République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abdelhamid Ibn Badis Mostaganem

Faculté des Sciences Exactes et de l'Informatique

Département de Mathématique et de l'Informatique

Mémoire de Master

en Informatique

Spécialité : *Ingénierie des Systèmes d'Informations*

Thème

Sentiment Communities Analysis on Social Networks

Encadré par

— Dr, Mokeddem Sid Ahmed

Réalisé par

— Mezarja Fouzia

2017/2018

Table des matières

Table des figures

Liste des tableaux

Résumé

La fouille de texte offre aux particuliers et aux entreprises un moyen d'exploiter d'une grande quantité d'informations. Elle a des valeurs commerciales très élevées. Le dernier décompte déclare qu'il se trouve plus de dix entreprises de haute technologie offrant des produits pour l'exploration de texte. La fouille de texte instruit aux développeurs les problèmes de gestion de texte non structuré et décrit comment créer des outils pour l'exploration de texte à l'aide de méthodes statistiques standard issues de l'intelligence artificielle et de la recherche opérationnelle. Ces outils peuvent être utilisés dans divers domaines, notamment le droit, les affaires et la médecine. Les sujets clés abordés comprennent l'extraction d'information, le regroupement, la catégorisation de texte, la recherche sur le Web, le résumé et les systèmes de requête en langage naturel. Ce mémoire présente des méthodes qui transforme des documents texte non structurés en une forme intermédiaire pour extraction des informations depuis les données du documents et aussi depuis Les données textuelles dans les médias sociaux .

Mots clés : Fouille de texte, Réseaux sociaux, Fouille de données ...

Introduction générale

La fouille de texte est une technologie émergente ambitionne généralement à découvrir des modèles de prédictions dans les données textuelles pour l'extraction des connaissances avantageuses et non prosaïques à partir de documents texte non structurés.

Évidemment, le prcessus le plus spontané de stocker des informations dans le texte. La fouille de texte a un potentiel plus orgueilleux que la fouille de données, En fait, une étude récente déclare que 80% des informations d'une entreprise sont contenues dans des documents texte. La fouille de texte est également une tâche beaucoup plus complexe (que la fouille de données) parce qu'elle traite des données de texte qui sont intrinsèquement non structurées et floues. Le texte et les documents peuvent être transformés en valeurs mesurées, telles que la présence ou l'absence de mots. L'exploration de texte est un domaine pluridisciplinaire qui inclut la recherche d'information, l'analyse de texte, l'extraction d'information, le regroupement, la catégorisation, la visualisation, la technologie de base de données, l'apprentissage automatique et l'exploration de données. Ce mémoire présente un cadre général pour la fouille de texte tel que le nettoyage de texte qui transforme les documents texte de forme libre en une forme intermédiaire et la distillation des connaissances qui déduit des modèles ou des connaissances de la forme intermédiaire. Le reste de ses papiers est organisé comme ceci. Ce mémoire présente quelques applications de la fouille de texte et le traitement de langage naturel. Parmi ces applications on présente la problématique de l'analyse des sentiments dans les réseaux sociaux. Par conséquent, la suite de ce mémoire est structurée comme suit : le chapitre 1 présente le processus de fouille de texte et un exemple d'application de synthèse de document, le chapitre 2 montre les méthodes utilisés pour la prédiction dans la fouille de texte, ensuite leurs applications dans les documents dans les réseaux sociaux. Enfin, récapitulant l'ensemble des chapitre avec l'idée principale du projet.

Chapitre 1

Fouille de Texte

1.1 Introduction

Avec l'air de la technologie ou toute information est stocké dans des bases de données par exemple l'achat de produit en ligne étant donné que toutes transactions en papier est devenu maintenant sous forme numérique. L'exploration de ces données est devenue une tendance émergente qui aide à produire des modèles pour améliorer la qualité de vie.

Les techniques de fouilles de données sont très développées dans ce domaine stratégique, elles s'attendent un format de données adéquat ce qui oblige une étape de transformation des données non structurées sous forme numérique (exemple : texte ; page web ; image et vidéo). Les experts en science de données donnent aux données collectés une signification à partir d'expériences passées [33].

1.2 Données Structurées

Les méthodes de fouille de données adoptent seulement des données structurées par contre les textes ne sont pas sous un format structuré, il ont besoins d'être préparer par un processus de transformation de données avant que toute méthode d'apprentissage puisse être appliquée. Le format des données structurées est présenté sous forme d'un tableau ou plus exact une matrice. La tâche de la collecte de données consiste à remplir les cellules formées par des lignes et des colonnes croisées d'une manière uniforme. Une ligne est un spécimen d'une expérience désuet par exemple dans le domaine d'éducation, il peut s'agir d'un seul étudiant. Une colonne est un exemple d'une mesure sur l'étudient (voir Table 1.1). On peut facilement ajouter des exemples, où chaque exemple est mesuré en utilisant les mêmes attributs mais il est un peut difficile d'ajouter une colonne parce qu'il faut vérifier tous les exemples précédent et appliquer cette nouvelle mesure pour chacun (voir Figure 1.1) [33]. Deux types de données sont :

— **Numérique Ordonner :** sont des attributs qui ont une valeur numérique permet de faire la comparaison entre eux (supérieur à, inférieure à). Par exemple le poids et le revenu.

— **Catégorique :** les attributs catégoriels sont des codes numériques non ordonnés qui

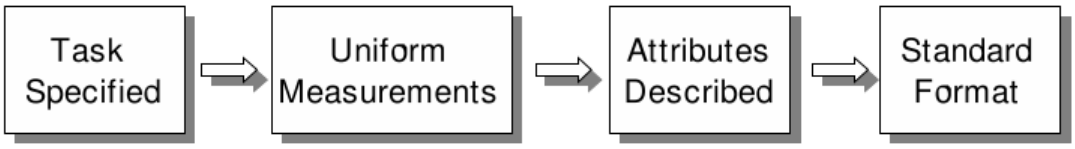

FIGURE 1.1 – Processus de structuration des données.

ont une définition dans un livre de codes. L'attribut catégoriel le plus commun est mesuré par une valeur booléenne (vrai ou faux) ou par exemple le sexe est mesuré par Homme ou Femme. Le tableau 1.1 illustre un exemple de données structurée.

Code	***Nom***	***Prénom***	***Age***	***Sexe***
639	Mohamed	Malik	23	H
949	Fatima	Smail	22	F

TABLE 1.1 – Exemple de données structurée.

1.3 Texte : Données non structurées ?

La présentation de données de la fouille de données classique et la fouille de texte est très différente. La représentation classique assimile les données sous forme d'un tableau où les mots sont des attributs et les documents sont des exemples, et les méthode d'extraction de textes assimilent les données sous forme XML ou JSON. De toute évidence le texte soit différent de chiffre. Ces représentations de texte sont similaire à celle de données, elles ne prennent pas en considération les propriétés spécifiques du texte telles que les concepts de la grammaire ou la signification des mots, seulement les informations de niveau bas telles que l'occurrence d'un nombre dans un texte ou leur apparence (s'il apparaît dans le texte on l'indique avec 1 sinon avec 0 (voir tableau 1.2)) après cette structuration de données les méthodes d'apprentissage sont appliquées [33].

1.4 Fouille de texte : pour quel type d'applications ?

Les méthode les plus étudiées de fouille de texte sont la clustering et la classification , Étant donné un échantillon antérieure étiqueté l'objectif est pour chaque document, nous devons deviner son étiquète à partir de l'expérience caché dans l'échantillon d'apprentissage. Ce processus est appelé prédiction ou classification.

méthode	***machine***	***du***	***niveau***
1	1	0	1
0	1	0	0
1	1	1	0

TABLE 1.2 – Tableau binaire de mots dans des documents

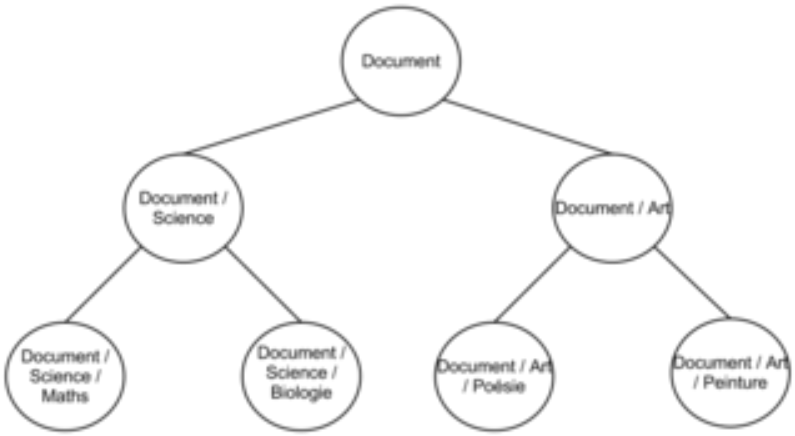

FIGURE 1.2 – classification de documents

Le concept de clustering concerne les documents qui ne sont pas clairement étiquetées. Notre tâche consiste à regrouper les documents que nous pouvons associer des étiquettes communes. La mesure de la similarité entre les documents est fondamentale pour la plupart des formes d'analyse de documents. Ces méthodes que nous avons assimilé auparavant ne font pas l'accent sur l'analyse linguistique, elles se base seulement sur Les relations statistiques et associatives [33].

1.4.1 Classification de documents

Dés que les données sont transformées sous la forme d'une matrice numérique, les méthode de fouille de données sont applicables. Les documents sont combiner en dossier, pour chaque dossier on associe un thème, lorsqu'un nouveau document provient, l'objectif est de le classer dans le dossier qui lui convient. Par exemple, nous pourrions avoir un dossier pour les documents art, poésie ou science et nous voulons affecter de nouveaux documents dans le bon dossier (Voir Figure 1.2) [33]. Notre objectif ultime est la prédiction, en commençant par un échantillon d'exemples antérieures vers de nouveaux exemples. Le programme d'apprentissage étudie les documents et trouve quelques règles importantes qui donneront des réponses correctes sur de nouveaux exemples. Mais comment savons nous que la prédiction de nouveaux exemples est réussite ?. Ces nouveaux exemples sont utilisés uniquement pour l'évaluation.

La solution idéale est de calculer la mesure de l'erreur pour l'attribution des exemples, nous pouvons facilement déterminer si la réponse d'un programme est bonne ou mauvaise. Les mesures classiques d'exactitude seront applicables, mais toutes les erreurs ne seront pas évaluées également. C'est pourquoi les mesures de précision telles que «rappel» et «précision» sont particulièrement importantes pour l'analyse documentaire.

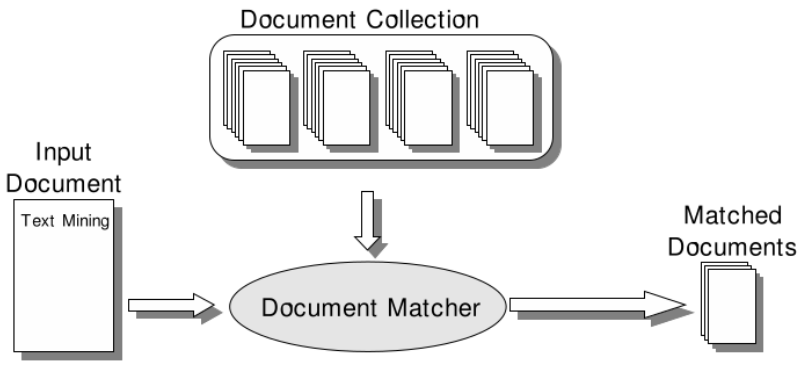

FIGURE 1.3 – Recherche d'Information

1.4.2 Recherche d'Information

Le problème de Recherche d'Information est connu plus fréquemment dans les documents en ligne, ou les documents sont stockés nous voulons récupérer à partir de ces documents des documents correspond à des mots clés lors d'une recherche (cas de moteur de recherche).

Un concept de base pour la recherche d'information est de mesurer la similarité : une comparaison est faite entre deux documents, mesurant à quel point les documents sont similaires. A titre de comparaison, même un petit nombre de mots saisis dans un moteur de recherche peut être considéré.

La matrice numérique est souvent utilisée dans cette tâche, le nouveau document est équivalent à une nouvelle ligne. La nouvelle ligne est comparée à toutes les autres lignes des documents existants dans le web, et les lignes les plus similaires et leurs documents associés sont les réponses (voir Figure 1.3)[33].

1.4.3 Clusetring et organisation des documents

Pour la catégorisation (classification) de texte, nous avons vu que l'objectif était de placer de nouveaux documents dans les dossiers appropriés. Ces dossiers ont été créés par un expert qui connaissait la structure de documents. Et si nous avons une collection de documents sans structure connue ?

Étant donné une collection de documents ,l'objectif général est de trouver un ensemble de dossiers de sorte que chacun contient des documents similaires. Le processus de regroupement équivaut à attribuer les étiquettes nécessaires à la catégorisation du texte. En terme de modèle de matrice numérique (tableau), le processus de regroupement consiste à ajouter une colonne correspondant à des étiquettes vrai ou faux pour les nouveau exemples. Le nombre d'étiquettes sera déterminer par l'algorithme de classification (voir Figure 1.4) [33].

1.4.4 Extraction de l'information

La représentation de données considère l'information en terme de mots, c'est une normalisation élémentaire qui est à la base de nombreuses applications. D'après cette représentation, les mesures sont peu profondes, elles sont basé seulement sur la présence et l'absence d'un

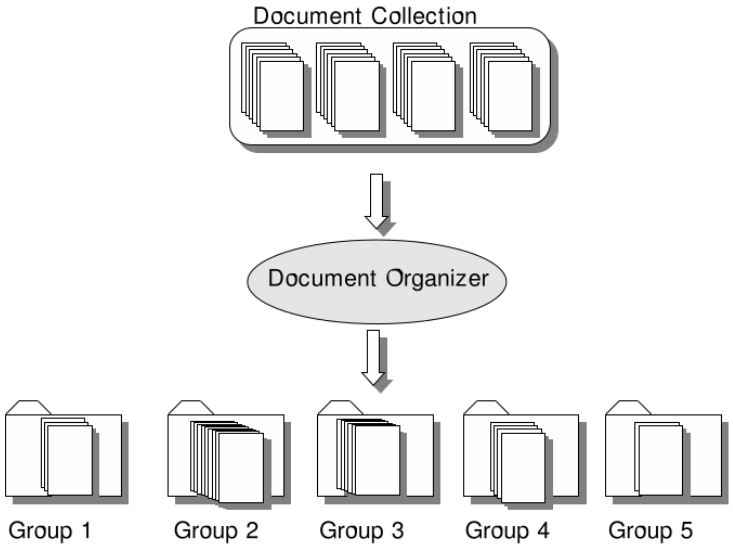

FIGURE 1.4 – Organisation des documents dans des groupes

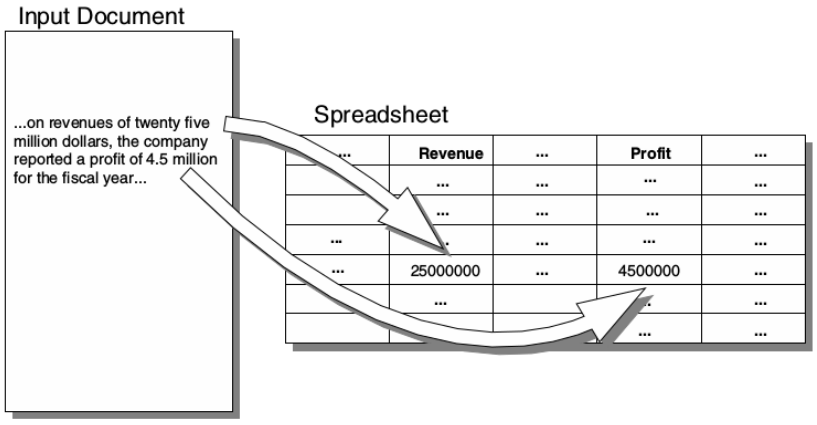

FIGURE 1.5 – Extraction des informations d'un document

mot, à l'opposé il est possible d'élaborer une représentation plus large et complexe avec des valeurs réelles. Une représentation très intuitive qui mesure l'occurrence simple des mots.

L'extraction d'informations tente prendre un document non structuré et de remplir automatiquement les valeurs dans d'un tableau informatif, L'attribut mesuré n'aura pas de position fixe dans le texte et ne pourra pas être décrit de la même manière dans différents documents. Les colonnes ne sont pas seulement des mots mais peuvent être des concepts de niveau supérieur trouvés par le processus d'extraction d'informations (voir Figure 1.5).

1.5 De l'information textuelle à la representation vectorielle

Pour extraire du texte, nous devons d'abord le transformer en une représentation structurée pour que les procédures d'exploration de données peuvent l'utiliser. Comme mentionné auparavant la représentation la plus utilisée est un tableau où on génère les attributs d'un

texte et nous avons une procédure soigneusement organisée pour remplir les cellules du tableau. Tout d'abord, bien sûr, nous devons déterminer la nature des colonnes (c'est-à-dire, les caractéristiques ou les attributs). Certaines caractéristiques utiles sont faciles à obtenir (par exemple, un mot tel qu'il apparaît dans le texte) et certaines sont beaucoup plus difficiles (par exemple, la fonction grammaticale d'un mot dans une phrase telle qu'un sujet, un objet, etc.). Ultérieurement, nous verrons comment obtenir les caractéristiques couramment générés à partir du texte.

1.5.1 Standardisation des documents

Dés que les documents sont amassé, ils s'en trouver dans une variété de format. Par exemple certains documents ont été généré par un traitement de texte avec son propre format propriétaire, d'autres ont été généré à l'aide d'un simple éditeur de texte et aussi certains sont scannés et stockés sous forme d'images. Nettement si nous voulons traiter tous ces documents, nous devons les convertir en un format standard XML ou JSON.

Communauté de traitement de texte, a adopté XML (Extensible Markup Language) comme format d'échange standard, Brièvement, XML est un moyen standard d'insérer des balises sur un texte pour en identifier les parties. Bien que les balises puissent être imbriquées dans d'autres balises à une profondeur arbitraire,les balises viennent en début et en fin de paire. Ils sont entourés d'équerres et l'étiquette de fin comporte une barre oblique immédiatement après l'équerre d'ouverture. Dans un document, il peut y avoir plusieurs autres balises pour marquer les sections du document : $< DOC >, < DATE >$ $, < SUBJECT >, < TOPIC >, < TEXT >$. $< HEADLINE >$ *et* $< BODY >$. De plus, pour la classification de texte ou le regroupement, on utilise $< SUBJECT >$ $, < HEADLINE >$ et pour la génération des caractéristiques à partir d'une section on utilise $< TOPIC >$.

De nombreux logiciels de traitement de texte permettent d'enregistrer des documents au format XML pour convertir des documents existants sans avoir à les traiter manuellement.

Certains systèmes OCR (reconnaissance optique de caractères) peuvent convertir les documents encodés tant que d'image, mais ils peuvent introduire des erreurs dans le texte et doivent être utilisés avec précaution.

JSON(Javascript Object Notation), est un langage d'échange de données qui soit lisible par l'utilisateur et facile à analyser et utiliser par les ordinateurs. JSON est adapté aux applications JavaScript, fournissant ainsi des gains de performance significatifs par rapport à XML, ce qui nécessite des bibliothèques complémentaire pour extraire des données à partir de modèle d'objet des documents (Document Object Model DOM). JSON est estimé à analyser cent fois plus vite que XML dans les navigateurs modernes. Mais malgré ces performances remarquables JSON incluse le manque du support d'espace de noms, manque de validation d'entrée et d'extensibilité. Chaque objet est un espace de noms et son clés est indépendant de tous les autre objets même exclusif d'imbrication.

$$\{"Nom" : "Ali", "Prenom" : "Mohamed", "Age" : 23\} \tag{1.1}$$

L'objectif de standardisation des documents est que les logiciels d'extraction de données

ont besoin de lire les données dans un seul format, et non dans les nombreux formats différents à l'origine, Pour récolter des informations à partir d'un document [33].

1.5.2 Tokenisation

Les documents sont en format XML et ils sont prêts pour identifier les caractéristiques utiles pour l'extraction d'information.

La première étape consiste à découper les caractères de texte en mots «Token». Chaque token est considéré en tant que *type*, le nombre de token et plus grand que le nombre de type. À titre d'exemple dans cette phrase «nous ne savons pas quoi faire de cette courte vie, et pourtant nous en désirons une autre qui soit éternelle» y en a deux token épelés 'nous' mais ce sont les deux instance de type 'nous' qui apparaît deux fois dans la phrase.

Parfois quelques caractères sont des délimiteurs d'un token et parfois non selon leurs position dans le texte. Les espaces, les tabulations et les retours à la ligne. Une virgule ou un point entre deux nombres est normalement considéré toujours comme un token et pas un délimiteur (Par exemple 5,8 ou 3,6). Un trait d'union entre deux nombres peut être un symbole de soustraction ou un séparateur (par exemple, 555-1212). Le processus de tokenisation dépend de la langue ou chaque langue a ses spécificités par rapport aux délimiteurs et mots [33].

1.5.3 Lemmatisation

Une fois qu'un flux de caractères est segmenté en séquence de token, la prochaine étape consiste à convertir les tokens en format standard elle se fait par un processus appelé lemmatisation qui a pour objectif de réduire le nombre de type distinct dans un texte et d'augmenter la fréquence d'occurrence de certains types individuels. Par exemple le mot 'présenter' et le mot 'présentation' sont deux instances de type 'présenter'.

lemmatisation

Dans les langues, les mots apparaissent dans plusieurs forme, par exemple 'livre' et 'livres' sont deux formes de même mot, il est avantageux d'éliminer cette variation avant de commencer le traitement. Lorsque la normalisation se limite à la régularisation des variantes grammaticales telles que singulier / pluriel et présent / passé, le processus est appelé «lemmatisation».

Dans la terminologie linguistique, on appelle cela «analyse morphologique». Dans certaines langues comme l'espagnol analyse morphologique est relativement simple par contre en anglais est un peu compliquée à cause de nombreuses formes de mots irréguliers et une orthographe non intuitive.

Il n'y a pas de règle précise pour faire cette normalisation à titre d'exemple :

En Allemand : 'angegeben' (participe passé de verbe déclarer) : 'angeben'.

En anglais : 'sought' : 'seek', 'rebelled' : 'rebel', 'belled' : 'bell'.

En français : acheté, achat, achetée : acheter.

Main sans connaître les informations grammaticales ça peut produire quelques erreurs en raison de l'ambiguïté, comme exemple en anglais 'bored' adjective 'he is bored' ou la conjugaison du verbe 'bore' en passé ou le passé du verbe 'bear'.

Enfin, tandis que la lemmatisation n'est pas parfaite, elle identifiera correctement un nombre significatif de types [33].

Extraction de racine

l'objectif de cette opération est d'atteindre une forme de racine sans préfixes et suffixes flexionnels ou dérivationnels. Par exemple, "normalisation" est réduite à la norme "norme". Le résultat final d'une telle radicalisation agressive est de réduire de façon très drastique le nombre de types dans une collection de textes. De plus, des mots ayant la même signification fondamentale sont regroupés.

1.5.4 Représentation vectorielle pour la prédiction

Les caractéristiques d'un document sont les mots ou les tokens qu'il contient. En vue de résoudre le problème de catégorisation du documents, il suffit de choisir les caractéristiques qui représente les tokens de chaque documents les plus fréquents sans aucune analyse approfondie du contenu linguistique des documents.

L'ensemble collectif des caractéristiques est couramment appelé un dictionnaire. Les tokens ou les mots du dictionnaire constituent la base de la création d'un tableau de données numériques (matrice numérique) correspondant à la collection de documents qui influence les méthodes d'apprentissage. Une ligne représente un document, une colonne représente une caractéristique (un mot) et une cellule est une mesure d'une caractéristique (correspondant à la colonne) d'un document (correspondant à la colonne). Les mesures de cellules sont binaires correspondant à la présence ou l'absence d'un mot dans le document.

Si une méthode d'apprentissage peut traiter les dimensions élevées d'un tel dictionnaire global, ce simple modèle de données peut être très efficace. La vérification des mots est simple car nous ne vérifions pas chaque mot du dictionnaire. Nous construisons une table de hachage des mots du dictionnaire et voyons si les mots du document sont dans la table de hachage. De grands échantillons de documents numériques sont facilement disponibles. Cela nous donne confiance que de nombreuses variations et combinaisons de mots apparaîtront dans l'échantillon.Cette attente suggère de consacrer moins de temps à la préparation des données pour rechercher des mots similaires ou supprimer des mots faibles.

Mais, dans de nombreuses circonstances, nous pouvons vouloir travailler avec un dictionnaire plus petit. L'échantillon peut être relativement petit, ou un dictionnaire volumineux peut être lourd. Dans de tels cas, nous pourrions essayer de réduire la taille du dictionnaire par diverses transformations d'un dictionnaire et de ses mots constitutifs. Selon la méthode d'apprentissage, plusieurs de ces transformations peuvent améliorer les performances prédictives (Voir Figure 1.6).

La prédiction nécessite une une colonne supplémentaire pour la réponse correcte (ou classe) pour chaque document. Lors de la préparation des données pour une méthode d'apprentissage, ces informations seront disponibles à partir des étiquettes de document. Nos

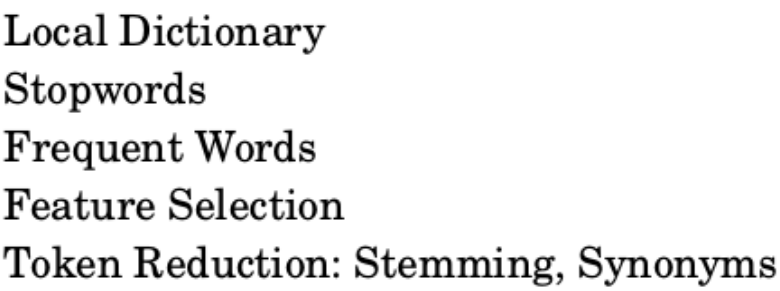

FIGURE 1.6 – Méthodes de transformations d'un dictionnaire.

étiquettes sont généralement binaires, et la plus petite classe est presque toujours la plus intéressante. Au lieu de générer un dictionnaire global pour les deux classes, nous pouvons considérer seulement les mots trouvés dans la classe que nous essayons de prédire. Si cette classe est beaucoup plus petite que la classe négative, ce qui est typique, un tel dictionnaire local sera beaucoup plus petit que le dictionnaire global.

Une autre réduction évidente de la taille du dictionnaire consiste à compiler une liste de mots vides et à les supprimer du dictionnaire. Ce sont des mots qui n'ont presque jamais de capacité prédictive, tels que 'a', 'ça' et 'eux'. Ces mots communs peuvent être supprimés avant le processus de génération de caractéristiques, mais il est plus efficace de générer les fonctionnalités en premier, d'appliquer toutes les autres transformations et, à la toute dernière étape, de rejeter celles correspondant aux mots vides.

Les informations sur la fréquence de mots peuvent être très utiles pour réduire la taille du dictionnaire et peuvent parfois améliorer les performances prédictives pour certaines méthodes. Les mots les plus fréquents sont souvent des mots vides et peuvent être supprimés. Les mots les plus fréquemment utilisés sont souvent les mots importants qui doivent rester dans un dictionnaire local. Les mots très rares sont souvent des fautes de frappe et peuvent également être rejetés.

Au lieu de placer tous les mots possibles dans le dictionnaire, nous pourrions suivre le chemin du dictionnaire imprimé et éviter de stocker toutes les variations du même mot. La raison en est que toutes les variantes se réfèrent réellement au même concept. Il n'y a pas besoin de singulier et pluriel. De nombreux verbes peuvent être stockés sous leur forme de l'indicatif. En étendant le concept, nous pouvons également mapper des synonymes sur le même token.

Nous avons habitué à représenter les données texte dans un tableau binaire indiquant l'existance ou l'absence de mot dans une collection de documents. Pour obtenir la meilleure précision prédictive, nous pourrions envisager d'autres transformations à partir de cette représentation. Les transformations possibles qui peuvent améliorer les performances prédictives tels que : la fréquence réelle d'occurrence d'un mot dans un document, si un mot apparaît dix fois dans un document, ce compte est entré dans la cellule. Nous avons toutes les informations d'une représentation binaire, et nous avons des informations supplémentaires pour contraster avec d'autres documents. Pour certaines méthodes d'apprentissage, le résultat est légèrement meilleur. Cela peut également conduire à des solutions plus compactes car il inclut le même espace de solution que le modèle de données binaires, mais les informations de fréquence supplémentaires peuvent aboutir à une solution plus simple. Cela est

particulièrement vrai pour certaines méthodes d'apprentissage dont les solutions n'utilisent qu'un petit sous-ensemble des mots du dictionnaire.

Dans l'ensemble, les fréquences sont utiles pour la prédiction mais ajoutent de la complexité aux solutions proposées. Un compromis qui fonctionne assez bien est d'avoir un système à trois valeurs pour les entrées de cellules : un ou zéro comme dans la représentation binaire, avec la possibilité supplémentaire d'un 2 (un mot apparaît 2 fois ou plus dans un document), cette méthode ajoute des informations de fréquence sans ajouter beaucoup de complexité au modèle.

L'autre méthode consiste à compter la fréquence d'un mot dans un document est de modifier le compte par l'importance perçue de ce mot. La formulation **tf-idf** bien connue a été utilisée pour calculer des pondérations ou des scores pour des mots. Encore une fois, les valeurs seront des nombres positifs afin que nous capturons la présence ou l'absence du mot dans un document. Dans l'équation 1.2, nous voyons que le poids tf-idf assigné au mot j est le terme fréquence (c'est-à-dire, le nombre de mots) modifié par un facteur d'échelle pour l'importance du mot. Le facteur d'échelle est appelé fréquence de document inverse, donnée dans l'équation 1.3. Il vérifie simplement le nombre de documents contenant le mot j (c'est-à-dire, $df(j)$) et inverse la mise à l'échelle. Ainsi, lorsqu'un mot apparaît dans de nombreux documents, il est considéré comme non important et l'échelle est abaissée, peut-être proche de zéro. Lorsque le mot est relativement unique et apparaît dans quelques documents, le facteur d'échelle effectue un zoom vers le haut parce qu'il semble important.

$$tf - idf(j) = tf(j) \times idf(j) \tag{1.2}$$

$$idf(j) = log\left(\frac{N}{df(j)}\right) \tag{1.3}$$

Tous ces modèles de données sont des variations modestes du modèle binaire de base pour la présence ou l'absence de mots. Lesquelles des transformations de données sont les meilleures ? Nous ne donnerons pas une réponse universelle. L'expérience a montré que la meilleure précision de prédiction dépend de l'appariement d'une de ces variantes à une méthode d'apprentissage spécifique. La meilleure variante pour une méthode peut ne pas être celle d'une autre méthode. Est-il nécessaire de tester toutes les variations avec toutes les méthodes ? Lorsque nous décrivons les méthodes d'apprentissage, nous allons donner des directives pour les méthodes individuelles basées sur l'expérience de recherche générale. De plus, certaines méthodes ont une relation naturelle avec l'une de ces représentations, ce qui en ferait à elles seules l'approche préférée pour représenter les données.

Bien que nous décrivons les données comme remplissant un tableau, nous nous attendons à ce que la plupart des cellules soient nulles. La plupart des documents contiennent un petit sous-ensemble des mots du dictionnaire. Dans le cas de la classification de texte, un corpus de texte peut avoir des milliers de types de mots. Cependant, chaque document individuel ne contient que quelques centaines de mots uniques. Ainsi, dans le tableau, presque toutes les entrées de ce document seront nulles. Plutôt que de stocker tous les zéros, il est préférable de représenter le tableau comme un ensemble de vecteurs clairsemés, où une ligne est représentée par une liste de paires, un élément de la paire étant un numéro de colonne et l'autre étant la caractéristique non nulle correspondante. valeur. En ne stockant pas les zéros, les économies

0	12	6	0	$\Longrightarrow$(2,12)(3,6)
0	0	9	2	$\Longrightarrow$(3,9) (4,2)
0	1	5	3	$\Longrightarrow$(2,1) (3,5) (3,4)

TABLE 1.3 – Transformation d'un tableau à un vecteur clairsemé.

en mémoire peuvent être immenses. Les programmes de traitement peuvent être facilement adaptés pour gérer ce format [33].

Des étiquettes pour les bonnes réponses

Pour la prédiction, une colonne supplémentaire doit être ajoutée au tableau. Cette dernière colonne, contenant l'étiquette, ne semble pas différente des autres. C'est un un ou zéro indiquant que la bonne réponse est soit vrai soit faux. Quel est le label ? Traditionnellement, cette étiquette a été un sujet pour indexer le document. Les histoires sportives ou financières sont des exemples de sujets. Nous ne faisons pas cette distinction sémantique. Toute réponse qui peut être mesurée comme vraie ou fausse est acceptable. Ce pourrait être un sujet ou une catégorie. Tant que les réponses sont étiquetées correctement par rapport au concept, le format est acceptable. Bien sûr, cela ne signifie pas que le problème peut facilement être résolu. Dans le format vectoriel clairsemé, les étiquettes sont ajoutées à chaque vecteur séparément soit comme une (classe positive) soit comme une classe zéro (classe négative).

Marquage des Parts of Speech

Le texte est segmenté en token et en phrase donc ce qu'on doit faire avec le texte ?. On peut procéder directement à la génération de caractéristiques utiles pour effectuer l'extraction de données. Si l'objectif est plus spécifique, par exemple en reconnaissant les noms des personnes, des lieux et des organisations, il est généralement souhaitable d'effectuer des analyses linguistiques supplémentaires du texte et d'extraire des caractéristiques plus sophistiquées. À cette fin, la prochaine étape logique consiste à déterminer les Parts of speech (POS) de chaque token.

Dans toutes les langues les mots sont organisés en classes grammaticales ou POS, presque toutes les langues auront au moins les catégories que nous appellerions les noms et les verbes, le nombre exact de catégories se diffère d'une langue à une autre.

En anglais, certaines analyses peuvent utilisent six à sept catégories. La grammaire anglaise auraient au moins un nom, un verbe, un adjectif, un adverbe, une préposition et une conjonction. Dans les dictionnaires on trouve aussi les POS. Dans l'exemple précédent le mot 'bore' pourrait être un nom, un verbe au présent ou un verbe au passé [33].

Désambiguïsation du sens du mot

Les sens du mots en générale sont ambigus dès qu'ils sont isolés de leurs POS, en retournant à l'exemple précédent du mot en anglais 'bore' on peut pas déterminer son signification sans le référer à une phrase 'He is a bore' .

Les dictionnaires ordinaire la signification d'un mot n'est pas applicables par un programme informatique pour la désambiguïsation. Un grand projet axé sur la signification d'un mot est le dictionnaire **WordNet**, il ne fournit pas à lui seul un algorithme pour sélectionner une signification particulière pour un mot en contexte. Malgré un travail important sur une longue période de temps, il n'y a pas d'algorithmes qui puissent complètement désambiguïse un texte.

Reconnaissance d'expression

Une fois le texte est divisé en token, l'étape suivante consiste à regrouper ces token en une entité appelée phrase, cette dernière est utile pour créer une analyse partielle et pour identifier les entités nommées apparaissant. Il existe des corpus standard et des ensembles de tests pour développer et évaluer des systèmes de reconnaissance de phrases, ces systèmes sont censés balayer un texte et marquer les débuts et les fins des phrases dont les plus importantes sont les expressions nominales, les expressions verbales et les phrases propositionnelles il existe plusieurs conventions pour marquer un mot, on dénote un mot au sein d'une phrase avec I un mot au début d'une phrase adjacente à une autre phrase avec B et un mot à l'extérieur de toute phrase avec O. Les étiquettes I et B peuvent être étendues avec un code pour le type de phrase : I-NP, B-NP, I-VP, B-VP.

Reconnaissance d'entité nommée

La détection des entités nommées, elle repose en particulier sur la recherche de syntagmes nominaux, est la reconnaissance de types particuliers de locution nominale propre tells que des personnes, des organisations, des lieux et parfois de l'argent, des dates, des heures et des pourcentages. Cela ressemble beaucoup au problème de la reconnaissance de la phrase, le même type de modèle de codage de jetons peut être utilisé (personne B, localisation B, personne I, etc.). Cependant, le robuste problème comporte l'attribution de classe correcte pour chaque token.

Analyse syntaxique

Le traitement de texte le plus sophistiqué est accomplir une analyse complète pour une phrase, Nous entendons par là que chaque mot est assemblé à une structure unique, couramment un arbre. L'analyse nous aide à détecter la relation de chaque mot dans la phrase avec les autres mot, aussi sa fonction dans la phrase (sujet,objet. . .). Se trouve énormément types d'analyse, chacun associé à une théorie linguistique du langage.nous pouvons restreindre l'attention aux analyses dites «sans contexte». Cette analyse de ce type ressemble à un arbre de nœuds dans lequel les nœuds de feuilles sont les mots d'une phrase, les phrases dans lesquelles les mots sont groupés sont des nœuds internes, et il y a un nœud supérieur à la racine de l'arbre, qui a généralement l'étiquette S. L'objectif de cette analyse est de fournir des informations sur l'identification de phrase.

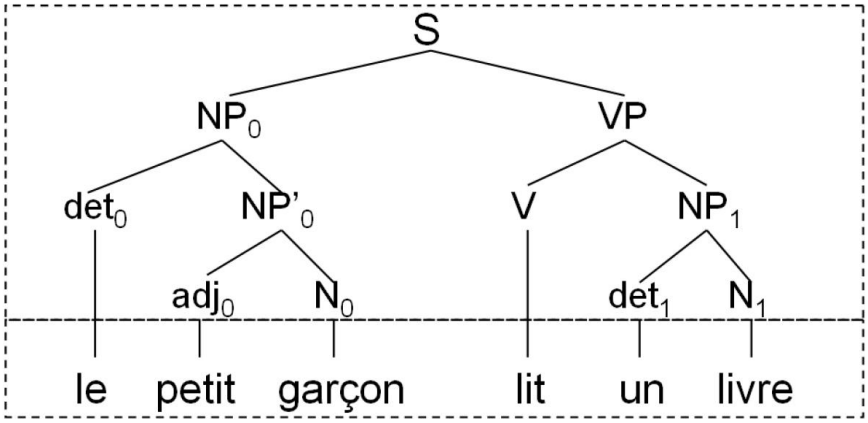

FIGURE 1.7 – Exemple 1 d'une analyse syntaxique.

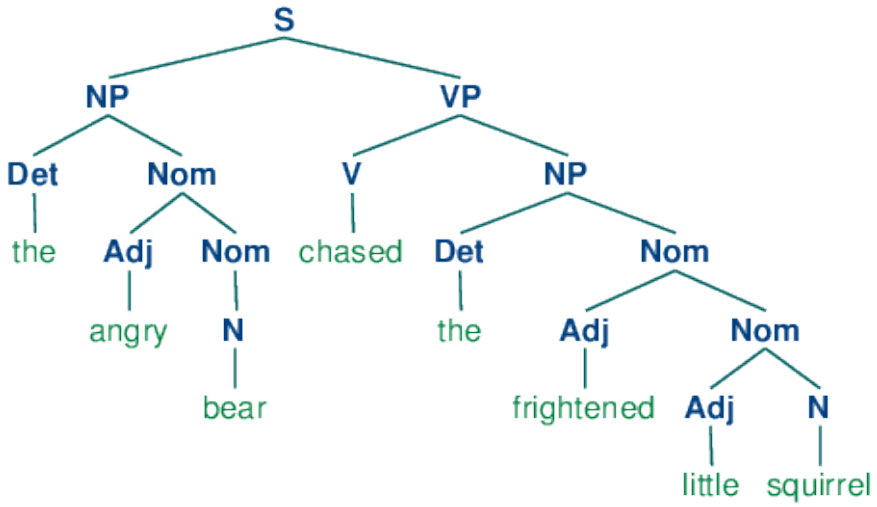

FIGURE 1.8 – Exemple 2 d'une analyse syntaxique.

1.6 Conclusion

Dans ce chapitre on a présenté le processus de fouille de texte, avec les différentes méthodologies de présentation des documents sous forme approprié pour leurs traitement en utilisant les techniques d'apprentissage. Cependant, on a illustré un exemple d'application connu la synthèse du texte. Dans le chapitre suivant, on va présenter les techniques d'apprentissage machine dans le contexte d'analyse des sentiments dans les réseaux sociaux.

Chapitre 2

Text mining dans les réseaux sociaux

2.1 Introduction

Le problème de classification est défini comme suit : nous avons un ensemble d'apprentissage, chaque enregistrement est étiqueté par une valeur de classe k qui contient des valeurs discrètes indexées $1..k$. Nous utilisons les données d'apprentissage pour former un modèle de classification qui relie chaque enregistrement par l'étiquette qui lui convient. Pour une instance de test le modèle va prédire une étiquette à cette instance. Le problème de classification de texte trouve dans plusieurs domaines, voici quelques exemples :

Filtrage des nouvelles et organisation : La majorité de services d'informations sont aujourd'hui de type électronique, il est difficiles de les organiser manuellement. Par conséquent, les méthodes d'automatisation sont utiles pour l'organisation des informations, Cette application est également appelée filtrage de texte [24].

Organisation et récupération de documents : Nombreuses méthodes supervisés sont utilisées pour l'organisation des documents dans plusieurs catégories, elles nécessitent de grandes bibliothèques numériques de documents, des collections Web, de la littérature scientifique ou même des flux sociaux. Les collections de documents organisées hiérarchiquement peuvent servir pour la navigation et la récupération [30].

Opinion Mining : Les opinions des clients sont aussi des documents courts qui aide à révéler des informations utiles sur les produits ou services des entreprises [27, 26, 19].

Classification des émail et filtrage des spam : Ça sert à classer les courriels en des courriels utiles et d'autres indésirables [23].

Une grande variétés de méthodes pour la classification des textes, ces méthodes sont basées généralement sur les domaines quantitatives et catégorielles. Puisque le texte peut-être converti en un modèle quantitative avec la fréquence des mots dans le corpus, il est possible de lui attribuer toutes ces méthodes. Quelques méthodes de classification de texte sont présentés dans la section suivante.

2.2 Méthodes de classification pour le text mining

2.2.1 Sélection de caractéristiques pour la classification de texte

Les tâche les plus intéressantes à accomplir avant toute opération de classification sont la représentation des documents et la sélection des caractéristiques. la sélection de caractéristiques est très importante dans la classification des textes, en raison d'une immense des entités textuelles et l'existence des entités non pertinentes (bruitées). En générale le texte peut-être représenter sous-forme un sac de mots avec leurs fréquence associée et aussi sous-forme une suite de chaînes. La plupart des méthodes de classification de texte au vue de sa simplicité à des fins de classification.

La sélection de caractéristiques est celle de l'élimination des mots d'arrêt et les mots vides qui ne sont pas discriminatoires à aucune des classes est utilisée dan les méthode supervisées et non supervisées. Une grande variété de méthodes de sélection de caractéristiques sont ci-dessous :

Indice de Gini

l'indice de Gini est l'une des méthodes qui détermine la discrimination d'une caractéristique. Soit $p_1(w)...p_k(w)$ est la fraction de la présence en classe sur l'ensemble des classes différentes pour le mot w. Par conséquent, nous avons l'indice de Gini pour le mot w, noté $G(w)$ est défini comme suit :

$$G(w) = \sum_{i=1}^{k} p_i(w) = 1 \tag{2.1}$$

La valeur de l'indice de Gini est toujours comprise dans la plage $[\frac{1}{k}, 1]$. La définition de $G(w)$ est un ensemble de documents qui sont partagés avec une classe particulière, la valeur la plus élevé représente un plus grand pouvoir discriminant du mot w. $p_1(w)...p_k(w)$ représentent les distributions globales des documents dans les différentes classes, avec k ensembles de classes [7].

Information mutuelle

Cette mesure est dérivée de la théorie d'information, elle propose une méthode formelle pour modéliser l'information mutuelle entre les caractéristiques et les classes. l'information mutuelle $M_i(w)$ entre le mot w et la classe i est basé sur la concurrence entre le mot est la classe. La concurrence est donnée par $P_i \times F(w)$, cette valeur en fonction de la corrélation. L'information mutuelle est défini comme suit :

$$M_i(w) = log\left(\frac{F(w) \times p_i(w)}{F(w) \times P_i}\right) = log\left(\frac{p_i(w)}{P_i}\right) \tag{2.2}$$

Si $M_i(w) > 0$ le mot w est positivement corrélé à la classe sinon est négativement corrélé à la classe. L'information mutuelle du mot avec les différentes classes est connu par l'information mutuelle globale est défini par les valeurs moyennes et maximales de $M_i(w)$ sur les différentes classes.

$$M_{avg}(w) = \sum_{i=1}^{k} P_i M_i(w) \tag{2.3}$$

$$M_{max}(w) = \max_i M_i(w) \tag{2.4}$$

Chacune de ces mesure détermine la pertinence du mot w, l'autre mesure détermine la corrélation positive du mot w avec l'une quelconque des classes [13].

Statistique χ^2

Cette mesure détermine le manque d'indépendance entre le mot w et la classe i. Soit n le nombre total de documents dans la collection, $p_i(w)$ la probabilité conditionnelle de la classe i pour les documents qui contiennent w, P_i soit la fraction globale des documents contenant la classe i, et $F(w)$ soit le global fraction de documents contenant le mot w. la formule est comme suit [10] :

$$\chi_i^2 = \frac{n \cdot F(w)^2 \cdot (p_i(w) \cdot P_i)^2}{F(w) \cdot (1 - F(w)) \cdot P_i \cdot (1 - P_i)} \tag{2.5}$$

une mesure globale χ^2 pour déterminer la corrélation du mot avec les différentes classes comme l'information mutuelle en utilisant les valeurs moyennes et maximales :

$$\chi_{avg}^2(w) = \sum_{i=1}^{k} P_i \chi^2(w) \tag{2.6}$$

$$\chi_{max}^2(w) = \max_i \chi_i^2(w) \tag{2.7}$$

2.2.2 Classification par arbre de décision

Un arbre de décision est une décomposition hiérarchique de l'espace de données, dans lequel se trouve une condition sur la valeur d'attribut pour diviser l'espace de données hiérarchiquement. Dans le cas du données textuelles la condition est la présence ou l'absence d'un ou plusieurs mots dans le document. La division de l'espace de données est appliquée de manière récursive dans l'arbre de décision, jusqu'à ce que les nœuds feuilles comportent un nombre minimum de données [2].

L'étiquette majoritaire en fonction de coûts trouvée dans les feuilles est utilisée pour la classification. Pour une instance de test donnée, on va parcourir l'arbre d'une façon descendante jusqu'à trouver le nœud de feuille pertinent. Tous les nœuds qui construisent l'arbre sont des mots trouvés dans la collection de textes. Pour construire l'arbre de décision il y a différents types d'approches sont les suivants :

Divisions d'attributs uniques : nous utilisons la présence ou l'absence de mots particuliers sur un nœud particulier de l'arbre pour effectuer la division. À n'importe quel niveau, nous choisissons le mot qui fournit la discrimination maximale entre les différentes classes. La mesure de l'indice de Gini ou le gain d'information sont utilisées pour déterminer le discrimination des niveaux.

Division multi-attributs discriminante : dans ce cas un test de discrimination est utilisé tel que le test de Fisher. Le choix du point de partage est choisi afin de maximiser la discrimination entre les différentes classes [1].

Il y a plusieurs algorithme d'arbre de décision tel que l'algorithme ID3, l'algorithme C4,5 ces algorithmes utilisent des subdivisions d'attributs uniques à chaque nœud, où la caractéristique ayant le gain d'information le plus élevé est utilisée aux fins du partage. L'algorithme d'arbre de décision basé sur l'approche bayésienne, les nœuds de feuilles de cet algorithme sont des probabilité de classe plutôt qu'une étiquette.

2.2.3 Classification basés sur des règles

Dans les classificateurs basé sur des règles, l'espace de données est transformé à un ensemble de règles, dans lesquelles le coté gauches est une condition sur l'espace de données et le coté droit est l'étiquette de classe. L'ensemble de règles est essentiellement le modèle généré à partir des données d'apprentissage. Pour une instance de test donnée, l'ensemble de règle sera déterminé tant que l'instance satisfaire la condition du coté gauche et l'étiquette sera déterminer tant que l'instance satisfait la règle.

Dans sa forme, le côté gauche de la règle est un ensemble de mots présent dans le document qui exprime une condition booléenne écrite en forme normale disjonctive. Un exemple d'une règle : $Toyota \cup Honda \Rightarrow Voitures$ [8].

Des critères sont utilisées pour générer l'ensemble de règle basé sur le support et la confiance :

Support

Ceci quantifie le volume de la règle, le nombre d'instances dans l'ensemble de données d'apprentissage qui sont pertinentes pour la règle. Par exemple, dans un corpus contenant 200 000 documents, une règle dans laquelle l'ensemble de côté gauche et le côté droit sont satisfaits par 60 000 documents ce qui représente plus de 20% des documents.

Confiance

Ceci quantifie la force de la règle, la probabilité conditionnelle de la règle le côté droit est satisfait si le côté gauche est satisfait. Pour une instance de test donnée, on va déterminer toutes les règles pertinentes pour cette instance, mais le problème c'est comment on va choisir l'étiquette convenable, si tous les étiquette de côté droit sont identiques alors l'étiquette est déterminée mais si elles ne sont pas identiques il y a plusieurs méthodes pour gérer ce conflit parmi eux l'étiquette qui a un grand nombre d'occurrence choisie comme la plus pertinente.

2.2.4 Classification probabiliste et Naive Bayes

Les classificateurs probabilistes utilisent le modèle de mélange, qui suppose que chaque classe est un élément de mélange. Chaque élément de mélange est un modèle génératif qui calcule la probabilité d'un terme dans une classe, ce type de classificateur est appelé un classificateur génératif. Le classificateur génératif le plus simple et le plus utilisé est le classificateur de Naive Bayes.Ce dernier calcule la probabilité a posteriori d'une classe, en fonction de la distribution des mots dans le document, il ignore la position réelle des mots dans le document [17, 5].

Modèle Bernoulli multivarié

Ce modèle utilise la présence ou l'absence d'un mot dans le document pour représenter un document.Puisque les caractéristiques à modéliser sont binaires, le modèle pour les documents de chaque classe est un modèle de Bernoulli multivarié [16].

Modèle multinomial

Ce modèle utilise la fréquences de termes dans un document en représentant un document avec un sac de mots. la probabilité conditionnelle d'un document donné à une classe est simplement un produit de la probabilité de chaque mot observé dans la classe correspondante.

2.2.5 Classification linéaire

Les classificateurs linéaires utilisent l'équation $p = AX + b$ où $X = (x_1..x_n)$ est le vecteur normalisé de fréquence des mots de document, $A = (a_1..a_n)$ est un vecteur de coefficients linéaires ayant la même dimensionnalité que l'espace caractéristique et b est un scalaire.

Les machines à vecteurs de support (SVM) sont des classificateurs qui cherche à trouver un bon séparateur linéaire entre les différentes classes [6].

La modélisation par régression (telle que la méthode des moindres carrés) est connue comme étant une méthode statique plus directe spécifique pour la classification des textes, cependant elle utilise les variables numérique plutôt que catégorique, plusieurs méthodes sont proposées pour adapter ces méthodes au cas de la classification des données textuelles [4].

les réseaux neuronaux simples (perceptron ou réseau à une seule couche) est un classificateur linéaire basé sur une fonction essentiellement linéaire calculée par un ensemble de neurones, il fonctionne bien pour les textes. Il est autant possible pour la séparation non linéaire.

2.2.6 Classification par SVM

Les machines à vecteurs de support ont été proposés pour les données numérique. L'objectif du SVM est de trouver des séparateurs qui séparent bien les différentes classes. l'avantage de la méthodes plus qu'elle détermine une direction optimale de la discrimination dans l'ensemble des caractéristiques, elle est robuste pour une dimension élevée. Elle est très efficace pour les données textuelles grâce à la nature clairsemée du texte dans laquelle peu de caractéristiques sont sans importance. SVM peut-être non linéaire, il construit une surface de décision non linéaire où les classes seront séparées,néanmoins dans la pratique le SVM linéaire est souvent utilisé à cause de leur simplicité et facilité de l'interprétation. La méthode SVM est flexible et plus utilisée dans le domaine de texte, par exemple dans les courriers électronique elle les classe comme spam ou non-spam. Elle offre des performances beaucoup plus robustes par rapport à de nombreuses autres techniques telles que l'augmentation des arbres de décision [29].

2.2.7 Classification par réseau de neurones

L'unité de base dans un réseau de neurones est un neurone, chaque neurone reçoit un ensemble de données en entrée qui sont désignées par le vecteur X qui signifie la fréquence de mots dans le $i$ème document. Chaque neurone est également associé à un ensemble de poids A, utilisés pour calculer une fonction de ces entrées. La fonction utilisée dans le réseau de neurone est la fonction linéaire suivante : $p_i = A \cdot X_i$ L'utilisation principal de réseau de neurones est quant toutes les classes peuvent ne pas être nettement séparées les unes des autres avec un séparateur linéaire. L'utilisation de plusieurs couches induit de telles limites de classification non linéaires. L'objectif de plusieurs couches est d'induire plusieurs limites linéaires par segments, qui peuvent être utilisées pour approcher des régions fermées appartenant à une classe particulière. les sorties des neurones dans les couches antérieures alimentent les neurones dans les couches ultérieures [9].

2.3 Classification des données du Web

la prolifération des technologies web et de réseaux sociaux a engendré une énorme quantité de données documentaires, tel que le web dans lequel les document sont liés entre eux à l'aide des hyper-liens. Les commentaires et les profils du texte des utilisateurs du réseau sociaux sont aussi un sorte de ces données. La classification dans ce cas base sur les informations de couplage car les documents qui assimilent le même sujet sont liés entre eux. Un sous-ensemble de nœuds de réseau sont étiqueté et les nœuds restantes sont classées à la base de leur liaison avec les autres.

Un réseau de nœuds basé sur le contenu est noté $G = (N, A, C)$, où N est l'ensemble des nœuds, A est l'ensemble des arêtes entre les nœuds, et C est un ensemble de documents texte. Chaque nœud de N correspond à un document texte en C, peut-être un document soit vide dans ce cas le nœud ne contient aucun contenu. Un sous-ensemble des nœuds de N sont étiquetés. La classification dans ce cas consiste à trouver les classes des autres nœuds. Le contenu et la structure joue un rôle dans la classification.

La méthode de classification d'hypertexte utilise le contenu et les étiquettes des page web voisines pour la classification, la présence d'un lien entre une page donnée et une autre étiquetée considérée comme une caractéristique du classificateur. Le problème se pose quant tous les voisins les plus proches ne sont pas étiquetées, alors il y a deux proposition :

Analyse de liaison améliorée dans le cas supervisé : Dans ce cas on suppose que tous les étiquettes sont connus, alors les étiquettes des voisins les plus proche sont considérées comme étiquette de classification.

Lorsque les étiquettes de classe des voisins les plus proches ne sont pas connues : une approche itérative est utilisée pour combiner la classification basée sur le texte et la liaison. Nous effectuons un premier étiquetage des documents voisins à l'aide du contenu du document. Ces étiquettes sont ensuite utilisées pour classer l'étiquette du document cible, en utilisant à la fois le texte local et les étiquettes de classe des voisins. Cette approche est

Rank	*Website*	*Rank*	*Website*
1	Google	6	**Blogger**
2	**Facebook**	7	Baidu
3	**Youtube**	8	**Wikipedia**
4	Yahoo!	9	**Twitter**
5	Windows Live	10	QQ.com

FIGURE 2.1 – Rapport de trafic Internet par Alexa

utilisée itérativement pour redéfinir les étiquettes du document cible et de ses voisins jusqu'à ce que la convergence soit atteinte.

2.4 Text mining dans les réseaux sociaux

Les médias sociaux tels que les blogs, les micro-blogs, les forums de discussion et les sites de partage multimédia sont utilisés comme un moyen de partage d'information de dernière heure, participer à des événements et se connecter à tout moment. Les sites de médias sociaux sont considérés comme des parties très important dans les applications web, qui représentent 50% des 10 premiers sites selon les statistiques d'Alexa (voir Table 2.1). Aussi les messages Twitter sont même enregistrés dans la US Library of Congress. Ces médias sociaux munirent une nanti information sur l'interférence humaine et le comportement collectif, attirant ainsi beaucoup d'attention de disciplines telles que la sociologie, les affaires, la psychologie, la politique, l'informatique, l'économie et d'autres aspects culturels des sociétés. La source de médias sociaux Wikipédia définit les médias sociaux comme suit :

> "Les médias sociaux sont des médias d'interaction sociale, utilisant des techniques de communication hautement accessibles et évolutives. C'est l'utilisation des technologies Web et mobiles pour transformer la communication en un dialogue interactif. "

Les médias traditionnels tels que les sont d'une forme unidirectionnel (du commerce au consommateur), c'est à dire l'information est créée à partir de sources médiatiques ou d'annonceurs et transmise aux consommateurs de médias. Par contre les technologies actuelles tel que web 2.0 fournisse au consommateurs des services, elles autorisent aux utilisateurs d'interagir et de contribuer les uns avec les autres dans un dialogue sur les médias sociaux de contenu généré par les utilisateurs dans une communauté virtuelle.

les sites Web de médias sociaux sont considérés comme les plus utilisées car ils comportent des différents types de services et des différents gabarit de données textes, images et vidéos etc. Par exemple les sites de partage de médias Flickr et Youtube permettent aux utilisateurs ordinaires de publier leurs images et vidéos de leurs vies quotidiennes. En conséquence, une grande quantité de données d'image et de vidéo est enregistrés dans les sites. D'autre part les dites de blogs permettent au utilisateurs de partager des données textuelles. Dans les sites de bookmarking social, les utilisateurs partagent entre eux des tags et des URL. Malgré tous

Category	*Representative Sites*
Wiki	Wikipedia, Scholarpedia
Blogging	Blogger, LiveJournal, WordPress
Social News	Digg, Mixx, Slashdot
Micro Blogging	Twitter, Google Buzz
Opinion & Reviews	ePinions, Yelp
Question Answering	Yahoo! Answers, Baidu Zhidao
Media Sharing	Flickr ,Youtube
Social Bookmarking	Delicious, CiteULike
Social Networking	Facebook, LinkedIn, MySpace

FIGURE 2.2 – types de médias sociaux

cette variétés de formats de données dans les médias sociaux le textes représente un rôle très important parce que la majorité des informations du sites de médias sociaux sont stockées en texte. Comme par exemple les services de micro-blogging permettent aux utilisateurs d'afficher de petites quantités de texte pour communiquer et partager des informations pour participer à des événements tels que la révolution égyptienne» et le «tremblement de terre et tsunami de Tohoku».

Analyse de texte consiste à découvrir des connaissances qui peuvent être trouvées dans les archives de texte. Ce domaine a reçu beaucoup d'attention en raison de sa large application, des techniques d'emprunt du traitement du langage naturel (NLP), de l'exploration de données (Data Mining (DM)), de l'apprentissage automatique (Machine Learning (ML)), de la recherche d'information (Information Retrieval (IR)) etc.

2.4.1 Aspects distincts du texte dans les médias sociaux

Le contexte générale de l'analyse de texte traditionnelle pour traiter un corpus de texte se limite à trois étapes (voir Figure 2.3) : le pré-traitement du texte, la représentation du texte et la découverte des connaissances. On va les expliquer ci dessous en utilisant un corpus de texte qui contient trois messages de micro-blogging :

"watching the King's Speech"

"I like the King's Speech"

"they decide to watch a movie"

Courte longueur : quelques sites Web de médias sociaux bornent la longueur des contenus conçus par les utilisateurs tels que les messages de microblogging, les revues de produits et les légendes d'images, etc. Twitter permet la longueur de chaque tweet est limitée à 140 caractères. De même, les commentaires Picasa sont limités à 512 caractères et les messages d'état personnels sur Windows Live Messenger sont limités à 128 caractères. De ce fait, ces messages courts ont joué des rôles de plus en plus importants dans les applications des médias sociaux.

Le traitement de courts textes est essentiel pour les méthodes d'analyse de texte. Ce dernier est constitué de quelques phrases ce qui pose des problèmes dans l'analyse de texte Par

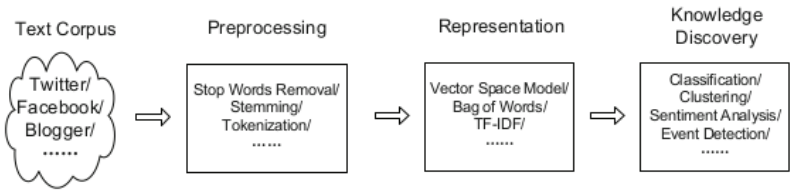

FIGURE 2.3 – Les phases d'analyse de texte.

exemple, l'étiquette «Tremblement de terre au Japon» ne contient aucun mot ou expression en rapport avec «Crise nucléaire», alors que nous apprenons que ces deux événements sont liés à des nouvelles récentes. Parce qu'ils n'ont pas de mots ou de phrases communs, il est très difficile pour les modèles et les méthodes basés sur BOW de construire des liens sémantiques entre eux.

Phrases non structurées Une différence importante entre le texte dans les médias sociaux et les médias traditionnels est la variance dans la qualité du contenu. Tout d'abord, la variance de la qualité provient des attitudes des gens lors de la publication d'un message de micro-blogging ou de répondre à une question dans un forum. Certains utilisateurs sont des experts pour le sujet et publient des informations très soigneusement, tandis que d'autres ne publient pas aussi haut de la qualité. Le principal défi posé par le contenu des sites de médias sociaux est le fait que la distribution de la qualité est très différente : des articles de très haute qualité aux contenus de mauvaise qualité, parfois abusifs. Cela rend les tâches de filtrage et de classement dans ces systèmes plus complexes que dans d'autres domaines.

Deuxièmement, lors de la rédaction d'un message, les utilisateurs peuvent utiliser ou créer de nouvelles abréviations ou acronymes qui apparaissent rarement dans les documents texte conventionnels. Par exemple, des messages tels que "Comment ?", "Bon 9" ne sont pas vraiment des mots, mais ils sont intuitifs et populaires dans les médias sociaux. Ils fournissent aux utilisateurs la commodité de communiquer les uns avec les autres, mais il est très difficile d'identifier avec précision la signification sémantique de ces messages. Outre les expressions non structurées, le texte est parfois "bruyant" pour un sujet spécifique. Par exemple, un passage QA dans Yahoo ! Réponses "J'aime Sony" devrait être des données bruyantes à un poste qui parle de la sortie iPad 2. Il est difficile de classer le passage dans les classes correspondantes sans tenir compte de son information contextuelle.

Information abondante : les médias sociaux en général exhibent une grande diversité de sources d'information. Par rapport au continu, il existe beaucoup d'informations non liées au contenu. Comme exemple, Twitter permet aux utilisateurs d'utiliser le symbole «#», appelé hashtag, pour marquer des mots-clés ou des sujets dans un Tweet, une image est généralement associée à plusieurs étiquettes qui sont caractérisées par différentes zone de l'image. Des informations sous formes des liens qui sont beaucoup utilisées entre les utilisateurs de réseaux sociaux tel que facebook et aussi Wikipedia fournit un moyen efficace pour les utilisateurs de rediriger vers la page de concept d'ambiguïté ou la page de concept de niveau supérieur.

$$\begin{bmatrix} watch \\ King' \\ Speech \\ decid \\ movi \end{bmatrix} = \begin{bmatrix} 0.4055 & 0 & 0.4055 \\ 0.4055 & 0.4055 & 0 \\ 0.4055 & 0.4055 & 0 \\ 0 & 0 & 1.0986 \\ 0 & 0 & 1.0986 \end{bmatrix}$$

Pré-traitement du texte

Le pré-traitement du texte désire à rendre les documents d'entrée plus compréhensibles pour faciliter la représentation du texte qui est nécessaire pour les autres phases de l'analyse de texte. Les méthodes traditionnelles de pré-traitement de texte consistent à enlever les mots d'arrêts (mots sans signification) et le stemming (par exemple "regarder", "regarder", "regardé" sont représentés comme "regarder"). Le résultat du prétraitement de texte pour les trois messages de micro-blogging sera :

"watch King' Speech"

"King' Speech"

"decid watch movi"

Représentation vectorielle

Dans cette phase on va modéliser le texte sous forme un vecteur numérique puis de les traiter avec des opérations algébriques linéaires. Cette ferme est appelée Sac de mots ("Bag Of Words" (BOW)) ou Modèle d'espace vectoriel ("Vector Space Model" (VSM). Dans le modèle BOW, le mot est représenté comme une variable numérique, la forme la plus utilisée est tf-idf :

$$tf - idf(j) = tf(j) \cdot idf(j) \tag{2.8}$$

$$idf(j) = \log(N/df(j)) \tag{2.9}$$

où : - $tf(w)$ est la fréquence du mot (le nombre d'occurrences de mots dans un document). - $df(w)$ est la fréquence du document (le nombre de documents contenant le mot). - N est le nombre de documents dans le corpus. - $tf - idf(w)$ est le poids relatif de l'entité dans le vecteur.

Construction de modèle

Après la transformation du corpus de texte en vecteurs numériques, nous pouvons lancer les méthodes d'apprentissage automatique ou d'exploration de données existantes, maintenant nous pouvons extraire des informations utiles du corpus de texte d'entrée.

2.5 Application de l'analyse de texte aux médias sociaux

Un certain nombre de méthodes ont été proposées pour traiter les données textuelles dans les médias sociaux avec de nouvelles fonctionnalités. Nous allons expliquer quelques

applications d'analyse de texte sur les médias sociaux dans cette section.

2.5.1 Détection d'événement

La détection d'événements consiste à surveiller une source de données et à découvrir l'occurrence d'un événement capturé dans cette source. Ces sources de données contiennent des images, des vidéos, de l'audio, des données spatio-temporelles, des documents texte. Le volume de données textuelles dans les médias sociaux accru de façon exponentielle, nous offrant ainsi de nombreuses possibilités de détection et de suivi des événements.

Des données textuelles dans les médias sociaux sont des capteurs du monde réel beaucoup de travails ont été fait pour détecter les événements du monde réel à partir de ces données. Sakaki étudie l'interaction en temps réel des événements sur Twitter. Il considère chaque utilisateur comme un capteur pour surveiller les tweets affichés récemment et pour détecter les tremblements de terre ou arc-en-ciel[28]. Les étapes pour détecter un événement sont comme suit : premièrement, un classificateur est formé en utilisant des mots-clés, la longueur du message et le contexte correspondant en tant que fonctionnalités pour classer les tweets dans des cas positifs ou négatifs. Deuxièmement, un modèle spatio-temporel probabiliste pour l'événement cible sera construit afin d'identifier la localisation de l'événement. Comme une application, les auteurs ont construit un système de signalement des tremblements de terre au Japon, où de nombreux tremblements de terre ont lieu chaque année.

l'objectif de la détection des événements dans les médias sociaux est d'améliorer la découverte des nouvelles traditionnelles. Un petit nombre de nouvelles qui est connu sous le nom de «nouvelles de dernière heure» reçoit l'attention des utilisateurs malgré que les reportages sont générés quotidiennement. Les éditeurs de journaux et de sites Web décident quelles histoires peuvent être classées plus haut et assignées dans un endroit important comme la première page pour qu'elles attirent l'attention. De la même manière, les services d'actualités sur le Web tels que Google Actualités , permettent aux utilisateurs d'accéder à des perspectives générales sur les reportages importants signalés en regroupant des articles dans des événements d'actualité connexes. Un grand problème quand les éditeurs décident automatiquement quelles sont les meilleures histoires à montrer. Un sondage mené par Technorati a découvert que 30% des blogueurs se considèrent comme des blogueurs sur des sujets liés à l'actualité. Les chercheurs ont suggéré d'utiliser blogosphère pour simplifier la détection et l'évaluation des nouvelles. Lee présente de nouvelles approches pour déterminer les titres importantes de la blogosphère pour un jour donné. Le système suggéré se constitue de deux composants basés sur le cadre de modèle de langage, la probabilité de requête et le titre d'actualité avant. Pour la vraisemblance de la requête, les auteurs proposent plusieurs approches pour estimer le modèle de langage de requête et le modèle de langage de titre de nouvelles. Ils suggèrent également plusieurs critères pour évaluer l'importance ou la pertinence de l'actualité pour un jour donné [31].

2.5.2 Réponse à une question collaborative

Les services de réponse aux questions collaboratives ont connu une évolution dés que l'épanouissement des médias sociaux. Ils regroupent un réseau d 'experts auto-proclamés pour répondre aux questions d'autres personnes. Une grande quantité de questions sont posées et répondues tous les jours sur des sites Web de questions et réponses (QA) tel que Yahoo !. Les services de réponse aux questions collaboratives sont conçu pour les utilisateurs qui recherche de conseils dans une situation particulière, pour recueillir des opinions, partager des connaissances techniques, pour se divertir et interagir avec la communauté. Ces services d'QA historiques ont créé leurs bases de données, cette approche facilite aux utilisateurs pour rechercher des informations au lieu d'une recherche sur le Web ou poser une nouvelle question dans un forum. Les utilisateurs peuvent trouver directement des questions ou des réponses historiques pertinentes dans les archives.

Les requêtes des utilisateurs reliées sémantiquement avec les questions les plus pertinentes pour que les utilisateurs puissent trouver des questions similaires et leurs réponses correspondantes. Wang propose une méthode basée sur les graphes pour effectuer des restitutions de questions à partie de segmentation des questions à plusieurs phrases. Les auteurs essayent d'abord de détecter les phrases question à l'aide d'un classificateur construit à partir de caractéristiques syntaxiques et lexicales, et utilisent des méthodes de similarité de pour calculer le score de proximité entre les phrases de question et de contexte [32].

Dans le but d'améliorer la gestion des archives de QA. Harper a essayé de déterminer les questions et les réponses avaient une valeur archivistique en comparant les différences entre les questions conversationnelles et les questions d'information. Les questions d'information sont des questions qui ont un but d'obtenir des informations dont le demandeur pourrait prendre des leçons comme exemple "Est-ce que boire du Coca est bon pour la santé ?". Les questions conversationnelles se réfèrent aux questions dans le but de vivifier la discussion où les utilisateurs peuvent viser à obtenir des opinions ou à s'exprimer. Un exemple est "Aimez-vous boire du Coca-Cola ?". Les auteurs montre des preuves que les questions conversationnelles ont généralement une valeur archivistique potentielle beaucoup plus faible que les questions d'information. En outre, ils ont utilisé des techniques d'apprentissage automatique pour classer automatiquement les questions en termes de conversation ou d'information du point de vue du processus sur les différences catégorielles, linguistiques et sociales entre les différents types de questions [12].

2.6 Sentiment Analysis et Opinion Mining

2.6.1 Définition d'opinion

Pour qu'on définit l'opinion, on va utiliser un segment de revue sur l'IPhone : "(1) J'ai acheté un iPhone il y a quelques jours. (2) C'était un si bon téléphone. (3) L'écran tactile était vraiment cool. (4) La qualité de la voix était claire aussi. (5) Cependant, ma mère était folle de moi comme je ne le lui avais pas dit avant que je l'achète. (6) Elle pensait aussi que le téléphone était trop cher et voulait que je le retourne au magasin. . . "

Exemple 1 : La question est : ce que nous voulons extraire de cette revue ? La première chose que nous remarquons est qu'il y a plusieurs avis dans cette revue. Les phrases (2), (3) et (4) expriment des opinions positives, tandis que les phrases (5) et (6) expriment des opinions ou des émotions négatives. Ensuite, nous remarquons également que les opinions ont toutes des cibles. La cible de l'opinion dans la phrase (2) est l'iPhone dans son ensemble, et les cibles des opinions dans les phrases (3) et (4) sont respectivement «écran tactile» et «qualité vocale» de l'iPhone. La cible de l'opinion dans la phrase (6) est le prix de l'iPhone, mais la cible de l'opinion / émotion dans la phrase (5) est "moi", pas iPhone. Enfin, nous pouvons également remarquer les détenteurs d'opinions. Le détenteur des opinions dans les phrases (2), (3) et (4) est l'auteur de la revue, mais dans les phrases (5) et (6) c'est "ma mère".

En général, les opinions peuvent être exprimées à propos de n'importe quoi, par exemple un produit, un service, un individu, une organisation, un événement ou un sujet, par une personne ou une organisation. Nous utilisons l'entité pour désigner l'objet cible qui a été mesuré. Formellement, nous obtenons ce qui suit :

Définition 1 : (Entité) Une entité e est un produit, un service, une personne, un événement, une organisation ou un sujet. Il est affilié à une paire, e : (T, W), où T est une hiérarchie de composants et W est un ensemble d'attributs de e. Une entité est montrée comme un arbre ou une hiérarchie. La racine de l'arbre est le nom de l'entité. Chaque nœud non racine est un composant ou un sous-composant de l'entité. Chaque lien est une partie de la relation. Chaque nœud est associé à un ensemble d'attributs. Une opinion peut être exprimée sur n'importe quel nœud et n'importe quel attribut du nœud.

Techniquement, il est très utile de simplifier cette définition pour deux raisons : Premièrement, le traitement du langage naturel est une tâche difficile. Pour étudier efficacement le texte à un niveau de détail arbitraire tel que décrit dans la définition est très difficile. Deuxièmement, pour un utilisateur ordinaire, il est trop complexe d'utiliser une représentation hiérarchique. Ainsi, nous simplifions et plaçons l'arbre à deux niveaux et utilisons le terme «aspects» pour désigner à la fois les composants et les attributs. Un exemple d'entité est le suivant :

Exemple 2 : une marque particulière de téléphone cellulaire est une entité, par exemple un iPhone. Il contient un ensemble de composants, par exemple une batterie et un écran, ainsi qu'un ensemble d'attributs, par exemple, la qualité de la voix, la taille et le poids. Le composant de batterie a également son propre ensemble d'attributs, par exemple la durée de vie de la batterie et la taille de la batterie.

Les détenteurs d'opinions ou sources d'opinions citent souvent explicitement la personne ou l'organisation et donnent leurs opinions. Il se trouve deux types d'opinions : les opinions régulières et les opinions comparatives. Les opinions régulières sont souvent simplement considérées comme des opinions dans la littérature de recherche. Un avis comparatif exprime une relation de similitudes ou de différences entre deux entités ou plus. Une opinion (ou une opinion régulière) est simplement un sentiment, une attitude, une émotion ou une appréciation positive, négative ou neutre concernant une entité ou un aspect.

Définition 3 : Un avis (ou opinion régulière) est un quintuple, $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$, où e_i est le nom d'une entité, un ij est un aspect de e_i, oo_{ijkl} est l'orientation de l'opinion sur l'aspect a_{ij} de l'entité e_i, h_k est le porteur d'opinion, et t_l est le moment où l'opinion est exprimée par h_k. L'orientation d'opinion de $ijkl$ peut être positive, négative ou neutre. Quand une opinion est sur l'entité elle-même dans son ensemble, nous utilisons l'aspect spécial GÉNÉRAL pour le désigner.

Cette définition fournit une base pour la transformation de texte non structuré en données structurées. Le quintuple nous donne les informations essentielles pour un riche ensemble d'analyses qualitatives et quantitatives d'opinions. Plus précisément, le quintuple peut être considéré comme un schéma pour une table de base de données. Pour extraire les opinions depuis un ensemble de documents, il faut suivre les étapes suivantes :

Étape 1 (extraction et regroupement d'entités) : extrayez toutes les expressions d'entité dans D et regroupez les expressions d'entités synonymes dans des clusters d'entités. Chaque grappe d'expression d'entité indique une entité unique e_i.

Étape 2 (extraction et regroupement d'aspect) : extrayez toutes les expressions d'aspect des entités et regroupez les expressions d'aspect en clusters. Chaque groupe d'expressions d'aspect de l'entité e_i indique un aspect unique a_{ij}.

Étape 3 (détenteur d'opinion et extraction de temps) : extraire ces informations du texte ou des données non structurées.

Étape 4 (classification du sentiment d'aspect) : Déterminez si chaque opinion sur un aspect est positive, négative ou neutre.

Étape 5 (génération quintuple d'opinion) : Produire tous les quintuples d'opinion $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ exprimés en D sur la base des résultats des tâches ci-dessus. On utilise un exemple de blog pour clarifier ces étapes :

Exemple 4 : (Blog Posting) Posté par : bigXyz le 4 novembre 2010 : (1) J'ai acheté un téléphone Motorola et ma copine a acheté un téléphone Nokia hier. (2) Nous nous sommes appelés quand nous sommes rentrés à la maison. (3) La voix de mon téléphone Moto n'était pas claire, mais l'appareil photo était bon. (4) Ma copine était plutôt contente de son téléphone et de sa qualité sonore. (5) Je veux un téléphone avec une bonne qualité de voix. (6) Je ne vais probablement pas le garder. L'étape 1 doit extraire les expressions d'entité "Motorola", "Nokia" et "Moto", et les groupes "Motorola" et "Moto" ensemble car ils représentent la même entité.

L'étape 2 devrait extraire les expressions d'aspect "caméra", "voix" et "son", et grouper "voix" et "son" ensemble, car ce sont des synonymes représentant le même aspect.

L'étape 3 devrait trouver le détenteur des opinions dans la phrase (3) pour être bigXyz (l'auteur du blog), et le titulaire des opinions dans la phrase (4) pour être la petite amie de

bigXyz. Il devrait également trouver l'heure à laquelle le blog a été publié, soit le 4 novembre 2010.

L'étape 4 devrait trouver que la phrase (3) donne une opinion négative à la qualité de la voix du téléphone Motorola, mais une opinion positive à son appareil photo. La phrase (4) donne des opinions positives sur le téléphone Nokia dans son ensemble ainsi que sur sa qualité sonore. La phrase (5) exprime apparemment une opinion positive, mais ce n'est pas le cas. Pour générer des quintuples d'opinion pour la phrase (4), nous devons également savoir ce que «son téléphone» est et ce que «son» désigne. Tout cela sont des problèmes difficiles.

L'étape 5 devrait finalement générer les quintuples d'opinion suivants : (Motorola, qualité de la voix, négatif, bigXyz, Nov-4-2010) (Motorola, appareil photo, positif, bigXyz, Nov-4-2010) (Nokia, GÉNÉRAL, positif, la petite amie de bigXyz, Nov-4-2010) (Nokia, qualité de la voix, positif, la petite amie de bigXyz, Nov-4-2010)

Définition 5 : (subjectivité de la phrase) Une phrase objective présente des informations factuelles sur le monde, tandis qu'une phrase subjective exprime des sentiments, des points de vue ou des croyances personnels. Par exemple, dans l'exemple ci-dessus, les phrases (1) et (2) sont des phrases objectives, tandis que toutes les autres phrases sont des phrases subjectives. Les expressions subjectives prennent plusieurs formes, par exemple, les opinions, les allégations, les désirs, les croyances, les soupçons et les spéculations.

Définition 6 : (Émotion) Les émotions sont nos sentiments et pensées subjectifs. Les gens ont 6 émotions primaires, c'est-à-dire l'amour, la joie, la surprise, la colère, la tristesse et la peur.

2.6.2 Résumé de l'opinion basée sur l'aspect

La majorité des applications minières d'opinion doivent analyser les opinions d'un grand nombre de détenteurs d'opinions. Une opinion d'une seule personne n'est pas suffisante pour l'action. Cela signifie qu'une forme de résumé des opinions est souhaitable. Les quintuples d'opinion définis ci-dessus constituent une excellente source d'informations pour générer des résumés à la fois qualitatifs et quantitatifs. Une forme commune de résumé est basée sur des aspects, et est appelée résumé d'opinion basé sur les aspects.

Par exemple, nous collectons toutes les révisions d'un téléphone cellulaire particulier, 125 commentaires ont exprimé des opinions positives sur le téléphone et 7 ont exprimé des opinions négatives. La qualité et la taille de la voix sont deux aspects du produit. 120 critiques ont exprimé des opinions positives sur la qualité de la voix, et seulement 8 avis ont exprimé des opinions négatives. Le résumé être visualisé à l'aide d'un graphique à barres. Une faiblesse d'un tel résumé basé sur le texte est qu'il n'est pas quantitatif mais seulement qualitatif, ce qui n'est pas accordé à des raisons analytiques. Par exemple, un résumé de texte traditionnel peut montrer "La plupart des gens n'aiment pas ce produit". Cependant, un résumé quantitatif peut indiquer que 60% des gens n'aiment pas ce produit et 40% d'entre eux l'aiment.

2.7 Classification du sentiment de document

Le principal sujet de recherche de l'opinion minière est concerné sur la classification des opinions. Il classe un document d'opinion (par exemple, une revue de produit) comme exprimant une opinion ou un sentiment positif ou négatif.

Définition 8 (Sentiment au niveau du document) Étant donné un document d'opinion d évaluant une entité e, déterminer l'orientation d'opinion sur e, c.-à-d., Déterminer oo sur l'aspect GÉNÉRAL dans le quintuple $(GENERAL, oo, h, t)$.

Hypothèse : La classification des sentiments présume que le document d'opinion d (par exemple, une revue de produit) exprime des opinions sur une seule entité e et que les opinions proviennent d'un seul détenteur d'opinion h. Cette hypothèse est admissible pour les évaluations de produits et de services par les clients, car chaque révision se focalise généralement sur un seul produit et est écrite par un seul évaluateur. Cependant, il peut ne pas être valable pour un forum et un blog, car dans un tel article, l'auteur peut exprimer des opinions sur plusieurs produits et les comparer en utilisant des phrases comparatives. La majorité des techniques existantes de classification des sentiments au niveau des documents sont basées sur l'apprentissage supervisé, bien qu'il existe également des méthodes non supervisées.

2.7.1 Classification basée sur l'apprentissage supervisé

La classification des sentiments peut certainement être énoncée comme un problème d'apprentissage supervisé avec trois classes, positive, négative et neutre. Les données sont principalement des revues de produits. Étant donné que chaque évaluation a déjà une notation attribuée par le réviseur (par exemple, de 1 à 5 étoiles), les données de formation et de test sont facilement disponibles. Par exemple, une critique avec 4 ou 5 étoiles est considérée comme une critique positive, une critique avec 1 ou 2 étoiles est considérée comme une critique négative et une critique avec 3 étoiles est considérée comme une critique neutre.

Toutes les méthodes d'apprentissage supervisé existantes peuvent être appliquées à la classification de sentiment, par exemple, la classification naïve baise et les machines vectorielles de support (SVM). [27] a adopté cette approche pour classer les critiques de films en deux catégories, positive et négative en utilisant un sac de mots individuels comme caractéristiques dans la classification. La tâche principale de la classification des sentiments est d'établir un ensemble efficace de fonctionnalités.

Mots et expressions d'opinion : Les mots d'opinion sont des mots couramment utilisés pour exprimer des sentiments positifs ou négatifs. Par exemple, beau, merveilleux, bon, et étonnant sont des mots d'opinion positifs, et mauvais, pauvres et terribles sont des mots d'opinion négatifs. Bien que de nombreux mots d'opinion soient des adjectifs et des adverbes, les noms (par exemple, les déchets, les ordures) et les verbes (par exemple, haineux et similaires) peuvent également indiquer des opinions. Il y a aussi des phrases d'opinion et des idiomes.

Négations : Clairement, les mots de négation sont importants parce que leurs apparences changent souvent l'orientation d'opinion. Par exemple, la phrase "Je n'aime pas cette caméra" est négative. Cependant, les mots de négation doivent être manipulés avec précaution car toutes les occurrences de ces mots ne signifient pas la négation. Par exemple, "not" in "not only" ne change pas la direction de l'orientation.

Au lieu d'utiliser une méthode d'apprentissage automatique standard, les chercheurs ont également proposé plusieurs techniques personnalisées spécifiquement pour la classification des sentiments, par exemple, la fonction de score (par exemple, 1-5 étoiles) basée sur des mots dans des revues positives et négatives.

Pour affecter l'étiquetage des documents on utilise les mots d'opinion dans la procédure de formation. Tan a utilisé des mots d'opinion pour étiqueter une partie des exemples informatifs et ensuite apprendre un nouveau classificateur supervisé basé sur ceux qui sont étiquetés.

2.7.2 Classification basée sur l'apprentissage non supervisé

Il n'est pas difficile d'imaginer que les mots et les phrases d'opinion sont les indicateurs dominants de la classification des sentiments. Ainsi, l'utilisation d'un apprentissage non supervisé basé sur de tels mots et phrases. On décrivons un algorithme de classification qui consiste en trois étapes :

Étape 1 : Il extrait des phrases contenant des adjectifs ou des adverbes en tant que adjectifs et les adverbes sont de bons indicateurs d'opinions. Cependant, bien qu'un adjectif isolé puisse indiquer une opinion, le contexte peut être suffisant pour déterminer son orientation d'opinion. Par exemple, l'adjectif «imprévisible» peut avoir une orientation négative dans une revue automobile, dans une expression telle que «direction imprévisible», mais il pourrait avoir une orientation positive dans une critique de film, dans une phrase telle que «intrigue imprévisible». Par conséquent, l'algorithme extrait deux mots consécutifs, où un membre de la paire est un adjectif ou un adverbe, et l'autre est un mot de contexte. Par exemple dans la phrase "Cette caméra produit de belles images", de "belles images" seront extraites

Étape 2 : Il estime l'orientation sémantique des phrases extraites à l'aide de la mesure d'information mutuelle ponctuelle (pointwise mutual information (PMI)) donnée dans l'équation suivante :

$$PMI(term_1, term_2) = \log_2 \left(\frac{Pr(term_1 \wedge term_2)}{Pr(term_1) \cdot Pr(term_2)} \right) \tag{2.10}$$

Ici, $Pr(term_1 \wedge term_2)$ est la probabilité de d'occurrence du terme 1 et du terme 2, et $Pr(term_1) \cdot Pr(term_2)$ donne la probabilité que les deux termes coexistent s'ils sont statistiquement indépendants. Le rapport entre $Pr(term_1 \wedge term_2)$ et $Pr(term_1) \cdot Pr(term_2)$ est donc une mesure du degré de dépendance statistique entre eux. Le log de ce ratio est la quantité d'informations que nous acquérons sur la présence de l'un des mots lorsque nous observons l'autre.

L'orientation d'opinion (SO) ou la sémantique d'une phrase est calculée en fonction de son association avec le mot de référence positif "excellent" et de son association avec le mot de référence négatif "pauvre" :

$$SO(phrase) = PMI(phrase,'excelent') - PMI(phrase,'poor') \tag{2.11}$$

Étape 3 : Après un examen, l'algorithme calcule le SO moyen de toutes les phrases de la revue et classe la révision comme recommandée si le résultat moyen est positif, et non recommandé autrement.

2.7.3 Subjectivité et classification des sentiments

Naturellement, les mêmes techniques de classification des sentiments au niveau du document peuvent également être appliquées à des phrases individuelles. La classification d'une phrase comme subjective ou objective est souvent appelée classification de la subjectivité. Les phrases subjectives résultantes sont également classées comme exprimant des opinions positives ou négatives, ce que l'on appelle la classification des sentiments au niveau de la phrase.

Étant donné une phrase s, deux sous-tâches sont exécutées :

Classification de subjectivité : Déterminer si s est une phrase subjective ou une phrase objective

Classification des sentiments au niveau de la phrase : Si s est subjectif, déterminez s'il exprime un opinion positive, négative ou neutre.

Notez que le quintuple (e, a, oo, h, t) n'est pas utilisé pour définir le problème ici car la classification au niveau de la phrase est souvent une étape intermédiaire. Dans la plupart des applications, il faut savoir quelles entités ou quels aspects des entités sont les cibles des opinions. Sachant que certaines phrases ont des opinions positives ou négatives, mais pas sur quoi, a un usage limité. Cependant, les deux sous-tâches sont toujours utiles parce que (1) elle filtre les phrases qui ne contiennent aucune opinion, et (2) après que nous connaissons les entités et les aspects des entités dont on parle dans une phrase, cette étape peut nous aider déterminer si les opinions sur les entités et leurs aspects sont positives ou négatives.

Une grande partie de la recherche sur la classification des sentiments au niveau de la phrase fait l'hypothèse suivante :

Hypothèse : La phrase exprime une opinion unique d'un seul tenant de l'opinion.

Cette hypothèse n'est appropriée que pour des phrases simples avec une seule opinion, par exemple, "La qualité d'image de cette caméra est incroyable." Cependant, pour les phrases complexes , une seule phrase peut exprimer plus d'une opinion. Par exemple, la phrase "La qualité d'image de cet appareil photo est incroyable, tout comme la durée de vie de la batterie, mais le viseur est trop petit pour un appareil aussi génial", exprime à la fois des opinions positives et négatives. Pour "qualité d'image" et "durée de vie de la batterie", la phrase est positive, mais pour "viseur", elle est négative. Il est également positif pour la caméra dans son ensemble (c'est-à-dire, l'aspect GÉNÉRAL).

De nombreux articles ont été publiés sur la classification de la subjectivité et la classification des sentiments au niveau de la phrase. Pour la classification de la subjectivité, il a appliqué l'apprentissage supervisé. Pour la classification de sentiment de chaque phrase subjective, il a utilisé une méthode classification non supervisée.

2.8 Conclusion

Dans ce chapitre on a présenté les méthodes de prédiction dans la fouille de texte. La particularité des documents dans les réseaux sociaux avec un exemple de illustratif. Enfin la problématique d'analyse des sentiments a été montrée dans les réseaux sociaux. Ce qui va etre l'objectif du reste du projet.

Chapitre 3

Titre chapitre 3

3.1 Introduction

L'objectif du projet est de réaliser une analyse des sentiments pour les tweets avec les fonctionnalités de bases suivantes (voir Figure 3.1) :

- Création de la base de données qui stockera les tweets.
- Authentification avec un compte Twitter : Authentification sur l'application avec un compte Twitter pour récupérer les clés pour l'utilisation des différentes APIs.
- Mise en place de la collecte (récupération, traitement, stockage) : Implémentation de l' API de recherche pour la collecte et implémentation du traitement pour chaque tweet (par exemple, tokenisation,élimination de mot vide et lemmatisation).
- Manipulation de la base de données : Développer le fait de pouvoir collecter des informations de tweets.
- Mise en place de score émotif de tweets : implémentation d'une fonction de calcule du score émotif en utilisant un dictionnaire émotionnel.
- Mise en place du Front End : Implémentation d'une interface pour les utilisateurs.

3.2 Twitter

est un réseau social de microblogage géré par l'entrepriseTwitter Inc. Il permet à un utilisateur d'envoyer gratuitement de brefs messages, appelés tweets, sur internet. Il a été créé le 21 mars 2006 par Jack Dorsey, Even williams, Biz Stone et Noah Glass, et lancé en juillet de la même année. Le service est rapidement devenu populaire, jusqu'à réunir plus de 500 millions d'utilisateurs dans le monde fin février 2012 . Au 5 mars 2017, Twitter compte 313 millions d'utilisateurs actifs par mois avec 500 millions de tweets envoyés par jour et est disponible en plus de 40 langues.

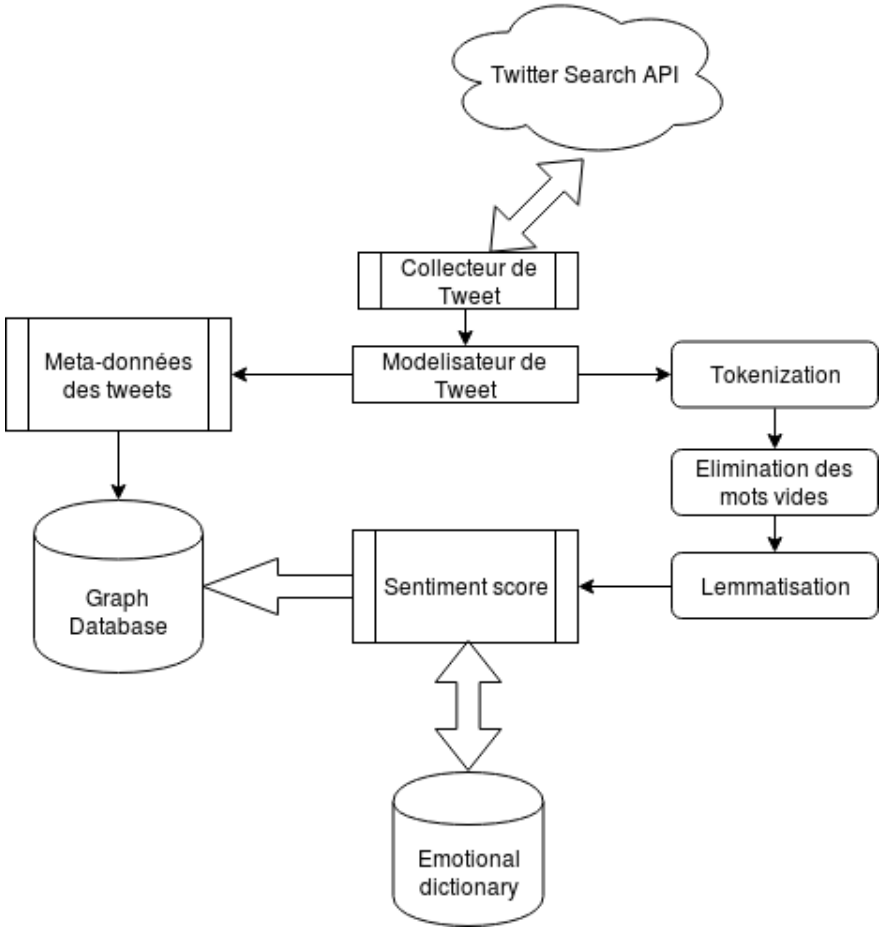

FIGURE 3.1 – Processus du mémoire

3.3 Twitter APIs

Twitter est ce qui se passe dans le monde et ce dont les gens parlent en ce moment. Vous pouvez accéder à Twitter via le Web ou votre appareil mobile. Pour partager les informations sur Twitter le plus largement possible, nous fournissons également aux entreprises, aux développeurs et aux utilisateurs un accès programmé aux données Twitter via nos API (interfaces de programmation d'applications).

À un niveau élevé, les API sont la façon dont les programmes informatiques «se parlent» afin qu'ils puissent demander et fournir des informations. Ceci est fait en permettant à une application logicielle d'appeler ce que l'on appelle un point de terminaison : une adresse qui correspond à un type spécifique d'information que nous fournissons (les extrémités sont généralement uniques comme des numéros de téléphone). Twitter permet l'accès à certaines parties de notre service via des API pour permettre aux gens de créer des logiciels qui s'intègrent à Twitter, comme une solution qui aide une entreprise à répondre aux commentaires des clients sur Twitter. Les données Twitter sont uniques à partir des données partagées par la plupart des autres plateformes sociales car elles reflètent les informations que les utilisateurs choisissent de partager publiquement. Notre plate-forme API offre un large accès aux données publiques Twitter que les utilisateurs ont choisi de partager avec le monde. Nous soutenons également les API qui permettent aux utilisateurs de gérer leurs

propres informations Twitter non publiques (par exemple, les messages directs) et de fournir cette information aux développeurs qu'ils ont autorisés à le faire.

3.3.1 accéder au données Twitter

Quand quelqu'un veut accéder à nos API, il doit enregistrer une application. Par défaut, les applications peuvent uniquement accéder aux informations publiques sur Twitter. Certains points de terminaison, tels que ceux chargés d'envoyer ou de recevoir des messages directs, nécessitent des autorisations supplémentaires de votre part avant de pouvoir accéder à vos informations. Ces autorisations ne sont pas accordées par défaut. vous choisissez par application si vous souhaitez fournir cet accès et pouvez contrôler toutes les applications autorisées sur votre compte. Les API Twitter incluent un large éventail de points de terminaison, qui se répartissent en cinq groupes principaux :

Comptes et utilisateurs : Nous permettons aux développeurs de gérer par programme le profil et les paramètres d'un compte, de mettre en sourdine ou de bloquer les utilisateurs, de gérer les utilisateurs et les abonnés, de demander des informations sur l'activité d'un compte autorisé, etc. Ces paramètres peuvent aider les services aux citoyens comme le Département de gestion des urgences du Commonwealth de Virginie qui fournit des informations aux résidents sur les réponses d'urgence et les alertes d'urgence.

Tweets et réponses (Twitter search API) : Nous mettons des tweets et des réponses publiques à la disposition des développeurs, et nous permettons aux développeurs de publier des Tweets via notre API. Les développeurs peuvent accéder aux Tweets en recherchant des mots clés spécifiques.

Messages directs (Direct Message API) : Nos points de terminaison Message Direct DM fournissent un accès aux conversations des utilisateurs qui ont explicitement accordé l'autorisation à une application spécifique. Nous ne vendons pas de messages directs. Nos API DM fournissent un accès limité aux développeurs pour créer des expériences personnalisées sur Twitter. Les entreprises peuvent créer ces expériences conversationnelles basées sur l'humain ou le chatbot pour communiquer directement avec les clients en matière de service client, de marketing et d'expérience d'engagement de la marque.

Créez des expériences client personnalisées avec notre plate-forme Direct Message. Ces Fonctionnalités :

- Envoyer et recevoir des messages directs
- Créez des messages qui s'affichent pour des scénarios spécifiques.
- Joindre des vidéos, des images et des GIF.
- Invite les utilisateurs pour des réponses structurées avec un menu d'options.
- Ajoutez des boutons pour créer des liens vers des sites Web, des liens vers des applications ou d'autres parties de Twitter.
- Propriétés pour aider à gérer la conversation entre plusieurs applications.
- Affiche une image et un nom de profil personnalisés dans un message direct.

Les publicités (Ads API) : Nous fournissons une suite d'API pour permettre aux développeurs d'aider les entreprises à créer et gérer automatiquement des campagnes publicitaires sur Twitter. Les développeurs peuvent utiliser des Tweets publics pour identifier des sujets et des intérêts, et fournir aux entreprises des outils pour mener des campagnes publicitaires afin d'atteindre les différents publics sur Twitter [1].

Outils de publication et SDK : Nous fournissons des outils pour les développeurs de logiciels et les éditeurs afin d'intégrer des chronologies Twitter, des boutons de partage et d'autres contenus Twitter sur les pages Web. Ces outils permettent aux marques d'intégrer des conversations publiques en direct de Twitter dans leur expérience Web et de faciliter le partage d'informations et d'articles sur leurs sites par leurs clients.

Dans l'ensemble de nos API et produits de données, nous prenons notre responsabilité de protéger sérieusement les données de nos utilisateurs. Nous maintenons des politiques et des processus stricts afin d'évaluer la façon dont les développeurs utilisent les données Twitter et de restreindre l'utilisation abusive de ces données. Lorsque nous apprenons qu'un développeur viole nos règles, nous prenons les mesures appropriées, qui peuvent inclure la suspension et la résiliation de l'accès aux API et produits de données de Twitter.

3.3.2 Twitter search API

L'API de recherche est désormais un élément essentiel de la programmation Twitter pour la collection de Tweets.

Elle a besoin d'OAuth pour se connecter à Twitter. L'API de recherche était limitée à l'adresse IP sans connexion requise, mais depuis la sortie de la version 1.1, vous devez vous connecter avec OAuth pour toutes les requêtes. L'API de recherche a des requêtes plus puissantes, elle dispose d'un ensemble assez riche d'opérateurs capables de filtrer les résultats en fonction d'attributs tels que l'emplacement de l'expéditeur, la langue et diverses mesures de popularité.L'API de recherche peut collecter une plus large gamme de données. La limite exacte des requêtes d'API de recherche n'est pas documentée, mais il est probable qu'une requête ne contienne pas plus de 15 à 20 mots-clés. D'un autre côté, vous pouvez effectuer jusqu'à 15 requêtes d'API de recherche par minute. Cela correspond à environ 250 mots-clés recherchés chaque minute, soit 15 000 mots-clés par heure.
Twitter search api est accessible sur :`https://api.twitter.com/1.1/search/tweets.json`
Exemples de demandes :

```
curl--requestGET--url'https://api.twitter.com/1.1/search/tweets.json?q=nasa&
result_type=popular'--header'authorization:OAuthoauth_consumer_key="consumer-key-for-
oauth_nonce="generated-nonce",oauth_signature="generated-signature",oauth_signature_
method="HMAC-SHA1",oauth_timestamp="generated-timestamp",oauth_token="access-token-fo
oauth_version="1.0"'
twurl/1.1/search/tweets.json?q=nasa&result_type=popular
```

1. L'API de publicité est accessible sur https ://ads-api.twitter.com

3.3.3 Connectez votre application à Twitter

Pour connecter votre application Auth0 à Twitter, vous devez générer des clés de consommateur et de secret dans une application Twitter, les copier dans vos paramètres Auth0 et activer la connexion.

Créez une application Twitter

1. Connectez-vous à **Twitter Application Management**.
2. Cliquez sur **Create New App**
3. Fournissez les informations requises. Pour Callback, entrez `https://YOUR_AUTHO_DOMAIN/login/callback`
4. Acceptez l'accord de développeur et cliquez sur Create your Twitter Application.
5. Une fois l'application créée, accédez à l'onglet Paramètres et vérifiez que l'option Autoriser cette application à être utilisée pour se connecter avec Twitter est sélectionnée.

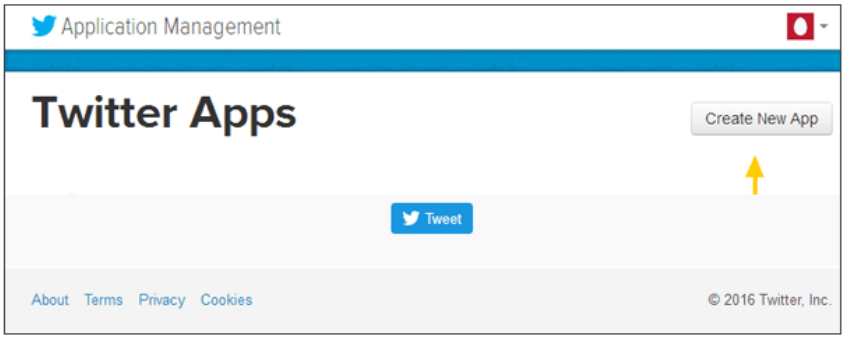

FIGURE 3.2 – Interface de création d'application Twitter

Obtenez votre Consumer Key et Consumer Secret

Votre Consumer Key et votre Consumer Secretr seront affichés dans l'onglet Keys and Access Tokens de votre application sur Twitter :

Copiez votre Consumer Key et votre Consumer Secret dans Auth0

1. Dans une fenêtre distincte, connectez-vous au Auth0 Dashboard et sélectionnez **Connections ¿ Social** dans la navigation de gauche.
2. Sélectionnez la connexion avec le logo Twitter pour accéder à la page Paramètres de cette connexion.
3. Copiez la **Consumer Key** et **Consumer Secure** à partir de l'onglet Keys and Access Tokens de votre application sur Twitter dans les champs de cette page sur Auth0.

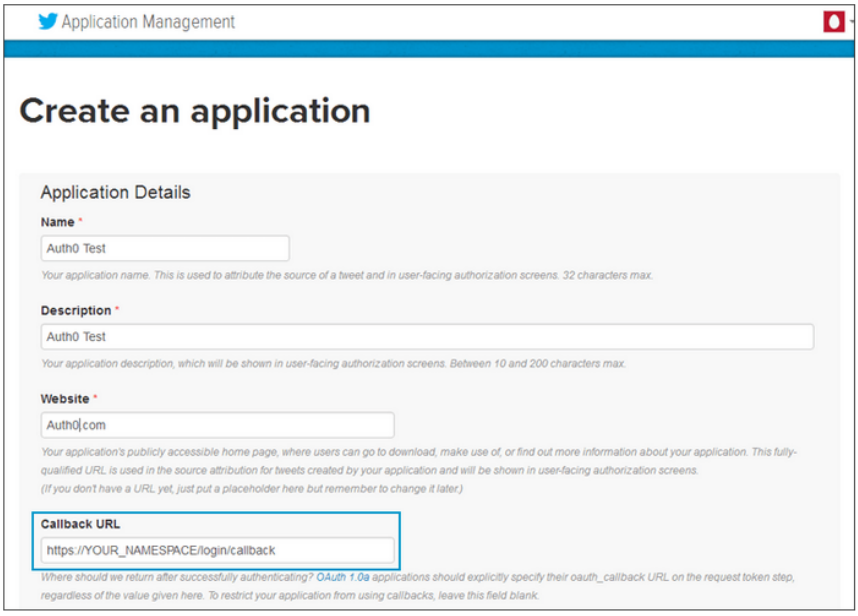

FIGURE 3.3 – Formulaire d'accès à l'application

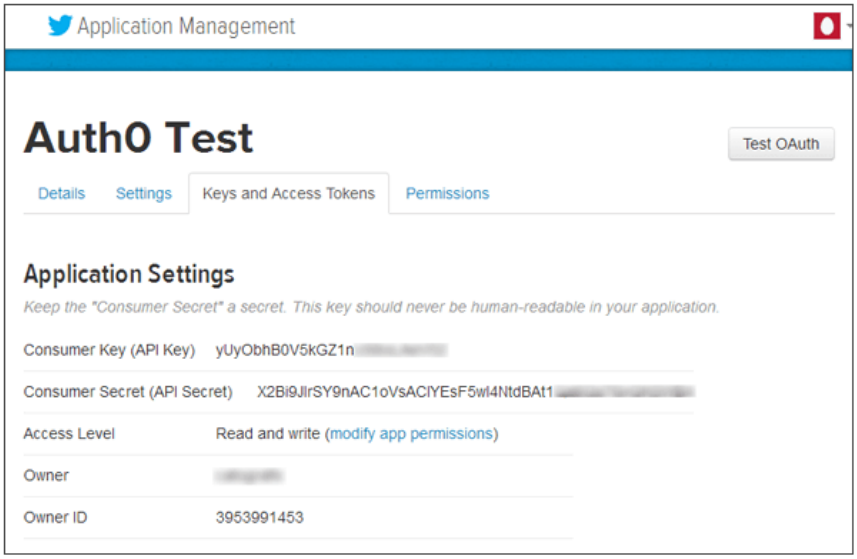

FIGURE 3.4 – Consumer Key de l'API

Activer la connexion

Accédez à l'onglet Applications de la connexion Twitter sur Auth0 et sélectionnez chacune de vos applications Auth0 existantes pour lesquelles vous souhaitez activer cette connexion :

3.4 Oscon graphique twitter

L'événement annuel d'informatique Oscon (O'Reilly Open Source Convention) qui discute des logiciels open source tels Linux, MySQL, Perl et Python et en utilisant l'API de recherche Twitter il a créer un graphique des utilisateurs , des tweets , des hashtags et des

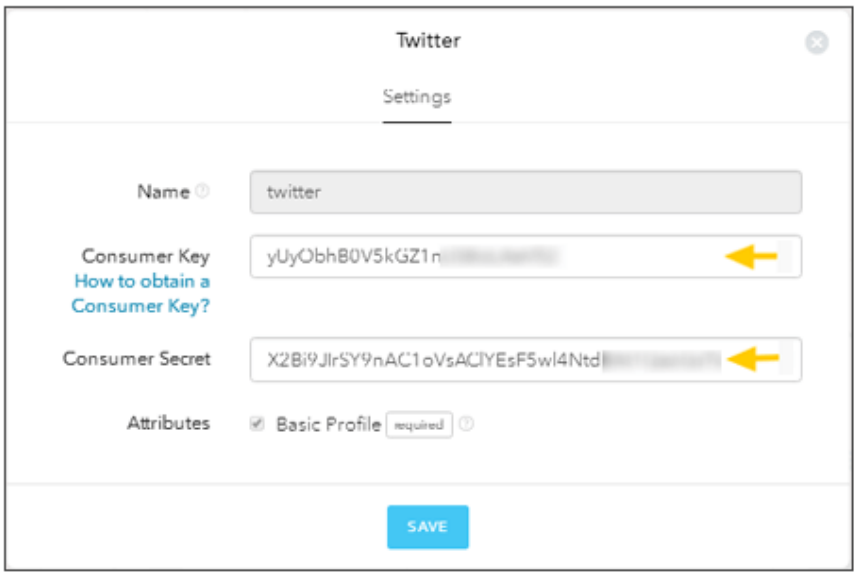

FIGURE 3.5 – Copie des clés de l'API

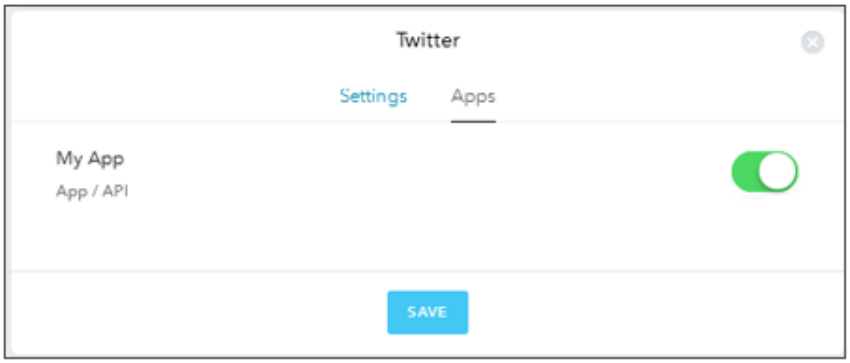

FIGURE 3.6 – Activation de la connexion

liens partagés à l'aide de neo4j. Le schéma ci_dessous explique le fonctionnement de twitter : L'utilisateur va poster un tweet, ce dernier peut mentionner un autre utilisateur. Le tweet peut être taggé par un hachtag comme il peut contenir des liens et utilise des source être un réponse d'un autre tweet ou retweeté par un autre tweet .

3.4.1 Méta données d'un tweet

Les tweets, considérés comme éphémères transportent un nombre impressionnant de métadonnées. Chaque tweet peut être incorporé dans un site web et donc doit contenir les informations pour se décrire de manière isolée de la timelime, fenêtre ou les tweets s'affichent au fur et à mesure de leur diffusion. Un tweet contient une liste de 31 métadonnées. Il connaît l'identité de son créateur (robot ou humain), la localisation de son endroit de création, la date, les retweets et bien d'autres petits éléments. Le texte d'un tweet représente moins de 10

```
1 {"id": "L'identifiant unique du tweet. Ces ID sont tries
```

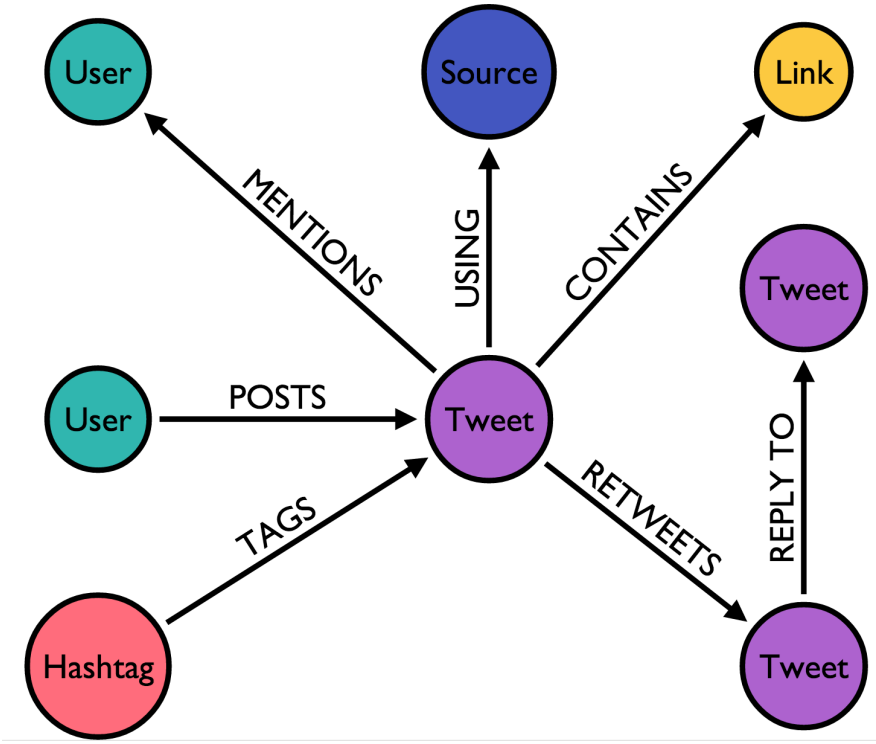

FIGURE 3.7 – Modèle de tweet

FIGURE 3.8 – exemple de réponse de l'api search

```
    grossierement et les developpeurs devraient les traiter
    comme opaques".
2
3 "text": "Le texte du tweet 140 caracteres."
4
5 "created_at": "Date de creation du Tweet."
6
7 "in_reply_to_user_id": "L'identifiant d'un tweet existant
```

```
   auquel ce tweet repond. Ne sera pas defini, sauf si l'auteur
    du tweet mentionne est mentionne."
8
9  "in_replyto_screen_name" : "Le nom d'ecran  de l'utilisateur
   ont repondu a l'auteur du tweet."
10
11 "in_reply_to_status_id" :  "l'ID de l'utilisateur ont repondu a
    l'auteur du tweet."
12
13 "Truncated" : "Tronque a 140 caracteres. Possible seulement de
   SMS."
14
15 "user" : {
16                 "id" : "L'identifiant de l'utilisateur de l'
                      auteur"
17                 "screen_name": "Le nom d'utilisateur de l'
                      auteur"
18                 "name" : "Le pseudonyme de l'auteur"
19                 "description": "La biographie de l'auteur"
20                 "url": "L'URL de l'auteur"
21                 "location": "La "localisation" de l'auteur C'
                      est un champ de texte de forme libre, et il
                      n'y a  aucune garantie sur le fait qu'il
                      peut etre geocode."
22                 "profile_background_color": "Rendre l'
                      information pour l'auteur. Les couleurs sont
                       codees en hexadecimal (RVB)"
23                 "profile_backgound_image_url":
24                 "profile_backgound_title":
25                 "profile_image_url":
26                 "profile_link_color":
27                 "created_at": "La date de creation de ce compte
                      "
28                 "contributors_enabled" : "Si ce compte a des
                      contributeurs actives"
29                 "favorites_count" : "Nombre de favoris que cet
                      utilisateur a"
30                 "statuses_count": "Nombre de tweets que cet
                      utilisateur a"
31                 "friends_count": "Nombre d'utilisateurs que cet
                       utilisateur suit"
32                 "time_zone": "Le fuseau horaire et le decalage
                      (en secondes) pour cet utilisateur
```

```
33                  lang: Langage selectionne de l'utilisateur"
34                  "protected": "Si cet utilisateur est protege ou
                        non. Si l'utilisateur est protege, ce tweet
                        n'est pas visible sauf pour les "amis"
35                  "followers_count": "Nombre d'abonnes a cette
                       utilisation"
36                  "geo_enabed": "Si cela a geo active"
37                  "verified": "Si cet utilisateur a un badge
                       verifie."
38          }
39 "contributors" : "Les ID utilisateur des contributeurs (le cas
       echeant)"
40 "place": {
41                   "id": "L'identifiant du lieu"
42                  "url": "L'URL pour aller chercher un polygone
                       detaille pour cet endroit"
43                  "name": "Les noms imprimables de ce lieu"
44                  "full_name": "Les noms imprimables de ce lieu"
45                  "place_type": "Le type de cet endroit peut etre
                        un quartier ou une ville. L'endroit associe
                        a ce Tweet"
46                  "country_code":  "Le pays dans lequel se trouve
                        cet endroit"
47                  "bounding_box":{
48                                          "coordinates": "
                                               longitude et
                                               latitude du lieu"
49                                          }
50          }
51 "source": "L'application qui a envoye ce tweet"
52
53 }
```

3.5 Analyse de texte

L'analyse naturelle du langage (NLP : Natural Language Processing) provient d'un processus automatique ou semi-automatique du langage humain. Le NLP fut développé autour de la recherche linguistique et des sciences cognitives, la psychologie, la biologie et les mathématiques. Dans le domaine particulier de l'informatique, la NLP est rattachée aux techniques de compilation, au théorie formelle du langage, à l'intéraction homme-machine, au "machine learning".

Python est un outil phénoménalement bon pour l'analyse de texte, et il y a quelques

bons outils que vous pouvez utiliser. Natural Language ToolKit (NLTK) est une boîte à outil permettant la création de programmes pour l'analyse de texte. Cet ensemble a été créé à l'origine par Steven Bird et Edward Loper, en relation avec des cours de linguistique informatique à l'Université de Pennsylvanie en 2001.

Exemple d'utilisation NLTK sur les tweets :
Tweet = "From pilot to astronaut, Robert H. Lawrence was the first African-American have chance to be selected as an astronautt"

— **Tokenisation** : Il s'agit du processus consistant à briser un flux de texte en plusieurs mots, phrases, symboles ou tout autre éléments significatifs dénommés Signes (tokens). Voilà ce que peut produire la fonction :

```
tokenizer = RegexpTokenizer(r'\w+')
tokenizer.tokenize(tweet)
```

la fonction retourne la liste contenant : ['From', 'pilot', 'to', 'astronaut', 'Robert', 'H', 'Lawrence', 'was', 'the', 'first', 'African', 'American', 'have', 'chance', 'to', 'be', 'selected', 'as', 'an', 'astronautt']

— **Élimination de mot vide :** Parfois, nous avons besoin de "raboter" des éléments inutiles afin que les données soient davatange traduisibles pour l'ordinateur. En NLP, de telles données (des mots) sont qualifiées par stop words. Par conséquent, ces mots n'ont aucune signification pour nous, et nous souhaiterions les retirer. La libraire NLTK contient quelques mots "d'arrêt" pour commencer ce traitement. Voilà ce que peut produire quand on applique la fonction sur le tweet précédant :

```
stop_words = set(stopwords.words('french'))
```

le résultat sera : ['From', 'pilot', 'astronaut', 'Robert', 'H', 'Lawrence', 'first', 'African', 'American', 'chance', 'selected', 'astronautt'].

— Lemmatisation la lemmatisation produit l'origine du mot comme par exemple le verbe (selected) il est conjugué au passé la lemmatisation donne l'infinitif du verb (select). Voilà ce que peut produire quand on applique la fonction sur le tweet précédant :

```
lemmatizer.lemmatize( )
```

['From', 'pilot', 'astronaut', 'Robert', 'H', 'Lawrence', 'first', 'African', 'American', 'chance', 'select', 'astronautt']

3.6 Base de données orientée graphe

Nous allons parler de l'avènement des bases de données orientées graphes en expliquant pourquoi les bases préexistantes ne peuvent répondre aux problématiques des données fortement connectées.

3.6.1 Données de plus en plus connectées

Nous n'avons jamais créés et stockés autant de données qu'à l'heure actuelle. Dès que nous naviguons sur Internet nous produisons des données, via les tracker pour les statistiques, les réseaux sociaux, nos achats en ligne, nos emails ...

Hier grâce aux ordinateurs, aujourd'hui via nos smartphones et demain avec l'Internet des objets, le volume de données ne va faire qu'augmenter. Bref, nous sommes en plein dans le BigData. Mais que se passerait-il si nous arrivions à mettre en relation toutes ces données ? Simplement, nous ferions comme Facebook, Google, LinkedIn ou eBay.

Ces grands acteurs du Web ont bien compris que connecter ou lier les données permet d'accroître considérablement leur valeur. Ainsi, nous ne sommes plus uniquement dans le Bigdata mais également dans l'ère du LinkedData : les données deviennent de plus en plus complexes et connectées.

Alors une question qui se pose : comment modéliser, stocker et requêter ces relations entre les données ?

3.6.2 Bases de données relationnelles

Dès qu'on doit stocker des données, on pense aux bases de données relationnelles. Mais sont-elles faites pour stocker des données connectées ?

Lorsqu'on veut créer une relation entre deux objets, il est fréquent de devoir créer une table n-tiers permettant de les lier.

Ceci implique que nous devons modifier le schéma de la base pour chaque nouvelle relation qu'on veut créer. Dans notre contexte, cette non-flexibilité allonge les temps de développement. De plus, en SQL, la description d'une relation se traduit par l'ajout d'un « JOIN ». Or, avec des données connectées, on obtient rapidement des requêtes avec beaucoup de JOIN, ce qui les rend complexes et donc difficilement maintenables.

De surcroît, plus on a de données dans une table et moins bonnes sont les performances. Pour calculer le résultat, les moteurs SQL font (quasiment) le produit cartésien de chacune des tables. Dans notre contexte, les bases relationnelles sont inappropriées, surtout si l'on veut faire du temps réel :

— schéma non flexible
— mauvaises performances
— code complexe et difficilement maintenable.

3.6.3 Bases de données orientées Graphes

Les bases de données orientées graphes vous permettent de modéliser, stocker et requêter en temps réel vos données connectées. Ici, on ne parle plus de table ou de document, mais de nœud et de relation. Le principe est simple : ce que vous Dès qu'on doit stocker des données, on pense aux bases de données relationnelles. Mais sont-elles faites pour stocker des données connectées ?

Lorsqu'on veut créer une relation entre deux objets, il est fréquent de devoir créer une table n-tiers permettant de les lier. Ceci implique que nous devons modifier le schéma de la

base pour chaque nouvelle relation qu'on veut créer. Dans notre contexte, cette non-flexibilité allonge les temps de développement. De plus, en SQL, la description d'une relation se traduit par l'ajout d'un « JOIN ». Or, avec des données connectées, on obtient rapidement des requêtes avec beaucoup de JOIN, ce qui les rend complexes et donc difficilement maintenables.

De surcroît, plus on a de données dans une table et moins bonnes sont les performances. Pour calculer le résultat, les moteurs SQL font (quasiment) le produit cartésien de chacune des tables. Dans notre contexte, les bases relationnelles sont inappropriées, surtout si l'on veut faire du temps réel :modélisez sur un tableau blanc est votre modèle physique !

Les bases de données orientées graphes ne sont pas la réponse à tout, il faut utiliser le bon outil pour le bon besoin. Voici une liste de questions à vous poser avant de partir sur une base de données orientée graphe :

— Vos données sont-elles dynamiques ?
— Vos données sont-elles connectées ?
— Avez-vous besoin d'un schéma flexible ?
— Devez-vous faire du temps réel ?

3.6.4 Présentation Neo4j

Neo4j est une base de données orientée graphe, libre (sous licence GPLv3) et écrite en Java. Développée par Neo Technology (une société suédoise dont le siège est aux US), les premières lignes de codes datent de l'année 2000 et la version 1.0 est sortie en 2010. Ceci en fait l'une des premières bases de données orientées graphes, mais aussi l'une des plus évoluées et robustes. Ses principales caractéristiques sont les suivantes :

— **transaction** : c'est une base de données transactionnelle, respectueuse des principes ACID ;
— **Haute disponibilité** : via la mise en place d'un cluster ;
— **Volume** : stocker et requêter des milliards de nœuds et de relations ;
— **Cypher** : un langage de requête graphe déclaratif, simple et efficace ;
— **schemaless** : pas de schéma préétabli.

3.6.5 Concepts de Neo4j

Les bases de données orientées graphes tournent autour de trois concepts : les nœuds, les relations et leurs propriétés.

Noeuds : L'unité fondamentale qui forme un graphe est le nœud. Les nœuds sont des enregistrements composés de propriétés de type clef/valeur, sans schéma préétabli. Généralement, ils représentent une entité du modèle. Pour différencier les nœuds, Neo4j apporte la notion de label. Ceux-ci permettent de donner un rôle ou un type à un nœud (un nœud peut avoir plusieurs labels).

Relations : Les relations entre les nœuds sont la clef de voûte des graphes, c'est ce qui permet de lier des données et de créer des structures comme des listes, des arbres, des maps

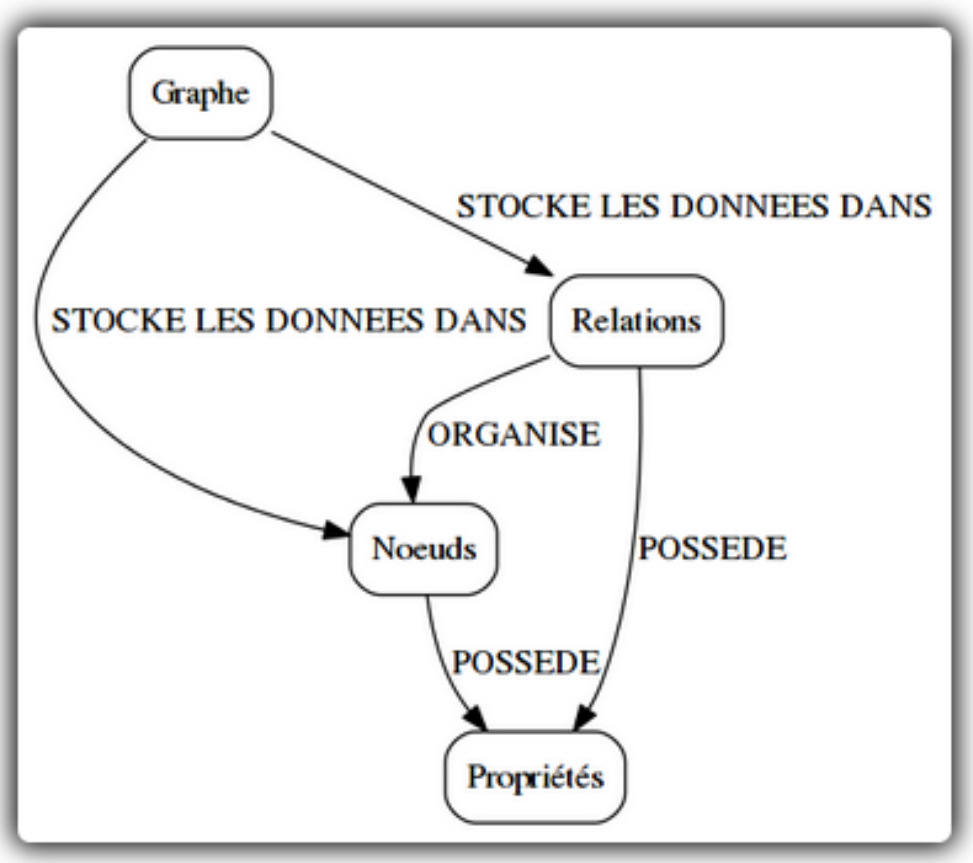

FIGURE 3.9 – Concepts de Neo4j

...

Neo4j les définit comme étant constituées d'un nœud de départ, d'arrivée (donc une relation avec un sens) et d'un type. De plus, tout comme les nœuds, elles sont également un enregistrement, et donc peuvent avoir des propriétés de type clef/valeur.

Propriétés : Nous avons vu que les nœuds et les relations peuvent avoir des propriétés. Leurs types possibles correspondent aux types primitifs de Java, ou à un tableau de type primitif.

Cypher : Cypher est un langage déclaratif permettant de requêter et mettre à jour le graphe. Inspiré du SQL, on y retrouve beaucoup de concepts familiers, comme les clauses WHERE, ORDER BY, SKYP, LIMIT ...

Son objectif est de permettre à l'utilisateur de définir des motifs, qui seront par la suite recherchés dans tout le graphe. Ainsi, si je veux les amis de mes amis, il faut décrire le motif suivant : Mais comment faire pour décrire ce genre de motif dans un langage textuel ? Tout simplement en faisant de L'ASCII art :

(moi) -[:AMI]-> (mesAmis) -[:AMI]-> (amisDeMesAmis)

Les nœuds sont représentés avec des parenthèses, ce qui ressemble à des cercles : () Si vous avez besoin d'identifier le nœud dans votre requête (dans une clause WHERE par exemple), il suffit de lui donner un nom : (monNoeud)

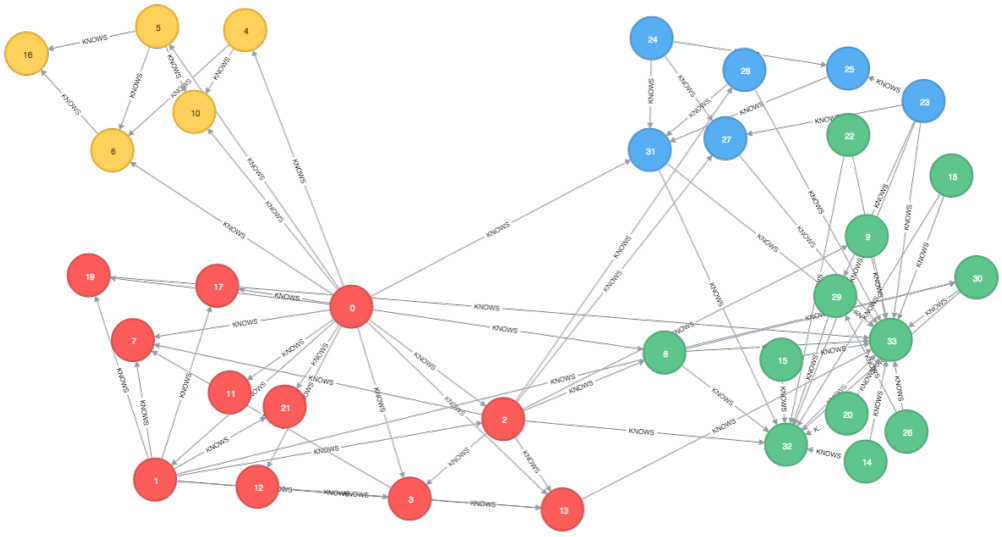

FIGURE 3.10 – Exemple de graphe Neo4j

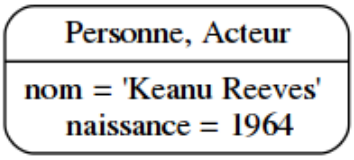

FIGURE 3.11 – Exemple de noeud Neo4j

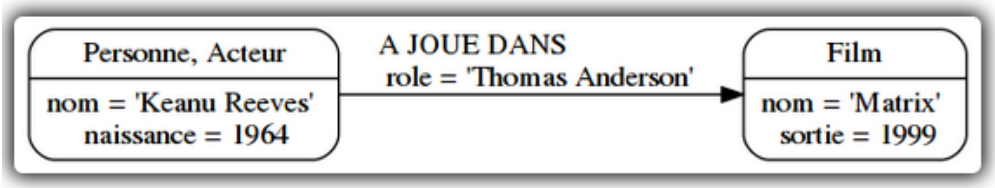

FIGURE 3.12 – Exemple de relations Neo4j

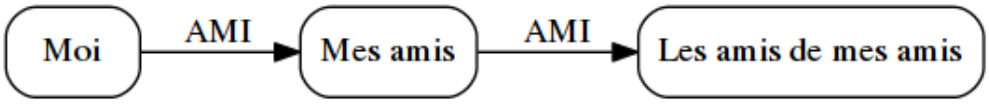

FIGURE 3.13 – Exemple illustratif cypher

Pour spécifier un label, il suffit de l'ajouter comme ceci : (monNoeud :monLabel). Voici quelques exemples :

1. () : n'importe quel nœud ;
2. (:Personne) : un nœud avec le label Personne ;
3. (n :Personne) : un nœud identifié dans la variable n avec le label Personne ;
4. (n :Personne :Acteur) : un nœud identifié dans la variable n avec le label Personne et Acteur.

Les relations sont représentées par deux tirets avec un '¿', ce qui ressemble à une flèche : –¿

Si vous avez besoin d'identifier la relation dans votre requête, vous pouvez lui donner un nom comme ceci : -[maRelation]-¿

Pour spécifier le type de la relation, il suffit de l'ajouter comme ceci : -[maRelation :MON_TYPE]-¿ . Voici quelques exemples :

1. (a)–(b) : n'importe quelle relation entre le nœud a et b (peu importe la direction) ;
2. (a)-[:AMI]-¿(b) : relation de type AMI depuis le nœud a vers le nœud b ;
3. (a)-[r :AMI—CONNAIT]-¿(b) : relation identifiée dans la variable r de type AMI ou CONNAIT depuis le nœud a vers le nœud b.

3.7 Score de sentiment :

3.7.1 Dictionnaire

L'analyse du sentiment de microblogs tels que Twitter a récemment gagné beaucoup d'attention. Les méthodes computationnelles pour évaluer le sentiment des algorithmes d'apprentissage comme des réseaux bayésiens naïfs, des machines vectorielles de soutien et des approches d'entropie maximum pour effectuer une analyse conceptuelle du texte en langage naturel. Une exigence pour ces approches est un texte de qualité suffisante pour permettre des évaluations précises du langage naturel. Certains chercheurs affirment que ce n'est pas disponible dans les extraits de texte tribunaux comme des tweets, des messages instantanés ou des SMS. Plusieurs chercheurs ont identifié une méthode alternative : l'une des approches d'analyse du sentiment les plus simples compare les mots d'une publication à une liste de mots étiquetés. Il existe plusieurs listes de mots affectifs, par exemple, ANEW (Normes affectives pour les mots anglais) développées avant l'avènement du microblogage et de l'analyse des sentiments. Notre dictionnaire des sentiments fournit des mesures de valence et d'excitation (arousal) pour environ 10 680 mots anglais. Chaque mot est évalué sur une échelle de neuf points allant de 1 à 9. Les mots inclus dans le dictionnaire ont été sélectionnés à partir de recherches antérieures qui les ont identifiés comme de bons candidats pour transmettre l'émotion. Par exemple, pour construire le dictionnaire ANEW, les volontaires ont été invités à lire un corpus de texte et à fournir une évaluation le long de chaque dimension pour chaque occurrence d'un mot reconnu par ANEW. Les notes attribuées à un mot commun sont combinées en une note moyenne et un écart-type des notes pour chaque dimension. Par exemple, pour le mot **house** , ANEW rapporte :

house , $v = (\mu : 7.26, \sigma : 1.72), a = (\mu : 4.56, \sigma : 2.41), fq = 591$

- **v** : valence (le type de l'adjectif si alla est positive ou negative).
- **a** : arousal (la physiologie de la personne conte cette adjectif si elle est haute ou faible).
- μ : valence moyenne / excitation moyenne
- σ : un écart-type
- **fq** : fréquence

3.7.2 Modèle du Russel

Des modèles émotionnels ont été proposés pour définir et comparer les états émotionnels. Ces modèles utilisent souvent des dimensions émotionnelles pour positionner les émotions sur un plan 2D. Les modèles les plus simples représentent la valence le long d'un axe horizontal, avec très désagréable d'un côté, très agréable de l'autre. Les modèles plus complexes utilisent plus d'une seule dimension. Par exemple, Russell a proposé d'utiliser la valence le long de l'axe horizontal et l'excitation (arousal) le long des axes verticaux. Les termes intermédiaires excited – depressed et distressed – relaxed sont des opposés polaires formés par des états intermédiaires de valence et d'excitation. Des modèles similaires ont été proposés par Watson et Tellegen (avec des axes de valence positifs et négatifs), Thayer (avec des axes de tension et d'énergie), et Larsen et Diener (avec des axes de valence et d'activation similaires à ceux de Russell).

3.7.3 Approche pour calculer le score de sentiment d'un tweet

On a le tweet qui se trouve dans le section précédente :
Tweet = "From pilot to astronaut, Robert H. Lawrence was the first African-American have chance to be selected as an astronautt".

Étape 1 : On applique les méthode de traitement de texte pour obtenir un tweet prêt a utilisé (la tokenisation, l'élimination du mot vide et le lemmatisation). Ça nous produit les lemmes suivantes : ['From', 'pilot', 'astronaut', 'Robert', 'H', 'Lawrence', 'first', 'African', 'American', 'chance', 'select', 'astronaut']

Étape 2 : On cherche dans le dictionnaire ANEW si se trouve les lemmes de notre tweet, après la recherche on a trouvé que le dictionnaire ANEW contient les mots (chance, astronaut).

Étape 3 : chance , $v = (\mu : 6,02, \sigma : 1,77), a = (\mu : 5,38, \sigma : 2,58)$.
astronaut , $v = (\mu : 6,66, \sigma : 1,60), a = (\mu : 5,28, \sigma : 2,11)$.
On calcule le score du sentiment à l'aide de la fonction suivante pour la valence et aussi pour l'excitation (arousal) :

$$Scoreemotionel = \frac{\sum_{i=1}^{N} \frac{\mu_i}{\sigma_i}}{\sum_{i=1}^{N} \frac{1}{\sigma_i}} \tag{3.1}$$

$Valence = 6,35$.
$Arousal = 2,93$.
On projette la valeur de valence sur l'axe horizontale et la valeur de arousal sur l'axe verticale dans le modèle du Russel. Le résultat sera (relaxed et excited).

3.8 Application

L'analyse de textes et de sentiments sur les réseaux sociaux est un enjeu important pour différentes activités telles que la mise en place de stratégies d'affaires ou de politiques publiques. Aujourd'hui les services de microblogging (twitter) attirent l'attention en raison des grandes masses de donnes qu'ils véhiculent. Analyser ces donnés c'est à dire que comprendre la grande quantité de messages recueillis à chaque instant et montrer des informations utiles aux utilisateurs est un problème difficile.Il est donc crucial de développer et connaître les méthodes efficaces de visualisation de texte et plus spécifiquement de sentiments pour aider les utilisateurs dans leur tâche d'analyse. On a développe une application web qui utilise twitter pour analyser ces tweets.

3.8.1 Environnement de travail

D'abord, on va donner une description concernant l'environnement notre application :

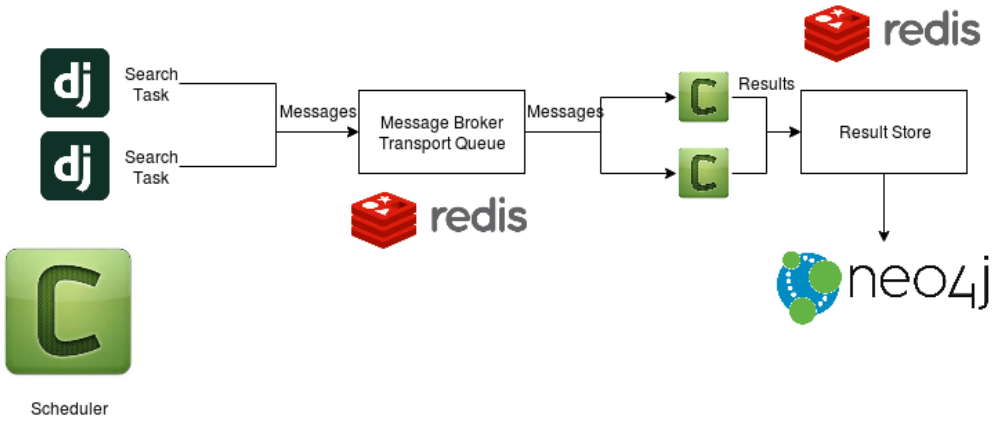

FIGURE 3.14 – Architecture de l'application

Environnement materiel

Dans le but d'aboutir notre application, nous avons utilisé un PC marque Acer, équipé d'un processeur multi-core I3, cadencé par une horloge d'une fréquence de 2.53GHz, avec 2.6 Gi Octets de RAM, un disque dur d'une capacité de 146.5 Gi Octets.

Environnement logiciel

Nous avons utilisé le langage de programmation Python. Python est un langage de programmation portable, dynamique, extensible, gratuit, qui permet une approche modulaire et orientée objet de la programmation.Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles. Pour se focaliser sur notre application et tirer profit des puissance du langage Python,nous avons utilisé les packages suivants :

Package CSV : CSV (Comma Separated Values) module pour lire et écrire des données au format CSV.

Package re : (Regular expressions) Ce module fournit des opérations correspondant aux expressions régulières.

Package numpy : numpy (NUMeric Python) est une bibliothèque numérique apportant le support efficace de larges tableaux multidimensionnels, et de routines mathématiques de haut niveau (algèbre linéaire, statistiques, .. etc.).

Package Nltk : Nltk (Natural Language Toolkit) est une plate-forme pour la création de programmes Python pour travailler avec des données de langage humain.

Package os : OS (Operating System) permet d'utiliser les fonctionnalités dépendantes du système d'exploitation.

Package time : le package qui manipule le temps de façon simple.

Package json JSON (JavaScript Object Notation), est un format d'échange de données léger inspiré par la syntaxe littérale d'objet JavaScript.

Package Twython : module qui Prend en charge les API Twitter normales et en streaming.

On a utilisé de plus des outils performants et tendances qui nous ont aidé à développer notre application sont les suivants :

Django : est un framework Python de code source ouvert. Il a pour but de rendre le développement web 2.0 simple et rapide.Il se constitue de :

La couche model : Django fournit une couche d'abstraction (les "models") pour structurer et manipuler les données de l'application Web.

La couche Vue : Une vue est le concept Django pour encapsuler la logique de traitement des requêtes utilisateur et de leurs réponses.

La couche template : fournit une syntaxe abordable pour la mise en page des données à présenter aux utilisateurs.

Redis : est une base de données open source de type clefs-valeurs. C'est en gros une grosse HashMap, mais avec des données structurées : des chaînes de caractères, des listes, des hash, des set, des set triés. L'utilisation de redis est très simple et la vitesse de lecture et d'écriture est vertigineuse (plus de 100 000 insertions par seconde). Toutes les opérations sont atomiques, vous ne risquez pas d'avoir des soucis de concurrence d'accès à vos données.

Par contre, il est impossible de requêter les valeurs comme on le fait habituellement avec un WHERE en MySQL, mais avec un peu d'astuce, d'habitude on arrive très vite à nos fins en demandant "la bonne clef".

Celery : est une file d'attente de tâches asynchrones basée sur le passage de messages distribués. Il est axé sur le fonctionnement en temps réel, mais prend également en charge la planification. Les unités d'exécution, appelées tâches, sont exécutées simultanément sur un ou plusieurs serveurs de travail. Les tâches peuvent s'exécuter de manière asynchrone (en arrière-plan) ou de manière synchrone (attendre jusqu'à la fin).

Neo4j : est une base de données orientée graphe, où on a stocké les données des tweets.

3.8.2 Interface

On a construit une interface web pour permettre à l'utilisateur de dialoguer avec notre application en utilisant le framework django et le framework bootstrap collaboré avec HTML,CSS et JS . Elle se compose de plusieurs pages on va les détailler si dessous :

Bootstrap : est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web.

HTML (HyperText Markup Language) : est le langage de balisage conçu pour mettre en forme le contenu des pages web.

CSS (Cascading Style Sheets) : elle décrit la présentation des documents HTML .

JS (Java Script) : est un langage de programmation de scripts principalement utilisé dans les pages web.

Page d'accueil :

La page d'accueil contient un navigateur qui nous dirige vers les autres page,et en plus des descriptions pour la partie technique et architecturale, ainsi tous les outils utilisés dans la conception de cette application avec les liens de leurs sites.

Page de recherche :

Dans la page de recherche se trouve le formulaire qui va nous aider à à collecter les tweets en tapant un tag désiré.

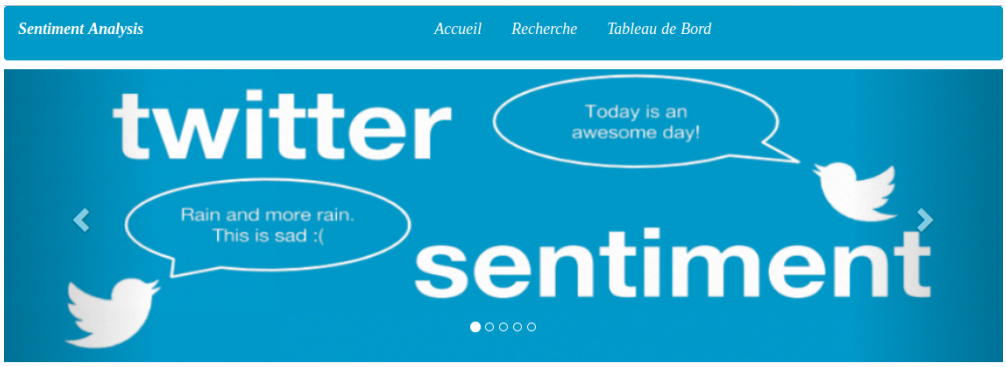

FIGURE 3.15 – Navigateur et Caroussel

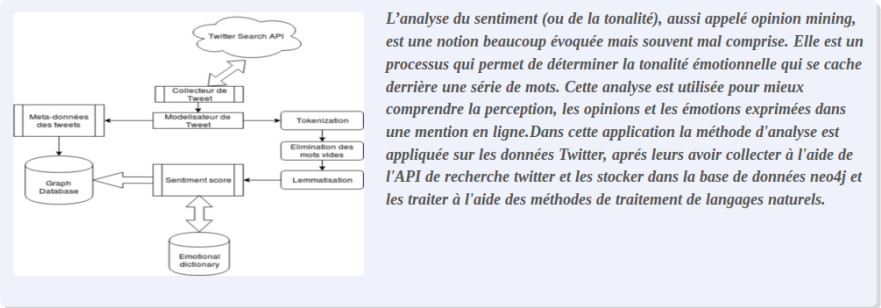

FIGURE 3.16 – Description du processus de l'application

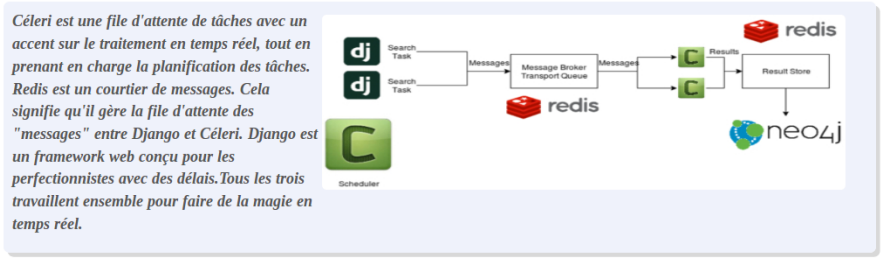

FIGURE 3.17 – Description d'architecture de l'application

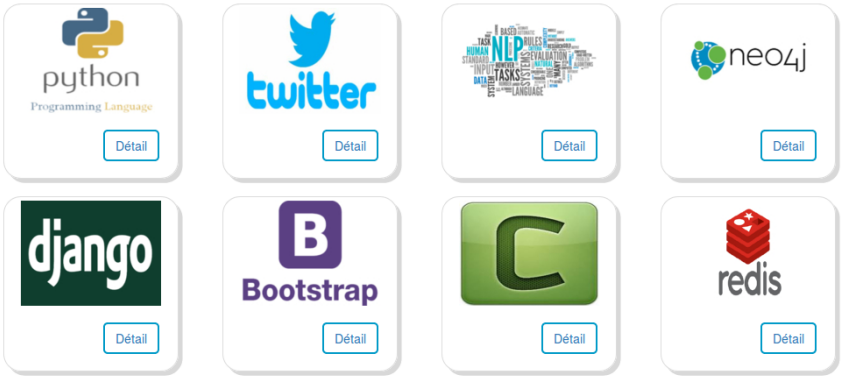

FIGURE 3.18 – Outils utilisés dans l'application

FIGURE 3.19 – Page de recherche

Page de tableau de bord :

Nuage de mots-clés (Word Cloud) : est une représentation visuelle des mots-clefs les plus utilisés dans les tweets qu'on a collecté. Généralement, les mots s'affichent dans des tailles et graisses de caractères d'autant plus visibles qu'ils sont utilisés ou populaires. on a développé ce nuage à l'aide de D3.js

FIGURE 3.20 – Exemple de word cloud

D3.js : (Data-Driven Documents) est une bibliothèque graphique JavaScript qui permet l'affichage de données numériques sous une forme graphique et dynamique.

prédiction positif et négatif : Elle classe l'ensemble des tweet collectés en tweet positifs et négatifs.

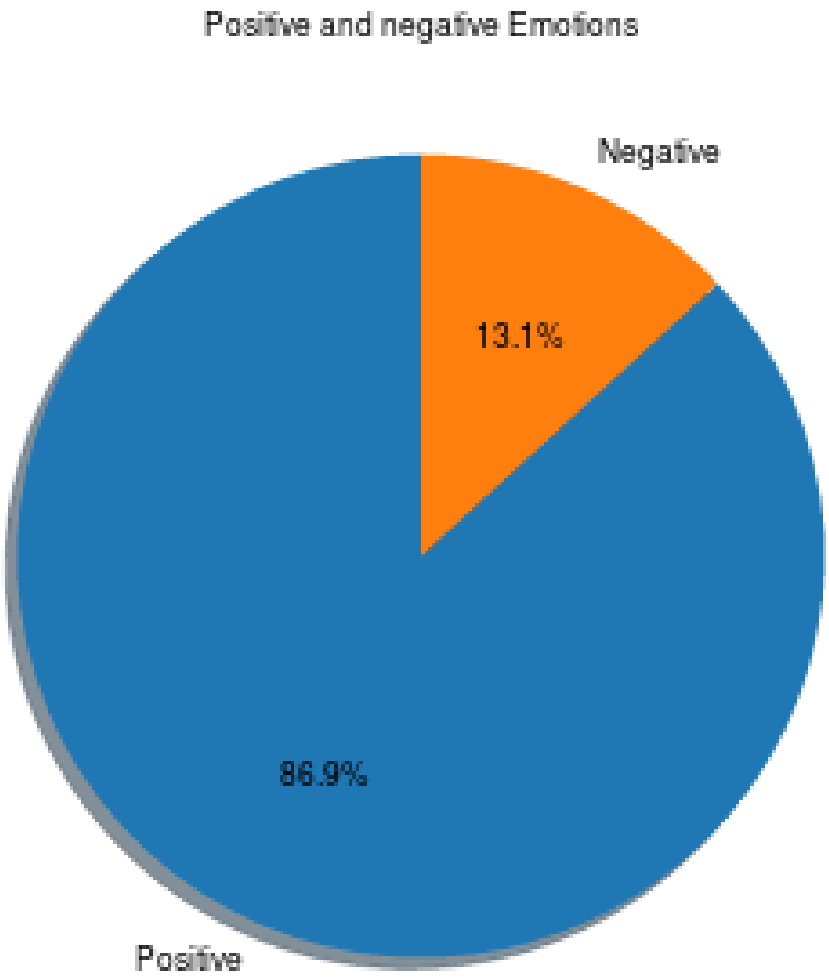

FIGURE 3.21 – Exemple de classification

FIGURE 3.22 –

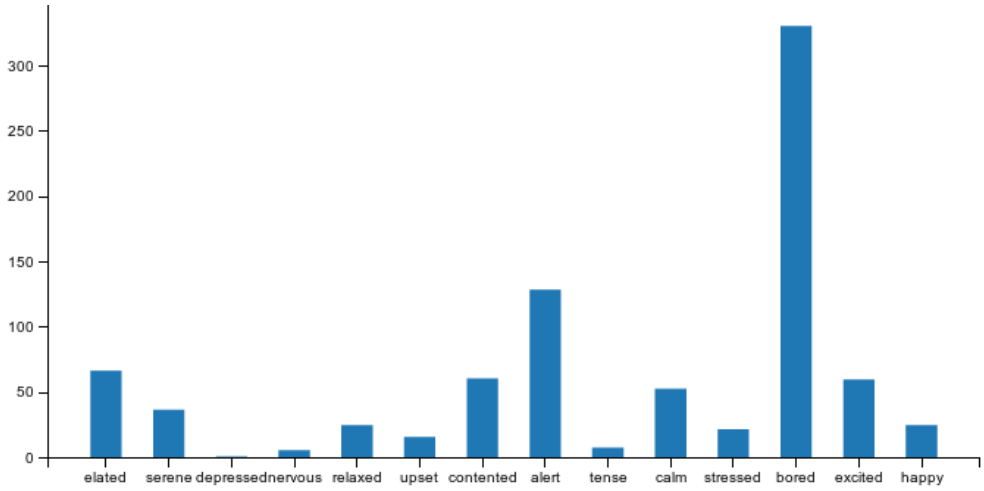

FIGURE 3.23 –

3.9 Conclusion

L'analyse des sentiments se réfère à l'extraction automatique de texte évaluative, qui aide à produire des résultats prédictifs. Dans ce mémoire nous avons étudié d'analyse des sentiments celle appliquée sur les données Twitter uniqument. on a appliqué une méthode qui calcule le score émentionnel des tweets aprés leurs avoir fait le prétraitement. Par ailleurs, le temps octroyé pour notre mémoire ne nous a pas permis d'explorer d'autres méthodes de l'analyse des sentiments telles que la méthode de Support Vector Machine (SVM)ou d'autre.

Conclusion générale

Les données textuelles dans les médias sociaux portent des informations abondantes. Elle fournissent des informations diverses et uniques sous forme de commentaires, d'articles et de tags. Les informations efficaces cachées dans les ressources textuelles des médias sociaux fournissent aux chercheurs de différentes règles des opportunités d'exploiter des modèles et des informations d'intérêt qui pourraient ne pas être évidents. Dans ce chapitre, on a discuté sur des aspects distincts des données textuelles dans les médias sociaux et de leurs défis, et a développé le travail actuel d'utilisation des méthodes d'analyse de texte pour résoudre les problèmes dans les médias sociaux. Un système de recherche en temps réel capable de trouver, de résumer et de suivre les dernières nouvelles ou événements mis à jour dans les communautés sociales sera très difficile mais utile. Comme nous l'avons discuté, le texte court joue un rôle très important dans les médias sociaux. D'une part, ces données textuelles contiennent moins d'informations que les documents standards ; d'un autre côté, il nous donne la possibilité d'utiliser des modèles traditionnels de NLP basés sur la syntaxe pour effectuer une analyse textuelle de niveau fin, qui prenait beaucoup de temps pour le texte standard.

Les données croisées se réfèrent ici à des données de différents formats ou données provenant de différentes ressources de médias sociaux. Les types de données de variance dans les médias sociaux, y compris le texte, l'image, le lien ou même les données multilingues, ont des relations latentes et des interactions entre les uns et les autres. En outre, un moyen efficace et efficient d'intégrer ces types de données sera très utile pour résoudre le problème de la rareté des données.

Le grand volume et la présentation compacte mais bruyante des données textuelles dans les médias sociaux empêchent les utilisateurs d'accéder facilement aux informations pour rechercher, naviguer et localiser facilement les messages spécifiques qui pourraient les intéresser. Trouver un moyen efficace de gérer ces types de données à grande échelle très difficile.

Bibliographie

[1] Agrawal, R., Chakrabarti, S., Dom, B. E., and Raghavan, P. (2001). Multilevel taxonomy based on features derived from training documents classification using fisher values as discrimination values. US Patent 6,233,575.

[2] Apte, C., Damerau, F., Weiss, S., et al. (1998). *Text mining with decision rules and decision trees*. Citeseer.

[3] Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM.

[4] Culotta, A. (2010). Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the first workshop on social media analytics*, pages 115–122. ACM.

[5] Culotta, A., McCallum, A., and Betz, J. (2006). Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296–303. Association for Computational Linguistics.

[6] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM.

[7] Fayyad, U. M. and Irani, K. B. (1992). The attribute selection problem in decision tree generation. In *AAAI*, pages 104–110.

[8] Feldman, R., Fresko, M., Kinar, Y., Lindell, Y., Liphstat, O., Rajman, M., Schler, Y., and Zamir, O. (1998). Text mining at the term level. In *European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 65–73. Springer.

[9] Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., and Trigg, L. (2009). Weka-a machine learning workbench for data mining. In *Data mining and knowledge discovery handbook*, pages 1269–1277. Springer.

[10] Fuller, C. M., Biros, D. P., and Delen, D. (2011). An investigation of data and text mining methods for real world deception detection. *Expert Systems with Applications*, 38(7) :8392–8398.

[11] Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM.

[12] Harper, F. M., Moy, D., and Konstan, J. A. (2009). Facts or friends ? : distinguishing informational and conversational questions in social q&a sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 759–768. ACM.

[13] Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

[14] Keller, F. and Lapata, M. (2003). Using the web to obtain frequencies for unseen bigrams. *Computational linguistics*, 29(3) :459–484.

[15] Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the special issue on the web as corpus. *Computational linguistics*, 29(3) :333–347.

[16] Kim, S.-B., Han, K.-S., Rim, H.-C., and Myaeng, S. H. (2006). Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11) :1457–1466.

[17] Lau, R. Y., Liao, S., Kwok, R. C. W., Xu, K., Xia, Y., and Li, Y. (2011). Text mining and probabilistic language modeling for online review spam detecting. *ACM Transactions on Management Information Systems*, 2(4) :1–30.

[18] Lin, S.-H., Yeh, Y.-M., and Chen, B. (2011). Leveraging kullback–leibler divergence measures and information-rich cues for speech summarization. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4) :871–882.

[19] Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

[20] Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2) :159–165.

[21] Luhn, H. P. (1960). Key word-in-context index for technical literature (kwic index). *Journal of the Association for Information Science and Technology*, 11(4) :288–295.

[22] Mani, I. and Maybury, M. T. (1999). *Advances in automatic text summarization*. MIT press.

[23] Meyer, D., Hornik, K., and Feinerer, I. (2008). Text mining infrastructure in r. *Journal of statistical software*, 25(5) :1–54.

[24] Mittermayer, M.-A. and Knolmayer, G. F. (2006). Newscats : A news categorization and trading system. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 1002–1007. Ieee.

[25] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking : Bringing order to the web. Technical report, Stanford InfoLab.

[26] Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10.

[27] Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2) :1–135.

[28] Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users : real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.

[29] Sasaki, M. and Shinnou, H. (2005). Spam detection using text clustering. In *Cyberworlds, 2005. international conference on*, pages 4–pp. IEEE.

[30] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1) :1–47.

[31] Soriano, J., Au, T., and Banks, D. (2013). Text mining in computational advertising. *Statistical Analysis and Data Mining : The ASA Data Science Journal*, 6(4) :273–285.

[32] Wang, J. H., Chung, E. S., and Jang, M. G. (2008). Semi-automatic construction method for knowledge base of encyclopedia question answering system. US Patent 7,428,487.

[33] Weiss, S. M., Indurkhya, N., Zhang, T., and Damerau, F. (2010). *Text mining : predictive methods for analyzing unstructured information*. Springer Science & Business Media.