UNIVERSITÉ DU QUÉBEC À MONTRÉAL

CLASSIFICATION SUPERVISÉE DE TEXTES COURTS ET BRUITÉS : APPLICATION AU DOMAINE DES MÉDIAS SOCIAUX

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

BILLAL BELAININE

AVRIL 2017

REMERCIEMENTS

Je tiens à exprimer ici ma reconnaissance à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire de maîtrise.

Tout d'abord, je tiens à remercier ma directrice de recherche, Professeure Fatiha Sadat, pour son aide précieuse, sa patience et son encouragement, mais surtout pour avoir cru en moi et m'avoir donné l'assurance de croire en moi-même.

Je tiens à remercier les professeurs Abdellatif Obaid et Hakim Lounis pour leurs précieuses évaluations de ce mémoire de maîtrise et recommandations ainsi que les autres professeurs du département d'Informatique de l'UQAM pour la qualité de leur enseignement lors de ma maîtrise.

Je tiens à remercier le Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG) et l'entreprise Nexalogie Envoronics pour avoir contribué dans le financement d'une partie de ce travail de recherche.

Je tiens également à remercier tous mes amis qui m'ont apporté leur soutien moral et intellectuel tout au long de ces années d'études. Finalement, je voudrais exprimer toute ma reconnaissance à l'égard de ma famille, à qui je dois tout et sans qui ce travail n'aurait jamais vu le jour.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

Les données massives (*Big data*) possèdent un important potentiel scientifique, spécifiquement dans les domaines du forage de données, apprentissage machine et traitement des langues naturelles.

Ce travail de recherche concerne l'analyse automatique de grandes masses de données non structurées et hautement bruitées, extraites des tweets, afin d'automatiser un système de classification de ces tweets.

Notre première contribution concerne le filtrage par catégories grammaticales et le prétraitement de ce genre de données hautement bruitées et courtes, comportant 140 caractères au maximum pour chaque tweet.
Notre deuxième contribution a trait à la reconnaissance des entités nommées (REN) dans les tweets, qui est une tâche très difficile. Ainsi, l'adaptation des outils linguistiques existants pour les langues naturelles, au langage bruité et non précis des tweets, est nécessaire.
Notre troisième contribution implique une segmentation des hashtags ainsi qu'un enrichissement sémantique à l'aide d'une combinaison de relations de WordNet, ce qui a aidé la performance de notre système de classification, notamment en désambigüisant les entités nommées, abréviations et acronymes. La théorie des graphes a été utilisée pour regrouper les mots extraits de WordNet et des tweets, en se basant sur les composantes connexes.

Notre système automatique de classification concerne les quatre catégories suivantes : politique, économie, sport et le domaine médical. Nous avons évalué et comparé plusieurs systèmes de classification automatique et constaté que l'étape de filtrage par catégorie grammaticale ainsi que la reconnaissance des entités nommées augmentent considérablement la précision de la classification jusqu'à 77.3%. De plus, un système de classification incorporant une segmentation des hashtags ainsi qu'un enrichissement sémantique à l'aide des deux relations de synonymie et d'hyperonymie de WordNet augmentent la précision de la classification jusqu'à 83.4%.

Mots clés :
Forage de données, classification, big data, médias sociaux, Twitter, WordNet, Hashtag.

INTRODUCTION GÉNÉRALE

Introduction

Le Traitement Automatique du Langage Naturel (TALN) est la discipline s'intéressant à l'automatisation du traitement de certains aspects du langage humain. Cette discipline a connu une croissance importante ces dernières années grâce aux avancées récentes en intelligence artificielle et est maintenant appliquée dans plusieurs domaines.
De nombreuses entreprises et chercheurs en linguistique informatique s'intéressent à l'analyse automatique du contenus. Cette discipline se retrouve au cœur des débats avec l'avènement des médias sociaux et le *Big data*.

En 2001, un rapport de recherche du Groupe Gartner (Laney, 2001) définit les enjeux inhérents à la croissance des données comme étant tridimensionnels selon la règle dite « des 3V » (volume, vélocité et variété). Ce modèle est encore largement utilisé aujourd'hui pour décrire ce phénomène (Lemberger *et al.*, 2015; Laney, 2001).

En effet, le Big data possède un important potentiel scientifique. Les chercheurs et autres professionnels cherchent aujourd'hui l'outil idéal leur permettant d'analyser automatiquement ces grandes masses de données hautement bruitées afin d'automatiser certaines tâches ou extraire l'information enfouie dans ces grandes masses d'information et ainsi développer des applications informatiques spécialisées en TALN pour ce genre de données. Aussi, les moteurs de recherche, les systèmes de traduction automatique et les assistants personnels intelligents dans les téléphones cellulaires découlent tous des recherches effectuées dans ce domaine.

À l'origine, les données extraites des médias sociaux sont issues de sources ouvertes obtenues à partir de blogues, de micro blogues, de forums de discussion, d'outils de clavardages, de jeux en ligne, d'annotations, de classements, de commentaires et de FAQ générées par des utilisateurs. Ces données possèdent de nombreuses propriétés.

Ce type de textes, rédigés par des auteurs différents dans une variété de langues et de styles, n'adoptent aucune structure précise et se présentent sous une multitude de formats. Par ailleurs, les erreurs typographiques et l'argot propres au clavardage sont maintenant courants sur les réseaux sociaux, notamment sur Facebook et Twitter. L'analyse et la veille de ce riche contenu sans cesse renouvelé donnent accès à une information précieuse que les médias traditionnels ne peuvent fournir (Melville et Sindhwani, 2009).

L'analyse sémantique des médias sociaux a ouvert la voie à l'analyse de données volumineuses, discipline émergente inspirée de l'analyse des réseaux sociaux, de l'apprentissage automatique, de l'exploration de données, de la recherche documentaire, de la traduction automatique (Gotti *et al.*, 2014), du résumé automatique (Farzindar et Roche, 2015) et du TAL plus globalement.

Par exemple, Twitter qui nous intéresse particulièrement dans ce travail, constitue une source continue et illimitée de données en langage naturel qui est particulièrement difficile à traiter avec les approches classiques de traitement automatique du langage naturel (TAL). Ce type de langage est très éloigné des normes du langage traditionnel, avec ses conventions (telles que les hashtags, les mentions, les retweet, etc.). Son lexique particulier est souvent grossier et contient des abréviations, des émoticons, des acronymes. Sa syntaxe est parcellaire dans le meilleur des cas. Les données extraites de Twitter sont hautement bruitées, non-structurées, et courtes (comportant au maximum 140 caractères par tweet).

La classification et catégorisation de documents est l'activité du traitement automatique des langues naturelles qui consiste à classer de façon automatique des ressources documentaires, généralement en provenance d'un corpus (Jaillet *et al.*, 2003).

Dans le cas des tweets, la classification consiste à annoter les différentes phrases d'un tweet avec des classes (exemple : sport, politique, éducation, etc.). Pour chaque classe C, on trouve des termes importants considérés comme des indicatifs pour la classe C (Liu, 2006). Par exemple, les termes loi, gouvernement, président et justice sont des indicatifs du sujet politique. Cependant, les textes courts des tweets ne fournissent pas assez d'occurrences de mots. Ainsi, les méthodes de classification qui utilisent les approches traditionnelles telles que les Sacs de mots sont limitées, car les mots ne se répètent pas assez et génèrent des matrices creuses, ayant des tailles indéterminées. Pour pallier à ce problème, nous proposons l'utilisation des méthodes destinées au prétraitement des tweets ainsi qu'une adaptation des méthodes traditionnelles de classification.

Les travaux existants sur la classification des messages courts intègrent chaque message avec des méta-informations à partir des sources d'information externe telles que Wikipédia (Genc *et al.*, 2011) et BableNet(Faralli *et al.*, 2015). Ces travaux vont jusqu'à l'utilisation des ontologies comme DBpedia (Cano *et al.*, 2013; Navigli et Ponzetto, 2012), ou WordNet ou autres bases lexicales (Montejo-Ráez *et al.*, 2014).

Dans notre travail nous procédons à l'analyse et la classification des textes extraits de Twitter. Nous sommes particulièrement concentrés sur les deux phases de prétraitement et de représentation des tweets avant d'utiliser un algorithme de classification adapté au Big data.

Problématique

Actuellement, il existe un grand intérêt académique et industriel pour le traitement automatique des langues naturelles, l'apprentissage machine, la traduction automatique ou l'extraction d'information telle que les entités nommées. La majorité de ces outils s'appuient sur des corpus relativement structurés et sans bruits. Cependant, les textes bruités comme les tweets, compliquent les tâches liées aux applications du TALN.

De plus, les réseaux sociaux comme Twitter ont des caractéristiques spécifiques, comme l'existence des métadonnées telles que les hashtags. Ceci rend notre tâche plus complexe, malgré que ces hashtags ont été définis par Twitter dans le but de regrouper les tweets selon leurs sujets de discussion (Farzindar et Roche, 2013).

Quelques travaux se sont concentrés sur le regroupement des tweets selon leurs hashtags, au lieu de décomposer ces hashtags et les traiter comme des éléments composés afin d'extraire les informations nécessaires et pertinentes des textes de Twitter.

Un deuxième problème est lié à la longueur des tweets. En effet, les textes de Twitter sont des textes courts, ne dépassent pas 140 caractères. Ces textes génèrent un problème durant l'analyse surtout que les mots ne se répètent pas suffisamment. Les fréquences des mots varient entre une répétition ou aucune, rendant l'approche traditionnelle comme les sacs de mots difficiles à réaliser et la matrice générée très creuse.
Sachant que les médias sociaux sont réputés à utiliser les variantes de langues et la langue de rue ou plutôt l'argot. Dans ce cas, les termes d'argot, les abréviations, les onomatopées, les acronymes et d'autres termes sont inventés par le grand publique. De plus, on retrouve dans un tweet non seulement des termes issus du néologisme mais aussi empruntés des autres langues étrangères causant ainsi un

sérieux problème à l'apprenant automatique.
L'identification de la langue d'un tweet est un problème majeur dans les applications TALN. On note aussi dans les tweets une richesse, qui concerne les métadonnées comme les émoticônes, les hashtags et l'existence des références comme, les URLs et les adresses utilisateurs de Twitter.

Dans ce travail de recherche, notre objectif est de remédier au problème de l'apprenant automatique, en développant un système dédié à la classification des tweets. On se base sur la segmentation des hashtags afin d'extraire le maximum d'informations et aussi sur la reconnaissance des entités nommées. On se base également sur la normalisation automatisée de ces données réelles et hautement bruitées en utilisant les dictionnaires et un algorithme raffiné avec le thésaurus WordNet. Le principal intérêt d'une tâche, comme celle-ci, est de simplifier et d'améliorer l'analyse d'un texte bruité. Une simplification opérée grâce à des outils de traitement automatique de la langue. Ces outils conçus avant tout pour le traitement de textes édités. En plus, pour régler les problèmes du nombre de mots ambigus extraits à partir de tweets, nous proposons la désambigüisation des textes courts à l'aide des relations sémantiques de la base de données lexicale WordNet.

Objectifs

Notre objectif global consiste à trouver une façon d'exploiter les contenus des tweets afin de les classifier selon les catégories pertinentes.
Nous visons les objectifs spécifiques suivants :

1. Reconnaissance des entités nommées afin d'améliorer la qualité de la classification.
2. Filtrage des textes en vue d'éliminer les mots vides non trouvés dans la liste traditionnelle des mots des tweets.
3. Segmentation des hashtags afin d'extraire l'information pertinente des tweets.

4. Étude des relations sémantiques afin de désambigüiser les mots des tweets et de remédier à des problèmes tels que de la polysémie (i.e. mots comprenant plusieurs sens).

Structure du document

Ce document est structuré comme suit. Dans le chapitre 1, nous présentons les concepts de base de la classification des documents. Le chapitre 2 présente l'état de l'art lié au sujet de recherche. Le chapitre 3 fournit une vue d'ensemble sur Twitter et ses caractéristiques. Les procédures de reconnaissance des entités nommées sont présentées dans le chapitre 4. Le chapitre 5 présente la base lexicale WordNet et ses différentes relations sémantiques. Nous discuterons la méthodologie suivie pour le prétraitement et la classification des tweets dans le chapitre 6. Les différentes évaluations et résultats des expérimentations sont abordés dans le chapitre 7. Enfin, nous présentons les conclusions de notre travail ainsi que les perspectives futures.

CHAPITRE I

CONCEPTS DE BASE SUR LA CLASSIFICATION DE TEXTES

1.1 Apprentissage machine

L'apprentissage machine est une tentative de comprendre et reproduire la faculté de l'apprentissage humain dans des systèmes artificiels. Il s'agit de concevoir des algorithmes capables, à partir d'un nombre important d'exemples, d'en assimiler la nature afin de pouvoir appliquer ce qu'ils ont ainsi appris aux cas futurs. Ainsi, le but essentiel de l'apprentissage machine est de déterminer la relation entre les objets et leurs catégories pour la prédiction et la découverte des connaissances (Silva et Ribeiro, 2009).
On distingue ainsi trois types d'apprentissage : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage semi-supervisé.

1.1.1 Apprentissage supervisé(Classification)

L'apprentissage supervisé (ou classification) consiste à construire un modèle basé sur un jeu d'apprentissage et des labels (nom des catégories ou des classes) et à l'utiliser pour classer des données nouvelles (Silva et Ribeiro, 2009; Joachims, 2002). Cette technique est utilisée dans plusieurs applications telles que les diagnostics médicaux, la prédiction des pannes et la détection des opinions trompeuses dans les réseaux sociaux.

Il existe plusieurs algorithmes et techniques utilisés pour la classification supervisée telles que :

- Classification Bayésienne : C'est une méthode de classification statistique qui se base principalement sur le théorème de Bayes. Elle est utilisée dans plusieurs applications telles que les applications de détection de pourriels (ou *Spams*) pour séparer les bons courriels des mauvais.

- Machine à vecteurs de support (*SVM*) : Il s'agit d'un ensemble de techniques destinées à résoudre des problèmes de discrimination (prédiction d'appartenance à des groupes prédéfinis) et de régression (analyse de la relation d'une variable par rapport à d'autres) (Silva et Ribeiro, 2009).

- Réseau neuronaux : c'est une technique de type induction c'est-à-dire que, par le biais d'observations limitées, elle essaye de tirer des généralisations plausibles. Elle est basée sur l'expérience qui se constitue une mémoire lors de la phase d'apprentissage (qui peut être aussi non supervisée) appelée entraînement (Silva et Ribeiro, 2009).

- Forêts d'arbres décisionnels (*Random Forest*) : C'est une application de graphe en arbres de décision permettant ainsi la modélisation de chaque résultat sur une branche en fonction des choix précédents. On prend ensuite la meilleure décision en fonction des résultats qui suivront. On peut considérer ceci comme une forme d'anticipation.

- Le Boosting : Il s'agit d'une méthode de classification émettant des hypothèses qui sont au départ de moindre importance. Plus une hypothèse est vérifiée, plus son indice de confiance augmente. Ce qui prend de l'importance dans la classification(Silva et Ribeiro, 2009).

1.1.2 Apprentissage non supervisé

L'apprentissage non supervisé (en anglais. *clustering*) vise à construire des groupes (clusters) d'objets similaires à partir d'un ensemble hétérogène d'objets (Silva et Ribeiro, 2009). Chaque cluster issu de ce processus doit vérifier les deux propriétés suivantes :

- La cohésion interne (les objets appartenant à ce cluster sont les plus similaires possibles).
- L'isolation externe (les objets appartenant aux autres clusters sont les plus distincts possibles).

Le processus de « *clustering* » repose sur une mesure précise de la similarité des objets qu'on veut regrouper. Cette mesure est appelée distance ou métrique. Le « *clustering* » est utilisé dans plusieurs applications telles que le traitement d'images, les études démographiques, la recherche génétique, le forage des données et l'analyse des opinion. On distingue plusieurs algorithmes de *clustering*, exemple :

- K-moyennes (*KMeans*) :Un algorithme de partitionnement des données en K groupes ou clusters. Chaque objet sera associé à un seul cluster. Le K est fixé par l'utilisateur.
- Fuzzy KMeans : Il s'agit d'une variante du précédent algorithme proposant qu'un objet ne soit pas associé qu'à un seul groupe.
- Espérance-Maximisation (EM) : Cet algorithme utilise des probabilités pour décrire qu'un objet appartient à un groupe. Le centre du groupe est ensuite recalculé par rapport à la moyenne des probabilités de chaque objet du groupe.
- Regroupement hiérarchique : deux sous-algorithmes en découlent : le « *bottom up* » qui a pour fonction d'agglomérer des groupes similaires, donc en

réduire le nombre (les rendre plus lisibles) et d'en proposer un ordre hiérarchique et le «*top down* » qui fait le résonnement inverse en divisant le premier groupe récursivement en sous-ensembles.

1.1.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé utilise un ensemble de données étiquetées et non-étiquetés. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées. L'utilisation de données non-étiquetées, en combinaison avec des données étiquetées, permet d'améliorer de façon significative la qualité de l'apprentissage. Un autre avantage vient du fait que l'étiquette de données nécessite l'intervention d'un utilisateur humain. Lorsque les jeux de données deviennent très grands, cette opération peut s'avérer fastidieuse. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, revêt un intérêt pratique évident et indiscutable (Zhu et Goldberg, 2009).

1.2 Classification de textes

La classification de textes est un domaine où les algorithmes sont appliqués sur des documents de texte. Cette tâche consiste à attribuer un document dans une ou plusieurs classes, en fonction de son contenu. En règle générale, ces classes sont triées sur le volet par les humains. Par exemple, considérons la tâche classifiant l'ensemble de documents comme bon ou mauvais. Dans ce cas, les catégories (ou étiquettes) « bon » et « mauvais » représentent les classes.
Certains applications populaires où la classification de textes est appliquée sont les suivantes (Chen *et al.*, 2014) :

- Classer les nouvelles comme Politique, Sports, Monde, Affaires, Style de vie.
- Classer les courriers électroniques comme Spam, Autre.

- Classer Les documents de recherche par type de conférence.
- Classer les critiques de films comme bons, mauvais et neutres.
- Classer les blagues comme drôles, pas drôles.

Pour qu'un classifieur apprenne à classer les documents, il faut une sorte d'apprentissage machine. À cet effet, les objets d'entrée sont divisés en données d'apprentissage et des données de test (essai). Les ensembles de données d'apprentissage sont ceux où les documents sont déjà étiquetés. Les ensembles des données d'essai sont ceux où les documents sont sans étiquettes. Le but est d'apprendre la connaissance de classes déjà marquées dans les données d'apprentissage et d'appliquer la connaissance tirée sur les données de test et de prédire l'étiquette de la classe d'essai avec précision. L'apprenant est responsable d'appliquer une fonction de classification (F) qui associe les documents (D) à la classe (C), comme suit :

$$F : D \rightarrow C \tag{1.1}$$

Le classificateur utilise ensuite cette fonction de classification pour classer l'ensemble des documents non marqués (Sanderson, 2010).

Le choix de la taille des données d'apprentissage et des tests est très important. Si le classifieur est alimenté par un petit nombre de documents afin de réaliser l'apprentissage, il ne peut pas acquérir des connaissances importantes pour classer les données de test correctement. Par ailleurs, si les données d'apprentissage sont trop importantes par rapport aux données de test, elle conduit à un problème appelé « Surapprentissage » (*Overfitting*)(Sriram *et al.*, 2010).

1.3 Représentation du texte

L'apprenant à besoin de comprendre le document par une fonction de classification. Cependant, la machine considère le texte comme des données non structurées. Pour l'apprenant, le document est simplement un texte brut. Par conséquent, il

est nécessaire de représenter le texte du document sous une forme structurée et formelle. La technique la plus courante pour représenter le texte est le modèle dit de Sac de mots (*Bag-of-word*). Dans cette technique, le texte est décomposé en mots. Chaque mot représente une caractéristique. Ce processus est également appelé *Tokenisation*, car le document est divisé en jetons qui sont des mots individuels. Notons que dans un tel modèle, l'ordre exact des mots est ignoré.

1.3.1 Réduction des vecteurs

La technique du Sac de mots place les mots dans un vecteur. Celui-ci devient trop grand. Il y a cependant plusieurs façons de le réduire :

- Élimination des mots vides (Stop Words) : cette étape consiste à enlever les mots qui n'ajoutent aucune valeur significative au document. Par exemple, des mots comme "a, an, the, if, for" peuvent être retirés à partir d'une liste (voir Section 6.9).
- Lemmatisation / racinisation : est une transformation des mots vers leur forme de racine ou de lemme. Les mots dans le texte existent sous une forme dérivée, représentée par cette racine ou lemme. La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son préfixe ou son suffixe, à savoir son radical. Contrairement, le lemme correspond à un mot réel de la langue. La racinisation (ou *stemming*) ne correspond généralement pas à un mot réel. Par exemple, « *ran* », « *running* », « *runs* » sont tout dérivés du mot « *run* ». Un algorithme couramment utilisé pour effectuer l'opération de racinisation pour la langue anglaise est dû à Porter (Porter, 1980). Dans la section 6.8 on utilise un outil de leminsation appelé *StanfordNLP*[1].

1. http ://nlp.stanford.edu/software/

D'autres techniques comprennent une représentation vectorielle à l'aide d'un modèle *TF-IDF*(Sanderson, 2010). *TF* fait référence à la fréquence de l'existence d'un mot dans un document, à savoir le nombre d'occurrences d'un mot particulier dans un document. Plus la fréquence du mot augment, plus le poids de la fonction *TF* augmente (Relation directe). Par exemple, si les documents ont pour sujet de discussion "la classification des tweets ", le terme «tweet» est très susceptible de se reproduire à plusieurs reprises. Par conséquent, pour réduire l'effet du mot "tweet", nous faisons usage de l'*IDF* (*Inverse Document Frequency*). La fréquence de document *DF* fait référence au nombre de documents de la collection qui contiennent un mot spécifique. Plus la valeur de *DF* augmente, plus on aura une réduction de l'importance de la fonction *IDF*. La fonction *IDF* est calculée comme suit :

$$IDF = \log \frac{N}{DF} \tag{1.2}$$

Ici, la variable N désigne le nombre total des documents dans le corpus.
Le score *TF-IDF*(*Term Frequency-Inverse Document Frequency*) pour une fonction est calculée comme suit :

$$TD\text{-}IDF = TF * IDF \tag{1.3}$$

1.4 Classification des textes courts

Les sections précédentes ont traité la classification des textes ou documents. Ces documents sont généralement grands et riches en contenu. Les techniques traditionnelles utilisant les sacs de mots fonctionnent bien avec ces données puisque l'occurrence de chaque mot est élevée et facile à extraire (le texte généralement est valide syntaxiquement et lexicalement). La fréquence des mots est cependant suffisante pour capturer la sémantique du document.

Contrairement au texte du document structuré, le texte des tweets n'est généralement pas valide syntaxiquement, contient des mots d'argot et est très court (ne

dépassant pas 140 caractères). Avec l'augmentation de la popularité des médias sociaux et la communication à travers le Web comme les messages de chat et les tweets, de l'information riche peut être extraite de la conversation concise entre les groupes de personnes. (Phan *et al.*, 2008; Hu *et al.*, 2009) ont présenté certains types de textes courts :

- Les messages SMS.
- Les légendes de l'image.
- Les messages du forum.
- Les descriptions des produits.
- Les avis des internautes sur divers produits.
- Les blogs et nouvelles (RSS) .
- Les messages Twitter.

1.5 Conclusion

Les techniques de classification traditionnelles s'adaptent aux textes longs. Cependant, lorsqu'il s'agit des textes courts, ces techniques traditionnelles ne fonctionneront pas aussi bien. Cela correspond à notre intuition, puisque ces techniques reposent sur la fréquence des mots seulement. Puisque le texte est trop court, ils n'offrent pas suffisamment de connaissances et de contexte sur le texte lui-même.

Ce chapitre a présenté les différentes techniques de classification pour des textes structurés. Dans le chapitre suivant, nous présentons l'état de l'art ainsi que les méthodes utilisées pour faire face aux problèmes des textes courts comme les tweets.

CHAPITRE II

ÉTAT DE L'ART

2.1 Introduction

Dans ce chapitre, nous présentons une revue de la littérature qui examine des différentes stratégies utilisées dans la classification de textes courts.

Nous évoluons dans un monde où l'information est centrale dans la mesure où l'ensemble de nos actions, interactions, personnels et professionnelles sont dépendants des informations à notre disposition. Accéder à une information pertinente, au bon moment, est un enjeu stratégique important pour en faire un bon usage. Ces dernières années, les blogues, les médias sociaux (Twitter, Facebook, LinkedIn, etc.) et autres flux tels que RSS se sont multipliés. Ces nouvelles formes de publication ont un grand potentiel en terme de vieille stratégie de publication. En effet, les professionnels de l'information peuvent les utiliser comme nouvelles ressources documentaires pour y rechercher de l'information pertinente (Rosoor *et al.*, 2011).

Certains travaux récents se sont intéressés à la classification des tweets et ont proposé des méthodes basées sur l'apprentissage automatique supervisé. Sriram *et al.* (2010) propose d'utiliser un petit ensemble de caractéristiques spécifiques au domaine, extraites du profil et du texte de l'auteur. L'approche proposée classe les tweets dans un ensemble prédéfini de classes génériques telles que les nouvelles,

les évènements, les avis, les offres et les messages privés portant des informations sur l'auteur et sur le domaine des caractéristiques spécifiques telles que la présence d'un raccourcissement des mots, des phrases temps-évènement, les opiniâtres mots, l'accent sur les mots, les symboles de monnaie et des pourcentages, et même l'existence de textes tels que "@UserName" présent au début du tweet ou dans le tweet lui-même.

La plupart des travaux lient la problématique de classification à l'élimination des problèmes de dispersion des données dans le modèle d'apprentissage (la taille des données extraites de textes). Une méthode intuitive pour ce faire est de gonfler le texte court avec des informations supplémentaires pour le faire apparaitre comme un grand document de texte. Par la suite, les algorithmes de classification ou de regroupement traditionnels peuvent être appliqués. Certains travaux (Bollegala *et al.*, 2007; Sahami et Heilman, 2006; Metzler *et al.*, 2007) se concentrent principalement sur l'intégration des messages de textes courts dans les moteurs de recherche comme *Google* et *Bing* pour extraire plus d'information liée au texte court. Pour chaque paire de textes courts, ils récupèrent des statistiques sur les résultats du moteur pour déterminer le score de similarité. Cependant, ces techniques nécessitent des approches d'homonymie supplémentaires pour traiter la polysémie.

Par exemple, «jaguar» et «voitures» sont très liés. Mais, lorsque la recherche du dictionnaire des synonymes ou une recherche sur le Web est effectuée, de nombreux résultats peuvent être liés à l'animal «jaguar» avec voiture. Par conséquent, il est nécessaire d'obtenir une rétroaction explicite de l'utilisateur afin de diriger le processus de recherche et l'inflation de texte.

Il est également impossible d'effectuer une recherche de similarité sémantique sur chaque paire de messages texte, car ça nécessite beaucoup de temps de traitement et ceci ne convient pas aux applications TALN en temps réel. Bien que ces tech-

niques identifient des termes prédominants entre les messages, il est nécessaire de calculer également des mots similaires qui sont très susceptibles de se produire dans le même contexte. L'avantage cependant d'utiliser la recherche Web par opposition à une recherche du dictionnaire des synonymes (exemple, utilisant *WordNet*) est que la méthode ne nécessite pas de taxonomie préexistante. Par conséquent, ces méthodes peuvent être appliquées dans de nombreuses tâches qui ne disposent pas de telle taxonomie (catégorisation des mots) ou ne sont pas mises à jour.

Un autre travail qui se base sur des ressources externes afin d'étende et d'élargir le contenu a été réalisé par Genc *et al.* (2011). Les auteurs ont proposé une technique de classification basée sur la ressource Wikipédia, afin de classer les tweets par un message de cartographie dans leurs pages Wikipédia les plus similaires. Ainsi, les messages sont mappés aux leurs pages Wikipédia les plus semblables, ensuite les distances sémantiques entre les messages sont calculées. Ces mesures sont basées sur les distances entre leurs pages Wikipédia les plus proches.

Il existe également des travaux qui se basent sur des ressources internes telles que les hyperliens, afin d'étendre et d'élargir les contenus ou de regrouper les utilisateurs.

Kinsella *et al.* (2011) ont utilisé la nature informelle des conversations pour donner un contexte à une conversation dans les textes courts et le recours fréquent des utilisateurs à des hyperliens externes pour comprendre plus le message. Leur stratégie consiste à examiner l'utilité de ces liens externes pour déterminer le sujet d'un individu. Dans leurs travaux, des hyperliens vers des objets qui relient les métadonnées disponibles sur le Web, y compris les produits *Amazon* et des vidéos *YouTube*, ont été utilisés.

Alors que tous ces travaux utilisent les caractéristiques des textes tweet ou méta-

informations provenant d'autres sources d'information, le travail de Lee *et al.* (2011) classe et regroupe les Sujets de Twitter en dix-huit catégories générales telles que le sport, la politique, la technologie, etc. Deux approches pour les sujets de la classification ont été utilisées : (i) l'approche traditionnelle du «sac des mots» pour la classification de texte (ii) et la classification fondée sur les réseaux. Dans cette dernière, ils ont identifié cinq grands sujets similaires pour une catégorie donnée sur la base du nombre d'utilisateurs influents communs.
Les catégories des sujets similaires et le nombre des utilisateurs influents communs entre le sujet donné et ses sujets similaires sont utilisés pour classer les catégories des données.
Sankaranarayanan *et al.* (2009) ont construit un système de traitement de nouvelles, appelé TwitterStand, qui identifie les tweets correspondant à la fin des dernières nouvelles. L'objectif de leur travail consiste à supprimer le bruit, de déterminer les groupes et les classes de tweet d'intérêt en utilisant des méthodes en ligne, et d'identifier les endroits pertinents associés aux tweets.

Contrairement à leurs travaux, nous ne nous basons pas sur un nombre des catégories spécifiques. Aussi, des méthodes basées sur la classification non supervisée existent, telles que celles proposées par Becker *et al.* (2011) qui distinguent et séparent les messages du tweet entre celles liés aux évènements du monde réel et celles liées aux non-évènements. Les auteurs ont utilisé une technique de clustering en ligne pour regrouper les tweets dans des sujets similaires et calculent les caractéristiques qui peuvent être utilisées pour apprendre à un classifieur ce qui distinguerait les classes évènements de celles du non-évènements.
Saif *et al.* (2012) ont introduit une approche qui se base sur la désambigüisation des entités nommées dans la phase d'apprentissage pour l'analyse des sentiments. Pour chaque entité extraite (par exemple le mot « *IPhone* ») à partir de tweets, ils ont ajouté son concept sémantique (par exemple, « produit Apple ») comme

caractéristique supplémentaire. Ils ont ensuite mesuré la corrélation du concept représentant avec le sentiment négatif ou positif.

Une autre forme d'exploitation des entités nommées avec la désambigüisation à l'aide des ressources externes est celle proposé par Michelson et Macskassy (2010). Leur approche exploite une base de connaissances de Wikipédia pour désambigüiser et classer les entités dans les tweets. Ils ont développé un «profil de sujet», qui caractérise les sujets d'intérêt des utilisateurs, et ont distingué les catégories qui apparaissent fréquemment et couvrent les entités nommées.

Une autre approche est basée sur l'exploitation des hashtags, reliant ainsi les textes ayant un sujet commun. Les hashtags sont des métadonnées, des annotations libres définies par les utilisateurs qui servent à marquer l'appartenance d'un message à un domaine particulier, et ainsi construire un canal implicite de communication. Wang *et al.* (2011) ont résumé les familles de hashtags en trois familles qui regroupent leurs sujets (sujet, sentiment, sentiment sujet) dans l'analyse.

Comme les hashtags sont des éléments essentiels dans les tweets et lient le sujet qu'il peut discuter, la plupart des systèmes d'analyse d'opinions cherchent à les incorporer dans leurs calculs.

Asur et Huberman (2010) montrent, comment l'on peut améliorer les techniques standards de classification supervisée en intégrant la polarité des hashtags les plus fréquents comme paramètre. Ces polarités ont été assignées manuellement.

Kouloumpis *et al.* (2011) ont employé une méthode similaire, mais ont rajouté les émoticons dans la détection de la polarité des tweets.

Cependant, la définition manuelle de la polarité des hashtags n'est pas très efficace et peut se révéler plutôt comme une opération coûteuse. Dave et Varma (2012) n'utilisent pas la polarité des hashtags qu'à la condition qu'ils appartiennent à une liste de mots connus à l'avance tels que : #efficace, #nul, #incapable, #visionnaire, etc.

Tous ces travaux ont prouvé à quel point l'utilisation des hashtags peut se révéler précieuse dans l'évaluation des messages dont la taille s'avère souvent trop courte pour que les méthodes traditionnelles fonctionnent de manière optimale. Les hashtags 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 des noms propres, des noms de lieux ou des phrases complètes sans souvent le moindre indice sur leur construction interne.
L'utilisation des hashtags, comme mots composés, et la décomposition de ces derniers a aidé à l'amélioration de la détection de la polarité des tweets (Brun et Roux, 2014). Liu (2010) a comparé l'intégration de la décomposition des hashtags et son effet sur un système de détection d'opinion. Cette comparaison est faite par rapport à un système basé sur un sac des mots.

2.2 Conclusion

Dans ce document nous avons présenté une vue générale sur la littérature qui concerne la classification des messages courts comme les tweets.
Dans le prochain chapitre, nous proposons notre méthode générale sur la classification des tweets. Le travail se base sur la catégorisation des tweets par la décomposition des hashtags avec la détection des entités nommées comme ressources internes. Ensuite, WordNet est utilisée comme ressource externe, afin de désambigüiser les mots et les entités nommées. On se base sur une représentation vectorielle d'un corpus de tweets classés en thèmes (économie, politique, sport, etc.). Chaque thème est représenté sous la forme d'un vecteur de mots. Chaque nouveau tweet est classé avec les autres vecteurs pour identifier le thème le plus proche. Une interface graphique a été conçue. Des expérimentations sur des jeux de données réelles soulignent la pertinence de notre proposition et ouvrent de nombreuses perspectives.
Dans le chapitre qui suit, nous discutons plus en détail les caractéristiques du

réseau social Twitter ainsi que sa structure générale.

CHAPITRE III

VUE D'ENSEMBLE DE TWITTER

3.1 Introduction

Twitter est un outil de microblogage géré par l'entreprise Twitter Inc. Il permet à un utilisateur d'envoyer gratuitement de brefs messages, appelés tweets, sur Internet, par messagerie instantanée ou par SMS. Et comme la taille d'un SMS ne dépasse pas 160 caractères, Twitter a limité la taille d'un tweet à 140 caractères dont 20 caractères réservés au nom de l'expéditeur (Gabielkov, 2016).

Selon les statistiques d'août 2016 [1], Twitter [2] a plus de 600 millions utilisateurs inscrits et reçoit plus de 500 millions de tweets par jour . La simple utilisation quotidienne de Twitter et la publication sur ce site ont fait de lui un moyen de communication de taille mondiale. Twitter est très important pour les gens de tous les horizons de la vie et de toutes les nationalités avec toutes les langues de la planète (Mendoza *et al.*, 2010).Twitter a joué et continue de jouer un rôle de premier plan dans les évènements sociopolitiques tels que le printemps arabe (Morstatter *et al.*, 2014) et le mouvement Occuppy Wall Street (Qu *et al.*, 2011) Twitter a également été utilisé pour recueillir les informations nécessaires pour

---

1. http ://www.statisticbrain.com/twitter-statistics/

2. https ://about.twitter.com/company

une bonne préparation de la population lors des grandes catastrophes naturelles, comme les tsunamis et les ouragans.

Twitter fournit une API gratuite pour différents objectifs et pour recueillir les données Twitter. La figure 3.1 montre une capture d'écran de la page d'accueil Twitter.

Figure 3.1 Capture d'écran de la page d'accueil de Twitter

3.2 L'architecture de Twitter

L'API Twitter (Application Programming Interface) est basée sur le service REST (Representational State Transfer) (Fielding, 2000).
L'architecture du type REST se compose de clients et de serveurs. Les clients lancent des demandes aux serveurs ; les serveurs traitent les demandes et renvoient des réponses appropriées.
Les demandes et les réponses sont construites autour du transfert de « représentations » des « ressources ». Une ressource peut être essentiellement tout concept cohérent et significatif qui peut être pris en compte. Une représentation d'une ressource est typiquement un document qui capture l'état actuel ou prévu d'une ressource. À un moment, un client peut-être soit en transition entre les états de

l'application ou «au repos». Un client dans un état de repos est capable d'interagir avec son utilisateur, mais ne crée pas de charge et ne consomme pas d'espace de stockage sur le serveur ni sur le réseau. Un concept important dans REST est l'existence de ressources (sources d'informations spécifiques), dont chacune est référencée avec un identifiant global (exemple, un URI dans HTTP). Dans le but de manipuler ces ressources, les composants du réseau (les agents utilisateurs et les serveurs) communiquent via une interface normalisée (par exemple, HTTP) et s'échangent des représentations de ces ressources (les documents réels de transport des informations).

L'API Twitter se compose de trois parties : deux API REST et une API en *streaming*[3]. Les méthodes de l'APIs REST Twitter permettent aux développeurs d'accéder aux données de base, en permettant les opérations de mise à jour, les données d'état et les informations utilisateurs. Les méthodes de recherche de l'API permettent aux développeurs d'interagir avec la recherche sur Twitter et suivent l'évolution des données. L'API en *streaming* fournit en temps quasi réel l'accès à un grand volume d'utilisateurs des Tweets sous forme échantillonnée filtrée.

3.3 Les concepts de Twitter

Différents concepts sont définis dans Twitter :

- Utilisateur

  un nom précédé d'arobase « @ » et est un lien direct vers un compte Twitter. L'utilisateur de ce nom il a la permission de voir tous ses tweets, sauf s'ils sont protégés. Chaque utilisateur peut consulter les mentions qu'il a reçues dans l'onglet « @ Connect ». Si un tweet débute par une mention, seuls les suiveurs du compte mentionné verront le tweet dans leur fil d'actualité (par exemple «@A » rédige un tweet en commençant par l'adresse de «@B

3. https ://dev.twitter.com/overview/documentation

», donc parmi les suiveurs de @A, seuls ceux qui suivent également «@B» liront le tweet depuis leur fil d'actualité)(Gabielkov, 2016). Les informations suivantes sont stockées pour chaque utilisateur :

1. La langue du tweet.
2. Le fuseau horaire de l'emplacement.
3. L'emplacement du Tweet (l'emplacement à partir duquel le tweet a été envoyé).
4. La photo du profil.
5. L'emplacement de l'utilisation.
6. La page web.
7. Une brève biographie.
8. Les liens favoris.

- Tweet

  un tweet est un message court, limité à 140 caractères. Cette restriction impose aux utilisateurs d'être concis dans ce qu'ils ont à dire. Ceci est également la raison pour laquelle les utilisateurs ont tendance à utiliser les abréviations (par exemple : «fr»-for, «cud» – could). Chose intéressante, est qu'il y a un ensemble riche et bien compris d'abréviations qui est étonnamment cohérent à travers les groupes d'utilisateurs, et même à travers d'autres supports électroniques tels que les SMS et les forumes de discussions (Sankaranarayanan *et al.*, 2009). Comme les utilisateurs veulent transmettre tout ce qu'ils ont à dire en 140 caractères, ils pourraient faire des erreurs d'orthographe et des tweets peuvent être sujet à des erreurs syntaxiques. Cela rend difficile le travail avec Twitter. La plupart du temps, les utilisateurs fournissent des liens vers des ressources externes quand ils ne peuvent pas transmettre l'information complète dans les 140 caractères. Ces liens URL vers des fichiers texte, audio ou vidéo sont appelés *artéfacts.*

3.4 Caractéristiques spéciales des tweets

- Référence à un autre utilisateur

  Pour faire référence à un autre utilisateur dans un tweet, le symbole "@" précède, un nom de cet utilisateur. Ce nom ne doit pas contenir des espaces. Lorsqu'un utilisateur se réfère à un autre utilisateur au début d'un tweet, le tweet devient un « message direct» (DM). Les messages directs ce sont des tweets publics conçus comme une correspondance entre deux utilisateurs du système. Twitter fournit une disposition pour afficher les messages directs destinés à l'utilisateur. Cela garantit que ceux-ci, qui ont généralement une plus grande priorité à l'utilisateur prévu, ne se perdent pas le flux écrasant d'autres tweets dans l'espace utilisateur. Lorsque la référence à un autre utilisateur ne se produit pas au début du tweet mais au milieu ou à la fin, il ne se qualifie pas pour un message direct, mais sert simplement comme un point de référence (Sriram *et al.*, 2010).

- Re-tweets

  Si un tweet est convaincant et assez intéressant, les utilisateurs peuvent le republier. Il devient ce qu'on appelle «re-tweeting». Un retweet est similaire au renvoi par courriel. Lorsqu'un utilisateur envoie un re-tweet il est considéré comme ayant approuvé ce contenu et partage son contenu avec ses partisans(Sriram *et al.*, 2010).

- Les hashtag

  Un « hashtag » commence toujours par le caractère « # » ; ce qui permet de le repérer très rapidement dans l'analyse des données (lors de la phase de tokenisation). Ces «hashtags» créent des problèmes durant l'analyse linguistique du texte. En effet, ils sont considérés comme des mots inconnus qui ne se trouvent pas dans les dictionnaires, car ils sont généralement des mots composés inventés par les utilisateurs de Twitter et leur sémantique

particulières se perdent dans le traitement des textes(Brun et Roux, 2014). Twitter permet aux utilisateurs d'étiqueter leurs tweets en utilisant les balises de « hashtag ». Ces balises sont de la forme "# <nom de tag>". Elles sont transmises comme mots-clés qui représentent le mieux le contenu du tweet. Les « hashtag » aident Twitter à regrouper ensemble les tweets similaires qui ont les mêmes balises de « hashtag ». Cela rend la recherche sur Twitter plus facile et plus rapide. Ainsi, les utilisateurs peuvent suivre un sujet d'intérêt particulier. La plupart des outils de recherche Twitter (Sankaranarayanan *et al.*, 2009) utilisent les hashtags pour améliorer la qualité de la recherche. Notons que le hashtag s'ajoute au nombre de caractères du tweet.

CHAPITRE IV

LA RECONNAISSANCE DES ENTITÉS NOMMÉES (REN)

Une entité nommée est une séquence de mots qui désignent une entité du monde réel. Des exemples d'entités sont : « Canada », « P. Elliott Trudeau » ou « Bell ». La tâche de reconnaissance des entités nommées, souvent abrégées REN (ou *NER* en Anglais, pour *Named Entity Recognition*), est l'identification d'abord des entités nommées du texte libre, ensuite leur classification dans un ensemble de types prédéfinis tels que personne, organisation ou lieu. Mais le plus souvent, cette tâche ne peut simplement pas être accomplie par correspondance des chaines dans des dictionnaires de lexique externe précompilés. Parce que les entités nommées d'un type d'entité ne forment pas un ensemble fermé et donc tout dictionnaire de lexique externe serait incomplet. Une autre raison est que le type d'entité nommée peut-être dépendant du contexte. Par exemple, « P. Elliott Trudeau » peut se référer à une personne ou un emplacement « l'aéroport international P. Elliott Trudeau » ou tout autre partage de même nom d'entité.

Par conséquent lorsqu'on détermine le type d'entité pour « P. Elliott Trudeau » apparaissant dans un document particulier, son contexte doit être considéré. La reconnaissance de l'entité nommée est probablement la tâche la plus fondamentale dans l'extraction de l'information. L'extraction des structures plus complexes telles que les relations et les évènements dépendent de l'identification précise de

l'entité désignée comme une étape de prétraitement.

Les types d'entités nommées les plus couramment étudiées sont personne, organisation et lieu , et qui ont définis par MUC-6 [1]. Ces types sont assez généraux pour être utiles dans des nombreux domaines d'application.

L'extraction des expressions de dates, les heures, les valeurs et les pourcentages monétaires, qui ont également été introduites par MUC-6, sont aussi étudiées sous NER [2], bien que strictement parlant ces expressions ne sont pas des entités nommées. Outre ces types d'entités générales, d'autres types d'entités sont généralement définis pour les domaines spécifiques. Par exemple, le corpus GENIA utilise une ontologie à grains fins pour classer les entités biologiques (Ohta *et al.*, 2002).

Les premières solutions à la reconnaissance d'entités nommées reposent sur des modèles fabriqués manuellement (Hobbs *et al.*, 1997). Ces modèles nécessitent une expertise humaine et un travail intensif. Les systèmes ultérieurs essaient d'apprendre automatiquement ces modèles à partir de données étiquetées. Des travaux plus récents sur la reconnaissance des entités nommées utilisent des méthodes d'apprentissage automatique statistique. Une première tentative est Nymble, un nom localisateur basé sur les modèles des chaines de Markov cachée (Bikel *et al.*, 1997). D'autres modèles d'apprentissage tels que les modèles de Markov d'entropie maximale (Rosenberg *et al.*, 2012), les machines à vecteurs de support (Apostolova et Tomuro, 2014) et les champs aléatoires conditionnels (Liu *et al.*, 2014) ont également été appliqués à la reconnaissance des entités nommées.

---

1. MUC-6 : the sixth in a series of Message Understanding Conferences

2. NER : Named-entity recognition

4.1 Les approches d'extraction des entités nommées

On distingue deux approches principales :

1. Approche à base de règles

   Les méthodes à base de règles pour la reconnaissance des entités nommées sont liées aux étapes suivantes :

   - Un ensemble de règles est soit défini manuellement ou appris automatiquement.
   - Chaque jeton dans le texte est représenté par un ensemble de fonctionnalités.
   - Le texte est ensuite comparé aux règles et une règle est déclenchée si une correspondance est trouvée.
   - Une règle est constituée d'un motif et une action. Un modèle est généralement une expression régulière définie sur les caractéristiques des jetons. Lorsque ce motif correspond à une séquence de jetons, l'action spécifiée est déclenchée. Une action peut étiqueter une séquence de jetons comme une entité, en insérant l'étiquette de début ou la fin d'une entité, ou l'identification de plusieurs entités simultanément. Par exemple, pour marquer toute séquence de jetons de la forme « Mr. X » où X est un mot capitalisé comme une entité de personne, la règle suivante peut être définie :

$$(token = \text{"Mr."}\ \ orthography\ type = FirstCap) \rightarrow person\ name) \tag{4.1}$$

   Le côté gauche de la règle ci-dessus représente une expression régulière qui correspond à toute séquence de deux jetons où le premier jeton est «Monsieur» et le deuxième jeton à la FirstCap de type orthographe. Le côté droit

indique que la séquence de jeton adapté doit être étiquetée comme un nom de personne.

Ce genre de méthodes basées sur des règles ont été largement utilisées (Soderland, 1999; Sarawagi, 2008). Communément utilisée pour représenter les caractéristiques de jetons comprenant le jeton lui-même, la balise de la catégorie grammaticale de mot, le type orthographe du jeton (par exemple, la première lettre en majuscules, toutes les lettres en majuscules, nombre, etc.), ou si le jeton est à l'intérieur d'un dictionnaire de lexique externe prédéfini.

Il est possible qu'une séquence de jetons correspondent à plusieurs règles, c'est-à-dire une entité qui est capable d'avoir deux étiquètes différentes. Par exemple l'entité « P. Elliott Trudeau » peut avoir une étiquète d'une personne ou bien une étiquète d'une organisation ou d'un lien en l'occurrence l'aéroport de Montréal « P. Elliott Trudeau ». Pour gérer ce genre de conflits, un ensemble de politiques doit être défini et respecté afin de contrôler la façon dont les règles doivent être tirées. L'approche consiste à ordonner les règles à l'avance afin qu'elles soient séquentiellement vérifiées où on donne des priorités pour chaque règle à exécuter (Aggarwal et Zhai, 2012).

2. Approche statistique

   L'approche de l'apprentissage statistique a pour principe de base la mise au point automatique des modèles d'analyse à partir de volumes importants de données. Ces méthodes sont dites statistiques (ou à base d'apprentissage), car elles apprennent des modèles d'analyse de textes à partir des corpus annotés. Ces modèles d'analyse peuvent prendre différentes formes telles que les arbres de décision, les ensembles de règles logiques, les modèles probabilistes ou encore les chaines de Markov cachées.

   Au regard de la reconnaissance d'entités nommées, un système *observant* plusieurs fois la présence de l'abréviation *Mme* devant un mot annoté comme

nom de personne dans le corpus d'apprentissage pourra facilement en déduire un modèle d'analyse. Ces systèmes à base d'apprentissage se sont considérablement multipliés (Ehrmann, 2008; Aggarwal et Zhai, 2012).

4.2 Comparaison entre les approches

Les avantages et les inconvénients respectifs de ces deux types d'approches sont connus. Entre autres, l'indispensable disponibilité de corpus annotés pour les premiers et le temps de développement leurs coûts de développement pour le deuxièmes.

L'annotation de corpus peut être toute aussi longue même si cela peut se faire par des personnes moins expertes. Hormis ces querelles de conception, l'intérêt se situe véritablement dans ce que chaque type de système est capable de faire et comment il peut fonctionner. Si un concepteur de règles ne peut, bien sûr, pas penser à toutes les exceptions, il peut en revanche prévoir des patrons plus ou moins complexes pour le captage d'éléments difficiles, ce qu'un système probabiliste ne peut pas faire. La précision est d'ordinaire plus importante dans les systèmes symboliques tandis que les systèmes à base d'apprentissage présentent l'avantage d'être plus flexibles quant à leur adaptation à une tache similaire, mais portant sur un autre domaine et d'être plus robuste sur des corpus difficiles (ou bruités). Cette partition entre avantages et inconvénients de telle ou telle approche se reproduit pour les systèmes de reconnaissance des entités nommées (Ehrmann, 2008).
Dans la section 6.7 nous avons choisi d'utiliser l'outil StanfordNER [3] construit par une approche de l'apprentissage statistique.

---

3. http ://nlp.stanford.edu/software/

4.3 Conclusion

Dans cette section, nous avons présenté des détails sur la reconnaissance des entités nommées (REN). Dans le chapitre suivant, nous abordons le sujet de l'enrichissement sémantique avec WordNet et plus précisément ses relations sémantiques.

CHAPITRE V

ENRICHISSEMENT SÉMANTIQUE AVEC WORDNET

5.1 Introduction

WordNet (Miller *et al.*, 1990) est un thésaurus créé principalement pour la langue anglaise basé sur des études de psycholinguistique. Cet outil a été développé à l'Université de Princeton. Il a été conçu comme une ressource de traitement de données qui couvre les catégories lexico-sémantique appelées « synsets ». Les synsets sont des ensembles de synonymes qui regroupent des éléments lexicaux ayant une similaire signification. Par exemple, les mots « *a board* » (un panneau) et « *a plank* » (une planche) regroupés dans l'ensemble de synsets {« *board* », « *plank* »}. Mais «*a board*» peut également indiquer un groupe de personnes (par exemple, un conseil d'administration). Pour désambigüiser ces homonymes de significations « *a board*» fera également partie du synset {« *board* », « *committee* »}.

La définition du synset varie de très spécifique à très générale. Les synsets les plus spécifiques réunissent un nombre limité de significations lexicales, alors que les synsets les plus généraux couvrent un très large nombre de significations.
L'organisation de WordNet à travers des significations lexicales au lieu d'utiliser des unités lexicales le rend différent des dictionnaires traditionnels. L'autre différence que présente WordNet par rapport aux dictionnaires traditionnels s'explique par la séparation des données en quatre catégories associées aux catégories gram-

maticales des mots : verbes, noms, adjectifs et adverbes. Ce choix d'organisation est motivé par des recherches psycholinguistiques sur l'association de mots aux catégories syntaxiques par des sujets humains. Chaque catégorie est organisée différemment des autres. Les noms sont organisés en hiérarchie, les verbes par des relations, les adjectifs et les adverbes par des hyperespaces N-dimension (Miller *et al.*, 1990).

Les avantages de l'utilisation de WordNet dans les travaux de l'analyse de texte permet de regrouper les mots ayant les mêmes sens pour faire face à la richesse morphologique des langues naturelles et de réorganiser les mots selon des relations hiérarchiques.

5.2 Les différentes relations sémantiques

La liste suivante énumère les relations sémantiques disponibles dans WordNet. Ces relations se rapportent aux concepts, mais les exemples que nous donnons sont basés sur les mots.

1. Synonymie : une liaison de deux concepts équivalents ou des relations étroite concepts (*frail /fragile*). Cette relation est symétrique.
2. Antonymie : une relation de liaison sur deux concepts opposés (exemple : *small / large*). Cette relation est symétrique.
3. Hyperonymie : une relation liant un concept à un concept plus général (exemple : *tulip /flower*).
4. Hyponymie : une relation liant un concept à un concept plus spécifique. Il est l'inverse d'hyperonymie. Cette relation peut être utile dans la récupération de l'information. En effet, si tous les textes traitant des véhicules sont recherchés, il peut être intéressant de trouver ceux sur les voitures ou motos.
5. Méronymie : une relation liant un concept-1 à un Concept-2 dont il est l'une

des parties. C'est le en face de la relation méronymie (exemple : *roue/voiture*).

5.2.1 Hyponymes / hyperonymes dans WordNet

X est un hyponyme de Y (et Y est un hyperonyme de X) si :

$$\begin{cases} \text{F(X) l'expression minimale compatible avec la phrase "A est F(X)"} \\ \textit{et} \\ \text{A est } F(X) \rightarrow A \textit{ est } F(Y) \end{cases} \tag{5.1}$$

En d'autres termes, l'hyponymie est la relation entre un terme spécifique et un terme générique exprimé par l'expression «est-un».

Exemple :

$$\begin{cases} X = cat \\ Y = animal \\ It \textit{ is a } cat \rightarrow It \textit{ is an } animal \end{cases} \tag{5.2}$$

Dans l'exemple, si nous remplaçons «*cat*» par l' hyperonyme «*animal*» la phrase reste valide (c'est-à-dire qu'elle conserve sa sémantique) selon la définition précédente. Par exemple,

Un chat est un hyponyme de l'animal et l'animal est un hyperonyme de chat.

Dans WordNet, l'hyponymie est une relation lexicale entre le sens des mots et plus précisément entre synsets qui sont des ensembles des synonymes. Cette relation est définie par :

X est un hyponyme de Y si «X représente une espèce de Y» est vrai.

Nous pouvons remarquer que l'hyponymie est une relation transitive et asymétrique, qui génère une hiérarchie descendante dans les thésaurus pour l'organisation des noms et des verbes.

L'hyponymie est représentée dans WordNet par le symbole '@', qui est interprété par «est-un» ou «est une sorte de»(Elberrichi *et al.*, 2008).

Par exemple :

$$\textit{It is a tree} \rightarrow \textit{It is a plant.} \tag{5.3}$$

5.3 Conclusion

Dans la section 6.11 du chapitre suivant nous avons choisi d'utiliser le WordNet dans notre travail afin d'améliorer le résultat et de faire une comparaison tout en utilisant ses relations sémantique. Dans le chapitre suivant, nous allons aborder le sujet de l'apprentissage machine pour les textes courts.

CHAPITRE VI

MÉTHODOLOGIE DE CLASSIFICATION DES TWEETS

Notre projet de recherche concerne la classification de textes hautement bruités et non structurés, extraits de Twitter. On peut constater que lors de la récupération de l'information sur le web, un des principaux risques qui peut surgir est le fait que cette information ne soit pas toujours fiable ou encore qu'elle soit écrite d'une manière incompréhensible. Ainsi, pour améliorer la performance de l'analyse de ces textes bruités et puisés depuis l'internet, un prétraitement et un nettoyage de ces textes s'avèrent indispensables.

La méthodologie proposée pour la classification des tweets se base sur des outils de la plateforme WEKA [1] tout en utilisant l'API Twitter [2] pour récupérer les tweets. Le processus de classification est illustré dans la figure 6.1. En effet, Ce processus est décomposé en trois tâches importantes.

---

1. http ://www.cs.waikato.ac.nz/ml/weka

2. https ://dev.twitter.com

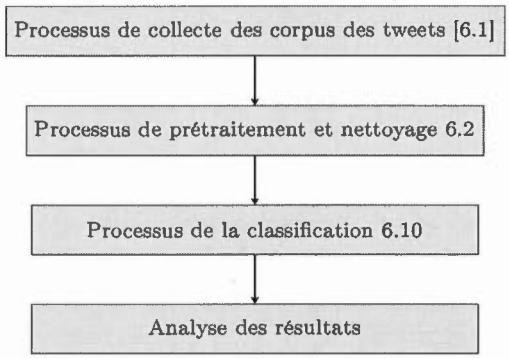


Figure 6.1 Le processus général de la méthodologie suivie dans la classification des tweets.

6.1 Corpus et outils utilisés

6.1.1 Acquisition du corpus des tweets

Cette étape nous permet d'obtenir les données d'apprentissage pertinentes à notre système. Le corpus est constitué d'un ensemble de tweets homogènes sur le fond (même thème global) et la forme (même format des tweets) en se basant sur l'API Twitter [3].

De plus, un nettoyage des données sera nécessairement exécuté en utilisant d'abord un détecteur de langue (Shuyo, 2010). Par contre, si les tweets sont écrits dans une langue autre que l'anglais, ils vont être automatiquement rejetés par le système. En effet, les tweets écrits dans différentes langues vont tout simplement créer un bruit et les mots de notre corpus vont être isolés (les tweets ne seront pas répérés, car WordNet [4] ne détecte pas les mots dans une autre langue que l'anglais). L'existence des mots dans une autre langue peut engendrer des résultats erronés.

---

3. https ://dev.twitter.com

4. http ://projects.csail.mit.edu/jwi/

La figure 6.2 montre le processus adopté pour collecter les tweets. On se base dans ce cas sur la détection de l'anglais par le pourcentage dans le texte donné a l'aide de l'outil language-detection [5]. Si le tweet contient les mots anglais moins de 80%, on le rejette, car ce tweet est bruité. Les tweets qui sont vides ou contiennent seulement des URLs seront supprimés du corpus pour qu'ils ne participent pas dans la construction du modèle de la classification.

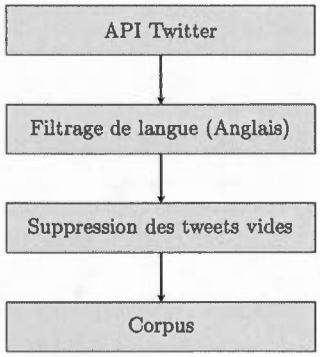


Figure 6.2 Processus de collection des tweets.

6.1.2 Préparation du corpus d'apprentissage

Pour effectuer des expérimentations et évaluations nécessaires (chapitre 7), nous commençons par collecter un corpus composé de plusieurs tweets. Ce corpus a été téléchargé par l'API Twitter en utilisant des requêtes en anglais qui contiennent des mots et des hashtag reliés à un domaine bien défini. La collection regroupe quatre domaines qui sont comme suit : sport, politique, économie et le domaine médical. Par exemple pour le domaine des sports nous avons les mots tels que football, basketball et soccer.

Le tableau 6.1 montre les statistiques sur le nombre des tweets, le nombre de termes et le nombre des lemmes à l'intérieur du corpus construit.

---

5. https ://github.com/shuyo/language-detection

Tableau 6.1 Statistiques sur le corpus des tweets

Économie	Nombre de termes simples	13 870
	Nombre de lemmes	7 938
	Nombre de tweets	2 504
domaine Médical	Nombre de termes simples	14 784
	Nombre de lemmes	12 138
	Nombre de tweets	2 415
Sport	Nombre de termes simples	16 112
	Nombre de lemmes	12 773
	Nombre de tweets	2 493
Politique	Nombre de termes simples	15 346
	Nombre de lemmes	11 976
	Nombre de tweets	2 497

6.2 Prétraitement des tweets

Malgré l'existence de plusieurs outils de traitement du langage naturel disponibles qui analysent les textes courts et les tweets, tels que par exemple TweetNLP [6], nous avons choisi StanfordNLP [7] comme outil de prétraitement des tweets qui se base sur des modèles [8] pour l'étiquetage grammatical construit par l'outil Gate [9] (un outil de langage naturel qui entraine des modèles statistiques capables d'analyser

6. http ://www.cs.cmu.edu/ ark/TweetNLP/

7. http ://nlp.stanford.edu/software/

8. https ://gate.ac.uk/wiki/twitter-postagger.html

9. https ://gate.ac.uk/

un texte à partir de son contenu syntaxique et morphologique).

Dans le but de construire notre outil, nous nous basons sur une architecture simple. L'architecture du prétraitement du tweet utilise un pipeline pour réduire le temps de prétraitement. Par conséquent, chaque tâche s'exécute dans une phase séparée. Nous utilisons l'outil StanfordNLP dans l'étape de prétraitement, comme l'illustre la figure 6.3.

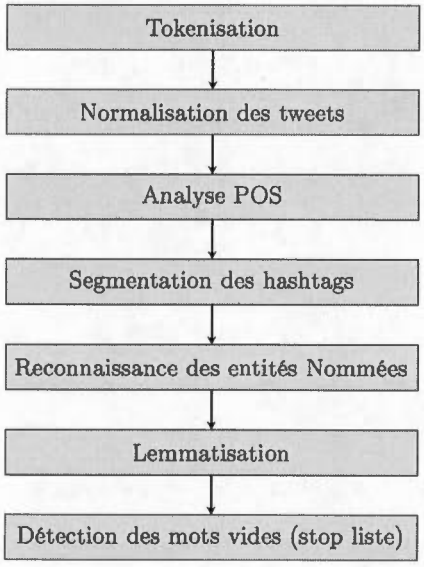


Figure 6.3 Processus de prétraitement des tweets.

6.3 Tokenisation

Dans l'analyse lexicale, la tokenisation est le processus de séparation d'un flux de texte en mots, phrases, symboles et d'autres éléments significatifs appelés jetons ou tokens. Pour effectuer cette tâche, nous avons utilisé l'outil StanfordNLP.

6.4 Normalisation lexical des tweets

La tâche de la normalisation consiste à réécrire le texte dans la langue standard ou proche de la langue standard. Notre but n'est pas de faire la correction orthographique et syntaxique, mais de réécrire le texte en se basant sur les erreurs lexicales fréquentes dans les médias sociaux. Pour rendre la tâche de normalisation de texte réalisable, nous avons découpé la tâche de la normalisation lexicale des messages anglais en sous tâches, comme suit :

1. Élimination des caractères en doublons, par exemple, le mot « *gooood* » est transformé en « *good* ». Pour résoudre ce genre de problème, nous avons utilisé un dictionnaire anglais [10] pour détecter les mots les plus proches au mot écrit. Pour ce faire, nous avons utilisé les expressions régulières endonnant la priorité aux caractères qui apparaissent en double.
2. La correction orthographique pour les erreurs fréquentes dans le web. Par exemple, quand on fait une recherche dans le dictionnaire anglais pour le mot «*scoll*», cela ne donne aucun résultat, car, le mot n'existe pas en anglais. Par conséquent, on le remplace systématiquement par le mot « *scroll*» car c'est le mot qui lui est le plus proche syntaxiquement.
3. La correction des erreurs fréquentes dans les médias sociaux, par exemple, on remplace le mot «*2day*» par« *today*» utilisant un dictionnaire [11] contenant les abréviations utilisées fréquemment dans les SMSs(Han *et al.*, 2013a).

6.5 Analyse grammaticale

L'étiquetage grammatical (*part-of-speech tagging* en anglais abrégé par *POS*) est un processus qui associe aux mots d'un texte les informations grammaticales

---

10. http ://gdt.oqlf.gouv.qc.ca/

11. https ://github.com/coastalcph/cs_sst/blob/master/data/res/emnlp_dict.txt

comme la partie du discours, le genre, le nombre, etc. à l'aide d'un outil informatique.

Les étiqueteurs grammaticaux qui analysent les textes courts et les tweets sont nombreux. On cite TweetNLP [12], TreeTagger [13].

Nous avons choisir StanfordNLP [14] comme outil de prétraitement des tweets qui peut utiliser des modèles [15] pour l'étiquetage grammatical construits par l'outil Gate [16].

6.6 Décomposition des hashtags

Un hashtag commence toujours par le caractère « # » ; ce qui permet de le repérer très rapidement dans l'analyse des données (lors de la tokenisation). Ces hashtags créent des problèmes durant l'analyse linguistique. En effet, ils sont considérés comme des mots inconnus et ne se trouvent pas dans les dictionnaires, car les hashtags sont généralement des mots composés inventés par les utilisateurs de Twitter et leur sémantique particulière se perd dans le traitement des textes. Or, dans un tweet dont la longueur ne peut dépasser 140 caractères, ignorer les hashtags peut conduire à une dégradation très forte de l'interprétation de celui-ci (Brun et Roux, 2014).

Généralement, l'utilisateur qui publie un tweet à propos d'un sujet donné, a tendance à mettre le plus possible d'hashtags qui sont en relation avec le sujet même. Cela, permet aux autres utilisateurs de trouver plus facilement le tweet publié

---

12. http ://www.cs.cmu.edu/ ark/TweetNLP/
13. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/
14. http ://nlp.stanford.edu/software/
15. https ://gate.ac.uk/wiki/twitter-postagger.html
16. https ://gate.ac.uk/

en faisant une recherche par les hashtags utilisés. Généralement, les utilisateurs n'utilisent pas le même hashtag pour un sujet particulier. Pour le sujet de #Newsmax_Media, nous avons découvert plusieurs hashtags sur le même sujet tels que : #News_Media,#VanRE,#vancouver...
Ces trois hashtags ont une relation avec les mots Newsmax, Media, vancouver, VanRE . Nous voulons extraire tous les mots qui composent ces hashtags, afin de le relier avec lès mots fréquents qui existent dans le dictionnaire [17] anglais standard comme suit :

#Newsmax_Media → (**Newsmax**, **Media**)
#News_Media → (**News**, **Media**)
#vancouver → (**vancouver**)
#VanRE → (**Van**, **RE**)

Ce traitement va nous aider à extraire le plus de mots possible à partir des hashtags et le faire relier avec les autres mots récupérés du tweets. Les sujets discutés dans les tweets sont généralement symbolisés par des hashtags qui sont le plus souvent des mots composés collés ensemble ou des sujets parlés dans le texte des tweets. Pour faire le lien entre les deux, nous avons eu besoin de faire cette segmentation.

L'algorithme de la segmentation proposé (voir Algorithme 1) est récursif et traite les hashtags dans la direction de la lecture de texte, c'est-à-dire de gauche vers la droite. Ceci nous permet de décomposer le problème en trois parties principales, comme suit :

1. Détecter des mots en se basant sur une délimitation avec une majuscule, utilisée pour marquer le début de chaque mot. Des mots qui commencent par

17. https ://github.com/dwyl/english-words

Algorithm 1 Segmenter les Hashtags

```
1: fonction SEGMENTERHASHTAG(hashtag : chaine) : Ensemble des chaines
2:     ens : Ensemble Vide
3:     si hashtag est vide  ou hashtag[0] ≠ # alors
4:         retourner ens                                    ▷ ensemble vide
5:     sinon
6:         ens.ajouter(hashtag.coupe(1, text.longeur)))
7:     fin si
8:     retourner SegmenterMots(ens)
9: fin fonction
10: fonction SEGMENTERMOTS(Ens : Ensemble des chaines) : Ensemble des
    chaines
11:     Ensemble1 : ensemble des chaines
12:     Ensemble2 : ensemble des chaines
13:     EnsembleTemp : ensemble des chaines
14:                         ▷ chercher les mots qui commence par des majuscules.
15:     pour Mot : Ens faire
16:         si contientMajuscule(Mot) alors
17:             motsDecouper ← decouperAnnotationMajuscule(Mot)
18:             motsSegmenter ← SegmenterMots(motsDecouperParAlphabet)
19:             Ensemble1.ajouterTous(decouperAnnotationMajuscule(motsSegmenter))
20:         fin si
21:     fin pour
22:             ▷ chercher les mots qui contient des caractère non alphabétique.
23:     pour Mot : Ensemble1 faire
24:         si ContientNonAlphabet(Mot) alors
25:             motsDecouper ← decouperParAlphabet(Mot)
26:             motsSegmenter ← SegmenterMots(motsDecouperParAlphabet)
```

Algorithm 2 Segmenter les Hashtags (suite)

```
27:             Ensemble2.ajouterTous(motsSegmenter)
28:         fin si
29:     fin pour
30:     ▷ chercher la plus petite nombre des mots qui compose le varaible Mot.
31:     pour Mot : Ensemble2 faire
32:         EnsembleTemp.ajouterTous(MaxMatsh (Mot,liste des mots anglais))
33:     fin pour
34:     retourner EnsembleTemp
35: fin fonction
```

une majuscule sont collés ensemble. Ce problème a été analysé par la fonction *decouperAnnotationMajuscule* dans la ligne 16 qui cherche les mots qui commencent par une majuscule. Et le séparer à l'aide des expressions régulières. Par exemple l'hashtag **#ParisClimateConference** construit à l'aide de trois mots collés ensemble et chaque mot commence par une majuscule.

2. Détecter des mots utilisant une délimitation avec des caractères spéciaux ou par des chiffres. Ce problème a été analysé par la fonction *decouperParAlphabet* qui cherche les mots qui sont séparés par des caractères non alphabétiques ou un nombre. Exemple**#3Novices,#Newsmax_Media** cette fonction a été construite à l'aide des expressions régulières qui détectent les caractères non alphabétiques, ou bien les chiffres dans le hashtag.

   Détecter les mots dans la séquence de lettres en minuscules faites en consultant le dictionnaire anglais [18]. Ce problème a été analysé par la fonction *maxMatsh* dans la ligne 31 qui cherche le plus petit nombre des mots qui

18. https ://github.com/dwyl/english-words

composent le mot dans le dictionnaire de gauche vers la droite, parce que tout simplement l'écriture de la langue anglaise se fait suivant cette orientation. Par exemple, le hashtag suivant **#renewableenergy** peut être décomposé de plusieurs façons, comme suit :

#renewableenergy → (**renew**, **able**, **energy**)

#renewableenergy → (**renewable**, **energy**)

Notre algorithme se base sur le nombre minimal de décompositions, car quand on lit une séquence de caractères collés ensemble, on essaie de trouver la chaine de caractères la plus longue. Pour cette raison, on a choisi le plus petit ensemble de mots qui compose la séquence.

Dans l'exemple précédent, on remarque que (**renewable**, **energy**) est la décomposition idéale. Le code de ce programme est fournis à l'annexe A.

6.7 La reconnaissance des entités nommées

La détection des entités nommées constitue une difficulté majeure dans cette étude. En fait, ces entités se compliquent avec les différents formats d'écriture utilisés. Par exemple, la date « 2016-03-10 » est différente de « 10 mars 2016 » au niveau orthographique, même si elles ont la même valeur sémantique. Les deux formats d'écriture ont le même sens. Donc, les entités nommées nécessitent une transformation vers un standard commun en utilisant une méthode de normalisation (Chang et Manning, 2012). Cette technique est utilisée dans différents projets ed recherches et domaines, comme le projet GNAT(Gene/protein named entity recognition and normalization software) (Wermter *et al.*, 2009) et le projet DNorm (Leaman *et al.*, 2013) qui se basent sur la domaine médical. Les données de type pourcentage, monnaie et temps, peuvent être transformées dans un format unique. L'API StanfordNLP, nous propose une normalisation pour les dates, les horaires, les pourcentages, l'argent et les mesures en les transformant dans un

format standard commun.

Parfois, dans une même phrase, nous découvrons une combinaison de mots qui peuvent désigner un sens unique, mais l'utilisation de ces mots en direct peut faire éloigner le sens de la phrase. Pour cela, nous essayons de détecter ces combinaisons de mots ensemble et on les garde en séquence. Généralement, ces données sont des entités nommées qui font parti du sujet de discussion. Les données de type location ou organisation n'acceptent pas le type de standardisation direct comme les dates. Pour cette raison, on cherche des synonymes proches à l'aide de WordNet s'ils existent dans le thesaurus.

6.8 La lemmatisation

La lemmatisation est l'étape qui désigne l'analyse lexicale chargée de faire regrouper les mots d'une même famille qui partagent le même suffixe lexical. Chacun des mots du texte se trouve ainsi réduit en une entité appelée « Lemme ». Ce lemme désigne la forme canonique des mots. La lemmatisation regroupe les différentes formes que peut avoir un mot. Par exemple, un nom en pluriel va être réduit au singulier, un verbe à son infinitif, etc.
La lemmatisation aide à regrouper les mots et les faire représenter avec les lemmes dans le but de réduire la dimension de l'espace des mots. Par conséquence, si les mots partageant un lemme on les considère comme un mot unique.

Dans notre travail, pour effectuer cette tâche, nous avons utilisé le lemmatiseur anglais de StanfordNLP [19].

19. http ://nlp.stanford.edu/software/

6.9 Détection des mots vides (stop liste)

Les mots vides (ou stop words) sont des mots qui sont tellement communs qu'il est inutile de les traiter ou de les utiliser dans une recherche d'informations. En Anglais, certains de ces mots sont « the », «is », « far», etc.
Un mot vide est un mot non significatif figurant dans un texte. La signification d'un mot s'évalue à partir de sa distribution (au sens statistique) dans une collection de textes. Un mot dont la distribution est uniforme sur les textes de la collection est dit « vide » et ne permets pas de distinguer les textes les uns par rapport aux autres.

En d'autres termes, un mot qui apparait avec une fréquence semblable dans chacun des textes de la collection n'est pas discriminant, car il ne permet pas de distinguer les textes les uns par rapport aux autres.
D'autre part, certains mots grammaticaux sont assez rares pour constituer des mots pleins.
La collection des mots vides [20] utilisés dans la classification des tweets est la même collection utilisée dans la recherche d'informations. Elle a pour le but de filtrer les tweets et d'extraire juste les mots pertinents afin de discriminer ces tweets par les mots qu'ils représentent.

6.10 La méthode de pondération

Quelle que soit la méthode de classification retenue, la première opération consiste à représenter les documents de façon à ce qu'ils puissent être traités automatiquement par les classifieurs. La plupart des approches se basent sur la représentation vectorielle des documents. Cette représentation est utilisée dans de nombreux autres domaines connexes de l'apprentissage automatique tels que par exemple,

---

20. http ://members.unine.ch/jacques.savoy/clef/englishST.txt

la fouille des textes, la recherche d'informations et le traitement automatique des langues.

6.10.1 La représentation vectorielle

La représentation vectorielle ou également appelée le Modèle vectoriel (*VSM* pour *Vector Space Model*) a été initialement développée pour le système SMART (Büttcher *et al.*, 2010). Le principe consiste à représenter chaque document de la collection comme un point de l'espace, autrement dit, un vecteur de coordonnées dans l'espace vectoriel. Les coordonnées correspondent en fait aux descripteurs composant le document. Dans la figure 6.4, quatre documents sont symbolisés dans un espace à trois dimensions (chaque dimension correspondant à un terme). Ainsi, deux points proches ($\overrightarrow{Tweet_1}$,$\overrightarrow{Tweet_2}$) dans l'espace vectoriel sont considérés comme des proches sémantiques (Albitar, 2013).

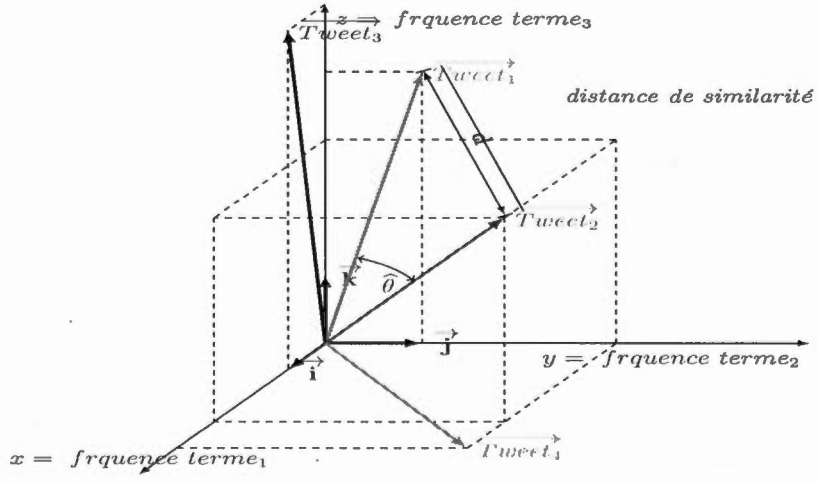


Figure 6.4 Représentation dans l'espace vectorielle avec trois termes.

Le système proposé repose sur une méthode automatique consistant, dans un premier temps, à représenter les tweets sous forme vectorielle. La méthode se décompose en deux phases que nous détaillons ci-dessous.

Une fois le corpus acquis, il sera représenté de manière vectorielle. Chaque tweet sera considéré comme un sac des mots.

Dans cette représentation dite « Saltonienne », un traitement préalable consistera à éliminer les mots inutiles (préposition, mots vides, etc.).

Chaque mot présent dans le corpus représentera une dimension dans l'espace vectoriel sur lequel nous nous appuierons pour effectuer la représentation.
Deux types de représentations peuvent alors être effectuées : une représentation fréquentielle (nombre d'occurrences des mots dans chaque tweet) et la mesure *TF-IDF* (Jones, 2004).

Dans les lignes qui suivent, nous allons appliquer la représentation fréquentielle à partir d'un corpus constitué des deux tweets suivants (cf. tableau 6.2) :

Tableau 6.2 Exemple de deux tweets

Twitte 1	Digital economy, intellectual property and small business in 2011.
Twitte 2	business ,Non-corporate crorepatis tripled in #Indiasince 2011.

La représentation saltonienne du corpus est donnée sous forme fréquentielle dans la Figure 6.3 :

Tableau 6.3 Représentation fréquentielle

	Digital	economy	intellectual	property	business	#corporate	crorepatis	tripled	#Indiasince	2011
tweet 1	1	1	1	1	1	0	0	0	0	1
tweet 2	0	0	0	0	1	1	1	1	1	1

Pour éviter d'obtenir des vecteurs trop creux, une phase d'élagage sera appli-

quée, et ce, tout en évitant qu'un mot se répète dans notre vecteur avec une conjugaison différente (lemmatisation). Avec cette représentation, les mots de la même famille peuvent être rassemblés (mots singuliers/pluriels, féminins/masculins, verbes conjugués, etc.). La représentation canonique des mots permet un regroupement de ce dernier.

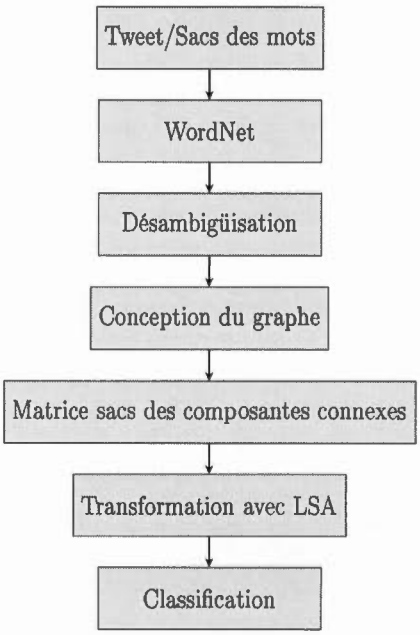


Figure 6.5 Processus de conception de la matrice des composantes connexes/ tweets.

6.11 La désambigüisation des tweets en utilisant WordNet

Il existe différentes méthodes de désambigüisation qui dépendent de l'analyse des textes, comme la classification automatique et les mesures de cooccurrence. Dans la présente étude, nous utilisions WordNet [21] pour l'expansion des tweets, ainsi

21. http ://wordnetcode.princeton.edu/wn3.1.dict.tar.gz

que pour la désambigüisation des entités nommées.

WordNet nous propose pour chaque mot, plusieurs sens appelés Synset, représentant des synonymes. Pour choisir le bon sens, on devrait chercher celui qui correspond le mieux aux contextes du texte (tweet).

Pour cela, nous avons adopté une méthode structurelle fondée sur la distance sémantique entre les concepts selon la formule suivante (Navigli, 2009) :

$$\hat{S} = \text{argmax}_{S \in Senses(w_i)} \sum_{w_j \in T : w_i \neq w_j} \text{max}_{S' \in Senses(w_j)} Score(S, S'). \qquad (6.1)$$

où T est l'ensemble des termes qui forment le tweet, w_i est le terme qu'on souhaite désambigüiser, $Senses(w_i)$ est l'ensemble des concepts candidats pour le terme w_i, ce qui correspond dans WordNet aux synsets qui contiennent ce terme.
$Score(S, S')$ est la fonction utilisée pour mesurer la similarité entre deux concepts S et S'(Audeh *et al.*, 2013).

Plusieurs méthodes existent pour mesurer la similarité entre deux concepts. Suite à plusieurs comparaisons, nous avons choisi une approche basée sur le parcours des arêtes du graphe (Wu et Palmer, 1994; Han *et al.*, 2013b). Cette approche suppose que la similarité entre deux concepts dépend de la profondeur des nœuds concernés et de leur ancêtre commun (Least Common Concept) par rapport à un nœud racine dans la ressource.
Cette technique a été introduite par Audeh *et al.* (2013) dans la recherche d'informations afin d'exécuter l'expansion des requêtes. Nous avons adopté cette technique pour faire enrichir les termes des tweets.

6.11.1 La sélection des termes

Une fois la désambigüisation des termes d'un tweet faite, l'étape suivante consiste à trouver les termes d'expansion les mieux adaptés pour chaque mot original.

La première étape consiste à chercher les synonymes dans le synset sélectionné par l'étape précédente. Dans la deuxième étape, on utilise le même technique d'expansion pour les synsets de l'hyperonymie. La désambigüisation peut se faire par un mot ou par une entité nommée.

6.11.2 La construction du graphe

Une fois que la sélection de synset choisi avec le sens le plus proche au contexte de tweet, nous regroupons les mots m extrait des tweets avec leurs synsets dans un graphe $G = (V, E)$, ce graphe est défini comme suite :

$$\begin{cases} V = \{m\} \cup Synset_m \ , \ m \in S_{tweet} \\ \\ E = \{(v_1, v_2) \in V^2, (v_1 \ Synonyme \ v_2) \vee (v_1 \ Hypronyme \ v_2) \vee \cdots\} \end{cases} \tag{6.2}$$

Le graphe G représente tous les fragments des mots extraits du graphe WordNet avec les relations entre les synsets :

- Les nœuds V correspondent à tous les mots extraits des tweets (S_{tweet}) avec les termes de tous les synsets choisis.
- Les arrêts E sont des relations WordNet utilisées dans nos expériences (Synonymie, Hypéronymie).

De cette façon, on a un graphe avec une connexité faible. Avec la possibilité de chercher les composantes connexes, une composante va contenir une connexité entre les mots représentant les sens des synsets dans WordNet. Ensuite, nous cherchons toutes les composantes connexes dans le graphe G. Chaque composante connexe va contenir des nœuds correspondants aux termes. Ces termes sont reliés par des arrêtes qui sont les relations WordNet.

L'idée est de regrouper les mots m_1 et m_2 et les relier par une arrête correspondant à une relation de synonymie avec un mot $m \in \{m_1\} \cup Synset_{m_1}$ et ce mot m a

une relation $m \in \{m_2\} \cup Synset_{m_2}$.
Autrement dit :

si $G'(V', E')$ *une composante connexe dans G(V,E)* $/V' \subset V \wedge \{m_1, m_2\} \in V'$

On a alors

$$\begin{cases} m \in \{m_1\} \cup Synset_{m_1} \\ m \in \{m_2\} \cup Synset_{m_2} \\ (\{m_1\} \cup Synset_{m_1}) \cap (\{m_2\} \cup Synset_{m_2}) \neq \phi \end{cases} \tag{6.3}$$

La matrice des sacs de mots va être représentée par les composantes connexes et les tweets. Pour ce faire, nous choisissons un mot surnommé « Représentant » qui va symboliser une composante et qui à son tour va représenter une dimension dans notre matrice. En d'autres termes, la matrice des sacs de mots devient un sac de représentants des mots ou un sac des composantes connexes. À titre d'exemple, dans la figure 6.6, le mot $football$ est le représentant de la composante suivante : $football \rightarrow [football, football\ game, soccer/sports, associationfootball, soccer]$

Chaque fois que nous trouvons un des mots de cette composante dans un tweet, nous incrémentons la fréquence de mot $football$. La taille des vecteurs des tweets va diminuer, car les dimensions des vecteurs au lieu d'être représentées par des mots seront représentées par les dimensions des composantes connexes de notre graphe G.

La figure 6.6 montre un graphe composé par deux composantes connexes représentées par le mot «$football$» et «*disco*» dans l'espace vectoriel.
Le tableau 6.4 montre la représentation d'une matrice de m tweets avec l'extrait des fréquences $f_{j,i}$ des mots trouvés dans les composantes connexes $Composante_i$ qui vont être représentées par un seul mot qu'on va nommer «Représentant».
L'attribut «Classe» fait référence à l'étiquette des tweets pour les assigner dans

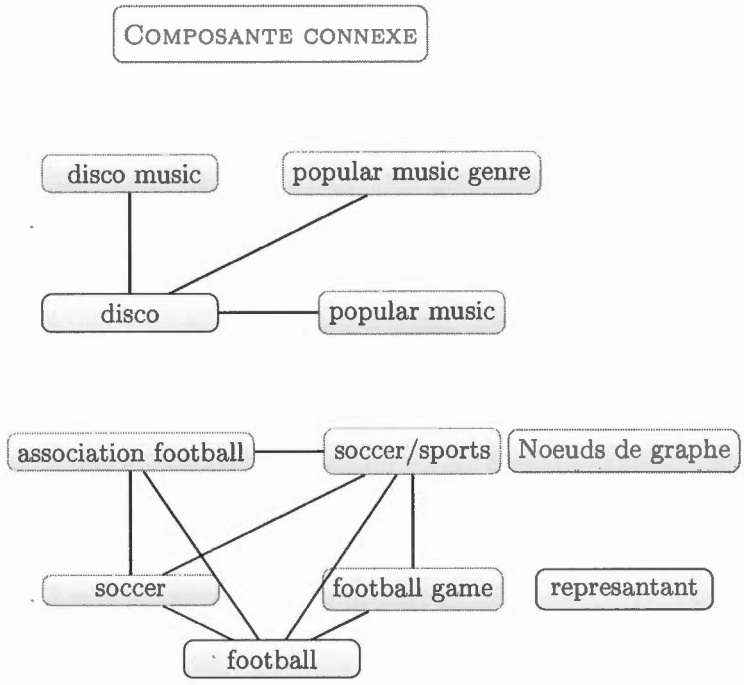


Figure 6.6 Graphe illustrant deux composantes connexes représentées par le mot *football* et *disco* dans l'espace vectoriel.

le modèle d'entraînement.

6.11.3 Réduction du rang avec ASL

Après avoir construit la matrice des occurrences, l'analyse semantique latente (ASL) (en anglais *Latente Semantic Analyses - LSA*) permet de trouver une matrice de rang plus faible, qui donne une approximation de cette matrice.
La logique de l'analyse sémantique latente consiste en une matrice lexicale qui contient le nombre d'occurrences de chaque mot dans chacun des documents. Pour extraire les relations sémantiques entre les mots à partir d'une matrice lexicale, l'analyse simple des cooccurrences brutes se heurte à un problème majeur. Même dans un grand corpus de textes, la majorité des mots sont relativement

Tableau 6.4 Représentation de la matrice des dimensions tweets/ composantes connexes.

Dimension des composantes / Vecteurs des tweets	Classe	Composante_1	$\cdots$	Composante_i	$\cdots$	Composante_n
Tweet_1	étiq_1	$f_{1,1}$	$\cdots$	$f_{1,i}$	$\cdots$	$f_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Tweet_j	étiq_j	$f_{j,1}$	$\cdots$	$f_{j,i}$	$\cdots$	$f_{j,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Tweet_m	étiq_m	$f_{m,1}$	$\cdots$	$f_{m,i}$	$\cdots$	$f_{m,n}$

rares. Il s'ensuit que les cooccurrences ne se répètent pas. Leur rareté les rend particulièrement sensibles à des variations aléatoires (Bestgen, 2004; Berry *et al.*, 1995).

L'ASL résout ce problème en remplaçant la matrice originale de fréquences par une approximation qui produit une sorte de lissage des associations. Pour cela, la matrice de fréquences fait l'objet d'une décomposition en valeurs singulières avant d'être recomposée à partir d'une fraction seulement de l'information qu'il contient. Les mots caractérisant les documents sont ainsi remplacés par des combinaisons linéaires ou « dimensions sémantiques » sur lesquelles peuvent être situés les mots originaux. Contrairement à une analyse classique, les dimensions extraites sont très nombreuses et non interprétables (Foltz *et al.*, 1998; Bestgen, 2004).

L'ASL ou l'ISL [22] est une technique à base algébrique qui a été utilisée dans le

22. LSI : Latent Semantic Indexing utilisé dans la recherche d'information

traitement du langage naturel. En particulier, la sémantique distributionnelle qui consiste à analyser les relations entre un ensemble de documents et les termes qu'ils contiennent en produisant un ensemble de concepts liés aux documents. L'ASL suppose que les mots de sens proches se produiront dans des pièces similaires dans le texte. Une matrice contenant les mots comptés dans les paragraphes (les colonnes représentent les mots uniques et chaque paragraphe est représenté dans une ligne) est construite à partir d'un gros morceau de texte et une technique mathématique appelée « la décomposition en valeurs singulières SVD [23]» qui a été utilisée pour réduire le nombre des lignes tout en préservant la structure de similitude entre les colonnes formées par deux rangées. Les valeurs proches de 1 représentent des mots très similaires alors que les valeurs proches de 0 représentent des mots très dissemblables. On peut justifier cette approximation par plusieurs aspects (Landauer *et al.*, 1998; Evangelopoulos *et al.*, 2012; Berry *et al.*, 1995) :

1. La matrice d'origine pourrait être trop grande pour les capacités de calcul de la machine.
2. La matrice d'origine peut être « bruitée » : des termes n'apparaissent que de manière anecdotique. La matrice sera nettoyée des vecteurs et des attributs qui peuvent être répétés dans les tweets, et ce, afin d'améliorer les résultats.
3. La matrice d'origine peut être « trop creuse » : elle contient les mots propres à chaque tweet plutôt que les termes liés à plusieurs tweets. C'est également un problème de synonymie.

Nous avons adopté la technique de la réduction matricielle pour réduire le nombre d'attributs fournis en les regroupant à l'aide du WordNet. Étant donnée une matrice X, on sait qu'il existe, deux matrices U et V orthogonaux et une matrice diagonale Σ telles que :

$$X = U\Sigma V^T \tag{6.4}$$

23. SVD : Singular Value Decomposition

D'après la théorie de l'algèbre linéaire (Landauer *et al.*, 2013), il existe une décomposition X de telle sorte que, U et V sont des vecteurs orthogonaux et Σ est une matrice diagonale. Ceci est appelé une décomposition en valeurs singulières :

$$\begin{cases} X = U\Sigma V^T \\ XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma\Sigma^T U^T \\ X^T X = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma\Sigma^T V^T \\ \textit{Déterminant}(XX^T - \sigma^2 I) = 0, I \textit{ la matrice d'identité} \end{cases} \tag{6.5}$$

Depuis $\Sigma\Sigma^T$ et $\Sigma^T\Sigma$ sont en diagonale, nous voyons que U doivent contenir les vecteurs propres de XX^T, alors que V doivent être les vecteurs propres de $X^T X$. Les deux produits ont les mêmes valeurs propres non nulles, données par les entrées non nulles $\Sigma\Sigma^T$. Pour trouver les valeurs propres il se fait de résoudre l'équation $\textit{Déterminant}(XX^T - \sigma^2 I) = 0$.

La décomposition se présente comme suit (Landauer *et al.*, 2013) :

$$(t_i^T) \rightarrow \overbrace{\begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}}^{X} = (\hat{t}_i^T) \rightarrow \overbrace{\begin{bmatrix} \begin{bmatrix} \\ u_1 \\ \\ \end{bmatrix} & \cdots & \begin{bmatrix} \\ u_l \\ \\ \end{bmatrix} \end{bmatrix}}^{U} \cdot \overbrace{\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_l \end{bmatrix}}^{\Sigma} \cdot \overbrace{\begin{bmatrix} \begin{bmatrix} v_1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} v_1 \end{bmatrix} \end{bmatrix}}^{V^T} \tag{6.6}$$

Nous traitons un tweet comme un « mini-document » et nous le comparons dans l'espace des concepts à un corpus pour construire une liste des documents les plus pertinents. Pour faire cela, il faut déjà convertir le tweet dans l'espace des concepts, en le transformant de la même manière que les documents.

Comment l'ASL a été utilisé ?

Nous avons représenté les composantes connexes au lieu de représenter les sacs de mots alors, l'espace va contenir un sac des représentants pour les composantes

connexes.

L'analyse sémantique latente (ASL) pose le problème sur la façon de trouver une représentation de la matrice $X = U\Sigma V^T$ dans un espace vectoriel réduit. Cette méthode tente de résoudre le problème en mettant une correspondance entre les représentants des mots et les tweets dans un espace de concept réduit. Pour ce faire, nous trions les valeurs propres et nous choisissons les K plus grands valeurs σ afin de réduire la matrice Σ. Notre nouvelle dimension devient alors K.

La figure 6.7 montre l'interface graphique qui permet de générer la matrice tweets /composante connexe de la classification, les dimensions après la transformation de l'LSA.

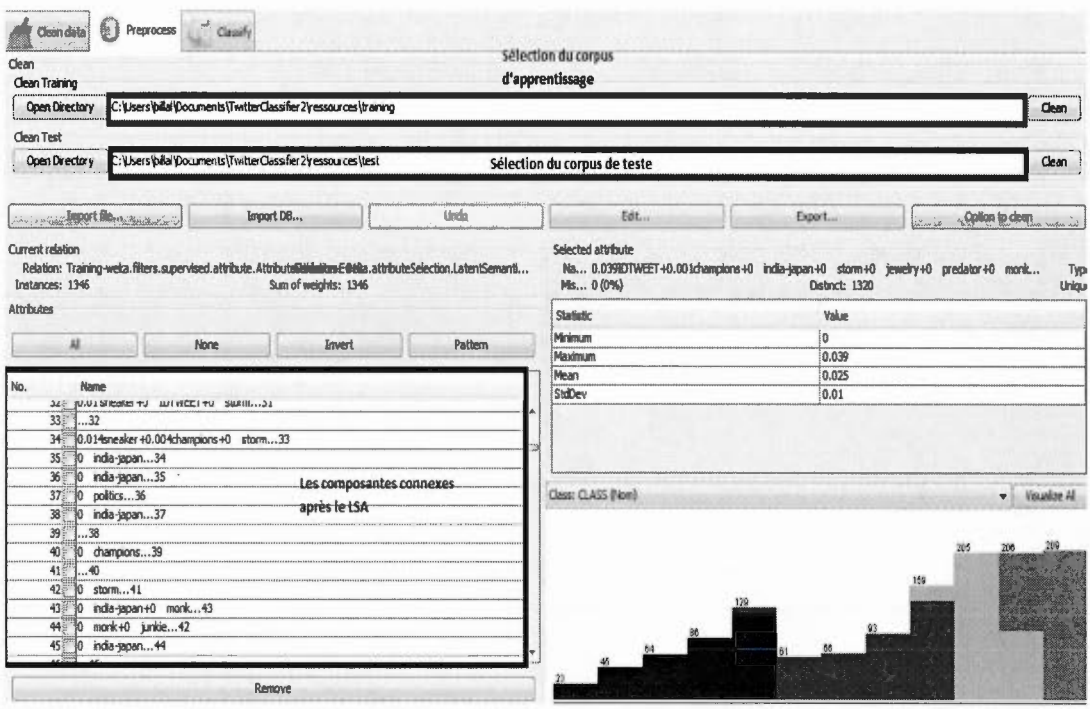


Figure 6.7 Interface utilisateur permettant de générer la matrice tweets/ composantes connexes

6.12 Le choix des classifieurs

La résolution des problèmes de la classification à grande échelle est cruciale dans de nombreuses applications telles que la classification de texte. Nous nous basons dans le choix du classifieur sur la rapidité et la simplicité. La classification linéaire est devenue l'une des techniques d'apprentissage les plus prometteuses pour les grosses données avec un grand nombre de cas et de fonctionnalités (Fan *et al.*, 2008). Nous nous basons sur LIBLINEAR [24] comme outil facile à utiliser pour traiter ces données. Cette outil prend en charge la régression logistique L2-régularisée (LR), L2-loss et L1-loss linéaires vecteurs machines de support (SVM) (Boser *et al.*, 1992). Il possède de nombreuses caractéristiques de la bibliothèque populaire de SVM (LIBSVM) (Chang et Lin, 2011) tel que la simplicité, la richesse de la documentation en plus d'être un logiciel libre (Sous license BSD [25]).

LIBLINEAR est très efficace pour l'entrainement des données sur des problèmes à grande échelle. Il ne faut que quelques secondes pour s'entrainer à un problème de classification de texte. Pour la même tâche, un autre classifieur SVM tel que LIBSVM s'exécute en plusieurs heures. En outre, LIBLINEAR est compétitif et plus rapide que les classificateurs linéaires tels que Pegasos (Shalev-Shwartz *et al.*, 2007).

---

24. http ://www.csie.ntu.edu.tw/ cjlin/liblinear

25. La nouvelle licence BSD(Berkeley Software Distribution License) approuvé par l'initiative Open source.

CHAPITRE VII

ÉVALUATION DE LA MÉTHODOLOGIE

Nous avons effectué plusieurs évaluations pour la classification des tweets. Aussi, nous avons étudié plusieurs cas et nous avons fait des comparaisons.

7.1 Les différentes évaluations

Dans les présentes évaluations, nous avons procédé par plusieurs stratégies de filtrage afin de comparer nos méthodologies :

1. Filtrage par catégories grammaticales : Filtrage par catégorie grammaticale : son rôle est de permettre de garder seulement les mots pertinents tels que les adjectifs, les noms et les verbes, tout en éliminant les mots vides. Le tableau 7.1 montre les catégories grammaticales ou parties du discours sélectionnées dans le filtrage.
2. Segmentation des hashtags : permet d'évaluer l'effet d'extraire les mots qui composent les hashtags sur les résultats de la classification.
3. Relations WordNet : permettent d'évaluer l'effet de l'utilisation des relations sémantiques de WordNet (synonyme, hyperonymie, synonyme avec hyperonymie) sur les résultats de la classification.
4. Reconnaissance des entités nommées : les entités nommées se trouvent dans le texte sous forme de mots composés ou d'abréviations. Ces entités ont

Tableau 7.1 Les catégories grammaticales(en anglais *Part of speech*-POS) utilisées dans le filtrage pour sélectionner les mots.

Abréviation (Catégorie grammaticale)	Description(Catégorie grammaticale)
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
JJ	Adjective

besoin de désambigüisation afin de déterminer la semantique et le sens du tweet.

Les résultats sont présentés dans le tableau 7.2 récapitulatif ci-dessous :

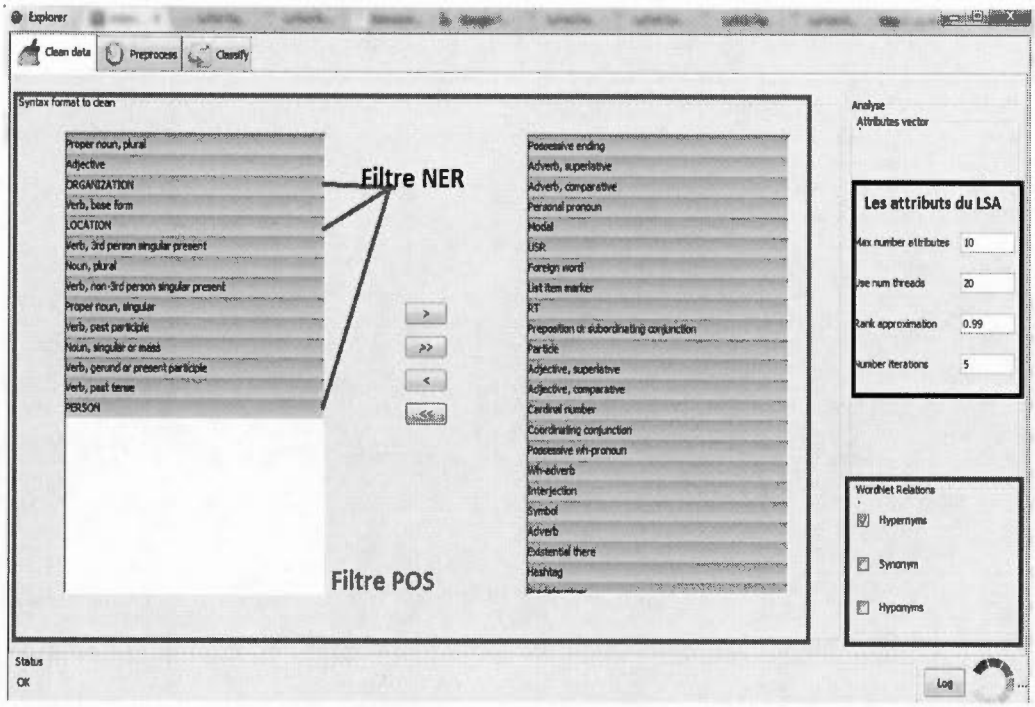


Figure 7.1 Interface de contrôle des expérimentations

7.2 Résultats

Dans un but d'étudier l'utilité de l'utilisation des entités nommées, nous avons exécuté deux types d'expérimentations, le premier type avec la reconnaissance des entités nommées et le deuxième type sans la reconnaissance des entités nommées. Le tableau 7.2 résume les résultats des évaluations.

7.2.1 Résultats sans la reconnaissance des entités nommées (¯REN)

1. Nous avons constaté que le filtrage par les catégories grammaticales (*Part Of Speech* en anglais) des tweets, améliore considérablement les résultats avec un gain de 29,9% en précision en utilisant la synonymie. Par ailleurs, on peut avoir jusqu'a 81.2% de gain en utilisant la relation de synonymie

Tableau 7.2 Différentes évaluations et résultats sans la reconnaissance des entités nommées(¯REN)

Type de filtre	Segmentation de hashtag	Relation WordNet	Précision sans ¯REN(%)
Tous les mots	Non	Synonymie	38.3
Filter Adj,NN,VB	Non	Synonymie	68.2
Tous les mots	Oui	Synonymie	42.1
Filter Adj,NN,VB	Oui	Synonymie	73.4
Tous les mots	Non	Hyperonymie	41.7
Filter Adj,NN,VB	Non	Hyperonymie	70.9
Tous les mots	Oui	Hyperonymie	43.2
Filter Adj,NN,VB	Oui	Hyperonymie	76.4
Tous les mots	Non	Synonymie&Hyperonymie	46.8
Filter Adj,NN,VB	Non	Synonymie&Hyperonymie	72.3
Tous les mots	Oui	Synonymie&Hyperonymie	51.3
Filter Adj,NN,VB	Oui	Synonymie&Hyperonymie	81.2

conjointement avec la relation d'hyperonymie. Ce gain peut être expliqué de façon presque naturelle par la suppression des mots sensibles, comme les déterminants et les adverbes qui font office de mots vides dans le texte. En effet, ces mots vides peuvent aboutir à un résultat dégradant de la classification.

L'utilisation des noms, des adjectifs et des verbes peut améliorer la classification selon les résultats obtenus, car ces mots sont nécessairement trouvés dans la liste de filtrage des mots des sujets de discussion des tweets. En effet, cette technique aide, de façon remarquable à l'élimination des mots rares telle que les adresses URL, les adresses électroniques et les adresses utilisateurs pour les comptes Twitter, ainsi que les émoticons qui ne donnent pas

une information pertinente sur le sujet du texte et qui donnent des composantes connexes élémentaires (contient un seul mot).

2. La segmentation des hashtags améliore le résultat avec un gain de 3,8% sans l'utilisation de filtrage par catégories grammaticales et fournit, par ailleurs, une amélioration de 5.2% avec l'utilisation de filtrage. Cela peut être expliqué par l'extraction des informations à partir des hashtags qui peuvent donner des informations plus pertinentes sur le sujet du tweet. En outre, l'information extraite d'un hashtag peut aider à améliorer la précision de la classification. Notant que la précision de la classification croît toujours avec la croissance de la précision du filtrage de la catégorie grammaticale.

3. L'enrichissement des mots avec la relation de synonymie donne un résultat de 68.2% en terme de précision en utilisant le filtrage sans l'utilisation de la segmentation et de 73.4% en terme de précision avec l'utilisation de la segmentation.

   Cependant, le résultat utilisant la relation de l'hyperonymie donne un résultat de meilleure précision comparativement à la synonymie avec un gain de plus de 2.7% en utilisant le filtrage sans l'utilisation de la segmentation et plus de 3% avec l'utilisation de la segmentation.

   La combinaison des deux relations sémantiques ensemble (synonymie et hyperonymie) améliore le résultat avec un gain de 1.4% de plus en utilisant le filtrage sans l'utilisation de la segmentation et le résultat donne un gain de 4.4% en terme de précision avec la segmentation. Nous notons que, peu importe, l'expérimentation utilisant l'hyperonymie donne une meilleure précision par rapport à la synonymie. Par contre, les deux relations ensemble améliorent nettement le résultat initial.

4. La segmentation des hashtags avec le filtrage de la catégorie grammaticale donne une meilleure amélioration de précision avec 81.2% en terme de

précision en utilisant la combinaison des deux relations, la synonymie avec l'hyperonymie.

La figure 7.2 montre un histogramme présentant une comparaison entre les différentes précisions qui se trouvent dans le tableau 7.3 .On remarque une augmentation progressive de la précision après l'utilisation de chaque critère tout en se basant sur la reconnaissance des entités nommées et la segmentation des hashtags avec un filtrage des catégories grammaticales.

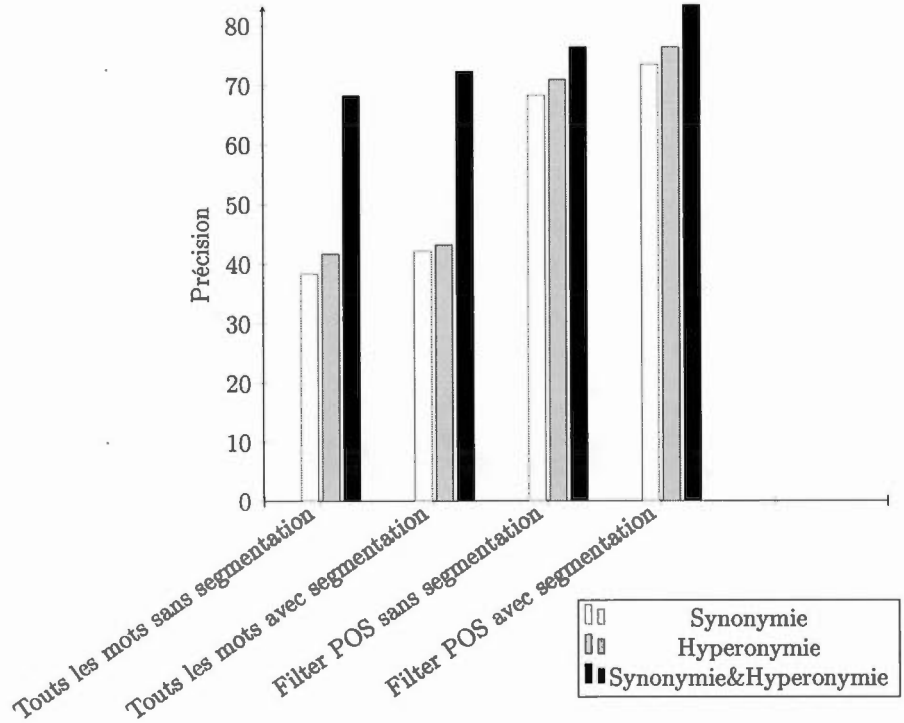


Figure 7.2 Représentation graphique des résultats obtenus avec l'utilisation de la reconnaissance des entités nommées($^{+}$REN).

L'histogramme dans la figure 7.3 montre une comparaison entre les différentes précisions de chaque classe en utilisant la reconnaissance des entités nommées et

la segmentation des hashtags avec un filtrage des catégories grammaticales.

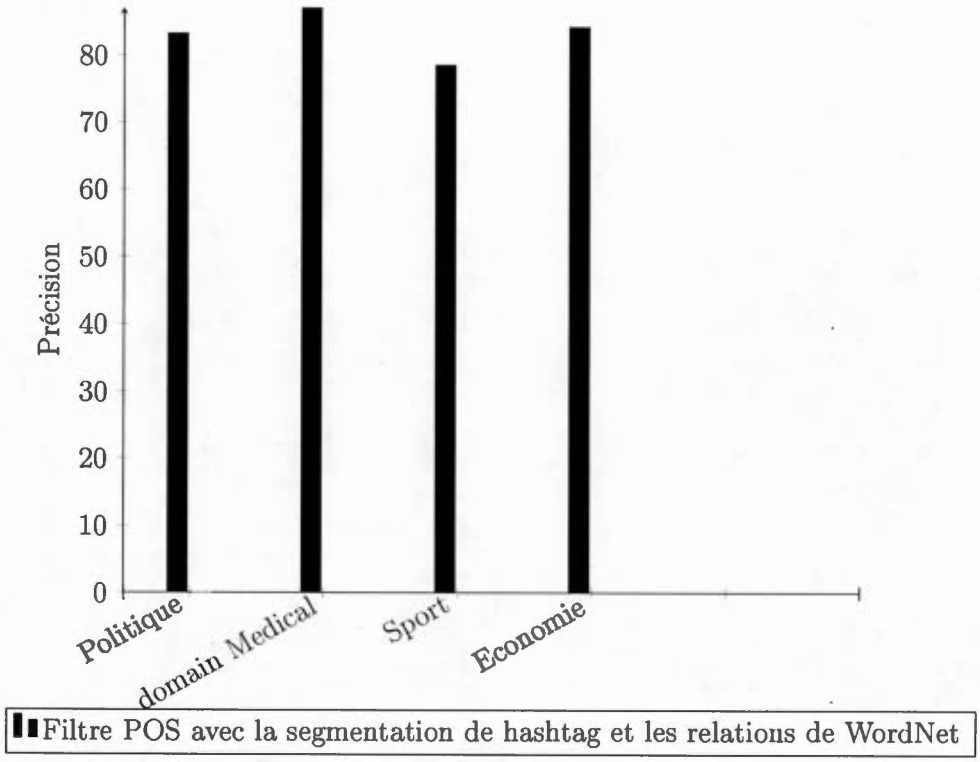


Figure 7.3 Représentation graphique du meilleur résultat obtenu, dépendamment de la catégorie.

7.2.2 Résultats avec la reconnaissance des entités nommées ($^{+}$REN)

En comparant les résultats des deux tableaux 7.2 et 7.3; c.-à-d. avec et sans la reconnaissance des entités nommées, on remarque les faits suivants :

1. L'utilisation du filtrage par les catégories grammaticales (en anglais, *Part Of Speesh*) conjointement avec l'utilisation de la reconnaissance des entités nommées, améliore les résultats avec un gain de 1.5 % en terme de précision

Tableau 7.3 Différentes évaluations et résultats avec la reconnaissance des entités nommées(+REN)

Type de filtre	Segmentation de hashtag	Relation WordNet	Précision +REN(%)	Gain(%)
Tous les mots	Non	Synonymie	36.2	-2.1
Filter Adj,NN,VB	Non	Synonymie	69.7	+1.5
Tous les mots	Oui	Synonymie	41.7	-0.3
Filter Adj,NN,VB	Oui	Synonymie	74.8	+1.4
Tous les mots	Non	Hyperonymie	40.6	-1.1
Filter Adj,NN,VB	Non	Hyperonymie	71.3	+0.4
Tous les mots	Oui	Hyperonymie	42.4	-0.8
Filter Adj,NN,VB	Oui	Hyperonymie	77.3	+0.9
Tous les mots	Non	Synonymie&Hyperonymie	45.3	-1.5
Filter Adj,NN,VB	Non	Synonymie&Hyperonymie	73.2	+0.9
Tous les mots	Oui	Synonymie&Hyperonymie	50.6	-0.7
Filter Adj,NN,VB	Oui	Synonymie&Hyperonymie	83.4	+1.2

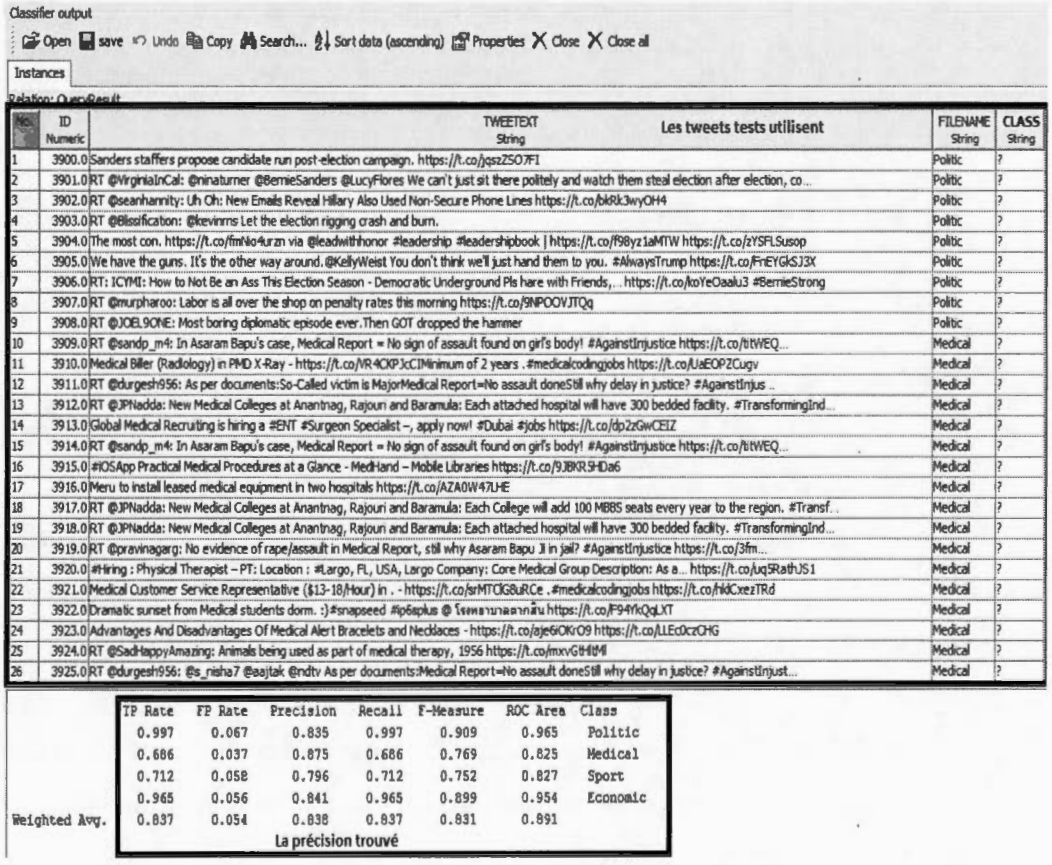
en utilisant la relation de synonymie et sans segmentation des hashtags et un gain de 1.4% en utilisant la segmentation.

2. En enrichissant les mots avec la relation d'hyperonymie, la précision augmente de 0.4% en utilisant le filtrage sans segmentation des hashtags et avec la segmentation on obtient un gain de 0.9%. C'est donc une amélioration de 0.5% en terme de précision.

3. La combinaison des deux relations, l'hyperonymie et la synonymie ont fait nettement augmenter la précision de façon remarquable. Au départ, avec le filtrage et sans l'utilisation de la segmentation des hashtags, la précision a augmenté de 0.9% avec la relation d'hyperonymie, et de 1.4% avec la relation de synonymie. Ceci rentre dans le cadre de l'expérimentation sans l'utilisation de la reconnaissance des entités nommées.

4. L'utilisation du filtrage avec les catégories grammaticales avec la reconnaissance des entités nommées a généré une augmentation de la précision dans toutes nos évaluations.

La figure 7.4 montre l'interface graphique de la classification qui utilise un classifieur de l'outil LIBLINEAR et en contre-partie retourne la précision trouvée dans cette dernière expérimentation.



No.	ID Numeric	TWEETEXT String	FILENAME String	CLASS String
1	3900.0	Sanders staffers propose candidate run post-election campaign. https://t.co/jqszZSO7F1	Politic	?
2	3901.0	RT @VirginiaInCal: @ninaturner @BernieSanders @LucyFlores We can't just sit there politely and watch them steal election after election, co...	Politic	?
3	3902.0	RT @seanhannity: Uh Oh: New Emails Reveal Hillary Also Used Non-Secure Phone Lines https://t.co/bkRk3wyOH4	Politic	?
4	3903.0	RT @Blissification: @kevinrns Let the election rigging crash and burn.	Politic	?
5	3904.0	The most con. https://t.co/fmNo4urzn via @leadwithhonor #leadership #leadershipbook | https://t.co/f98yz1aMTW https://t.co/zYSFLSusop	Politic	?
6	3905.0	We have the guns. It's the other way around.@KellyWeist You don't think we'll just hand them to you. #AlwaysTrump https://t.co/FnEYGkSJ3X	Politic	?
7	3906.0	RT: ICYMI: How to Not Be an Ass This Election Season - Democratic Underground Pls here with Friends,... https://t.co/koYeOaaku3 #BernieStrong	Politic	?
8	3907.0	RT @murpharoo: Labor is all over the shop on penalty rates this morning https://t.co/9NPOOVJTQq	Politic	?
9	3908.0	RT @JOEL9ONE: Most boring diplomatic episode ever.Then GOT dropped the hammer	Politic	?
10	3909.0	RT @sandp_m4: In Asaram Bapu's case, Medical Report = No sign of assault found on girl's body! #AgainstInjustice https://t.co/titWEQ...	Medical	?
11	3910.0	Medical Biller (Radiology) in PMD X-Ray - https://t.co/VR4OXP3cCIMinimum of 2 years .#medicalcodingjobs https://t.co/UaEOP2Cugv	Medical	?
12	3911.0	RT @durgesh956: As per documents:So-Called victim is MajorMedical Report=No assault doneStill why delay in justice? #AgainstInjus ...	Medical	?
13	3912.0	RT @JPNadda: New Medical Colleges at Anantnag, Rajouri and Baramula: Each attached hospital will have 300 bedded facility. #TransformingInd...	Medical	?
14	3913.0	Global Medical Recruiting is hiring a #ENT #Surgeon Specialist –, apply now! #Dubai #jobs https://t.co/dp2zGwCEIZ	Medical	?
15	3914.0	RT @sandp_m4: In Asaram Bapu's case, Medical Report = No sign of assault found on girl's body! #AgainstInjustice https://t.co/titWEQ...	Medical	?
16	3915.0	#iOSApp Practical Medical Procedures at a Glance - MedHand – Mobile Libraries https://t.co/9JBKR3HDa6	Medical	?
17	3916.0	Meru to install leased medical equipment in two hospitals https://t.co/AZA0W47LHE	Medical	?
18	3917.0	RT @JPNadda: New Medical Colleges at Anantnag, Rajouri and Baramula: Each College will add 100 MBBS seats every year to the region. #Transf...	Medical	?
19	3918.0	RT @JPNadda: New Medical Colleges at Anantnag, Rajouri and Baramula: Each attached hospital will have 300 bedded facility. #TransformingInd...	Medical	?
20	3919.0	RT @pravinagarg: No evidence of rape/assault in Medical Report, still why Asaram Bapu Ji in jail? #AgainstInjustice https://t.co/3fm...	Medical	?
21	3920.0	#Hiring : Physical Therapist – PT: Location : #Largo, FL, USA, Largo Company: Core Medical Group Description: As a... https://t.co/uq5RathJS1	Medical	?
22	3921.0	Medical Customer Service Representative ($13-18/Hour) in . - https://t.co/srMTCIG8uRCe .#medicalcodingjobs https://t.co/hkCxezTRd	Medical	?
23	3922.0	Dramatic sunset from Medical students dorm. :)#snapseed #ip6xplus @ โรงพยาบาลแกลงกลิ้น https://t.co/F94YkQqLXT	Medical	?
24	3923.0	Advantages And Disadvantages Of Medical Alert Bracelets and Necklaces - https://t.co/aje6iOKrO9 https://t.co/LLEc0czOHG	Medical	?
25	3924.0	RT @SadHappyAmazing: Animals being used as part of medical therapy, 1956 https://t.co/fmxvGtHtMl	Medical	?
26	3925.0	RT @durgesh956: @s_nisha7 @aajtak @ndtv As per documents:Medical Report=No assault doneStill why delay in justice? #AgainstInjust...	Medical	?

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.997	0.067	0.835	0.997	0.909	0.965	Politic
	0.686	0.037	0.875	0.686	0.769	0.825	Medical
	0.712	0.058	0.796	0.712	0.752	0.827	Sport
	0.965	0.056	0.841	0.965	0.899	0.954	Economic
Weighted Avg.	0.837	0.054	0.838	0.837	0.831	0.891	

Figure 7.4 Interface graphique de la classification avec la meilleure évaluation

7.3 Discussion

Les hashtags représentent une information pertinente au sujet de la discussion et du tweet. La segmentation de ce hashtag pourrait enrichir les composantes connexes qui peuvent contenir les mots du hashtags. Un exemple est le hashtag #ParisClimateConference qui contient le mot *climate*. Avant la segmentation, un tweet associé à ce hashtag ne partage aucun mot avec la composante connexe suivante :

climate→ *environmental condition*, *clime*, *climate*

Cependant, après la segmentation du hashtag le mot *climate* apparait dans le sac de mots de tweet qui contient ce hashtag :

$$\#\boldsymbol{ParisClimateConference} \rightarrow (\boldsymbol{Paris}, \boldsymbol{Climate}, \boldsymbol{Conference})$$

En effet, les tweets qui ont le mot *climate* soit dans le texte du tweet ou bien dans le hashtag lui-même vont partager ce mot dans la même composante connexe.
Notre filtrage qui se base sur la catégorie grammaticale (*Part-Of-Speech*-POS) aide vraiment l'amélioration de la précision et ainsi la qualité de la classification. Le filtrage sur la catégorie grammaticale se base sur le filtrage des noms, des adjectifs et des verbes et ces trois catégories principales se trouvent généralement dans le texte du tweet. Pour cette raison, nous constatons une bonne amélioration de la précision après le filtrage avec les catégories grammaticales.
Malgré l'existence des mots polysémiques, on remarque que plusieurs mots peuvent être employés dans un format grammatical différent et peuvent avoir différentes significations, mais la désambigüisation en utilisant WordNet(décrite dans la section 6.11) à aidé à désambigüiser les mots ainsi qu'à déterminer la sémantique et la catégorie grammaticale du mot si ce mot est un nom, verbe ou un adjectif dans le thésaurus WordNet.

Les entités nommées détectées par WordNet de types personne, organisation et lieu sont minimes, parce que, WordNet ne contient pas tous les noms des personnes possibles, mais contient des noms des célébrités par exemple «Barack Obama», «Hillary Clinton», etc. Ainsi que les organisations célèbres ou internationales. Cependant, WordNet contient une large bibliothèque des lieux.

Notre utilisation des entités nommées a aidé dans l'amélioration de la précision quand elle est utilisée conjointement avec le filtrage basé sur les catégories grammaticales. Ceci peut être expliqué par l'existence des compositions de mots qui désignent un sens unique d'une entité nommée telle que le mot «United state of America» synonyme du mot «United States» et de l'abréviation «USA». La figure 7.5 montre un sous-graphe de la relation de synonymie extrait de WordNet. L'ex-

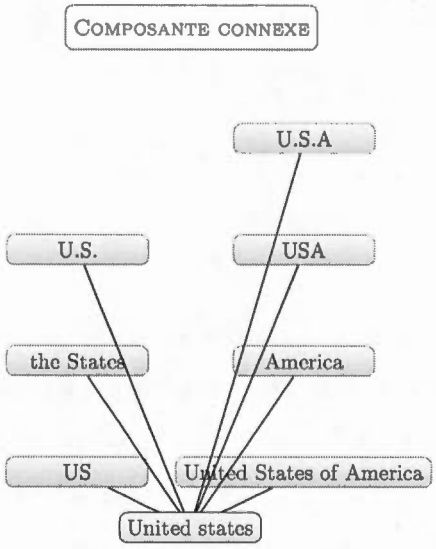


Figure 7.5 Sous graphe représentant une composante connexe représentée par les synonymes du mot composé *«United States»*.

traction des entités nommées en mots séparés peut affecter la qualité des résultats dans nos évaluations. Par contre, l'utilisation des entités nommées comme une composition de mots sous forme d'une séquence peut aider à désambigüiser cette

séquence à l'aide de WordNet.

Exemple, l'utilisation du mot *«United States»* comme un ensemble de mots séparés dans la recherche dans le thésaurus WordNet, donne les résultats suivants :

$$\begin{cases} United \rightarrow unite, unify \\ \text{State} \rightarrow government, authorities, regime \end{cases} \tag{7.1}$$

Cependant, l'utilisation de «United States» comme une composition de mots donne un résultat différent :

United States → United States, United States of America,
America, the States, $US, U.S., USA, U.S.A$

De plus, quelques abréviations et acronymes ont pu être détectés à l'aide de WordNet, tel que le mot *«FBI»* ou le mot *«UN»* :

FBI → Federal Bureau of Investigation, **FBI**
ou
UN → United Nations, **UN**

cette reconnaissance des entités nommées a permit de trouver un lien entre les entités nommées se trouvant dans les tweets sous différentes formes. De plus, la désambigüisation a fait un lien entre les tweets qui peuvent contenir l'un des synonymes ou hyperonymes, c'est-à-dire, un tweet qui a le mot *«UN»* (équivalent au mot *«United Nations»*) et un tweet qui a le mot *«United Nations»* seront regroupés ensemble par le sous-graphe (composante connexe *«UN»* et *«United Nations»*)). Il y aura danc réduction dans l'espace du concept. Plutôt que de représenter chaque entité dans une dimension, on représente l'abréviation avec ces synonymes dans une seule dimension.

CONCLUSION

Notre travail de recherche a apporté des réponses claires à des questions importantes. Tout d'abord, Nous avons introduit un processus (pipeline) de collection des tweets qui génère le modèle en se basant sur un détecteur de langue anglaise afin de minimiser le bruit dans les tweets multilingues.

Ensuite, nous avons introduit un processus (pipeline) de prétraitement comprenant la reconnaissance des entités nommées afin de réduire le temps de prétraitement. Ce processus se compose des tâches suivantes :

- Tokenisation
- Normalisation des tweets
- Analyse par les catégories grammaticales (Part-Of-Speech)
- Segmentation des hashtags
- Reconnaissance des entités nommées(REN)

Notre filtrage basé sur la catégorie grammaticale (Part-Of-Speech)a aidé l'amélioration de la précision et de la qualité de la classification.

Les mots extraits par la segmentation des hashtags ont enrichi la collection des mots extraits des tweets. Ces mots ont aussi aidé à améliorer la précision de la classification.

Nous avons introduit un concept basé sur les théories de graphe pour extraire les composantes connexes. Ces dernieres ont contribué à réduire de manière significative la matrice des sacs de mots (*Bag-Of-Word*). Aussi, nous avons réduit les sacs

de mots en des composantes connexes. Chaque composante contient plus d'un mot, et les mots de cette composante sont reliés par des relations sémantiques extraits de WordNet. C'est-à-dire que les mots vont être similaires.
La matrice réduite générée à partir des composantes connexes a contribué à l'amélioration de la performance de la classification des tweets.

La reconnaissance des entités nommées dans les tweets a amélioré la précision, lorsqu'utilisée parallèlement avec le filtrage des catégories grammaticales. Ainsi, l'utilisation des entités nommées se fait conjointement avec un filtrage des noms, des verbes et des adjectifs sinon la précision de la classification est touchée par une diminution.

Nous nous sommes basés sur le thésaurus WordNet afin d'enrichir sémantiquement les mots extraits des tweets. Cependant, les mots extraits des tweets ne sont pas couverts en totalité dans cette ressource. Dans les travaux futurs, nous pouvons nous baser sur un thésaurus plus large comme *Babelnet*(Navigli et Ponzetto, 2012) et explorer le *Word embeddings* sur l'analyse distributionnelle sémantique pour la désambigüisation des entités nommées, des acronymes et des abréviations.

Dans ce travail, nous n'avons pas pris en compte la détection des expressions polylexicales (EPLs), malgré que la désambigüisation de ces derniers peut apporter du changement dans la qualité de la classification.
Aussi, la segmentation des hashtags ne prend pas en compte les hashtags multilingues, par exemple #japan ぐるーヴ qu'on peut décomposer en des mots anglais et en des mots écrits dans une langue non-latine(la langue japonaise dans ce cas).
D'autres perspectives futures incluent l'apprentissage semi-supervisé pour la classification des tweets, qui nous aiderait à élargir le corpus d'apprentissage avec des tweets non annotés. Cette technique peut faciliter le travail de l'expert au niveau de l'étiquetage manuel du corpus. Nous sommes aussi très intéressés par

les réseaux de neurones et l'apprentissage en profondeur dans l'élaboration d'un système de classification semi-supervisé.

ANNEXE A

PROGRAMME DE SEGMENTATION

Cette annexe contient une classe Java permettant de détecter les mots qui composent un hashtag en utilisant un dictionnaire anglais.

```
package intoxicant.analytics.coreNlp;

import cmu.projectclassifier.util.Util;
import java.io.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import org.apache.commons.io.FileUtils;
import org.apache.commons.lang.math.NumberUtils;

/**
 * Segmenteur est une classe java proposée pour
 * la segmentation, Elle traite les hashtags dans
 * la direction de la lecture de texte, c'est-à-dire de
 * gauche vers la droite. Elle nous permet de décomposer le
 * problème en trois parties principales suivantes:
 * 1.Utiliser un dictionaire "large-word-list.txt" dans une
```

```
 * expression régulière après qu'on trie les mots dans un
 * ordre décroissant.
 * 2.Utiliser les caractères non alphabétiques pour séparer
 * les mots.
 * 3.Utiliser les caractères écrits en majuscules pour
 *  séparer les mots.
 * @author Billal Belainine
 */
public class Segmenteur {

//déclaration des constantes
private final String FILE_NAME = "large-word-list.txt";
private final String FILE_NAME_SORT_COPY =
   "large-word-list-sort.txt";
private final String FILE_NAME_PATTERN_COPY =
   "large-word-list-pattern.txt";
static final String NON_ALPHA = "[\\d.]+|\\D+";
static final String ANNOTATION_MAJUSCULE =
   "([A-Z]*)([A-Z][a-z]+)|([a-z]+)";
//déclaration des variables
static private Segmenteur segmenteur = null;
static String pattern = "";
static List<String> largeListMots;
static Pattern p = null;

public class ComparateurDESC implements
   java.util.Comparator<String> {

    public int compare(String s1, String s2) {
```

```
        return Integer.compare(s2.length(), s1.length());
    }
}
/**
* Constricteur: Sengloton Construis une expression régulière
* contenant le dictionnaire trié en ordre croisant, afin de
* cherche la combinaison des mots maximaux pour forme
* l'hashtag
*/
public Segmenteur() {
try {
  if (!new File( FILE_NAME_PATTERN_COPY).exists()) {
    if (!new File( FILE_NAME_SORT_COPY).exists()) {
          this.largeListMots = FileUtils.readLines(
                                    new File(Segmenter.class
                                    .getClassLoader()
                                    .getResource(FILE_NAME)
                                    .getPath()));
          Collections.sort(largeListMots,
                              new ComparateurDESC());
          FileUtils.writeLines(
                              new File(Segmenteur.class
                              .getClassLoader()
                              .getResource(".").getPath()
                              + FILE_NAME_SORT_COPY)
                              , largeListMots);
    } else {
         largeListMots = FileUtils.readLines(
                             new File(Util.class
```

```
                        .getClassLoader()
                        .getResource(FILE_NAME_SORT_COPY)
                        .getPath()));
            }
            pattern = "(";
            for (int i = 0; i < largeListMots.size(); i++) {
                if (i != largeListMots.size() - 1) {
                    pattern += largeListMots.get(i) + "|";
                } else {
                    pattern += largeListMots.get(i);
                }
            }
            pattern += ")";
            FileUtils.writeStringToFile(
                      new File(Util.class.getClassLoader()
                      .getResource(".").getPath()
                      + FILE_NAME_PATTERN_COPY), pattern);
        } else {
            pattern = FileUtils.readFileToString(
                              new File(Util.class
                              .getClassLoader()
                .getResource(FILE_NAME_PATTERN_COPY)
                              .getPath()));
        }
        pattern = "^" + pattern + pattern + "+$";
        p = Pattern.compile(pattern
                          ,Pattern.CASE_INSENSITIVE);
    } catch (IOException ex) {
        Logger.getLogger(Segmenter.class.getName())
```

```
                .log(Level.SEVERE, "Erreur! :" +
    ex.getMessage(), ex);
     }
}

/**
 * La fonction MaxMatch cherche le plus petit nombre des mots
 * qui composent le mot dans le dictionnaire de gauche vers
 * la droite, parce que tout simplement l'écriture de la
 * langue anglaise se fait suivant cette orientation.exemple,
 * le hashtag suivant #renewableenergy ->(renew,able,energy)
 * elle utilise un dictionnaire "large-word-list.txt" dans
 * une expression régulière après qu'on trie les mots dans
 * un ordre décroissant afin de favoriser les mots langues.
 * @param tokenText mot d'un hashtag
 * @return liste des mots d'un hashtag
 * @throws IOException
 */
static public List<String> getMaxMatch(String tokenText)
    throws IOException {

     List<String> crudeSegments = new ArrayList<>();
     Matcher m = p.matcher(tokenText);
     if (m.matches()) {
         while (!tokenText.isEmpty()) {
             String token = "";
             m = p.matcher(tokenText);
             while (m.find()) {
                 token = m.group(1);
```

```
                crudeSegments.add(token);
            }
            tokenText = tokenText.replaceFirst(token, "");
        }
    }
    if (crudeSegments.isEmpty()) {
        crudeSegments.add(tokenText);
    }
    return crudeSegments;
}

/**
 * Cette fonction cherche les mots qui commencent par une
 * majuscule est le sépare à l'aide des expressions
 * régulières. Par exemple l'hashtag #ParisClimateConference
 * construit à l'aide de trois mots collés ensemble et
 * chaque mot commence par une majuscule.
 * @param tokenText mot d'un hashtag
 * @return liste des mots d'un hashtag
 */
public static List<String> decouperParAlphabet
                            (StringBuilder tokenText) {

   Matcher m = Pattern.compile(NON_ALPHA).matcher(tokenText);
   List<String> resultats = new LinkedList<>();
    while (m.find()) {
        resultats.add(m.group());
    }
    return resultats;
```

```
}

/**
 * Cette fonction a été construite à l'aide des expressions
 * régulières qui détectent les caractères non alphabétiques,
 * ou bien les chiffres dans le hashtag. Ensuite, elle
 * l'utilise comme un séparateur des mots.
 * @param crudeSegments liste des mots d'un hashtag
 * @return liste des mots d'un hashtag
 */
public static List<String> decouperAnnotationMajuscule
                              (List<String> crudeSegments) {
    List<String> nouveauxSegs = new LinkedList<String>();
    for (int i = 0; i < crudeSegments.size(); i++) {
        Matcher m = Pattern.compile(ANNOTATION_MAJUSCULE)
                           .matcher(crudeSegments.get(i));
        boolean isFragment = false;
        while (m.find()) {
            isFragment = true;
           for (int j = 1; j <= m.groupCount(); j++) {
                String fragmnet = m.group(j);
             if (fragmnet != null && !fragmnet.isEmpty()) {
                    nouveauxSegs.add(fragmnet);
             }
           }

        }
        if (!isFragment) {
            nouveauxSegs.add(crudeSegments
```

```
                        .get(i).toLowerCase());
            }
        }
        return nouveauxSegs;
}

/**
 * fonction segmente un hashtag est retourne une liste des
   mots
 * @param text un hashtag
 * @return liste des mots d'un hashtag après la segmentation
 */
public static List<String> segmenterHashtag(String text) {
    if (segmenteur == null) {
        segmenteur = new Segmenteur();
    }
    // Détient des segments bruts de la fraction de nombre
    List<String> crudeSegments = new ArrayList<String>();

    // Détient des jetons complètement segmentés
    List<String> nouveauxSegs = new ArrayList<String>();
    List<String> finalSegments = new ArrayList<String>();

    // Définit le jeton en minuscules
    StringBuilder tokenText = new StringBuilder(text);

    // Vérifie si mot est un hashtag
    if (tokenText.charAt(0) == '#') {
        // Supprime le caractère '#'
```

```
tokenText = tokenText.deleteCharAt(0);
if (tokenText.length() > 5) {
// Divise le texte en segments du jeton bruts
// lorsqu'il existe un nombre
// ex: "iwant2eatfood" -> ['iwant', '2', 'eatfood']
    nouveauxSegs.addAll(
                        decouperParAlphabet(tokenText)
                        );
// Divise le texte  en segments du jeton bruts
// lorsqu'il existe un caractere majuscule
// ex:"iWantFood"->['i', 'want', 'food']
    crudeSegments.addAll(
              decouperAnnotationMajuscule(
                                  nouveauxSegs
                                        )
                       );
    nouveauxSegs.clear();
// Segments de la liste des segments bruts
// eg: temp[0] = ['iwant'] -> segments = ['i','want']
    for (int i = 0; i < crudeSegments.size(); i++) {
// Si l'élément brut est un nombre,ajoutez-le
// à la liste
    if (NumberUtils.isNumber(crudeSegments.get(i))) {
          finalSegments.add(crudeSegments.get(i));
    } else {
       try {
// Si l'élément brut n'est pas un nombre, segmenter
// l'élément et ajoutez-le à la liste
         nouveauxSegs = getMaxMatch(
```

```
                                        crudeSegments.get(i)
                                            );
                    } catch (IOException ex) {
                     Logger.getLogger(Segmenter.class.getName())
                        .log(Level.SEVERE, "Erreur!"
                        + ex.getMessage(), ex);
                    }

            // Ajoute une nouvelle liste des segments aux
            // segments finaux
                    if (nouveauxSegs != null) {
                         finalSegments.addAll(nouveauxSegs);
                    } else {
                         return null;
                    }
                 }
                }
            } else {
                finalSegments.add(tokenText.toString());
            }
        }
        return finalSegments;
    }
    }
```

RÉFÉRENCES

Aggarwal, C. C. et Zhai, C. (2012). *Mining text data.* Springer Science & Business Media.

Albitar, S. (2013). *De l'usage de la sémantique dans la classification supervisée de textes : application au domaine médical.* (Thèse de doctorat). Univ. Aix-Marseille, France. Thèse de doctorat dirigée par Espinasse, Bernard et Fournier, Sébastien Informatique.

Apostolova, E. et Tomuro, N. (2014). Combining Visual and Textual Features for Information Extraction from Online Flyers. Dans *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1924–1929., Doha, Qatar. Association for Computational Linguistics.

Asur, S. et Huberman, B. A. (2010). Predicting the Future with Social Media. Dans *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, 492–499.

Audeh, B., Beaune, P. et Beigbeder, M. (2013). Expansion sémantique des requêtes pour un modèle de recherche d'information par proximité. Dans C. Soulé-Dupuy (dir.). *INFORSID 2013*, pages 83–90., Paris, France.

Becker, H., Naaman, M. et Gravano, L. (2011). Beyond Trending Topics : Real-World Event Identification on Twitter. *International AAAI Conference on Weblogs and Social Media, 11*, 438–441.

Berry, M. W., Dumais, S. T. et O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37*(4), 573–595.

Bestgen, Y. (2004). *Analyse sémantique latente et segmentation automatique des textes.* Cahiers du Cental. Presses universitaires de Louvain : Louvain-la-Neuve

Bikel, D. M., Miller, S., Schwartz, R. et Weischedel, R. (1997). Nymble : A High-performance Learning Name-finder. Dans *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, 194–201., Stroudsburg, PA, USA. Association for Computational Linguistics.

Bollegala, D., Matsuo, Y. et Ishizuka, M. (2007). Measuring Semantic Similarity Between Words Using Web Search Engines. Dans *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, 757–766., New York, NY, USA. ACM.

Boser, B. E., Guyon, I. M. et Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. Dans *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, 144–152., New York, NY, USA. ACM.

Brun, C. et Roux, C. (2014). Décomposition des «hash tags» pour l'amélioration de la classification en polarité des «tweets». Dans *Proceedings of TALN 2014*, 473–478. Association pour le Traitement Automatique des Langues.

Büttcher, S., Clarke, C. L. et Cormack, G. V. (2010). *Information retrieval : Implementing and evaluating search engines*. Mit Press.

Cano, A. E., Varga, A., Rowe, M., Ciravegna, F. et He, Y. (2013). Harnessing Linked Knowledge Sources for Topic Classification in Social Media. Dans *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT'13, 41–50., New York, NY, USA. ACM.

Chang, A. X. et Manning, C. (2012). SUTime : A library for recognizing and normalizing time expressions. Dans N. C. C. Chair), K. Choukri, T. Declerck, M. U. DoÄŸan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, et S. Piperidis (dir.). *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Chang, C.-C. et Lin, C.-J. (2011). LIBSVM : A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, *2*(3), 1–27.

Chen, Y., Balke, W.-T., Xu, J., Xu, W., Jin, P., Lin, X., Tang, T. et Hwang, E. (2014). *Web-Age Information Management : WAIM 2014 International Workshops : BigEM, HardBD, DaNoS, HRSUNE, BIDASYS, Macau, China, June 16-18, 2014, Revised Selected Papers*, volume 8597. Springer.

Dave, K. S. et Varma, V. (2012). Identifying Microblogs for Targeted Contextual Advertising. Dans *Sixth International AAAI Conference on Weblogs and Social Media*.

Ehrmann, M. (2008). *Les entités nommées, de la linguistique au TAL*. (Thèse de doctorat). Univ. Paris 7,France.

Elberrichi, Z., Rahmoun, A. et Bentaallah, M. A. (2008). Using WordNet for Text Categorization. *Int. Arab J. Inf. Technol.*, *5*(1), 16–24.

Evangelopoulos, N., Zhang, X. et Prybutok, V. R. (2012). Latent semantic analysis : five methodological recommendations. *European Journal of Information Systems*, *21*(1), 70–86.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. et Lin, C.-J. (2008). LIBLINEAR : A Library for Large Linear Classification. *J. Mach. Learn. Res.*, *9*, 1871–1874.

Faralli, S., Stilo, G. et Velardi, P. (2015). What women like : A gendered analysis of twitter users' interests based on a twixonomy. Dans *Ninth International AAAI Conference on Web and Social Media*.

Farzindar, A. et Roche, M. (2013). Les défis de l'analyse des réseaux sociaux pour le traitement automatique des langues. *Traitement Automatique des Langues*, *54*(3), 7–16.

Farzindar, A. et Roche, M. (2015). Les défis du traitement automatique du langage pour l'analyse des réseaux sociaux. *TAL*, *54*.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. (Thèse de doctorat). University of California, Irvine.

Foltz, P. W., Kintsch, W. et Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, *25*(2-3), 285–307.

Gabielkov, M. (2016). *How information propagates on Twitter ?* (Theses). Université Nice Sophia Antipolis,France.

Genc, Y., Sakamoto, Y. et Nickerson, J. V. (2011). *Discovering Context : Classifying Tweets through a Semantic Transform Based on Wikipedia*, 484–492. Springer Berlin Heidelberg : Berlin, Heidelberg

Gotti, F., Langlais, P. et Farzindar, A. (2014). Hashtag Occurrences, Layout and Translation : A Corpus-driven Analysis of Tweets Published by the Canadian Government. 2254–2261.

Han, B., Cook, P. et Baldwin, T. (2013a). Lexical Normalization for Social Media Text. *ACM Trans. Intell. Syst. Technol.*, *4*(1), 1–5.

Han, L., L. Kashyap, A., Finin, T., Mayfield, J. et Weese, J. (2013b). UMBC_EBIQUITY-CORE : Semantic Textual Similarity Systems. Dans *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1 : Proceedings of the Main Conference and the Shared Task : Semantic*

Textual Similarity, 44–52., Atlanta, Georgia, USA. Association for Computational Linguistics.

Hobbs, J. R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M. et Tyson, M. (1997). 13 FASTUS : A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. *Finite-state language processing*, p. 383.

Hu, X., Sun, N., Zhang, C. et Chua, T.-S. (2009). Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge. Dans *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, 919–928., New York, NY, USA. ACM.

Jaillet, S., Teisseire, M., Chauché, J. et Prince, V. (2003). Classification Automatique de Documents. Dans *INFORSID'03 : INFormatique des Organisations et Systèmes d'Information et de Décision*, 87–102., Nancy (France).

Joachims, T. (2002). *Learning to classify text using support vector machines : Methods, theory and algorithms*. Kluwer Academic Publishers.

Jones, K. S. A. (2004). IDF term weighting and IR research lessons. *Journal of Documentation*, *60*(5), 521–523.

Kinsella, S., Passant, A. et Breslin, J. G. (2011). *Topic Classification in Social Media Using Metadata from Hyperlinked Objects*, 201–206. Springer Berlin Heidelberg : Berlin, Heidelberg.

Kouloumpis, E., Wilson, T. et Moore, J. D. (2011). Twitter sentiment analysis : The good the bad and the omg! *Fifth International AAAI Conference on Weblogs and Social Media*, *11*, 538–541.

Landauer, T. K., Foltz, P. W. et Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2-3), 259–284.

Landauer, T. K., McNamara, D. S., Dennis, S. et Kintsch, W. (2013). *Handbook of latent semantic analysis*. Psychology Press.

Laney, D. (2001). 3D Data Management : Controlling Data Volume, Velocity, and Variety. [Online; accessed 30- July -2016]. Récupéré de `http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf`

Leaman, R., Doğan, R. I. et Lu, Z. (2013). DNorm : Disease Name Normalization with Pairwise Learning to Rank. *Bioinformatics*.

Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A. et Choudhary, A. (2011). Twitter Trending Topic Classification. Dans *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, 251–258., Washington, DC, USA. IEEE Computer Society.

Lemberger, P., Batty, M., Morel, M. et Raffaëlli, J.-L. (2015). *Big Data et machine learning : Manuel du data scientist.* Dunod.

Liu, B. (2006). *Web Data Mining : Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Secaucus, NJ, USA : Springer-Verlag New York, Inc.

Liu, B. (2010). *Sentiment analysis and subjectivity*, chapitre 26, 627–666. Chapman and Hall/CRC, (second éd.)

Liu, F., Rahimi, A., Salehi, B., Choi, M., Tan, P. et Duong, L. (2014). Automatic Identification of Expressions of Locations in Tweet Messages using Conditional Random Fields. Dans *Proceedings of the Australasian Language Technology Association Workshop 2014*, 171–176., Melbourne, Australia.

Melville, P. et Sindhwani, V. (2009). Active Dual Supervision : Reducing the Cost of Annotating Examples and Features. Dans *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, 49–57., Stroudsburg, PA, USA. Association for Computational Linguistics.

Mendoza, M., Poblete, B. et Castillo, C. (2010). Twitter Under Crisis : Can We Trust What We RT ? Dans *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, 71–79., New York, NY, USA. ACM.

Metzler, D., Dumais, S. et Meek, C. (2007). *Similarity Measures for Short Segments of Text*, 16–27. Springer Berlin Heidelberg : Berlin, Heidelberg

Michelson, M. et Macskassy, S. A. (2010). Discovering Users' Topics of Interest on Twitter : A First Look. Dans *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, AND '10, 73–80., New York, NY, USA. ACM.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. et Miller, K. J. (1990). Introduction to WordNet : An On-line Lexical Database*. *International Journal of Lexicography*, *3*(4), 235–244.

Montejo-Ráez, A., Martínez-Cámara, E., Martín-Valdivia, M. T. et Ureña-López, L. A. (2014). Ranked wordnet graph for sentiment polarity classification in twitter. *Computer Speech Language*, *28*(1), 93–107.

Morstatter, F., Pfeffer, J. et Liu, H. (2014). When is It Biased? : Assessing the Representativeness of Twitter's Streaming API. Dans *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, 555–556., New York, NY, USA. ACM.

Navigli, R. (2009). Word Sense Disambiguation : A Survey. *ACM Comput. Surv.*, *41*(2), 1–10.

Navigli, R. et Ponzetto, S. P. (2012). BabelNet : The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, *193*, 217–250.

Ohta, T., Tateisi, Y. et Kim, J.-D. (2002). The GENIA Corpus : An Annotated Research Abstract Corpus in Molecular Biology Domain. Dans *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, 82–86., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Phan, X.-H., Nguyen, L.-M. et Horiguchi, S. (2008). Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. Dans *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, 91–100., New York, NY, USA. ACM.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

Qu, Y., Huang, C., Zhang, P. et Zhang, J. (2011). Microblogging After a Major Disaster in China : A Case Study of the 2010 Yushu Earthquake. Dans *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, 25–34., New York, NY, USA. ACM.

Rosenberg, D. S., Klein, D. et Taskar, B. (2012). Mixture-of-Parents Maximum Entropy Markov Models. *CoRR*, *abs/1206.5261*.

Rosoor, B., Sebag, L., Bringay, S., Poncelet, P. et Roche, M. (2011). Quand un tweet détecte une catastrophe naturelle... Dans *VSST'10 : Veille Stratégique Scientifique et Technologique*, 1–15., Toulouse, France.

Sahami, M. et Heilman, T. D. (2006). A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. Dans *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, 377–386., New York, NY, USA. ACM.

Saif, H., He, Y. et Alani, H. (2012). *Semantic Sentiment Analysis of Twitter*, 508–524. Springer Berlin Heidelberg : Berlin, Heidelberg

Sanderson, M. (2010). Christopher D. Manning, Prabhakar Raghavan, Hinrich SchÃtze, Introduction to Information Retrieval, Cambridge University Press. 2008. ISBN-13 978-0-521-86571-5, xxi + 482 pages. *Natural Language Engineering*, *16*, 100–103.

Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D. et Sperling, J. (2009). TwitterStand : News in Tweets. Dans *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, 42–51., New York, NY, USA. ACM.

Sarawagi, S. (2008). Extraction d'Information. *Trouvé. bases de données Trends*, *1*, 261–377.

Shalev-Shwartz, S., Singer, Y. et Srebro, N. (2007). Pegasos : Primal Estimated sub-GrAdient SOlver for SVM. Dans *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, 807–814., New York, NY, USA. ACM.

Shuyo, N. (2010). Language Detection Library for Java. [Online; accessed 30-July -2016]. Récupéré de `http://code.google.com/p/language-detection/`

Silva, C. et Ribeiro, B. (2009). *Inductive inference for large scale text classification : kernel approaches and techniques*, volume 255. Springer.

Soderland, S. (1999). Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, *34*(1), 233–272.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H. et Demirbas, M. (2010). Short Text Classification in Twitter to Improve Information Filtering. Dans *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, 841–842., New York, NY, USA. ACM.

Wang, X., Wei, F., Liu, X., Zhou, M. et Zhang, M. (2011). Topic Sentiment Analysis in Twitter : A Graph-based Hashtag Sentiment Classification Approach. Dans *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, 1031–1040., New York, NY, USA. ACM.

Wermter, J., Tomanek, K. et Hahn, U. (2009). High-performance gene name normalization with GeNo. *Bioinformatics*, *25*(6), 815–821.

Wu, Z. et Palmer, M. (1994). Verbs Semantics and Lexical Selection. Dans *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, 133–138., Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhu, X. et Goldberg, A. B. (2009). Introduction to Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *3*(1), 1–130.