans une table de taille fixe. Par exemple, le calcul du reste d'une division «  $n : d$  » rend toujours une valeur comprise entre  $[0..d-1]$ , ce qui permet de ranger dans une table de dimension «  $d$  » n'importe quelle valeur.

Un « hash tag » commence toujours par le caractère « # » ce qui permet de le repérer très rapidement dans le flux d'analyse. Ces « hash tags » posent des problèmes très particuliers à l'analyse linguistique. En effet, ils sont considérés comme des mots inconnus et leur sémantique particulière se perd dans le traitement complet. Or, dans un « tweet » dont la longueur ne peut excéder 144 caractères, ignorer les « hash tags » peut conduire à une dégradation très forte de l'interprétation de celui-ci. En effet, les « hash tags » sont souvent employés soit pour véhiculer des informations sur l'auteur du tweet, soit pour répondre à un autre tweet, soit pour introduire un contre-point au message, sous la forme d'une remarque ironique ou amusante.

Voici quelques exemples de « hash tags » trouvés dans nos corpus: #MariagePourTous, #AvecSarkozy, #voteHollande, #placeaupleuple etc.

Les « hash tags » dans nos corpus tombent peu ou prou dans quatre catégories :

- a. Les hash tags qui correspondent à un seul mot de la langue
- b. Les hash tags qui correspondent à un nom propre

---

<sup>1</sup> ANR 2012-CORD-002-01

- c. Les hash tags qui correspondent à un groupe de mots
- d. Les hash tags impossibles à décoder, le plus souvent des acronymes ou des chiffres.

Les deux premières catégories sont très simples à analyser et ne requièrent pas de traitement très lourd. Il suffit de disposer de lexiques adéquats pour les traiter. La dernière catégorie elle contient une information qui pose déjà des problèmes à des humains pour les comprendre, nous n'avons pas essayé de les interpréter plus avant. Nous nous sommes en revanche concentrés sur la catégorie c) pour tenter de découper en mots le contenu. Peu de travaux s'intéressent à la segmentation des « hash tags », citons [Bakliwal et al. 2012] qui détectent les mots de polarité positive ou négative dans un « hash tag », par exemple « happy » dans « #Iamhappy », mais concluent qu'une segmentation serait plus appropriée, par exemple pour pouvoir détecter les négations. Très récemment, [Maynard & Greenwood 2014] segmentent également les « hash tags » pour améliorer la détection des tweets sarcastiques.

### 3.1 Découpage

Il existe nombre de méthodes pour découper des chaînes en unités lexicales, en particulier dans des langues agglutinatives comme l'allemand, où les mots composés sont non seulement nombreux mais peuvent souvent être construits librement. Un mot comme « Geburtstagfest », qui signifie « fête d'anniversaire » est composé de trois segments : Geburt (être né), Tag(le jour) et Fest (fête). Il existe des règles précises qui gouvernent la composition de ces mots, en particulier ici l'utilisation d'un « s » entre Geburt et Tag. [Koen et al 03] proposent différentes méthodes pour redécouper cette chaîne en ses racines premières, en se basant sur leurs fréquences au sein d'un corpus ou d'un lexique. Ils obtiennent ainsi une précision de l'ordre de 83%. Notre problème est très proche de celui du découpage d'un mot composé allemand, mais nous avons préféré utiliser des méthodes plus simples pour résoudre notre problème. Ainsi, pour découper les hash tags composés de plusieurs mots, nous avons utilisé différentes techniques selon la nature du « hash tag » lui-même. L'exemple le plus simple est l'utilisation de majuscule au début des mots pour les différencier entre eux. Dans ce cas, il suffit de repérer celles-ci pour le découper : #MariagePourTous donne « mariage pour tous ».

Cette méthode échoue souvent lorsque la chaîne est en minuscule mais contient un nom propre telle que : #avecHollande. On utilise alors une seconde méthode qui consiste à analyser la chaîne en deux phases. Dans un premier temps, on analyse la chaîne depuis le début et on tente de repérer la sous-chaîne la plus longue appartenant à nos lexiques. Nous choisissons la chaîne la plus longue pour éviter certains écueils. Par exemple dans le hash tag « #placeauple », on veut éviter d'arrêter le traitement après avoir détecté « peu », mais au contraire aller jusqu'au bout pour isoler « peuple ». On effectue la même opération en partant aussi de la fin. Analyser depuis la fin ou depuis le début ne donne pas forcément les mêmes résultats. Ce travail à priori simple présente une certaine combinatoire parfois difficile à contrôler. Nous travaillons d'ailleurs sur des méthodes plus sophistiquées, basées sur l'utilisation d'automates pour effectuer la reconnaissance des sous-chaînes.

### 3.2 Evaluation

On évalue ensuite les trois découpages en comptant le nombre de mots inconnus et on garde celui qui offre le score le plus faible. Nous avons appliqué cette méthode sur notre corpus et nous avons obtenu une précision d'environ 80% pour les 1132 « hash tags » extraits de nos 20.000 tweets. Cette précision a été obtenue en vérifiant chacun des découpages produits à la main. Pour information, sur les 1132 hash tags, 524 présentaient une possibilité de découpage (soit 46%), et parmi eux 160 (soit 30% de ces mots composés ou 14% du total) utilisaient une délimitation avec majuscule.

## 4 Intégration dans un système de détection d'opinion

### 4.1 Modèle d'une opinion

D'un point de vue formel, notre système de détection d'opinion adopte la représentation d'une opinion selon le modèle proposé par [Liu 2010], où une opinion est un prédicat d'arité 5 de la forme  $(O_j, f_{jk}, SO_{ijkl}, h_i, t_i)$  avec :

- $O_j$  est l'objet cible de l'opinion (le concept principal)  $f_{jk}$  est un aspect associé à l'objet  $O_j$
- $SO_{ijkl}$  est la valeur (positive ou négative) de l'opinion exprimée par le locuteur  $h_i$  à propos de l'aspect  $f_{jk}$
- $h_i$  désigne le locuteur

- $t_i$  est le moment où l’opinion a été exprimée

Notre système d’extraction est basé sur un analyseur syntaxique qui fournit pour chaque énoncé un ensemble de dépendances syntaxiques. Celles-ci sont alors réinterprétées pour produire des relations sémantiques qui nous servent à instancier nos prédicats.

## 4.2 Détection de l’opinion

Nous utilisons notre analyseur syntaxique comme composant fondamental pour extraire les dépendances syntaxiques profondes à partir desquelles sont construites nos relations sémantiques, à partir desquelles les prédicats sont instanciés. Elles sont encodées sous la forme suivante : OPINION[POLARITE](POLAR-PREDICAT,OPINION-CIBLE), où OPINION est le nom de la relation sémantique, POLARITE un trait associé avec la dépendance, dont les valeurs sont : “POSITIVE” ou “NEGATIVE”. POLAR-PREDICAT est l’expression dans l’énoncé portant la polarité de l’opinion et OPINION-CIBLE, la cible de l’opinion.

Ces relations sémantiques sont en fait produites par des règles particulières au sein de notre analyseur robuste, qui combine à cette fin, informations lexicales spécialisées et dépendances syntaxiques. En particulier notre lexique comprend aussi bien des mots du domaine dont la polarité est connue que les hash tags extraits préalablement. Si les lexiques ont été construits en appliquant des techniques de clustering et de classifications sur de larges corpus ([Brun 2012]), les règles sémantiques sont développées manuellement ([Brun 2011]). Ces règles prennent la forme suivante :

If (SUJET(#1[!polarité:!, #2]) $\implies$  SENTIMENT[polarité](#2,#1)

Cette règle peut se lire de la façon suivante. Si un verbe donné (ici #1) portant une certaine polarité est sujet d’un nom #2, alors on produit une dépendance sémantique SENTIMENT dont la polarité sera celle de ce verbe. (!polarité :! déclenche une percolation du trait à partir du verbe, lequel est alors capturé par la dépendance).

Exemples :

- Ces enchiladas#2 m’émerveillent#1 sans cesse.
- Cette voiture#2 m’a profondément déçue#1.

Environ une centaine de règles ont été ainsi écrites pour couvrir un large éventail de structures identifiées dans les corpus d’apprentissage. Notre système appartient à la même famille que celui de [Kim et Hovy 2006] ou [Wu et al. 2009], qui utilisent aussi des dépendances syntaxiques pour lier la source et la cible des opinions. De fait, avec ce type d’approche, il devient possible d’intégrer le traitement de phénomènes complexes comme la négation par exemple.

## 4.3 Adaptations aux tweets

Dans une première passe, nous avons détecté tous les hash tags présents dans le corpus. Puis nous avons appliqué notre mécanisme de découpage sur chacun d’entre eux. Comme nous disposons à l’interne d’une grammaire d’opinion, enrichie d’un vocabulaire spécifique au domaine que nous traitons, nous avons pu l’appliquer sur ces décompositions de façon à attribuer à chacun de ces hash tags une polarité positive ou négative. De cette façon, nous avons pu composer un lexique de hash tags, considérés comme des noms, chacun associé avec un trait particulier pour indiquer sa polarité, et éventuellement sa cible.

A partir des 896 hash tags correctement décomposés, nous avons obtenu les résultats suivants :

- 215 hash tags encodant à la fois la polarité et la cible comme: #VotezHollande,#HollandeHonte, #SarkoOnTaime
- 304 hash tags encodant seulement la polarité : #Abruti, #CasseToi
- 377 hash tags encodant seulement le sujet, parmi lesquels 169 sont des entités nommées.

Une fois cette analyse effectuée, nous les avons enregistrés dans un lexique spécialisé sous la forme suivante:

- “#SarkoPipo”:    noun[negative=+,target=”Sarkozy”].
- “#VoteHollande” : noun[positive=+, target=”Hollande”].
- “#CasseToi” :    noun[negative=+].

Ce lexique a été alors intégré à notre grammaire d’opinion avant d’appliquer celle-ci à notre corpus de tweets. De cette façon, lors du traitement linguistique, les hash tags sont interprétés comme des noms par les couches basses de la grammaire (segmentation et analyse morphologique) puis enrichis avec les informations de polarité provenant de ces nouveaux lexiques spécialisés. De cette façon, les données de polarité introduites par les hash tags peuvent être intégrées dans le calcul de polarité de chacun des aspects dans les tweets.

## 5 Expériences et résultats

Pour évaluer l'impact de l'intégration de la polarité des hash tags et des cibles dans notre système de détection d'opinion, nous avons procédé à plusieurs expériences de classification sur notre corpus Imagiweb de tweets annotés. Ce corpus comprend donc 3920 tweets dont les opinions sont annotées en polarité (ici positive ou négative) et en cible, comprenant un total de 392 hash tags décomposés. La mesure de performance choisie est l'exactitude de la classification. Nous avons comparé le comportement de notre système sur la détection de la polarité des tweets une première fois sans les hash tags, et une seconde fois en les intégrant. Nous avons utilisés les sorties de notre analyseur pour entraîner un SVM avec différents paramétrages (SVMLight, [Joachims 1999]), de tels classifieurs s'étant montrés performants pour des tâches de classification de textes. Pour chaque expérience, la classification est effectuée sur le même ensemble de données d'entraînement et de test, extraites de façon aléatoire du corpus initial, tandis que les résultats sont calculés via une procédure de validation croisée en 10 découpages différents (ten-fold). Le jeu de test correspond à 10% du corpus initial, et le jeu d'entraînement au 90% restant. Chaque ensemble comprend la même distribution de tweets positifs et négatifs.

- Expérience 1 (référence) est basé sur l'approche en sac de mots.
- Expérience 2 (hash tags) rajoute les hash tags et leur polarité
- Expérience 3 (opinions) rajoute les opinions détectées sans les hash tags
- Expérience 4 (opinions+hashtags) intègre les opinions détectées et les hash tags.

L'exactitude moyenne observée lors d'une validation croisée est estimée selon une erreur moyenne quadratique. La table suivante résume le résultat des quatre expériences.

Expériences	Exactitude %
Expérience 1 : sac de mots	80,1
Expérience 2 : hash tag	82,6
Expérience 3 : opinions	80,2
Expérience 4: opinions+hash tags	82,2

Alors que l'utilisation des relations d'opinion dans l'entraînement de la classification n'a pas d'impact significatif, l'utilisation des hash tags améliore l'exactitude de la classification d'environ 2%.

Pour confirmer ce résultat, nous avons effectué les mêmes expériences avec un sous-ensemble du corpus initial, dans lesquels ne sont conservés que les tweets contenant des hash tags :

Expériences	EXACTITUDE %
Expérience 1 : sac de mots	79,9
Expérience 2 : hash tag	84,6
Expérience 3 : opinions	80,1
Expérience 4: opinions+hash tags	84,7

Dans ce dernier cas, l'amélioration sur la tâche de classification est passée à 4,8% par rapport à la référence. Ces résultats prouvent sans ambiguïté l'impact significatif de l'intégration des hash tags et de leur polarité dans la classification des tweets.

## 6 Conclusion

Dans ce papier, nous avons proposé un mécanisme de décomposition automatique et d'analyse des « hash tags » de façon à pouvoir utiliser l'information sémantique dont ils sont porteurs dans une tâche de classification. Les différentes expériences que nous avons conduites montrent que ces hash tags ont un impact important dans la polarité des tweets.

De plus, notre méthode présente l'avantage d'être totalement automatique, et permet d'exploiter efficacement près de 80% des « hash tags » complexes présents dans les tweets.

## Références

- AIT-MOKTHAR, S., CHANOD, J.P. (2002). Robustness beyond Shallowness: Incremental Dependency Parsing. *Numero Special du NLE Journal*.
- ASUR, S., & HUBERMAN, B. A. (2010). Predicting the future with social media. Actes de *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference* (Vol. 1, pp. 492-499). IEEE.
- AKSHAT BAKLIWAL, PIYUSH ARORA, SENTHIL MADHAPPAN, NIKHIL KAPRE, MUKESH SINGH, AND VASUDEVA VARMA. (2012). Mining sentiments from Tweets. Actes du *3eme Workshop in Computational Approaches to Subjectivity and Sentiment Analysis (WASSA '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 11-18.
- BRUN C. (2011). Detecting Opinions Using Deep Syntactic Analysis. Actes de *RANLP, Recent Advances in Natural Language Processing*, Hissar, Bulgaria, September 12-14, 2011.
- BRUN C. (2012). Learning Opinionated Patterns for Contextual Opinion Detection, Actes de *Coling2012*, Décembre 2012, Bombay, Inde.
- DAVE, K. S., & VARMA, V. (2012, May). Identifying microblogs for targeted contextual advertising. Actes de *Sixth International AAAI Conference on Weblogs and Social Media*.
- DAVIDOV D., OREN TSUR, AND ARI RAPPOPORT. (2010). Enhanced sentiment learning using Twitter hashtags and smileys. Actes de *COLING '10*. ACL, Stroudsburg, PA, USA, 241-249.
- DEVEAUD, R., & BOUDIN, F. (2012). *LIA/LINA at the INEX 2012 Tweet Contextualization track*. Initiative for the Evaluation of XML Retrieval (INEX).
- HU. M., LIU, B. (2004). Mining and summarizing customer reviews. Actes de *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*, Seattle, Washington, USA.
- JOACHIMS T. (1999). *Making large-Scale SVM Learning Practical*. Advances in Kernel Methods – Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press.
- KOEHN, P., & KNIGHT, K. (2003). Empirical methods for compound splitting. Actes de *the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1* (pp. 187-193). Association for Computational Linguistics.
- KOULOUMPIIS, E., WILSON, T. ET MOORE, J. (2011), Twitter Sentiment Analysis: The Good the Bad and the OMG!, in Lada A. Adamic; Ricardo A. Baeza-Yates & Scott Counts, ed., 'ICWSM', The AAAI Press.
- LAKE, T., & FITZGERALD, W. (2011). Twitter Sentiment Analysis. *Western Michigan University, Kalamazoo, MI*, For client William Fitzgerald.
- LEVIN, BETH. (1993). English Verb Classes and Alternations A Preliminary Investigation. University of Chicago Press, Chicago and London.
- LIU, B. (2010). Sentiment Analysis and Subjectivity, *Chapter of Handbook of Natural Language Processing*, 2nd edition.
- MAYNARD D, GREENWOOD, M.A. (2014). Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. Actes de *LREC'2014*, Reykjavik, Finlande.
- RAMAGE, D., DUMAIS, S., & LIEBLING, D. (2010). Characterizing microblogs with topic models. Actes de *International AAAI Conference on Weblogs and Social Media* (Vol. 5, No. 4, pp. 130-137).
- RUPPENHOFFER, J., ELLSWORTH M, PETRUCK M, JOHNSON C., ET SCHEFFCZYK J. (2005). FrameNet II: Extended theory and practice. *Technical report*, ICSI.
- THELWALL, M., BUCKLEY, K., ET PALTOGLOU, G. (2011). Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology*, 62(2), 406-418.
- WANG XIAOLONG, FURU WEI, XIAOHUA LIU, MING ZHOU, ET MING ZHANG. (2011). Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. Actes de *CIKM '11*, Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis, and Ian Ruthven (Eds.). ACM, New York, NY, USA, 1031-1040.
- WU, YUANBIN, ZHANG, QI, HUANG, XUANJING, HUANG ET WU, LIDE. (2009). Phrase Dependency Parsing for Opinion Mining. Actes de *EMNLP'09, Vol3*. Association for Computational Linguistics, Stroudsburg PA, USA, 1533-1541.