République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida USDB

Faculté des sciences, département d'informatique

MEMOIRE DE MASTER

En Informatique

Spécialité : Système Informatique Et Réseaux

Thème

ANALYSE DE SENTIMENTS DANS LES DONNEES SOCIALES MASSIVES EN UTILISANT L'APPRENTISSAGE AUTOMATIQUE

LIEU DU STAGE : Centre de recherche sur l'information scientifique et technique

Réalisé par :

- **SAHNOUNE Youcef Abdelali**
- **GUERROUMI Ayoub**

Encadré par :

- Dr.ALIANE HASSINA, MRA CERIST
- Dr.CHERIGUENE SORAYA

Soutenu le 9.12.2020 devant le jury composé de :

Mme GEUSSOUM Dalila	Université de Blida 1	Présidente
Mme HIRECHE Célia	Université de Blida 1	Examinatrice

Année Universitaire 2019 – 2020

Remerciement

Tout d’abord et avant tout, nous tenons à remercie dieu le tout puissant de nous avoir donné la force, la volonté et le courage de réaliser ce modeste travail.

Nos vifs sincères remerciements S’adressent spécialement à,

« Mme. ALIANE »

Dont nous avons eu la chance de l'avoir comme Professeur, Encadreur et qui a bien voulu nous a confié ce riche travail d'expérience et nous a guidés à chaque étape de sa consécration. Vous m'avez toujours réservé un chaleureux accueil, malgré vos obligations et les contraintes professionnelles. Vos talents ainsi que vos compétences et votre sens du devoir m’ont marqué à jamais. Vos encouragements inlassables, votre amabilité, votre gentillesse et votre patience méritent toute notre attention. Veuillez trouver ici l’expression de notre estime et considération.

Et à « Mme. Cheriguene»

Nous tenons à exprimer nos profondes gratitudes à Mme. Cheriguene, pour avoir encadré et dirigé ce travail. Nous la remercie tout d’abord pour sa disponibilité et ses réponses dans les bons moments. Ses précieux conseils, son exigence et ses commentaires ont permis d'améliorer grandement la qualité de ce travail. Merci Madame !

Nous remercions aussi,

Les membres du jury de nous avoir fait l’honneur de juger cette thèse. Veuillez accepter l'expression de notre profonde gratitude.

Enfin,

A toutes les personnes qui ont contribué de près ou de loin, d’une manière directe ou indirecte à l’élaboration de ce travail de fin d’études

Dédicace

Je dédie ce modeste travail accompagné d'un profond amour :

À celle qui m'a arrosé de tendresse et d'espoirs, à la source d'amour Incessible, à la mère des sentiments fragiles qui ma bénie par ces prières…. Ma mère.

À mon support dans ma vie, qui m'a appris, m'a supporté et ma dirigé vers la gloire…. Mon père.

À mes chères sœurs que je les souhaite à leur tour d'arriver là et rendre nos parents heureux.

À toutes les personnes de ma grande famille et à tous mes amis.

Mais aussi à mon Professeur de 1er année Math et Informatique Monsieur HABANI d'avoir m'encourager, me conseiller et cru en moi quand j'étais dans le flou, je vous dis

Merci !

M. Sahnoune Youcef Abdelali

Dédicace

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

A toute ma famille, pour leur soutien tout au long de mon parcours universitaire, Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible, Merci d'être toujours là pour moi.

M. Geurroumi Ayoub

Résumé

L’analyse des sentiments est l’utilisation de langage naturel pour automatiser la classification de sentiment à partir d’un texte généralement non structuré.

L'extraction de l'opinion par l'analyse des ‘Big Social Data’ a connu une croissance explosive en raison de leur nature interactive, en temps réel. C’est donc dans ce contexte que nous nous intéressons particulièrement aux méthodes d’analyse du Big data. La problématique qui se pose et qui a fait naissance à notre recherche est comment peut-on analyser les données sociales massives car ces données sont si grandes qu’elles en deviennent difficiles à gérer avec les outils classiques.

Dans ce mémoire, nous expérimentons trois techniques d’apprentissage automatique, à savoir Naïves Bayes, Régression Logistique et Machine à vecteurs de support pour l’analyse des Tweets dans un grand ensemble de données en anglais. Pour valider notre étude nous avons utilisé un très grand ensemble de données contenant 1600000 Tweets partitionné en positif et négatif. Nous avons utilisé apache Spark comme Framework et le système de fichier distribuer HDFS de Hadoop pour le stockage et le traitement de l’ensemble de données.

Plusieurs expérimentations ont été effectuées pour avoir une meilleure performance des algorithmes utilisés et ce, en introduisant différentes stratégies de prétraitement. L’étude comparative avec d’autres systèmes de classification existants dans la littérature a montré que nos algorithmes sont compétitifs. En effet notre système est capable d’atteindre une moyenne de précision de 76.60%.

Mots clés : analyse des sentiments, donnes massives, apprentissage automatique, réseaux social, donnée social massive.

Abstract

Sentiment analysis is the use of natural language to automate sentiment classification from generally unstructured text.

The mining of public opinion through analysis of 'Big Social Data' has experienced explosive growth due to its interactive, real-time nature. In this context that we are particularly interested in big data analysis methods. The problem that arises and that gave birth to our research is how we can analyze massive social data because this data is so large that it becomes difficult to manage with traditional tools.

Therefore, the proposed solutions for big data sentiment analysis largely depend on the use of two methods based on machine learning techniques and lexicon based methods.

This report discusses the use of three machine learning techniques Naïve Bayes, Regression Logistics and Support Vector Machine (SVM) for the analysis of Tweets in English. To validate our study we used a very large dataset containing 1 600 000 Tweets partitioned into positive,negative.
We used Apache Spark as the framework and Hadoop's HDFS Distribute File System for the storage and processing of the dataset.

Several experiments have been made to have a better performance and to assess the relevance of our system, using different preprocessing strategies such as Stop Words Removing, Stemming and Tokenisation. The comparative study with other classification systems existing in the literature confirmed that our algorithms are precise and very competitive. Indeed, our system is capable of achieving an average precision of 76.60%.

Keywords : sentiment analysis, Big data, machine learning, social networks, Big social data

ملخص

شهد التنقيب عن الرأي العام من خلال تحليل "البيانات الاجتماعية الكبيرة" نموًا هائلاً بسبب طبيعته التفاعلية في الوقت الفعلي. في الواقع، ترتبط بيانات وسائل التواصل الاجتماعي ارتباطًا وثيقًا بالحياة الشخصية "لرواد الأنترنيت" لذا يمكن استخدامها لمرافقة الأحداث المهمة (الانتخابات، شراء المنتجات، إلخ)، من خلال تتبع سلوكهم. لذلك في هذا السياق، نحن مهتمون بشكل خاص بطرق تحليل البيانات الضخمة. تكمن المشكلة في أن هذه البيانات كبيرة جدًا بحيث يصعب إدارتها باستخدام الأدوات التقليدية.

يناقش هذا التقرير استخدام ثلاث تقنيات من تقنيات التعلم الآلي لتحليل التغريدات باللغة الإنجليزية. للتحقق من صحة دراستنا، استخدمنا مجموعة بيانات كبيرة جدًا تحتوي على 1600000 تغريدة مقسمة إلى إيجابية وسلبية. استخدمنا لتخزين ومعالجة مجموعة البيانات Hadoopالخاص بـ HDFS ' كإطار عمل ونظام 'Apache Spark
تم إجراء العديد من التجارب للحصول على أداء أفضل ولتقييم مدى ملاءمة نظامنا باستخدام استراتيجيات معالجة مسبقة مختلفة مثل إزالة الكلمات والترميز.

أكدت الدراسة المقارنة مع أنظمة التصنيف الأخرى الموجودة في الأدبيات أن خوارزمياتنا دقيقة وتنافسية للغاية. في الواقع، نظامنا قادر على تحقيق متوسط دقة يبلغ 76.60٪.

الكلمات المفتاحية: تحليل المشاعر، البيانات الضخمة، التعلم الآلي، الشبكات الاجتماعية، البيانات الاجتماعية الضخمة

Table de matières

Liste des figures

Liste des tableaux

Introduction Générale

Nous vivons dans un monde numérique où nous sommes devenus accro à l'internet plus particulièrement aux réseaux sociaux. Avec le développement des appareils intelligents et l'apparition du web 2.0, la donnée numérique est en augmentation exponentielle, des tonnes de données sont générées par les 'socionautes' à travers le monde en des fractions de seconde. Ces derniers expriment leurs idées, opinions sur des divers sujets tels que le marketing, la santé, la finance et la politique.

Au fil des années 2000 des chercheurs ont pensé que l'exploitation et le traitement de ces données volumineuses peut s'avérer utile pour la prise de décision, soit pour les organisations et les entreprises en analysant l'attitude et le sentiment des clients envers leurs produits, services soit pour les consommateurs en prenant en considération les avis des autres.

L'analyse de sentiments est un des nouveaux défis apparus en traitement automatique des langues avec l'avènement des réseaux sociaux sur le WEB. Profitant de la quantité d'information maintenant disponible, la recherche et l'industrie se sont mises en quête de moyens pour analyser automatiquement les opinions exprimées dans les textes.

Dans ce contexte, plusieurs travaux de recherche ont été menés en exploitant les données des différents réseaux sociaux comme Facebook, Twitter et Instagram, dans les différentes langues (Français, Anglais, Chinois …etc.) avec des techniques différentes comme l'utilisation de lexiques et de l'apprentissage automatique.

Dans notre recherche, nous avons utilisé les données de la plus grande plateforme de microblogging Twitter. Nous avons expérimenté trois algorithmes d'apprentissage automatique les plus utilisés en analyse de sentiments à citer, Naïve Bayes, Régression Logistique et Machine à vecteurs de support sur un grand dataset. Pour gérer et stocker notre large ensemble de données qui contient 1 600 000 Tweets nous avons utilisé le système de fichier distribué de Hadoop 'HDFS' et nous avons utilisé PySpark qui est la collaboration d'Apache Spark et de Python pour la programmation.

Ce mémoire se compose de cinq chapitres. Le premier couvre le concept du BIG DATA. Dans le deuxième chapitre nous décrivons et montrons l'importance de l'analyse de sentiments dans le 'Big Social Data. Puis à la fin de ce chapitre, nous présenterons les méthodes d'apprentissage automatique existant dans ce contexte. Et le chapitre 3 a été consacré à la classification de sentiment et son processus bien détaillé, à la fin de ce chapitre nous définissons l'apprentissage automatique et ses différents algorithmes de classification. Dans le 4ème chapitre nous décrivons notre système et l'architecture Big data. Enfin dans le chapitre 5 nous parlons de l'implémentation et l'expérimentation, nous y discutons aussi les résultats obtenus. Et nous parachèverons notre travail avec une conclusion.

Chapitre 1 : Big data

1 Introduction

De nos jours, presque toutes nos actions laissent une trace numérique. Nous générons des données chaque fois que nous allons en ligne, lorsque nous transportons nos smartphones équipés d'un GPS, lorsque nous communiquons avec nos amis par le biais de médias sociaux ou d'applications de chat, et lorsque nous faisons des achats. On pourrait dire que nous laissons des empreintes numériques avec tout ce que nous faisons qui implique une action numérique, ce qui est presque tout. De plus, la quantité de données générées par les machines augmente rapidement. Les données sont générées et partagées lorsque nos appareils domestiques « intelligents » communiquent entre eux ou avec leurs serveurs domestiques. Les machines industrielles dans les usines et les usines du monde entier sont de plus en plus équipées de capteurs qui recueillent et transmettent les données.

Le terme « Big Data » désigne la collecte de toutes ces données et notre capacité à les utiliser à notre avantage dans un large éventail de domaines, y compris les affaires.

C'est donc dans ce contexte que nous nous intéressons dans ce 1er chapitre à la définition de ce concept très important pour la compréhension de contexte.

2 Définition Big Data

Le Big Data s'agit d'un ensemble de technologie, d'architecture ,d'outil et de procédure permettant à une organisation de très rapidement capter, traiter et analyser une grande quantités de données.

L'expression « Big Data » date de 1997 selon l'Association for Computing Machinery. En 2001, l'analyste du cabinet Meta Group Doug Laney décrivait les Big Data d'après le principe des « trois V » :

- le Volume de données de plus en massif ;
- la Variété de ces données qui peuvent être brutes, non structurées ou semi-structurées ;
- la Vélocité qui désigne le fait que ces données sont produites, récoltées et analysées en temps réel.

D'autres organisations et praticiens du Big Data (par exemple, des chercheurs, des ingénieurs, etc.) ont étendu ce modèle 3V à un modèle 4V en incluant un nouveau « V » : Valeur (Value), fait référence au processus d'extraction d'informations précieuses à partir de grands ensembles de données sociales. Ce modèle peut même être étendu à 5V si les concepts de Véracité, qui se réfère à l'exactitude et la précision des informations. Derrière toute pratique de gestion de l'information se cachent les doctrines fondamentales de la qualité des données.

3 Défis Big data

Les opportunités sont toujours suivies de défis. D'une part, le Big Data apporte de nombreuses opportunités attractives. D'autre part, nous sommes également confrontés à de nombreux défis lorsque nous traitons des problèmes de Big Data, les difficultés résident dans la capture, le stockage, la recherche, le partage, l'analyse et la visualisation des données. Si nous ne pouvons pas surmonter ces défis, le Big Data deviendra un minerai d'or, mais nous n'avons pas les capacités pourl'explorer, l'un des principaux défis d'analyse Big data c'est d'arriver à stocker et analyser ce

gros volume de données, Le Big Data a changé la façon dont nous capturons et stockons les données, y compris le dispositif de stockage de données, l'architecture de stockage de données, le mécanisme d'accès aux données. Comme nous avons besoin de plus de supports de stockage et d'une vitesse d'E / S plus élevée pour relever les défis, ce stockage doit faciliter la mise à l'échelle du hardware (scalable), pour ça certaine utilise des outils de stockage distribué comme les base de données NoSQL, Hadoop, Spark, En relation avec le volume se trouve la vélocité, un défi qui pousse certaines vers une nouvelle génération d'outils analytiques réduisant fortement le temps nécessaire pour générer des rapports, L'organisation Apache a quelques solutions populaires, parmi lesquelles Spark et Kafka. Spark est idéal pour le traitement par lots et le traitement en flux, Ces outils aident à prendre des décisions rapides grâce à des analyses en temps réel. Quant à la gestion des données, vu que ces données sont variées avec différents formats, l'exploitation peut s'avérer très difficile, pour pallier ce problème, des outils d'intégration conçus pour faciliter le processus, Il s'agit souvent d'un système de stockage de fichiers comme Amazon S3, qui collecte les données sous sa forme brute et les transformer plus tard.

Toutefois, de nombreuses entreprises déclare qu'il n'arrive pas encore à réussir à relever ce défi tellement c'est pénible. La sécurité est une autre dimension, Pour les applications liées au Big Data, les problèmes de sécurité des données sont plus délicats pour plusieurs raisons. Premièrement, la taille du Big Data est extrêmement importante, canalisant les approches de protection.

Deuxièmement, cela entraîne également une charge de travail beaucoup plus lourde de la sécurité. De plus, la plupart des Big Data sont stockés de manière distribuée et les menaces des réseaux peuvent également aggraver les problèmes.

4 Outils d'analyse Big data

Dans cette section nous parlons des outils du Big Data dans les deux traitements par lots et par flux

4.1 Basés sur le traitement par lots

Apache Hadoop est l'un des outils Big Data basés sur les processus par lots les plus célèbres et les plus puissants. Il fournit des infrastructures et des plates-formes pour d'autres applications spécifiques Big Data. Un certain nombre de systèmes de Big Data spécifiés (tableau 1) sont construits sur Hadoop et ont des utilisations spéciales dans différents domaines, par exemple, l'exploration de données et l'apprentissage automatique utilisés dans les affaires et le commerce [7].

Tableau 1 - Outils Big Data basés sur le traitement par lots [7].

Nom	**Usage spécifique**	**Avantage**
Apache Hadoop	Infrastructure et plateforme	Évolutivité, fiabilité et exhaustivité élevées
Dryad	Infrastructure et plateforme	Moteur d'exécution distribué haute performance, bonne programmabilité
Apache Mahout	Algorithmes d'apprentissage automatique en entreprise	Bonne maturité
Jaspersoft BI Suite	Logiciel de Business Intelligence	BI rentable et en libre-service à grande échelle
Pentaho Business Analytics	Plateforme d'analyse commerciale	Robustesse, évolutivité, flexibilité dans la découverte des connaissances
Skytree Server	Apprentissage automatique et analyse avancée	Traitez des ensembles de données massifs avec précision à grande vitesse
Tableau	Visualisation de données, analyse commerciale,	Tableaux de bord plus rapides, intelligents, adaptés, beaux et faciles à utiliser

Karmasphere Studio and Analyst	Espace de travail Big Data	Analyses collaboratives et basées sur des normes, et libre-service
Talend Open Studio	Gestion des données et intégration d'applications	Environnement graphique facile à utiliser basé sur les éclipses

- **Apache Hadoop et MapReduce**

Apache Hadoop est l'une des plates-formes logicielles les mieux établies qui prennent en charge les applications distribuées gourmandes en données. Il implémente le paradigme de calcul nommé MapReduce. La plate-forme Apache Hadoop (voir figure 1) se compose du noyau Hadoop, du système de fichiers distribué MapReduce et Hadoop (HDFS), ainsi que d'un certain nombre de projets connexes, notamment Apache Hive, Apache HBase, etc.

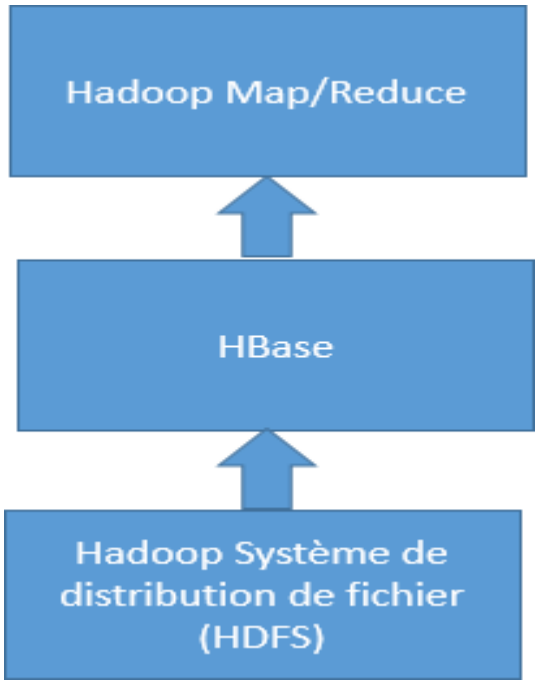

Figure 1 - Architecture du système Hadoop [7].

MapReduce, qui est un modèle de programmation et d'exécution pour le traitement d'un grand volume de données, basé sur la méthode diviser pour mieux régner et fonctionne en décomposant récursivement un problème complexe en de nombreux sous-problème, ces sous-problèmes sont affectés à un groupe de nœud de travail et résolus de manière distincte et parallèle. Enfin, les solutions aux sous-problèmes sont ensuite combinées pour donner une solution au problème d'origine.

- **Apache mahout**

Apache Mahout, vise à fournir des techniques d'apprentissage automatique évolutives et commerciales pour des applications d'analyse de données à grande échelle et intelligentes, De nombreuses grandes sociétés telles que Google, Amazon, Yahoo ! IBM, Twitter et Facebook, ont mis en œuvre des algorithmes d'apprentissage automatique évolutifs dans leurs projets. Beaucoup de leurs projets impliquent des problèmes de Big Data et Apache Mahout fournit un outil pour atténuer les grands défis. Les algorithmes de base de Mahout, y compris le clustering, la classification, l'extraction de motifs, la régression, la réduction de dimension, les algorithmes évolutifs et le filtrage collaboratif basé sur les lots, s'exécutent au-dessus de la plateforme Hadoop via le cadre MapReduce. Ces algorithmes dans les bibliothèques ont été bien conçus et optimisés pour offrir de bonnes performances et capacités.

4.2 Basé sur le traitement par flux

Hadoop réussit bien à traiter une grande quantité de données en parallèle. Il fournit un mécanisme de partitionnement général pour distribuer la charge de travail d'agrégation sur différentes machines.

Néanmoins, Hadoop est conçu pour le traitement par lots. Il s'agit d'un moteur polyvalent mais pas d'un moteur en temps réel et hautes performances, car il existe une latence élevée dans ses implémentations. Pour certaines applications de données de flux, telles que le traitement de fichiers journaux, l'industrie avec capteur, la machine à machine (M2M) et la télématique nécessitent une réponse en temps réel pour le traitement d'une grande quantité de données de flux. Dans ces applications, le traitement de flux pour l'analyse en temps réel est fortement

nécessaire. Les plates-formes Big Data, telles que SQLstream, Storm et StreamCloud, sont spécialement conçues pour l'analyse des données de flux en temps réel. Le traitement en temps réel signifie que le traitement des données en cours nécessite fortement une latence de réponse très faible. Par conséquent, il n'y a pas trop d'accumulation de données à la dimension temporelle pour le traitement. En général, les Big Data peuvent être collectées et stockées dans un environnement distribué, pas dans un centre de données. La caractéristique de latence élevée de Hadoop rend presque impossible l'analyse en temps réel. Plusieurs outils Big Data basés sur le traitement de flux ont été développés ou en cours de développement. Storm est l'une des plates-formes les plus connues. Parmi les autres, citons S4, SQLstream, Splunk, Apache Kafka et SAP Hana.

Tableau 2 - Outils Big Data basés sur le traitement par flux [7].

Nom	**Usage spécifique**	**Avantage**
Storm	Système de calcul temps réel	Mise à l'échelle, tolérance aux fautes, configuration facile
S4	Traitement de flux continus illimités	Plateforme éprouvée, distribuée, évolutive, tolérante aux pannes et enfichable
SQLstream s-Server	Applications de capteurs, M2M et télématiques	Plateforme Big Data de streaming en temps réel, basée sur SQL

Splunk	Collecter et exploiter les données de la machine	Environnements dynamiques rapides et faciles à utiliser, évolutifs du laptop au centre de données
Apache Kafka	Système de messagerie de publication-abonnement distribué	Flux à haut débit de données d'activité immuables
SAP Hana	Plateforme pour les entreprises en temps réel	Calcul rapide en mémoire et analyse en temps réel

- **Apache Storm**

Storm est un système de calcul en temps réel distribué et tolérant aux pannes pour le traitement de données de streaming illimité. Il est très facile à configurer et à utiliser, et garantit que toutes les données seront traitées. Il a de nombreuses applications, telles que l'analyse en temps réel, le système d'exploitation interactif, l'apprentissage automatique en ligne, le calcul continu, le RPC distribué. Pour implémenter le calcul en temps réel sur Storm, les utilisateurs doivent créer différentes topologies. Une topologie est un graphique de calcul et peut être créée et soumise dans n'importe quel langage de programmation. Un cluster Storm se compose de deux types de nœuds de travail, un nœud maître et plusieurs nœuds esclaves. Ces derniers implémentent deux types de démons : Nimbus et Supervisor. Nimbus est en charge de la distribution et la planification des travaux affectant les tâches aux nœuds esclaves, il surveille l'ensemble du système. En cas de défaillance du cluster, le Nimbus le détectera et réexécutera la tâche correspondante. Le supervisor se conforme aux tâches assignées par Nimbus, et démarre ou arrête les processus de travail selon les besoins sur la base des instructions de Nimbus. Un autre type de démon appelé Zookeeper joue un rôle important pour coordonner le système. Il enregistre tous les états du Nimbus et des superviseurs sur le disque local.

AHA Rahnama 2014 [4] propose un système distribué qui marche en temps réel pour le big data. Des algorithmes d'analyse en ligne sont utilisés pour garantir l'aspect temps réel, une plate-forme de calcul distribué basé sur apache Storm, et Vertical Hoeffding Tree un arbre de décision parallèle basée sur VFDT (very fast décision tree) pour une classification parallèle sur un environnement distribué.

- **Apache Kafka**

Kafka fonctionne comme un outil pour gérer le streaming et les données opérationnelles via des techniques analytiques en mémoire pour obtenir une prise de décision en temps réel. En tant que système de messagerie de publication-abonnement distribué, Kafka a quatre caractéristiques principales : messagerie persistante avec des structures de disque O (1), haut débit, prise en charge du traitement distribué et prise en charge du chargement parallèle des données dans Hadoop. Il a déjà de nombreuses utilisations dans un certain nombre de sociétés différentes en tant que pipelines de données et outils de messagerie. Ces dernières années, l'activité et les données opérationnelles ont joué un rôle important pour extraire les fonctionnalités des sites Web. Les données d'activité sont l'enregistrement de diverses actions humaines en ligne, telles que le contenu d'une page Web, le contenu de copie, la liste de clics et la recherche de mots clés. Il est utile de déconnecter ces activités dans un fichier standard et de les agréger pour une analyse ultérieure. Les données opérationnelles sont des données décrivant les performances des serveurs, par exemple, l'utilisation du processeur et des E / S, les temps de demande, les journaux de service, etc. La découverte des connaissances des données opérationnelles est utile pour la gestion des opérations en temps réel. Kafka combine le traitement hors ligne et en ligne pour fournir un calcul en temps réel et produire une solution ad hoc pour ces deux types de données.

5 Big Social data

Les médias sociaux sont devenus l'une des sources de données les plus représentatives et pertinentes pour le Big data. Les données des médias sociaux sont

générées à partir d'un grand nombre d'applications Internet et de sites Web, dont les plus populaires sont Facebook, Twitter, LinkedIn, YouTube, Instagram, Google, Tumblr, Flickr et WordPress. La croissance rapide de ces sites Web permet aux utilisateurs d'être connectés et a créé une nouvelle génération de personnes, qui sont enthousiastes à l'idée d'interagir, de partager et de collaborer en utilisant ces sites. Le mot Big Social data Proviennent d'unir deux domaines :

Les médias sociaux et le Big data. Par conséquent, il peut être défini comme processus et méthodes, qui sont conçus pour fournir des connaissances sensibles et pertinentes à tout utilisateur ou entreprise, à partir de sources de données de médias sociaux, lorsque les sources de données peuvent être caractérisées par leurs différents formats et contenus, leur très grande taille et la génération en ligne ou en continu d'informations.

Enfin, pour résumé les trois domaines de base pour le Big Social data : les médias sociaux comme source naturelle pour l'analyse des données, ces lieux de prises de parole des internautes, espace d'échanges et de discussions ; le Big data en tant qu'environnement de traitement parallèle et massif ; et l'analyse des données comme un ensemble d'algorithmes et de méthodes utilisés pour extraire et analyser les connaissances.

6 L'analyse de sentiments et le Big social data

Les médias sociaux et leurs applications correspondantes permettent à des millions d'utilisateurs d'exprimer et de diffuser leurs opinions sur un sujet et de montrer leurs attitudes en aimant ou en détestant le contenu. Toutes ces actions qui s'accumulent constamment sur les médias sociaux génèrent des données à volume élevé, à grande vitesse, à grande variété, à valeur élevée et à variabilité élevée, appelées big data. En général, ce type de données fait référence à un ensemble massif d'opinions qui pourraient être analysées, traitées pour déterminer les tendances des personnes dans le domaine numérique.

Plusieurs chercheurs ont montré un vif intérêt pour l'exploitation des données massives sociales afin de décrire, déterminer et prédire les comportements humains dans plusieurs domaines. Le traitement de ce type de donnée implique diverses pistes de recherche, notamment l'analyse de sentiments qui est le sous-domaine

de l’analyse de texte. En fait, près de 80% des données Internet sont du texte, par conséquent, l'analyse de texte est devenue un élément clé pour le sentiment du public et l'exploration de l'opinion. L'analyse des sentiments, vise à déterminer le sentiment des gens sur un sujet en analysant leurs publications et différentes actions sur les réseaux sociaux.

7 Conclusion

Dans ce chapitre nous avons défini le concept du Big data et sa relation avec les réseaux sociaux ‘BIG SOCIAL DATA’, nous avons aussi dévoilé les outils de traitement dans un environnement Big data que nous utiliserons par la suite pour notre implémentation.

Chapitre 2 : Analyse de sentiments

1 Introduction

Il existe de nombreux types d'analyses de sentiments allant des systèmes qui se concentrent sur la classification de la polarité (positif, négatif, neutre) aux systèmes qui détectent des émotions (en colère, heureux, triste, etc.) ou identifient des intentions (par exemple, intéressé, pas intéressé). Dans ce mémoire nous intéressons pour la classification de la polarité.

2 Définition de l'analyse de sentiment

L'analyse de sentiments est un sous domaine de l'analyse de texte qui est tout simplement l'utilisation du traitement du langage naturel et les techniques de calcul pour automatiser l'extraction et la classification des sentiments à partir d'un texte généralement non structuré. Dans leur état de l'art qui fait référence en la matière [1] Pang et Lee définissent l'analyse des sentiments comme « le traitement informatique du sentiment, de l'opinion et de la subjectivité dans le texte
». Ici le « traitement informatique » se rapporte à un processus automatisé et algorithmique. « Sentiment » peut être considéré comme un synonyme de l'opinion individuelle.

L'analyse des sentiments s'intéresse à l'orientation d'une opinion par rapport à une entité ou à un aspect d'une entité. Elle fait référence aux algorithmes et techniques de l'intelligence artificielle, utilisées pour extraire la polarité à partir du texte : s'il est positif, neutre, ou négatif. L'analyse des sentiments est un type d'exploration de données qui mesure l'inclination des opinions par le biais du traitement du langage naturel, de la linguistique computationnelle et de l'analyse de texte utilisées pour extraire et analyser des informations subjectives sur le Web. Les données analysées quantifient les sentiments ou réactions du grand public envers certains produits, personnes ou idées et révèlent la polarité contextuelle de l'information).

Il existe trois niveaux d’analyse qui sont :

- Niveau du document : détermine la polarité d'un texte entier. L'hypothèse est que le texte n'exprime qu'une seul opinion sur une seule entité (par exemple, un seul produit).

- Niveau de la phrase : détermine la polarité de chaque phrase contenue dans un texte. L'hypothèse est que chaque phrase dans le texte exprime une opinion unique sur une entité unique.

- Niveau des aspects : Effectue une analyse plus fine que les autres niveaux. Il est basé sur l'idée qu'une opinion consiste d'un sentiment et une cible (d'opinion). Par exemple, la phrase «La voiture est très rapide, mais le freinage est mauvais et il faut encore travailler sur les problèmes de sécurité» évalue trois aspects : voiture (positif), freinage (négatif) et la sécurité (négative).

2.1 Cas des sentiments basés sur les aspects

Nous avons détaillons seulement ce niveau d'analyse car c'est le niveau le plus difficile et le plus compliqué. Pour une cible donnée, l'identification de la polarité des opinions basées sur des aspects consiste à déterminer la polarité associée à chaque aspect de la cible. Cependant, les aspects d'une cible ne sont généralement pas connus a priori, et ils varient d'une cible à une autre. Par exemple, les aspects d'un téléphone mobile sont sa batterie, son appareil photo, son poids, etc., alors que les aspects d'un film de cinéma sont son histoire, son jeu d'acteurs, ses décors, etc. Ainsi, une difficultée additionnelle dans l'analyse de sentiments basés sur des aspects est d'extraire les aspects dans un premier temps. La seconde étape, similaire à l'identification de la polarité d'opinions, associe une polarité aux différents aspects extraits. On distingue dans la littérature deux types de travaux sur l'analyse de sentiments basés sur les aspects Certains travaux séparent le problème en deux phases et traitent l'une de ces phases ou les deux : extraction des aspects de la cible dans le texte et identification des sentiments (ainsi que leur polarité) associées aux aspects.

D’autres travaux proposent de découvrir conjointement les aspects et les opinions. Nous détaillons dans le reste de cette section les approches permettant l’extraction des aspects et les approches identifiant les opinions exprimées vis-à-vis des aspects.

2.1.1 Extraction des aspects

L’extraction d’aspects peut être considérée comme une instance du problème d’extraction d’informations : l’objectif est d’inférer des informations structurées (la liste des aspects) à partir de données non structurées (les textes subjectifs). Certains travaux ont ainsi opté pour une approche symbolique basée sur des règles et sur les parties de discours.

2.1.2 Identification des sentiments associées aux aspects

De nombreux travaux associent l’expression du sentiment aux adjectifs et aux adverbes. Ainsi, les mots du sentiment associés à un aspect peuvent être détectés en considérant les adjectifs et adverbes situés à proximité des mots dénotant des aspects (extraits lors de la phase précédente).

3 Identification de la polarité

Un sentiment est ciblé. Cette cible peut être de diverses natures suivant le type de données d’opinions concernées. Par exemple, une critique en ligne cible généralement un produit commercial (par exemple, un téléphone mobile) ou un service (par exemple, un hébergement dans un hôtel). Ainsi, on peut considérer qu’un texte d’opinion dont la polarité est positive révèle l’approbation globale de l’auteur du texte vis-à-vis de la cible, et inversement un texte négatif dénote une dépréciation globale.

Cette considération présuppose implicitement qu'un texte subjectif est homogène : il est soit totalement positif, soit totalement négatif (éventuellement neutre) ça dépend de la cible étudiée, à partir de cette considération, nous avons deux vues :

3.1 Opinions de polarité unique

Ce point de vue tient compte du l'opinion général exprimé, de sorte que le texte est lié à une seule pensée et ne forme pas un mélange de points de vue différents (hétérogènes) .Cette considération est plus efficace en raison du document court. Par exemple, avis sur les produits ou services [25]

3.2 Opinions basées sur les aspects

La désignation de la polarité d'une opinion basée sur des aspects pour une cible donnée, elle consiste à identifier la polarité de chaque un de ces aspects, néanmoins, les aspects d'une cible ne sont pas généralement connus préalablement, et ils varient d'une cible à une autre. Par exemple, les aspects d'une machine à laver sont : sa rapidité, sa consommation d'électricité, sa quantité à laver, etc..., alors que les aspects d'un ordinateur sont : son poids, sa vitesse, son espace de stockage, sa durée de vie, ...etc., il y a des travaux séparent l'opération en deux étapes, la première est l'extraction des aspects dans un premier temps, puis l'identification de l'orientation de chaque aspect, des autres travaux proposent à modéliser ces deux étapes conjointement [25].

4 Motivations de l'analyse de sentiments

Depuis l'apparition du phénomène populairement nommé « Web 2.0 », les internautes se sont vus offrir de nouvelles possibilités en matière d'interaction et de sociabilité en ligne. Les nouveaux services proposés permettent aux utilisateurs d'Internet de générer leur propre contenu et ainsi exprimer leurs opinions, par exemple sous la forme de billets de blog, ou encore par l'intermédiaire de postes sur des plateformes de réseaux sociaux telles que Twitter et Facebook, Les

consommateurs disposent de forums sans précédent, influents et puissants pour partager leurs expériences et exprimer leurs opinions (positives ou négatives) sur tout produit ou service. Les entreprises peuvent répondre aux demandes des consommateurs en surveillant et en analysant les opinions pour améliorer leurs produits [Zabin & Jefferies (2008)]. Malheureusement, le risque de changer d'opinion est élevé. Par conséquent, il est nécessaire de disposer d'un système capable d'analyser automatiquement le comportement général des consommateurs afin de mieux comprendre comment les clients perçoivent les différents produits et services. Bien que savoir quels aspects du produit et quels sont les produits qui ont été appréciés ou non permet à l'entreprise de corriger ses défauts ou en proposer une version améliorée dans le futur.

De l'autre côté les clients potentiels veulent également connaître les opinions des utilisateurs existants avant d'utiliser un service ou d'acheter un produit, En effet, selon l'enquête [comScore, Horrigan (2008)], 81% des internautes ont étudié au moins un produit et environ 80% d'entre eux ont exprimé leur opinion selon laquelle d'autres personnes ont une influence significative sur leur décision d'achat. Cela représente beaucoup de monde. Environ 30% des personnes ont exprimé leur opinion sur les produits, services ou personnes en ligne via le système de notation, ce qui n'est pas anodin pour le nombre. Par conséquent, cela est dû au fait que les utilisateurs ont exprimé leur intérêt pour les opinions sur les produits et services.

5 Défis de l'analyse de sentiment

Étant donné que l'analyse de sentiments est une instance de l'analyse de textes, il est légitime de se demander ce qui rend cette première tâche spécifique et difficile. Le niveau de difficulté d'une tâche de classification de textes est lié aux différences lexicales entre les classes : plus les classes utilisent un vocabulaire distinct, plus il sera facile d'assigner un texte à la bonne classe.

Cependant, la différence entre un texte de sentiment positif et un texte de sentiment négatif est plus subtile. Il est possible que les deux textes traitent le même thème (par exemple, un film de cinéma donné), induisant ainsi une grande similarité lexicale entre les classes positives et négatives .De plus, bien que certains mots tels que «super» et «mauvais» semblent indiquer de manière fiable la classe d'opinions,

ignorer la négation et le contexte de ces adjectifs faussera la classification. Par exemple, « Il y a une grève des chauffeurs de bus aujourd'hui, Il va falloir que je prenne un taxi pour aller travailler. Super ! ». Sans un contexte explicite ou une compréhension de la culture, le mot « super » sera potentiellement catégorisé comme un sentiment positif par une machine, ce qui serait correct dans un autre contexte mais pas dans notre exemple. Par ailleurs, une opinion peut être exprimée de manière implicite (« Je ne reviendrai pas dans ce restaurant. ») ou peut même inclure de l'ironie (« Bravo les verts pour votre excellent match !»).

6 Applications de l'analyse de sentiments

Comme le rappellent Pang et Lee dans « Opinion Mining and Sentiment Analysis » [1], « ce que les autres pensent » est régulièrement convoqué dans tout processus décisionnel, que ce soit en vue de l'achat d'un bien, dans le contexte d'une élection, dans le secteur tourisme, médecine ou encore pour évaluer la réputation de son entreprise.

Des recherches récentes indiquent que le nombre de personnes et d'entreprises utilisant des applications de médias sociaux comme outil de gestion de la relation client a considérablement augmenté [29]. C'est la raison de voir un grand nombre d'avis, de plaintes, de critiques et de compliments publiés et partagés quelques secondes seulement après la sortie d'un nouveau produit. L'analyse de ces informations aide :

6.1 Les entreprises et les organisations :

L'entreprise aujourd'hui peut s'adapter à cette tendance croissante afin d'atteindre certaines valeurs commerciales telles que l'augmentation du nombre de clients ; l'amélioration de la fidélité des clients, la satisfaction des clients et la réputation de l'entreprise ; et réaliser des ventes et des revenus totaux plus élevés (Batrinca et Treleaven, 2014).

6.2 Les consommateurs et les clients :

Il est devenu trivial que lors de l'achat d'un produit ou l'utilisation d'un service, le consommateur s'appuie et prend toujours en considération l'avis des autres clients ayant utilisés le même produit et ceci lui aide à prendre la bonne décision.
L'infographie 2 ci-dessous, présente quelques domaines d'application dans le domaine d'analyse des sentiments.

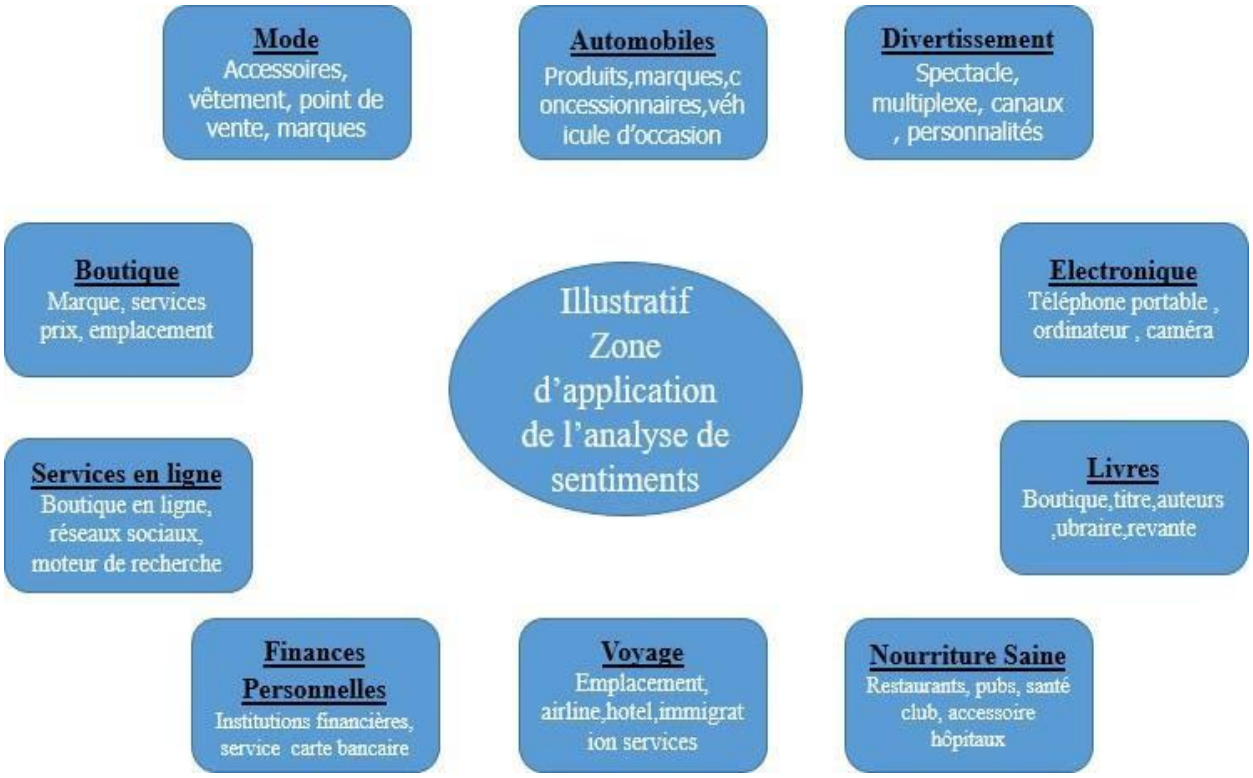

Figure 2 - Domaine d'application d'analyse de sentiment

7 Détection de subjectivité

Une tâche préliminaire à l'analyse des opinions contenues dans une collection de textes consiste à détecter les documents ou portions de documents subjectifs, c'est-à-dire exprimant des opinions. En effet, certains documents peuvent s'avérer purement factuels (par exemple, un article de presse relatant un évènement sportif)

alors que d'autres documents mentionnant des sujets plus polémiques reflètent les opinions de leurs auteurs (par exemple, un essai politique). De plus, un document exprimant une opinion n'est pas nécessairement subjectif dans son intégralité. Par exemple, une critique en ligne sur un téléphone mobile peut contenir une phrase telle que « J'ai commandé le modèle blanc. », qui ne porte aucune marque d'opinion et pourrait aussi bien être utilisée dans une critique positive que dans une critique négative.

La détection de subjectivité constitue ainsi un problème à part entière – qui se révèle en réalité être souvent plus difficile que l'analyse subséquente de la polarité des sentiments Mihalcea et al [6]. Afin d'encourager la recherche sur ce problème, la campagne d'évaluation TREC a proposé en 2006 une tâche de recherche d'opinions sur les blogs [8]. Dans le cadre de cette tâche, un document est jugé subjectif s'il contient « une expression explicite d'opinion ou de sentiment vis-à-vis de la cible, révélant une position personnelle de l'auteur » (traduit de l'anglais). Le but de TREC Blog 2006 était ainsi d'identifier les documents (c'est-à-dire les posts de blogs) à la fois pertinents vis-à-vis d'un sujet donné et subjectifs. D'autres travaux ont porté sur la détection de subjectivité au niveau de la phrase [9] ou au niveau des expressions [10] plutôt qu'au niveau du document. Pang et Lee [1] proposent une méthode à deux phases de classification, la première subjective-objective, dans la deuxième phase ils gardent que la partie subjective pour faire une classification positive-négative, en effet cette méthode améliore la classification de polarité en suppriment les phrases objectives.

De manière générale, les approches pour la détection de subjectivité reposent sur une combinaison des méthodes suivantes :

L'utilisation de lexiques de mots d'opinions externes, construits manuellement ou automatiquement [8].

L'exploitation de marqueurs linguistiques tels que les parties du discours (part of speech) en considérant par exemple les pronoms et les adjectifs comme des

Marques de subjectivité [9] ;

La mise en œuvre de classifieurs supervisés tels que les machines à vecteurs de support (SVM) et classifieurs bayésiens naïfs [9] ;

L'application de méthodes symboliques basées sur des règles et des motifs, définis manuellement ou automatiquement [10] ;

Une fois que les documents ou fragments de texte subjectifs ont été détectés, en utilisant une de ces deux approches, les sentiments qui y sont exprimées peuvent en être extraites et leur polarité identifiée.

8 Sources des données

Savoir d'où viens la donnée à analyser est très important. C'est pour cela dans cette section nous citons les trois sources de données utilisées dans la littérature

8.1 Micro-Blogging

Un outil de communication très populaire parmi les utilisateurs d'Internet est de microblogging. Des millions de message apparaissent tous les jours dans les sites Web populaires pour microblogging comme Twitter, utilisés comme source de données pour la classification des sentiments.
Le but des micro-blogs est de diffuser plus fréquemment des informations en se limitant au minimum utile, ils permettent la diffusion en temps réel d'informations jugées pertinentes par leurs éditeurs.

8.2 Sites D'avis

Donner aux utilisateurs de générer des avis sur les produits et services qu'ils ont achetés est une pratique largement disponible sur Internet. Les données utilisateurs sont analysées pour la classification de sentiment recueillies à partir des sites comme www.gsmarena.com (avis mobiles), www.amazon.com (commentaires sur les produits), www.CNET download.com (avis sur les produits), qui accueille des millions de commentaires sur les produits par les consommateurs.

8.3 Blogs

Le nom associé à l'univers de tous les sites de blog est appelé la blogosphère. Les gens écrivent sur les sujets qu'ils veulent partager avec les autres sur un blog. Un blog est facile à créer en raison de sa simplicité, sa forme libre et sa nature inédite. Nous trouvons un grand nombre de poste sur presque tous les sujets d'intérêt sur la blogosphère. Ces blogs permettent à ses auteurs, appelés blogueurs, d'exprimer une opinion subjective et sont la plupart du temps ouverts aux commentaires des lecteurs. Les blogs sont utilisés comme sources d'opinion dans la plupart des études portant sur l'analyse des sentiments.

9 Méthodes de classification des sentiments

Il existe trois approches de classification de sentiments qui sont les suivantes :

9.1 Approches lexicales

L'approche à base lexique définit un ensemble de règles dans un type de langage de programmation (script) qui identifie la subjectivité, la polarité ou le sujet d'une opinion. Cette approche peut utiliser diverses entrées, telles que les techniques classiques de traitement naturel du langage (NLP) comme la racinisation, tokenisation, POS (part of speech) et Chunking.

D'autre part les approches lexicales utilisent des dictionnaires de mots subjectifs, considérés comme des références universelles à partir de l'anglais.

Ces dictionnaires peuvent être généraux comme le General Inquirer, Sentiwordnet, Opinion Finder, NTU Sentiment Dictionary (NTUSD), etc. Ils peuvent également être construits automatiquement en fonction des corpus. Dans ces dictionnaires, une polarité est associée a priori à chacun des mots. Quel que soit le contexte dans lequel il sera inséré, le mot devrait ainsi avoir toujours la même polarité. On donne ensuite au document un score d'opinion en fonction de la présence de mots issus de ces dictionnaires dans le texte.

La figure 6 ci-dessous présente la fenêtre principale d'un exemple de ces dictionnaires General Inquirer et ses principales fonctionnalités.

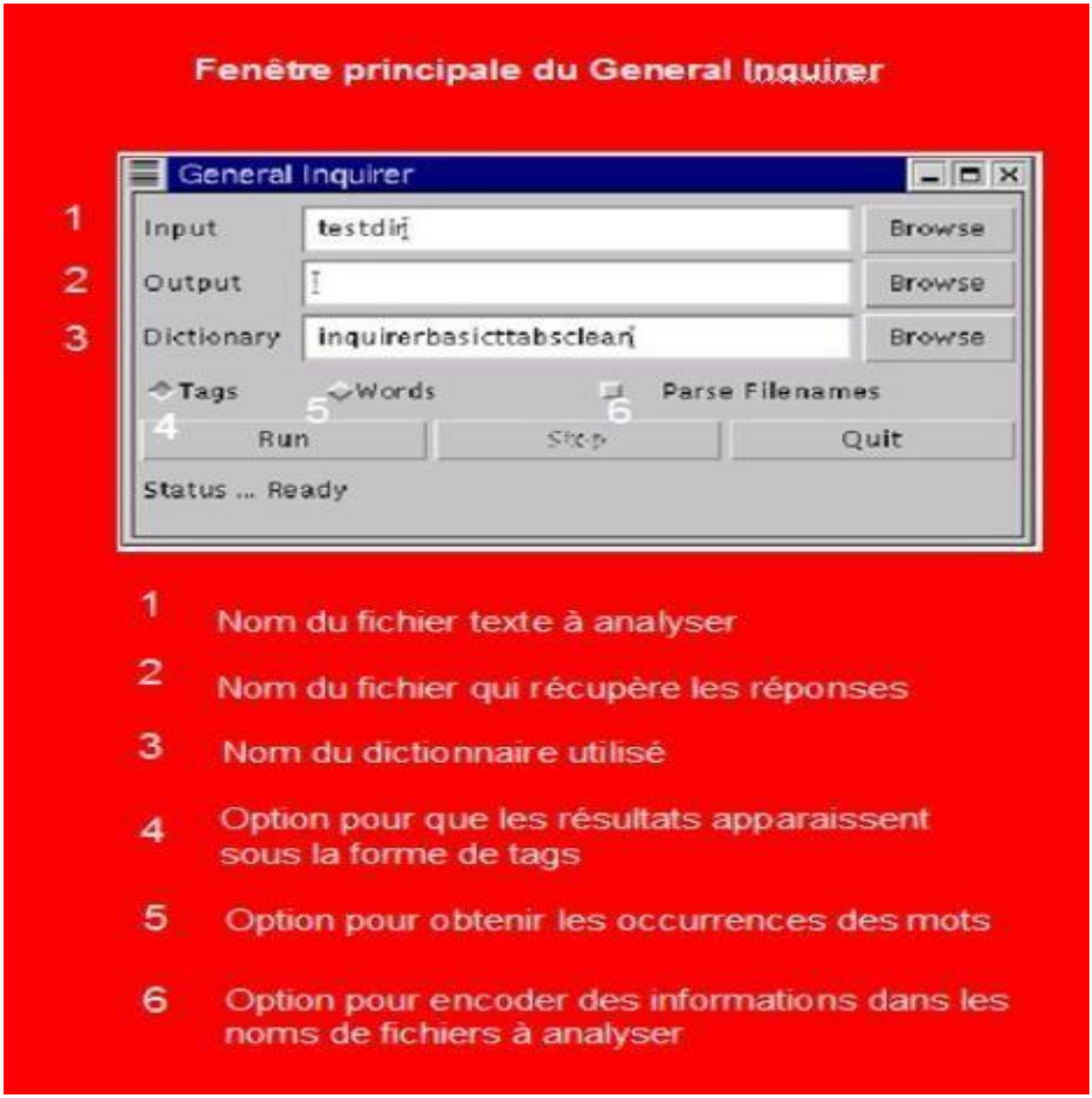

Figure 3 - Fenêtre principale de dictionnaire General Inquirer [28]

En dehors de General Inquirer deux autres dictionnaires sont couramment utilisés :

WordNet Affect : ressource linguistique pour la représentation lexicale de connaissances sur les affects pour l'anglais (1 903 concepts liés à un état mental ou émotionnel);

SentiWordNet : ressource lexicale dédiée aux systèmes de classification de textes d'opinion. Il s'agit d'une base de lexiques similaire à WordNet, mais elle est étendue avec les informations lexicales sur le sentiment de chaque synset contenues dans WordNet. Synset sont les regroupements de mots synonymes qui expriment le même concept. WordNet a été utilisé pour la collecte de synonymes, tandis que SentiWordNet a été utilisé pour identifier l'orientation sémantique de chaque phrase ou fonctionnalité extraite.

Ces dictionnaires ont été constitués de différentes façons :

1 à la main ;
2 à partir de corpus ;
3 à partir de dictionnaires existant.

Limites de l'analyse de sentiment à partir des lexiques

- les dictionnaires affectent une tonalité positive ou négative à un mot, sans tenir compte du contexte, c'est-à-dire du texte environnant, comme des paramètres de la communication située ;
- les dictionnaires ont tendance à éliminer les mots à valence ambiguë a priori ;
- le traitement des expressions ambiguës reste à faire et demande de faire appel à d'autres principes et à d'autres techniques ;
- lorsque la négation n'est pas prise en compte (ce qui peut paraître étonnant mais qui existe encore), le score de polarité peut être complètement faussé ;

9.2 Approche par apprentissage automatique

La tâche d'analyse de sentiments est généralement modélisée comme un problème de classification, dont le but est de renvoyer la classe correspondante (positive,

négative ou neutre) d'un texte donné(en cas d'analyse de polarité). Différentes techniques sont utilisées par cette approche pour classifier les sentiments, bien que ça défère selon les deux types d'apprentissage automatique à savoir l'apprentissage supervisé et non supervisé. Nous clarifions mieux cette approche dans les sections 2 et 3 du prochain chapitre.

9.3 Approche hybride

Dans les approches hybrides, les approches basées sur le lexique et l'apprentissage automatique peuvent fonctionner en parallèle pour calculer deux polarités de sentiment. Les résultats obtenus à partir des méthodes à base lexique et des méthodes basées sur l'apprentissage automatique sont ensuite combinés pour fournir une polarité de sentiment final et aboutir des résultats très précis. Dont l'approche à base de règle fait d'abord une analyse du texte phrase par phrase en le nettoyant, par exemple, diviser les phrases en mots, et jeter les caractères inutile (Tokenisation), Après par des méthodes statistiques de la deuxième approche commence l'analyse et l'extraction des sentiments. Il est également possible de concevoir un modèle d'analyse des sentiments en incorporant à la fois des méthodes basées sur le dictionnaire (lexique) et l'apprentissage automatique à différentes étapes du modèle [11] ; [12] ; [13] ;).

Le modèle du plus proche voisin sensible aux sentiments (SANN) est une combinaison d'approches basées sur un dictionnaire et basées sur l'apprentissage qui classe initialement un texte comme une revue subjective ou objective. Si le texte est objectif, alors la tâche de l'analyse des sentiments est terminée. Cependant, si le texte est subjectif, il est ensuite classé comme positif ou négatif. Pour un texte avec une polarité nulle, l'étiquette neutre est attribuée [14].

10 Travaux connexes

Nous présentons dans cette section un aperçu des travaux existants en analyse des sentiments. Étant donné le volume important de littérature en la matière, cet état de l'art sur l'analyse des sentiments n'a pas vocation à être exhaustif, mais nous allons quand même couvrir les principaux concepts afin de fournir le contexte de l'analyse des sentiments.

Zhao et al.2012 [24] ont travaillé dans un environnement BIG DATA et ont utilisé le modèle Naïve Bayes + émoticons pour classer les Tweets. Ils ont achevé une performance équivalente à 58.3%. F-score a été utilisé comme une mesure de performance.

Kessler and Nicolov 2009 [27] ont utilisé comme source de données les Blogs et comme modèle d'apprentissage ont utilisé la machine à vecteur de support SVM, et ont obtenu un F-score égale à 69.8%.

Pang, Lee, and Vaithyanathan 2002 [17] ont obtenu une précision très compétitive 82.9% pour la classification des données issue des critiques du film IMDb en utilisant le SVM.

Certains utilisent un modèle multi varié de Bernoulli, c'est-à-dire un réseau bayésien sans dépendance entre les mots et les caractéristiques de mots binaires (par exemple Larkey et Croft 1996, Koller et Sahami 1997). D'autres utilisent un modèle multinomial, c'est-à-dire un modèle de langage uni-gramme avec des nombres de mots entiers (par exemple Lewis et Gale 1994, Mitchell 1997).

Hasan et al. [20] ont adopté des caractéristiques uni grammes pour la classification binaire des sentiments et atteint 79% en utilisant le Naïve Bayes pour les tweets traduites en ourdou.

Par exemple [10] a utilisé des caractéristiques uni grammes pour la classification binaire des sentiments et a obtenu une précision de 88,94% en utilisant le SVM.

Kennedy et Inkpen [11] ont adopté les caractéristiques uni gramme et bi gramme pour la classification binaire des sentiments et ont atteint une précision de 84,4% pour les données de revue de film [21] en utilisant le SVM.

Wan [18] s'est intéressé à la classification binaire avec des fonctionnalités uni gram et bi gram et a atteint une précision de 86,1% pour les données de révision des produits Amazon traduites.

Dans [19], Akaichi a utilisé une combinaison de caractéristiques unigramme, bigramme et trigramme et a obtenu une précision de 72,78% en utilisant le SVM.

Gautam et Yadav [15], ont utilisé le SVM avec le modèle d'analyse sémantique pour la classification binaire des sentiments des textes Twitter et ont atteint une précision de 89,9% en utilisant des caractéristiques unigrammes.

Pang, Lee et Vaithyanathan [3] ont introduit un système qui base sur l'apprentissage automatique, ce dernier utilise unigrammes, bigrammes et les adjectifs en tant que caractéristiques, ils ont utilisé également SVM, Maximum Entropie et Naïve Bayes pour la classification des films selon les critiques. Leur conclusion était que la pondération binaire (0,1), donnait plus de précision que la fréquence du terme lorsqu'il est utilisé avec des unigrammes et SVM.

Toutes ces études qui utilisaient des caractéristiques en n-grammes ont généralement atteint une précision de 70 à 90%, et le modèle le plus efficace était le SVM.

Essam Al-Mansouri Sean Amos [5] ils ont fait un projet d'Utilisation de réseaux neuronaux artificiels et analyse de sentiment pour prédire vers le haut mouvement en Stock Prix.

Le but de son projet était de concevoir et de mettre en place un système d'apprentissage automatique permettrait de prédire avec précision si le prix d'une action serait plus élevé 65 minutes dans le futur.

Chapitre 3 : classification de sentiments

1 Introduction

La classification d'opinions ou de sentiments est essentiellement un problème de catégorisation de textes. Puisqu'il s'agit d'un problème de classification de texte, toute méthode d'apprentissage supervisé existante peut être appliquée, par exemple, la Régression Logistique, le Naïve Bayes et la Machine à vecteurs de support...etc. Pour cela dans ce chapitre nous allons expliquer les différentes étapes de processus d'analyse ou classification de sentiments.

Comme mentionné avant dans le chapitre 2 (analyse de sentiments), deux grands types de méthodes sont utilisés pour cette tâche, les approches linguistiques et les approches basé sur l'apprentissage automatique. Nous nous intéressons à l'application de la 2éme approche (apprentissage automatique) donc il est important de définir l'apprentissage automatique, son fonctionnement et ses types dans ce qui suit.

2 Processus de classification de sentiments

La fouille d'opinions peut être considérée comme un processus en plusieurs étapes qui prend en entrée un ensemble de textes sur une cible (par exemple, un produit ou une personnalité) et fournit en sortie un résumé agrégeant les sentiments exprimés dans le texte vis-à-vis de la cible ou éventuellement vis-à-vis des aspects de la cible [Dave et al. 2003]. L'aspect d'une cible est une caractéristique, un attribut ou un élément composant de la cible.

1. récupération des données,
2. prétraitement des données,
3. extraction des caractéristiques,

4. classification (i.e. identifier si le sentiment exprimé dans chaque texte subjectif est positive, négative ou neutre à l'égard de la cible et éventuellement à l'égard de ses aspects).

Le processus de fouille d'opinions ou d'analyse des sentiments passe par quatre étapes comme l'illustre la Figure 4

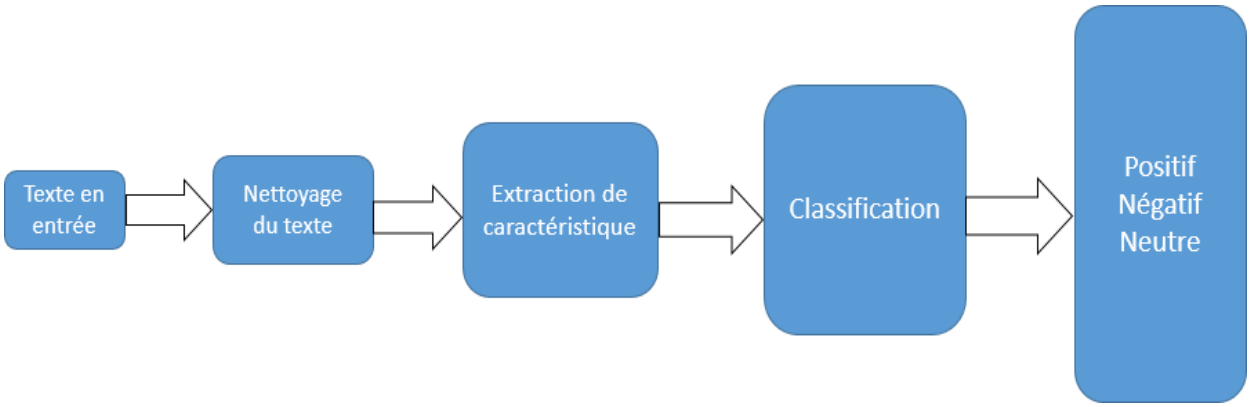

Figure 4 - Processus complet de l'analyse de sentiment

2.1 Récupération des données

La récupération des données nécessite l'identification et la définition de la source de données, par exemple, un portail de fournisseur de services commerciaux ou un réseau de médias sociaux. Pour collecter les données d'examen de ces sources, un mécanisme d'exploration Web spécifique est nécessaire pour extraire les données, puis les enregistrer dans une base de données en tenant compte du format des données [16] ; [13]. Après avoir collecté des données dans une base de données, les données doivent être extraites d'un ensemble de champs de données hétérogènes. Par exemple, dans le cas des données TripAdvisor, une revue est intégrée dans un document HTML récupéré, qui est composé de différents éléments, tels que les pieds de page ou les en-têtes, les

balises et le texte de la revue lui-même ([16]; [13]). Le texte doit être extrait à l'aide d'expressions appropriées. Chaque revue extraite contient une ou plusieurs phrases reflétant l'opinion de la critique.

2.2 Prétraitement

Dans l'étape du prétraitement différentes tâches sont effectuées, tel que la division d'un texte en phrases, la division d'une phrase en mots, la tokenisation, le filtrage des mots vides, le balisage de partie de discours (POS), le stemming et la transformation en minuscules / majuscules sont effectuées sur les données, pour les préparer à l'étape suivante (c.-à-d., l'extraction des caractéristiques) (Schmunk et al. 2014 [13]). Le balisage POS est une tâche de prétraitement importante qui fait généralement partie de l'analyse des sentiments en attribuant à chaque mot une étiquette particulière (par exemple, nom, verbe et adjectif).

La technique la plus courante pour remplacer et filtrer les chaînes de caractères est une expression régulière [Thompson 1968] dont nous avons utilisé dans notre contribution. Elle fournit un moyen de filtrer les erreurs et la surutilisation, et plus généralement elle fournit un bon moyen de reconnaître les sous-chaînes de caractères dans les chaînes de caractères. Les expressions régulières sont considérées comme une technique d'appariement de chaînes de caractères très puissante qui est utilisée dans de nombreuses fonctions de recherche et de remplacement "search and replace" des applications.

2.3 L'extraction de fonctionnalités

L'extraction de fonctionnalités est connue comme le processus de dérivation d'un ensemble de valeurs discriminantes, informatives et non redondantes pour représenter numériquement une revue ou un texte. Il existe plusieurs technique pour extraire les caractéristique parmi ces techniques on trouve la technique basée sur des occurrences de terme, appelées fréquence de terme (TF) ou fréquence de document inversée de fréquence (TF-IDF). En utilisant la technique d'extraction de fonctionnalités TF, les revues ou les phrases sont converties en une « matrice de documents terminologiques » [17] ; [16], la technique des N-gramme, l'intégration des mots (Word embedding en anglais), la représentation par sac de mots (bag of words en anglais) la

représentation par stemme et la représentation par lemme. Nous détaillant ci-après les plus couramment utilisés.

2.3.1 Term Frequency - Inverse Document Frequency (TF-IDF)

Le modèle de pondération TF-IDF (Term Frequency - Inverse Document Frequency) dans le modèle vectoriel, un document est représenté sous forme d'un vecteur dans un espace engendré par tous les termes d'indexation. La dimension de cet espace est le nombre de termes d'indexation de la collection de document. Les coordonnées d'un vecteur document sont les poids des termes d'index dans ce document. Un poids plus important est donné aux mots caractéristiques d'un document présenté sous forme d = (w1, w2, w3, …, wn) (Saldarriaga 2010).

Dans, un premier temps, il est nécessaire de calculer la fréquence d'un terme (TF). Celle-ci correspond au nombre d'occurrences de ce terme dans le document considéré. Ainsi, pour le document dj et le terme ti, la fréquence du terme dans le document est donnée par l'équation suivante :

$$TFi,j = ni,j \,/\, \sum k\ nk,j$$

$\boldsymbol{ni}$, j : est le nombre d′occurrences du terme ti dans di.

$\sum \boldsymbol{nk}$, J : est le nombre de termes dans le document.

La fréquence inverse de document (Inverse Document Frequency) mesure l'importance du terme dans l'ensemble du corpus. Elle consiste à calculer le logarithme de l'inverse de la proportion de documents du corpus qui contiennent le terme. Elle est définie de la manière suivante :

$IDFi$ = log (dj :) D représente le nombre total de documents dans le corpus.

dj :$ti \epsilon dj$: est le nombre de documents dans lesquels le terme ti apparait.

Enfin, le poids s'obtient en multipliant les deux mesures :

$$TFi - IDFi,j = TFi,j * IDFi$$

2.3.2 Sac de mots (Bag of words)

Un modèle de sac de mots, ou BoW en abrégé, est un moyen d'extraire des caractéristiques du texte pour les utiliser dans la modélisation, comme avec les algorithmes d'apprentissage automatique.

L'approche est très simple et flexible et peut être utilisée de multiples façons pour extraire des fonctionnalités à partir de documents. Un sac de mots est une représentation de texte qui décrit l'occurrence de mots dans un document. Cela implique deux choses :

1. Un vocabulaire de mots connus.
2. Une mesure de la présence de mots connus.

Cela s'appelle un «sac» de mots, car toute information sur l'ordre ou la structure des mots dans le document est supprimée. Le modèle se préoccupe uniquement de savoir si les mots connus apparaissent dans le document, et non à l'endroit du document.

2.3.3 Word Embedding

Le Word Embedding est essentiellement une forme de représentation de mots qui relie la compréhension humaine du langage à celle d'une machine. Il désigne un ensemble de techniques de Machine Learning qui visent à représenter les mots ou les phrases d'un texte par des vecteurs de nombres réels, décrits dans un modèle vectoriel.

Ces nouvelles représentations de données textuelles ont permis d'améliorer les performances des méthodes de traitement automatique des langues, comme la modélisation de sujets ou l'analyse de sentiments [26].

Le Word Embedding repose sur la théorie linguistique fondée par Zelling Harris et connue sous le nom de Distributional Semantics. Cette théorie considère qu'un mot est caractérisé par son contexte, c'est à dire par les mots qui l'entourent. Ainsi, des

mots qui partagent des contextes similaires partagent également des significations similaires. Les algorithmes de Word Embedding sont le plus souvent employés pour décrire des mots à travers de vecteurs numériques, mais ils peuvent également être utilisés pour construire des représentations vectorielles de phrases entières, de données biologiques comme les séquences d'ADN, ou encore des réseaux représentés comme des graphes [26].

2.3.4 N-grammes

Un *n*-gramme est une sous-séquence de *n* éléments construite à partir d'une séquence donnée. L'idée semble provenir des travaux de Claude Shannon en théorie de l'information. Son idée était que, à partir d'une séquence de lettres donnée (par exemple « par exemple ») il est possible d'obtenir la fonction de vraisemblance de l'apparition de la lettre suivante. À partir d'un corpus d'apprentissage, il est facile de construire une distribution de probabilité pour la prochaine lettre avec un historique de taille n. Cette modélisation correspond en fait à un modèle de Markov d'ordre n où seules les dernières observations sont utilisées pour la prédiction de la lettre suivante.

Les grammes sont des mots qui sont fréquemment répétés dans le corpus. Les Uni-grams (un seul mot, comme « produit ») se sont révélés plus performants que les bi-grams (deux mots consécutifs, comme « produit-chimique ») dans la catégorisation des critiques de produit en utilisant la polarité des sentiments, alors que les bi-grams et tri-grams (trois mots consécutifs) se traduisent par une meilleure classification de la polarité des critiques de produits [27].

2.4 Algorithmes de classification

Il existe de nombreuses méthodes dans l'apprentissage automatique pouvant être explorées. Nous avons exploré les trois plus utilisé (Régression Logistique, Naïve Bayes et Machine à vecteurs de support). Nous détaillant chaque algorithme dans la section 2.4 du chapitre suivant.

Nous nous définissons d'abord l'apprentissage automatique qui est un volet très important dans le domaine de l'analyse de sentiment, son fonctionnement et ses types pour faciliter la compréhension des méthodes utilisé par la suite.

3 Les types de l'apprentissage automatique

L'apprentissage automatique (Maching Learning) est une sous branche de l'intelligence artificiel (IA), qui se base sur des méthodes mathématique et statistique pour permettre aux machines d'apprendre à partir d'un modèle de données, c'est-à-dire de s'améliorer à résoudre les problèmes sans être programmés pour chacun. La figure 5 ci-dessous illustre les différentes étapes de l'apprentissage automatique.

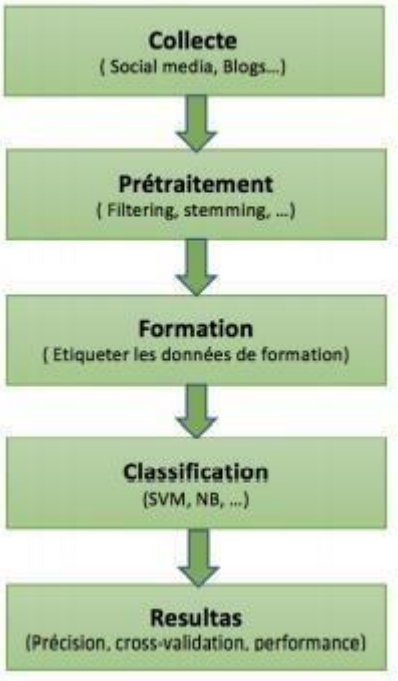

Figure 5 - les différentes étapes de l'apprentissage automatique

L'apprentissage automatique implique deux principaux systèmes d'apprentissage qui définissent ses différents modes de fonctionnement. Il s'agit de :

3.1 Apprentissage supervisé (Classification)

Dans notre contribution nous avons utilisé l'apprentissage supervisé qui consiste à construire un modèle basé sur un jeu d'apprentissage et des labels (nom des catégories ou des classes) et à l'utiliser pour classer des données nouvelles. L'algorithme est

entraîné par un ensemble de données qui est déjà étiqueté et qui a une sortie prédéterminée.

Cette technique est utilisée dans plusieurs applications telles que les diagnostics médicaux, la prédiction des pannes, l'analyse de sentiment et la détection des opinions trompeuses dans les réseaux sociaux.

3.2 Apprentissage non supervisé

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

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

Le processus de « clustering » repose sur une mesure précise de la similarité des objets qu'on veut regrouper. Cette mesure est appelée distance ou métrique.

Le « clustering » est utilisé dans plusieurs applications telles que le traitement d'images, les études démographiques, la recherche génétique, le forage des données et l'analyse des opinions. On distingue plusieurs algorithmes de clustering, exemple :

- K-moyennes (KMeans) : Un algorithme de partitionnement des données en K groupes ou clusters. Chaque objet sera associé à un seul cluster. Le K est fixé par l'utilisateur.
- Fuzzy KMeans : Il s'agit d'une variante du précédent algorithme proposant qu'un objet ne soit pas associé qu'à un seul groupe.
- Espérance-Maximisation (EM) : Cet algorithme utilise des probabilités pour décrire qu'un objet appartient à un groupe. Le centre du groupe et ensuite recalculé par rapport à la moyenne des probabilités de chaque objet du groupe.

4 Conclusion

Le processus de classification de sentiment est découpé essentiellement à quatre étapes acquisition des données, prétraitement, extraction de fonctionnalité et la classification. Dans ce chapitre nous avons expliqué et détaillé chacune.

Le nombre de données textuelles augmente de façon exponentielle, il devient donc plus important de développer des modèles pour analyser automatiquement le texte en générale et les sentiments particulièrement. De ce fait nous avons présenté dans ce chapitre les types et les techniques de l'apprentissage automatique.

CHAPITRE 4 : Apprentissage automatique pour l'analyse de sentiments Big Data

1 Introduction

Dans ce chapitre nous parlerons du processus de l'analyse de sentiment en détaillent tous les étapes à savoir le prétraitement, l'extraction de fonctionnalité et l'étape de classification dont nous avons expliqué le fonctionnement de chaque algorithme d'apprentissage supervisé utilisé à savoir Régression Logistique, Naïve Bayes et Machine à vecteurs de support. Ensuite nous avons dévoilé les critères utilisés pour la mesure de performance des systèmes de classification de sentiments.

2 Prétraitement

Habituellement, les données brutes collectées à partir de différentes sources sont très volumineuses, redondantes et consomment la capacité de stockage en conservant les données non pertinentes. De plus, certaines méthodes analytiques nécessitent un certain niveau de qualité des données. En effet, la construction d'un bon modèle repose sur la qualité des données à analyser.

Le prétraitement des données est le processus de nettoyage et de préparation du texte pour la catégorisation. Les textes en ligne contiennent généralement beaucoup de bruit et des parties non informatives telles que des balises HTML, des scripts et des publicités.

De plus, au niveau des mots, de nombreux mots dans le texte n'ont pas d'impact sur l'orientation générale de celui-ci. Garder ces mots rend la dimensionnalité du problème élevée et donc la classification plus difficile puisque chaque mot dans le texte est traité comme une dimension. Voici l'hypothèse d'un prétraitement correct des données: réduire le bruit dans le texte devrait aider à améliorer les performances

et accélérer le processus de classification, aidant ainsi à l'analyse des sentiments en temps réel.

2.1 Filtrage

L'opération de filtrage a pour but la suppression des métadonnées contenues dans les messages, nous pouvons citer comme exemples :

- **Liens URL** : Si le message contient un lien URL (exemple : http://bit.ly/KCairo) l'opération de filtrage le supprime, étant donné qu'il ne contient aucune information qui influe sur le sentiment exprimée dans le message.

Figure 6 - Exemple d'un tweet contenant lien url.

- **Les symboles** : dans le cas des tweets et des posts, souvent les utilisateurs utilisent des symboles (exemple : le hashtag « # ») pour mettre en évidence un mot précis, ces symboles doivent être supprimés de façon à pouvoir utiliser les mots par la suite.

Figure 7 - Exemple de Tweet avec des hashtags.

Après analyse du tweet, le processus de filtrage supprime le hashtag Tweet devient : (FH2012) a la place de #FH2012

• **Les noms d'utilisateurs** : nous pouvons les détecter à travers les liens hypertexte, sur les tweets comme pour les posts. Les utilisateurs se servent du symbole « @ » afin de mentionner d'autres utilisateurs ou pages. Pour le cas de Twitter, le symbole apparait dans le tweet donc il est facile de supprimer le nom d'utilisateurs. Par contre pour Facebook, l'opération doit tenir compte des liens hypertextes qui indiquent la page pour effectuer la suppression.

Le filtrage peut comprendre d'autres tâches qui dépendent de la nature du corpus et de l'objectif du travail.

Figure 8 - Exemple de tweet portant des noms d'utilisateurs.

2.2 Tokenisation

Dans cette partie, l'opération de Tokenisation est l'acte de décomposer une séquence de chaînes en morceaux tels que des mots, des mots-clés, des phrases, des symboles et d'autres éléments appelés jetons (tokens). Les jetons peuvent être des mots individuels, des phrases ou même des phrases entières.

Dans le processus de tokenisation, certains caractères comme les signes de ponctuation sont ignorés. Les jetons deviennent l'entrée d'un autre processus comme l'analyse syntaxique et l'exploration de texte. La tokenisation est utilisée en informatique, où elle joue un rôle important dans le domaine d'analyse de texte, d'où l'utilisation de cette fonction dans notre projet.

Figure 9 - Tweet composé du texte et des signes de ponctuation.

Après analyse du tweet, le processus de tokenisation transformer le texte en liste de tokens (mot, ponctuation,..) donc le tweet de la figure devient : (‘watching’ , ‘my’ , ‘segment’ , ‘on’ , ‘FOX’ , ‘and’ , ‘cringing’ , ‘.’ , ‘.’ , ‘.’ , ‘.’ , ‘Listening’, ‘to’ , ‘my’ , ‘voice’ , ‘on’ , ‘tv’ , ‘is’ , ‘SO’ , ‘painful’ , ‘.’ , ‘Do’ , ‘I’ , ‘really’, ‘sound’ , ‘like’ , ‘such’ , ‘a’ , ‘valley’ , ‘girl’ , ‘ ?’ , ‘!’ , ‘ ?’ , ‘ !’).

2.3 Suppression des mots vides

Dans ce cas, nous effectuons une suppression des mots qui n'influent pas sur l'opinion exprimée dans le message et qui, de plus, augmente considérablement et inutilement le nombre de mots dans le vocabulaire. Ce sont spécifique pour chaque langue. Ces mots en anglais sont :

- Les conjonctions de coordination (for, and, nor, but, or, yet, So).
- Les déterminants (a/an, the, this, that, these, those).
- Les prépositions (at, in, to).

Figure 10 - Exemple de Tweet contentant des mots vides.

Après analyse du tweet de la figure, la phrase devient : (willing stand, vote, organize, will finish started)

Pour cibler et éliminer les mots vides, nous avons utilisé la fonction StopWordsRemover de Python.

3 L'extraction de fonctionnalité

Comme nous l'avons mentionné dans le troisième chapitre, il existe plusieurs représentation pour l'extraction de fonctionnalité ça dépend des besoins et de l'application

D'après les articles que nous avons eu l'opportunité de lire, nous avons constaté que l'utilisation des classifieurs Maching Learning est meilleure avec la technique de pondération TF-IDF et les classifieurs du Deep Learning donne des résultats meilleures avec la technique Word Embedding).C'est pour cela nous avons utilisé la technique TF pour le codage.

Un document dans le codage TF est représenté comme un vecteur de pondération, dans lequel chaque poids de composant est calculé sur la base du schéma Fréquence du terme (TF) t f i j. Le poids p i j du terme ti dans le document dj est le nombre d'occurrence de ti dans dj, noté f i j. La normalisation pourrait également être appliquée par l'équation suivante, où le maximum est calculé sur tous les termes qui apparaissent dans le document dj :

$$Tf\ i\ j = f\ i\ j\ /\ Max\ \{f1j, f2j, \ldots, f\ |E|\ j\}$$

4 Classification

Dans cette section nous présentons les trois modèles d'apprentissage automatique (supervisé) que nous avons utilisé pour la classification de sentiment à citer logistique régression, naïves bayes et machine à vecteurs de support.

4.1 Régression Logistique

En raison de ses bonnes performances dans le classement automatique, cet algorithme est devenu une méthode de classement très populaire. Il peut calculer la probabilité d'appartenir à la catégorie k, $Pr\ (Pr = k \mid X = x)$ comme suit :

$$Pr\ (Y = k|X = x) = e\ \beta x\ 1 + e\ \beta x$$

Où β est le vecteur de coefficients de régression qui doivent être estimés avec des exemples d'entraînement en utilisant, par exemple, la méthode de moindres carrés. La catégorie assignée sera celle dont la probabilité est la plus grande, c'est-à-dire :

$$\hat{y} = argmax\ \_\ (Y = k|\ x)\ k \in \{1,\ldots,\}$$

La régression logistique a bénéficié de nombreuses recherches et est devenue une méthode pratique dans de nombreux grands systèmes d'entreprise, en particulier dans les grandes sociétés Internet comme Google et Yahoo, qui l'utilisent pour apprendre de grands ensembles de données.

4.2 Naïve Bayes

Les méthodes Naïve Bayes sont un ensemble d'algorithmes d'apprentissage supervisé basés sur l'application du théorème de Bayes avec l'hypothèse «naïve» d'indépendance conditionnelle entre chaque paire de caractéristiques étant donné la valeur de la variable de classe. [22]

Le théorème de Bayes permet de calculer la probabilité qu'une donnée appartienne à une classe donnée, compte tenu de nos connaissances préalables. Le théorème de Bayes est énoncé comme suit :

P (classe | données) = (P (données | classe) * P (classe)) / P (données).

Où P (classe | données) est la probabilité de classe compte tenu des données fournies.

Subséquemment, nous testons nos modèles en minant les fonctions utilisées antérieurement tel que Tokenize, SwRemovedTest …etc.

4.3 Machine à vecteurs de support

Une machine à vecteurs de support, traduction littérale pour Support Vector Machine, est un algorithme d'apprentissage automatique supervisé qui peut être utilisé à des fins de classification et de régression. Le SVM appartient à la catégorie des classificateurs linéaires (qui utilisent une séparation linéaire des données), et qui dispose de sa méthode à lui pour trouver la frontière entre les catégories. [29]

Pour que le SVM puisse trouver cette frontière, il est nécessaire de lui donner des données d'entraînement. A partir de ces données, le SVM va estimer l'emplacement le plus plausible de la frontière : c'est la période d'entraînement, nécessaire à tout algorithme d'apprentissage automatique.

Une fois la phase d'entraînement terminée, le SVM a ainsi trouvé, à partir de données d'entraînement, l'emplacement supposé de la frontière. En quelque sorte, il a « appris » l'emplacement de la frontière grâce aux données d'entraînement. Qui plus est, le SVM est maintenant capable de prédire à quelle catégorie appartient une entrée qu'il n'avait jamais vue avant, et sans intervention humaine (comme c'est le cas avec le triangle noir dans la Figure 19) : c'est là tout l'intérêt de l'apprentissage automatique.

L'exemple ci-dessous illustre le fonctionnement des SVM dont nous avons une nouvelle entrée (le triangle noir) pour la classifier soit avec la catégorie des ronds rouges ou des carrés blues.

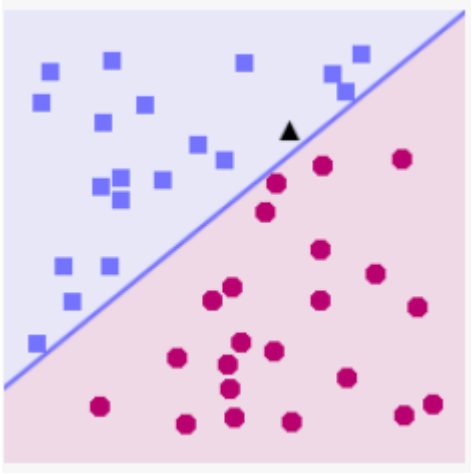

Figure 11 - Machine à vecteur de support

5 Performance

Dans ce qui suit, nous présentons les métriques et méthodes couramment utilisées pour l'évaluation des « classifieurs ».

5.1 La précision et le rappel

La précision et le rappel sont deux critères de mesures statistiques évaluant les « classifieurs », aussi appelés valeur prédictive (précision) et sensibilité (rappel). Nous notons :

VP : le nombre d'éléments correctement étiquetés positifs (vrai positif)

FN : le nombre de classifications incorrectes d'exemples positifs (faux négatifs)

FP : le nombre d'éléments qui ont été incorrectement étiquetés positifs (faux positifs)

VN : le nombre de classifications correctes d'exemples négatifs (vrai négatif)

- Dans une tâche de classification de sentiment, la précision P d'une classe est le nombre de vrais positifs divisé par le nombre total d'éléments catégorisés positifs :

$$P = VP / (VP+FP)$$

- Le Rappel R dans ce domaine est défini comme le nombre de vrais positifs divisé par le nombre total d'éléments qui appartiennent effectivement à la classe positive.

$$R = VP / (VP+FN)$$

5.2 F-score

F-score, est une mesure apprécié de la performance d'un test qui combine à la fois la précision et le rappel. Elle est généralement utilisée pour comparer différents classificateurs avec une seule mesure.

$$F = 2 \times (p \times r / p+r)$$

F-score est également appelé F1-score ou F-mesure, est la moyenne harmonique pondérée de la précision et du rappel :

$$F = 2 / (1/ p +1/r)$$

6 Conclusion

Dans ce chapitre nous avons présenté le processus d'analyse de sentiments d'analyse en détaillant chaque étape, et nous avons expliqué le fonctionnement de nos méthodes d'apprentissage utilisé. Enfin nous avons cité trois mesures de performance très utilisé pour l'évaluation des classifieurs.

Dans le chapitre suivant, nous présentons notre contribution dont nous commençons avec l'architecture générale du système proposé puis nous présentons les outils et les bibliothèques utilisés pour le traitement des « Big social data» ainsi que les expérimentations que nous avons menées afin de mesurer la robustesse et la performance de la méthode de classification proposée.

Chapitre 5 : implémentation et résultats

1 Introduction

Nous avons consacré ce chapitre à la description du système proposé en détaillant tous les étapes comme l'architecture proposé, le prétraitement, l'extraction de fonctionnalité et la classification en utilisant l'apprentissage supervisé. Nous allons parler aussi de tous les outils et les bibliothèques des données massives (Big data) et d'apprentissage automatique (maching learning) qui ont été utilisés dans notre implémentation. Nous allons également décrire notre environnement de travail.

Pour l'analyse de sentiments, nous avons utilisé l'approche basée apprentissage automatique. Nous avons travaillé avec un grand ensemble de donnée et utiliser Twitter comme source de donnée.

Nous utilisons trois modèles d'apprentissage automatique (Régression Logistique, Naïve Bayes et Machine à vecteurs de support) pour déterminer la polarité d'un sentiment dans un tweet dans un environnement Big data, en utilisant apache Spark comme Framework et l'architecture maître-esclave.

Nous discuterons les résultats obtenus et les comparerons à l'état de l'art.

2 Description du système

Dans cette section nous allons présenter notre programme en faisant la correspondance avec le processus d'analyse de sentiment détaillé dans le troisième chapitre. Le schéma ci-dessous représente notre méthodologie de travail et les différentes étapes de processus d'analyse de sentiment.

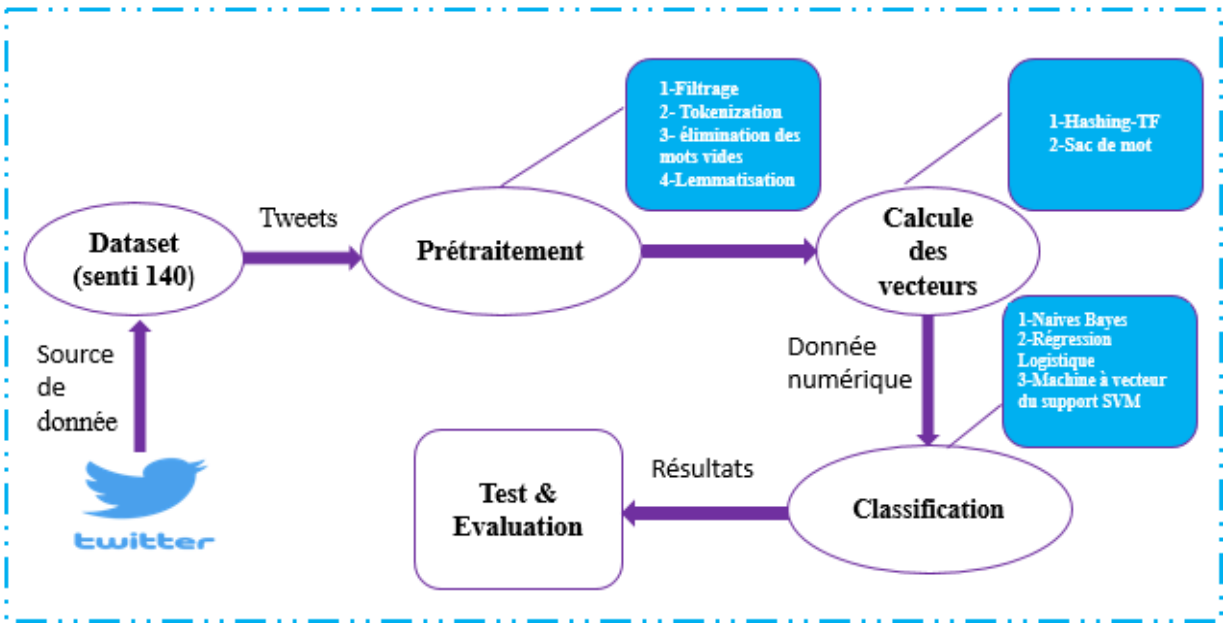

Figure 12 - Processus général de notre système pour l'analyse de sentiments.

3 Environnement de travail

D'abord, Dans cette section nous allons mentionner et décrire les outils et le langage utilisés dans notre implémentation.

Une étape intéressante de ce projet était la mise en place de l'environnement matériel et logiciel nécessaire pour la conception, le développement et le test de notre système.

Dans ce qui suit, nous présenterons l'environnement logiciel et matériel exploité dans notre projet.

3.1 Environnement matériel

- Ordinateur portable : HP
- Système d'exploitation : linux
- Processeur : Core I5

- ❖ Mémoire : 16 G RAM

- ❖ Disque dur : 256 Gb SSD

3.2 Environnement logiciel

Nous avons utilisés le langage de programmation Python, la version 3.7.

Python est un langage de programmation portable, dynamique, extensible, gratuit, syntaxe très simple, code plus court que C ou Java, multi thread, orienté objet, évolutif, interprété et interactif. La syntaxe de Python est très simple et combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles.

- **Environnement de développement Jupyter**

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, R, ou encore Scala2. Jupyter est une évolution du projet IPython. Jupyter permet de réaliser des calepins ou notebooks, c'est-à-dire des programmes contenant à la fois du texte en markdown et du code en Scala, Python... Ces calepins sont utilisés en science des données pour explorer et analyser des données.

- **Anaconda (Python Distribution)**

Anaconda est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique), qui vise à simplifier la gestion des paquets et de déploiement.

- **PySpark**

Est écrit dans le langage de programmation Scala. PySpark a été publié pour prendre en charge la collaboration entre Apache Spark et Python, il s'agit en fait d'une API Python pour Spark.

De plus, PySpark aide à connecter aux ensembles de données dans Apache Spark et le langage de programmation Python. Ceci a été réalisé en tirant parti de la bibliothèque Py4j .PySpark LogoPy4J est une bibliothèque populaire qui est intégrée à PySpark et permet à python de s'interfacer dynamiquement avec des objets JVM.

PySpark a plusieurs bibliothèques pour écrire des programmes efficaces. De plus, il existe diverses bibliothèques externes qui sont également compatibles.

- **Hadoop**

Hadoop est un Framework libre et open source écrit en Java destiné à faciliter la création d'applications distribuées (au niveau du stockage des données et de leur traitement) et échelonnables (scalables) permettant aux applications de travailler avec des milliers de nœuds et des pétaoctets de données.

Dans notre projet nous avons travaillé avec la version 3.1.0, et vu que nous avons abîmé une très large Dataset de 1 600 000 tweets, nous avons tri parti du système de fichiers distribué, HDFS qui a été conçu pour stocker de très gros volumes de données sur un grand nombre de machines équipées de disques durs banalisés.

- **Système de fichiers distribué Hadoop (HDFS)**

HDFS est très essentiel pour le Big Data. Les données sont désormais trop nombreuses pour être stockées de façon centralisée, spécialement à cause du coût et des contraintes de capacité de stockage. Mais, grâce à la nature distribuée de ce dernier, il est possible de répartir les données sur différents serveurs afin de réaliser des économies. En raison de sa capacité et de sa fiabilité énorme, HDFS est un système de stockage très approprié pour le Big Data. Ce système augmente les capacités de gestion des données du cluster HDFS Hadoop et permet ainsi un traitement efficace des données volumineuses. Parmi ses principales

caractéristiques, on compte la possibilité de stocker des téraoctets ou même des pétaoctets de données.

3.3 Bibliothèques et Packages

Nous avons utilisé certains packages et bibliothèques de Big Data définie sur PysPark et du Maching Learning à savoir :

- **Package Re (Regular expressions)**

Ce module fournit des opérations correspondant aux expressions régulières. Les expressions régulières utilisent le caractère barre oblique inverse ('\') pour indiquer des formes spéciales ou pour permettre l'utilisation de caractères spéciaux sans invoquer leur signification particulière. Nous l'avons utilisé dans la section du prétraitement (filtrage).

- **Package Tokenizer**

Chaque mot est un jeton (token) lorsqu'une phrase est "tokenisée" en mots, il divise une phrase à une liste des mots séparés.

- **Bibliothèque Matplotlib**

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques pour une compréhension meilleure.

Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy.

- **Bibliothèque NLTK**

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant le traitement automatique des langues.

Il fournit des interfaces faciles à utiliser pour plus de 50 ressources corporelles et lexicales telles que WordNet. Ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, tokenisation, stemming, étiquetage et l'analyse.

4 Présentation de corpus

Il s'agit de le grand ensemble de données sentiment140. Il contient 1 600 000 Tweets extrait du la plateforme web 'kaggle'. Les Tweets ont été annotés (0 = positif, 4 = négatif).

Chaque entrée de notre ensemble de données est structurée comme suit :

- Sentiment : tag qui peut être : positive, négative
- Item id : un identifiant d'un tweet
- Date : la date de publication du tweet
- User : l'utilisateur qui a publié le tweet
- Texte : le contenu publié par les utilisateurs

La capture ci-dessous montre une ligne de l'ensemble de donnée avant le prétraitement.

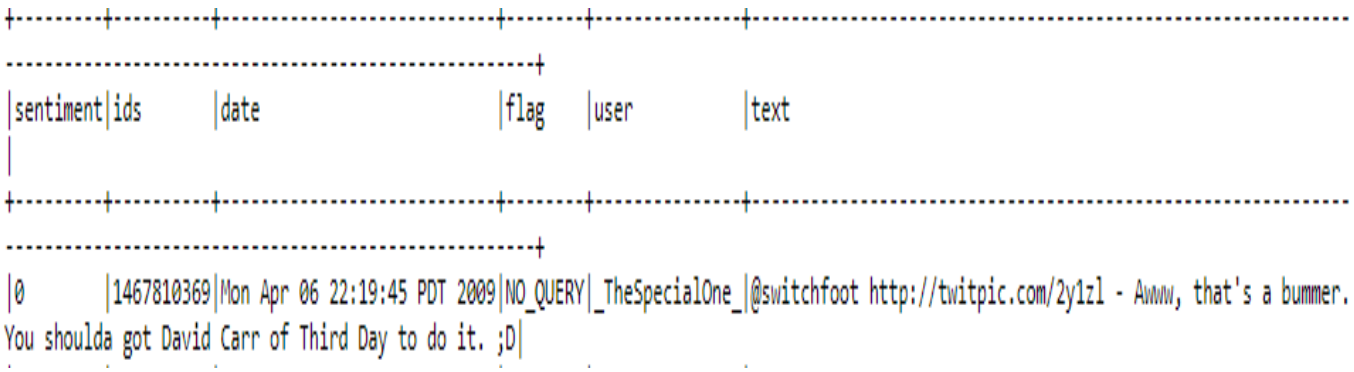

Figure 13 - l'ensemble de donnée annoté avant le prétraitement

L'histogramme ci-dessous généré à l'aide de la bibliothèque Matplotlib de python dévoile la distribution de notre dataset qui est déjà annoté avec 50% positif et 50% négatif.

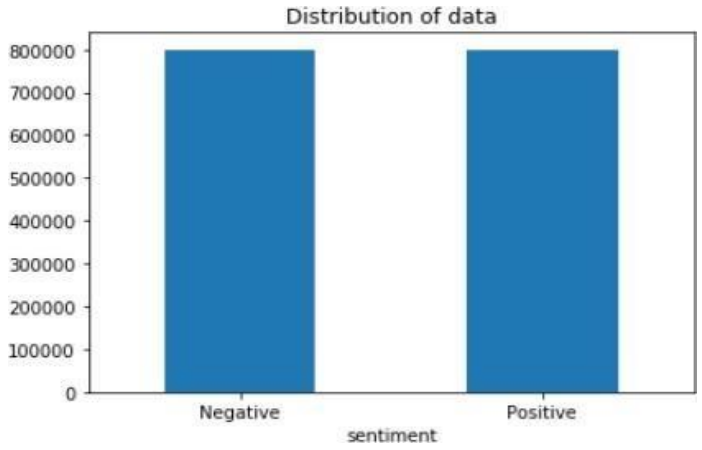

Figure 14 - Distribution de l'ensemble de données.

- Finalement, après avoir mentionné toutes les ressources qui ont été utilisées pour le développement de notre programme. Nous allons ci-après présenter et discuter la phase de classification, qui est fondamentalement l'objectif principal de ce projet.

5 Prétraitement de l'ensemble de donnée

Avant d'entamer l'étape de prétraitement qui est l'étape la plus importante nous allons tous d'abord, dévoiler les importations nécessaires pour le fonctionnement du système.

```
import findspark
findspark.init('/home/ayoubgrm/spark')
#import modules
from pyspark.sql import SparkSession, functions as F
from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark.ml.classification import LogisticRegression, NaiveBayes
from pyspark.ml.feature import HashingTF, Tokenizer, StopWordsRemover, Word2Vec
from pyspark.sql.functions import monotonically_increasing_id

# nltk
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer, SnowballStemmer

# utilities
import re
import numpy as np
import pandas as pd

# plotting
from wordcloud import WordCloud
import matplotlib.pyplot as plt

#create Spark session
appName = "Sentiment Analysis in Spark"
spark = SparkSession.builder.appName(appName).config("spark.some.config.option", "some-value").getOrCreate()
```

Figure 15 - les importations des package et bibliothèques

Pour le prétraitement nous avons utilisé une combinaison des traitements les plus importants à savoir la lemmatisation, élimination des mots vides, Tokenisation, et nous avons utilisé des expressions régulières pour le filtrage (suppressions des URL, @utilisateur …etc.) comme le montre la capture d'écrans 18 ci-dessous. Tous d'abord nous affichons le dictionnaire des émojis.

- Dictionnaire des émojis :

```
# Defining dictionary containing all emojis with their meanings.
emojis = {':)': 'smile', ':-)': 'smile', ';d': 'wink', ':-E': 'vampire', '>-)':
          'evilgrin', ':(': 'sad', ':-(': 'sad', ':-<': 'sad', ':P': 'raspberry',
          ':-O': 'surprised', ':-*': 'kissing', ':-@': 'shocked', ':-$': 'confused',
          ':-\\': 'annoyed', ':-#': 'mute', '(((H)))': 'hugs', ':-X': 'kissing',
          '`:-)': 'smile', ':^)': 'smile', ':-&': 'confused', '<:-)': 'smile',
          ':->': 'smile', '(-}{-)': 'kissing', ':-Q': 'smoking','$_$': 'greedy',
          '@@': 'eyeroll', ':-!': 'confused', ':-D': 'smile', ':*)': 'smile',
          ':@': 'shocked',':-0': 'yell', ':-----)': 'liar', '%-(': 'confused',
          '(:I': 'egghead', '|-O': 'yawning', ':@)': 'smile', 'O.o': 'confused',
          '<(-_-)>': 'robot', 'd[-_-]b': 'dj', '~:0': 'baby', '-@--@-': 'eyeglass',
          ":'-)": 'sadsmile', '{:-)': 'smile', ';)': 'wink', ';-)': 'wink',
          'O:-)': 'angel','O*-)': 'angel','(:-D': 'gossip', '=^.^=': 'cat'}
```

Figure 16 - dictionnaire des émojis

```python
def preprocess(textdata):
    processedText = []

    # Create Lemmatizer and Stemmer.
    wordLemm = WordNetLemmatizer()
    snowStem = SnowballStemmer("english")

    # Defining regex patterns.
    urlPattern        = r"((http://)[^ ]*|(https://)[^ ]*|( www\.)[^ ]*)"
    userPattern       = '@[^\s]+'
    alphaPattern      = "[^a-zA-Z0-9]"
    sequencePattern   = r"(.)\1\1+"
    seqReplacePattern = r"\1\1"

    for tweet in textdata:
        tweet = tweet.lower()

        # Replace all URls with 'URL'
        tweet = re.sub(urlPattern,'URL',tweet)
        # Replace all emojis.
        for emoji in emojis.keys():
            tweet = tweet.replace(emoji, "EMOJI" + emojis[emoji])
        # Replace @USERNAME to 'USER'.
        tweet = re.sub(userPattern,'USER', tweet)
        # Replace all non alphabets.
        tweet = re.sub(alphaPattern, " ", tweet)
        # Replace 3 or more consecutive letters by 2 letter.
        tweet = re.sub(sequencePattern, seqReplacePattern, tweet)

        tweetwords = ''
        for word in tweet.split():
            if len(word)>1:
                # Lemmatizing the word.
                word = wordLemm.lemmatize(word)
                # Stemming the word.
                word = snowStem.stem(word)
                tweetwords += (word+' ')

        processedText.append(tweetwords)

    return processedText
```

Figure 17 - prétraitement de l'ensemble de données

5.1 Suppressions des données inutiles

Dans notre cas nous devrons supprimer les colonnes item id, sentiment, date et user de la figure 14 car nous n'aurons pas besoin lors de processus d'analyse, le résultat est affiché dans la capture d'écrans suivante :

```
# create preprocessed spark dataframe
processedtext = preprocess(text)
dataset['text'] = processedtext
pdf = spark.createDataFrame(dataset)
pdf.show()
```

```
+---------+--------------------+
|sentiment|                text|
+---------+--------------------+
|        0|kinda miss my mom...|
|        1| USER huh uhu  i ...|
|        1|awake since six o...|
|        1|this is one of th...|
|        1|  USER hmm  why not |
|        1|can t get enough ...|
|        1|just had a facebo...|
|        1|thank god we fina...|
|        1| USER  really  lo...|
|        1|          USER why  |
|        1|is going to stone...|
|        1|  USER i luv it too |
|        0|ahh EMOJIshocked ...|
|        0| USER i sit here ...|
|        0|at work going men...|
|        0|nearlly finished ...|
|        0|today was not a g...|
|        1|i am special aren...|
|        1|another lovely da...|
|        1|had  great weeken...|
+---------+--------------------+
only showing top 20 rows
```

Figure 18 - l'ensemble de donnée après le prétraitement

5.2 Tokenisation

Le code de tokenisation de notre système est dévoilé dans la capture d'écrans ci-dessous

```
# additional spark dataframe preprocessing
# 1. tokenization
tokenizer = Tokenizer(inputCol="text", outputCol="SentimentWords")
tokenizedTrain = tokenizer.transform(trainingData)
tokenizedTrain.show(truncate=False, n=2)
```

Figure 19 – Tokenisation

Pour bien illustré l'étape de tokenisation nous avons pensé à montrer les deux premières lignes de l'ensemble de donnée après la tokenisation. Le résultat est affiché dans la colonne SentimentWords de la figure 23.

```
+---------+--------------------------------------------------------------------+--------------------------------------------------
---------------------------------+
|sentiment|text                                                                |SentimentWords
|
+---------+--------------------------------------------------------------------+--------------------------------------------------
---------------------------------+
|0        |00 09 am and still not done pack tire so how are you guy do fill me in |[00, 09, am, and, still, not, done, pack, tire, so, ho
w, are, you, guy, do, fill, me, in]|
|0        |00 about to sleep can find my remot                                 |[00, about, to, sleep, can, find, my, remot]
|
+---------+--------------------------------------------------------------------+--------------------------------------------------
---------------------------------+
only showing top 2 rows
```

Figure 20 - l'ensemble de donnée après la tokenisation

5.3 La lemmatisation

La lemmatisation est par définition une action consistant à l'analyse lexicale d'un texte avec pour but de regrouper les mots d'une même famille. La figure ci-dessous illustre la lemmatisation de notre programme.

```
for word in tweet.split():
    if len(word)>1:
        # Lemmatizing the word.
        word = wordLemm.lemmatize(word)
```

Figure 21 - Lemmatisation de notre système

6 Résultats et discussion

Dans cette section nous présentons toutes les étapes de test qui ont été effectuées depuis le début. Nous utilisons différentes méthodes et techniques dans chaque test, en diversifiant dans l'étape de prétraitement qui est un facteur très critique dans le processus de l'analyse de sentiment.

Les classifieurs classiques Régression Logistique, Naïves Bayes et Machine à vecteurs de support forment les algorithmes que nous avons choisis pour mener cette étude, en raison de leur bonne performance et leur utilisation intensive. Nous allons ci-après présenter les résultats obtenus ensuite nous les comparerons à d'autres études antérieures.

6.1 Test et Evaluation

Avant de commencer les tests nous devons d'abord diviser l'ensemble de données en données de test et données d'entrainement pour la phase d'apprentissage car c'est la base de l'apprentissage automatique. La capture ci-dessous montre la division de notre dataset en 70% pour l'entrainement et 30% pour le test.

```
#divide data, 70% for training, 30% for testing
dividedData = pdf.randomSplit([0.7, 0.3], seed=12345)
trainingData = dividedData[0] #index 0 = data training
testingData = dividedData[1] #index 1 = data testing
train_rows = trainingData.count()
test_rows = testingData.count()
print ("Training data rows:", train_rows, "; Testing data rows:", test_rows)
```

```
Training data rows: 1119896 ; Testing data rows: 480104
```

Figure 22 - Division de l'ensemble de données

6.1.1 Test

Vu que nous avons travaillé avec une très grande Dataset (senti140) qui contient 1 600 000 tweets, nous avons songé à faire nos tests en prenant une partie de cet ensemble de données à savoir 100 000 tweets et effectué les différentes sous étapes du filtrage jusqu'à l'obtention d'une meilleure précision.

Nous allons expliquer simplement chaque test de chaque classifieur dans ce qui suit :

1. Régression Logistique :

❖ Les cinq premiers tests ont été effectués avec une partie de l'ensemble de donnée (100 000 tweets) en déversant dans l'étape 'Filtrage' de prétraitement

- Dans le 1er test nous avons essayé d'éliminer seulement les url dans le processus de prétraitement qui possède en principe cinq sous-étapes discutés dans le chapitre précédent (lemmatisation, élimination des mots vides … etc.). Nous avons obtenu une précision de 0.7152.

  Dans le 2eme test nous avons pensé à ajouter l'élimination des @ utilisateurs car nous admettons que ça n'as aucun impact sur le processus de catégorisation des sentiments, on estimant une amélioration de la précision, bien heureusement c'était le cas dont nous avons obtenus une précision de 0.7184.

- Pour le 3eme test nous avons pensés à prendre en considération les émoticônes dont nous avons créé un dictionnaire des émoticônes, en plus des url et @ utilisateurs mais cela a affectait négativement sur les résultats dont nous avons eu une très petite chute de 0.7184 à 0.7161.

- Après cela pour le 5eme test nous voulions voir l'importance du processus de prétraitement donc nous avons lancé une exécution sans cette étape, et effectivement la précision a chuté encore plus bas à 71%.

Finalement nous avons effectué une expérimentation avec notre ensemble de donnée complet (senti140) et tous les étapes du prétraitement dont le pourcentage de précision a élevé jusqu'à 76.52%.

2. Naïves bayes :

Les cinq premiers tests ont été effectués avec une partie de l'ensemble de donnée (100 000 tweets).

- Nous avons lancé les tests avec exactement les mêmes procédés du premier classifieur. Les résultats obtenus confirment la conclusion faite au-dessus, dont le processus de prétraitement est très important dans l'analyse de sentiment.
- Le classifieur Bayes a atteint une précision maximal de 75.90% avec tous le traitement inclus (élimination des mots vide, suppression des url, suppression des @nom d'utilisateurs et la liste des émoticônes), sans prétraitement a achevé une précision de 74.98%.

- Dans le premier test dont nous avons supprimé seulement les url il a aboutis une précision meilleure de 75%.

❖ Finalement nous avons effectué une expérimentation avec notre ensemble de donnée complet (senti140) dont le pourcentage de précision a élevé jusqu'à 76.68%.

3. Régression Logistique & Naïves Bayes & Machine à Vecteurs de Support :

La comparaison des modèles est dévoilée dans le tableau suivant, on résumant tous les résultats obtenus avec les différents sous étapes de filtrage :

3.1 Avec une partie de l'ensemble de donnée (100 000 tweets)

Tableau 3 - comparaison des deux classifieurs avec 100 000 tweets

Filtrage	**Logistique Régression**	**Naïves Bayes**	**Support à vecteur de machine**
url	71.52%	**75%**	**72.69%**
url + user	71.84%	**74.97%**	**72.81%**
url + user + emoji	71.61%	**74.96%**	**72.80%**
Tout prétraitement	73.71%	**75.90%**	**74.46%**
Sans prétraitement	71%	**74.78%**	**72.03%**

Le résumé des résultats d'évaluation expérimentale des classifieurs classiques avec un système de pondération de type HashingTF, et l'ensemble de données de 100 000 tweets présenté dans le tableau 3 concluait comme suit :

1) Le classifieurs Naïves Bayes atteint un meilleur niveau de précision dans tous les tests avec la combinaison des différentes sous étapes de filtrage on comparant avec le classifieurs Régression Logistique et Support à vecteur de machine.

2) La liste des émoticônes a produit une petite chute pour les deux classifieur dont le Naïves bayes a chuté d'une précision de 75% à 74.96%. Pareillement le classifieurs SVM et Régression Logistique qui à diminuer de 71.84% à 71.61%.

3) **Le taux de changement de la précision entre les différents tests est vraiment petit, ce qui implique que le filtrage (Url, @utilisateur, émojis…) n'as pas un impact sérieux sur la performance**.

❖ **Après avoir lu d'autres articles nous avons constaté que c'est les traitements linguistiques (élimination des mots vides, lemmatisation, tokenisation) qui peuvent impacter sur la performance des systèmes. Donc nous avons songé à refaire deux autres tests pour les trois classifieurs.**

1. **Sans la lemmatisation :**

Pareillement à l'élimination des mots vides l'étape de lemmatisation est une étape très critique dans le processus de prétraitement. L'exécution de notre programme sans cette étape a connu une chute considérable dans les mesures de performance dont nous avons eu une précision équivalente à 73.76% pour le Naïve Bayes et 70% pour le model Régression Logistique et 72.28 pour la machine à vecteur de support (SVM en anglais), comme le dévoile les captures d'écrans ci-dessous.

```
Class positive
-----------------
precision = 0.6904109589041096
recall = 0.7528005974607916
F1 Measure = 0.7202572347266881

Class negative
-----------------
precision = 0.7828083989501312
recall = 0.7252279635258359
F1 Measure = 0.752918901861786

Accuracy-----------------
0.7376005361930295
```

Figure 23 - la précision de Naïves Bayes sans lemmatisation

```
Class positive
-----------------
precision = 0.7184931506849315
recall = 0.6856209150326797
F1 Measure = 0.7016722408026755

Class negative
-----------------
precision = 0.6843832020997376
recall = 0.717331499312242
F1 Measure = 0.7004701141705842

Accuracy-----------------
0.7010723860589813
```

Figure 24 - la précision de Régression Logistique sans lemmatisation

```
Class positive
-----------------
precision = 0.7349315068493151
recall = 0.7091870456047588
F1 Measure = 0.7218298015472586

Class negative
-----------------
precision = 0.7112860892388452
recall = 0.7369136641740313
F1 Measure = 0.723873121869783

Accuracy-----------------
0.7228552278820375
```

Figure 25 - la précision de machine à support de vecteur sans lemmatisation

2. **Sans l'élimination des mots vides :**

L'étape de l'élimination des mots vides est une étape très importante dans le processus de prétraitement, pour le confirmer nous avons effectué un test en gardant les mots vides dans le corpus, la précision a chuté jusqu'à 72.97% pour le Naïve Bayes, 69.47 pour le modèle Régression Logistique et 70.50% pour la machine à vecteur de support comme le montre les captures d'écrans ci-dessous.

```
Class positive
-----------------
precision = 0.6923076923076923
recall = 0.7038770053475936
F1 Measure = 0.69804441498177

Class negative
-----------------
precision = 0.6971975393028025
recall = 0.6854838709677419
F1 Measure = 0.6912910877668587

Accuracy-----------------
0.6947050938337802
```

Figure 26 - la précision de Régression Logistique

Sans éliminations des mots vides.

```
Class positive
-----------------
precision = 0.7100591715976331
recall = 0.7463718037318591
F1 Measure = 0.7277628032345014

Class negative
-----------------
precision = 0.7491455912508544
recall = 0.7130774235523748
F1 Measure = 0.7306666666666668

Accuracy-----------------
0.7292225201072386
```

Figure 27 – la précision de Naïves Bayes

Sans éliminations des mots vides.

```
Class positive
-----------------
precision = 0.715318869165023
recall = 0.7087947882736156
F1 Measure = 0.712041884816754

Class negative
-----------------
precision = 0.6944634313055366
recall = 0.7011732229123534
F1 Measure = 0.6978021978021979

Accuracy-----------------
0.7050938337801609
```

Figure 28 - la précision de machine à vecteur de support

Sans éliminations des mots vides.

Le tableau 3 ci-dessous rassemble les résultats des trois modèles sans les traitements linguistiques*.*

Tableau 4 - Comparaison des trois classifieurs sans les traitements linguistiques

Prétraitement	**Régression Logistique**	**Naïves Bayes**	**Machine à vecteurs de support**
Sans lemmatisation	**70.10%**	***73.76%***	**72.28%**
Sans élimination Des mots vides	**69.74%**	**72.92%**	**70.50%**

- Nous pouvons valider que les traitements linguistiques ont vraiment un impact considérable sur le processus d'analyse de sentiment car nous avons eu des résultats très minimales par rapport aux résultats avec tous le prétraitement.

3.2 Avec tout l'ensemble de données (1 600 000 tweets)

Tableau 5 - comparaison des trois classifieurs avec tout l'ensemble de données

Prétraitement	**Régression Logistique**	**Naïves Bayes**	**Machine à vecteurs de support**
Tous les prétraitements	76.52%	76.68%	76.75%

En ce qui concerne le tableau 4, les résultats des trois classifieurs en utilisant tout l'ensemble de données dont le Naïves Bayes a atteint une précision intéressante de 76.68% de l'autre côté le modèle Régression Logistique a abouti une précision proche de 76.52% et le modèle machine à vecteurs de support a atteint la meilleure précision 76.75%.

D'après le tableau nous voyons clairement que le classifieur Machine à vecteur de support surclasse les deux autres modelés, de plus, Naïves Bayes est plus performant que Régression Logistique.

6.1.2 Evaluation

Après avoir expliqué les mesures de performances, nous allons présenter deux exemples pour chaque classifieur avec l'ensemble de données complet en illustrant la matrice de confusion et la méthode d'évaluation de Framework Spark.

```
# Train our classifier model using training data
nb = NaiveBayes(labelCol="sentiment", featuresCol="features",
                        smoothing=1.0, modelType="multinomial")
NBmodel = nb.fit(numericTrainData)
print ("Training is done!\nEvaluation in progress ...")
model_Evaluate(NBmodel)
```

```
Training is done!
Evaluation in progress ...
+----------+---------+
|prediction|sentiment|
+----------+---------+
|0.0       |0        |
|0.0       |0        |
|0.0       |0        |
|0.0       |0        |
|0.0       |0        |
+----------+---------+
only showing top 5 rows

Class positive
-----------------
precision = 0.7663208918242503
recall = 0.7671962745400499
F1 Measure = 0.7667583333333332

Class negative
-----------------
precision = 0.7672971700275455
recall = 0.7664220511902631
F1 Measure = 0.7668593609435912

Accuracy-----------------
0.7668088580807492
```

Figure 29 - Précision Rappel et F1-score de Naïves Bayes

Nous pouvons voir en détail les différents critères de mesure de performance (Précision Rappel et F1-score) dans les deux classes positives et négatives. Nous remarquons que la performance dans le classement des sentiments positif est

presque égale à celle de la classe négative. Autrement dit le classement des deux classes est équilibré.

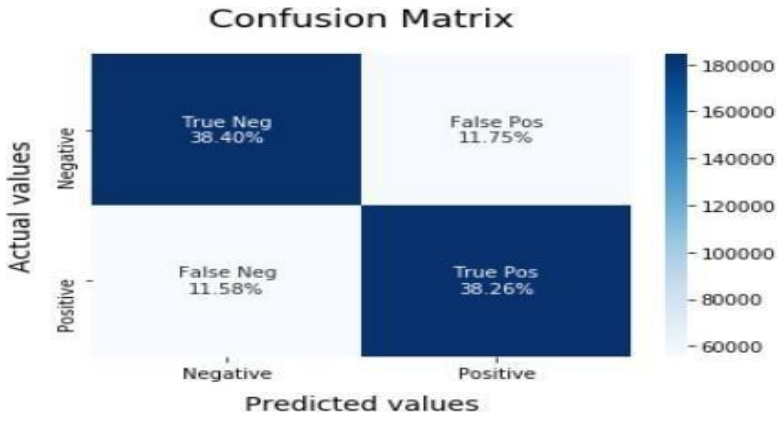

Figure 30 - Matrice de confusion Naïves Bayes

La matrice de confusion appuit nos conclusion discutées juste avant, dont nous voyons clairement que la prédiction de la classe négative est presque égale à la prédiction de la classe positive avec une petite différence de 0.14% entre le vrai positif et le vrai négatif, pareil pour le coté des faux positif et faux négatif dont l'écart est très minimal aussi 0.17%.

```
# Train our classifier model using training data
lr = LogisticRegression(labelCol="sentiment", featuresCol="features",
                        maxIter=10, regParam=0.01)
LRmodel = lr.fit(numericTrainData)
print ("Training is done!\nEvaluation in progress ...")
model_Evaluate(LRmodel)
```

```
Training is done!
Evaluation in progress ...
+----------+---------+
|prediction|sentiment|
+----------+---------+
|0.0       |0        |
|0.0       |0        |
|1.0       |0        |
|0.0       |0        |
|0.0       |0        |
+----------+---------+
only showing top 5 rows

Class positive
-----------------
precision = 0.7797174113110433
recall = 0.7578930751420754
F1 Measure = 0.7686503595114011

Class negative
-----------------
precision = 0.750744894089604
recall = 0.77302061342533
F1 Measure = 0.7617199309118661

Accuracy-----------------
0.76523628213887
```

Figure 31 - Précision Rappel et F1-score de Logistique Régression

Nous avons obtenus une précision de 77.90% pour la classe positive, un peu moins pour la classe négative 75.16% par contre le rappel est plus bon dans cette dernière avec 77.27% et 75.84 pour la classe positive, en parlant de F1 mesure ou F1-score nous observons qui sont presque égaux dans les deux classe avec un écart de 6%.

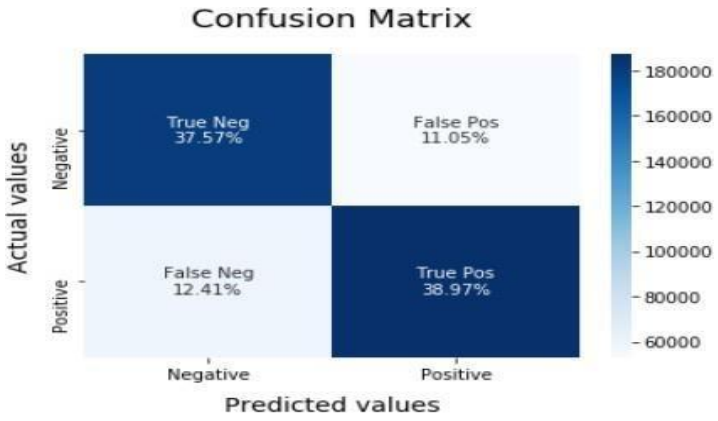

Figure 32 - Matrice de confusion Logistique Régression

La matrice de confusion soutiens nos observation de la figure 27, dont nous voyons clairement que la prédiction de la classe positive est plus bon que dans la classe négative avec un vrai positif de 38.97% et un vrai négatif de 37.57% qui est 1.4% moins bon, en outre le faux positif est égale à 11.05% et le faux négatif est 1.36% plus élévé (moins bon car on est dans le sens des faux) avec 12.22%.

Une classification parfaite donnerait la coordonnée (0,1) pour le rappel et la précision, représentant 100% de sensibilité (aucun faux négatif) et 100% de spécificité (aucun faux positif).

Dans le tableau 5 ci-dessous nous allons résumer tous les mesures de performance dans les deux classes positive, négative pour les trois classifieurs utilisés.

Tableau 6 - Mesures de performances des algorithmes d'apprentissage utilisés

Modèles de classification		Précision	Rappel	F1 – mesure	Accuracy (précision final des deux classes)
Naïves Bayes	Classe positive	76.63%	76.71%	76.67%	76.68%
	Classe négative	76.72%	76.64%	76.68%	
Régression Logistique	Classe positive	77.97%	75.78%	76.86%	76.52%
	Classe négative	75.07%	77.30%	76.17%	
Machine à vecteurs de support	Classe positive	78.59%	75.81%	77.18%	76.75%
	Classe négative	74.90%	77.76%	76.31%	

6.2 Comparaison des méthodes antérieurs sur l'analyse des sentiments dans le Big Social Data

La classification des sentiments sur les réseaux sociaux en générale et Twitter particulièrement a suscité un intérêt croissant dans la recherche ces dernières années, afin de mesurer la robustesse et la performance de notre system de classification, nous avons extrait les résultats des articles que nous avons eu l'opportunité de lire et les comparer avec nos modelés.

Tableau 7 - Performances des méthodes antérieurs d'analyse de sentiments

Approches	Etudes	Ensemble de donnée utilisé	Méthodes, classifieurs	Performance
Basé sur lexique	(Balahur et al. 2013)	critiques d'articles de journaux	Sentiwordnet	82.0%(précision)
	(Jaap, Kamps et al. 2004)		wordnet	60,4% (précision)
	(Edison and Aloysius 2017)	Twitter	Synset	73,27% (précision)
	(Rehman and Bajwa 2017)	sites web	Dictionnaire anglais traduit en urdu	66% (précision)
Basé sur l'apprentissage Automatique	**(Guerroumi et Sahnoune 2020)**	**Twitter**	**Régression Logistique, Naïves Bayes, SVM**	**76.52%, 76.68%, 76.75% (précision)**
	(Zhao et al. 2012)	Twitter	Naïve Bayes + émoticons	58,3 % (F-score)
	(Jin, Wei, Hay Ho, Hung, and K. Srihari, Rohini 2009)	commentaire d'Amazon	Bootstrapping	74.8% (F-score)
	(Kessler and Nicolov 2009)	Blog	Support Vector Machine SVM	69.8% (F-score)
	(Pang, Lee, and Vaithyanathan 2002) [17]	Critique du film IMDb	SVM	82.9% (précision)

Le tableau ci-dessus résume les pourcentages de performance des différentes technologies utilisées pour l'analyse de sentiments produites par des tests menés par différents chercheurs dans différents domaines. Nous pouvons constater qu'une meilleure approche n'existe pas, la performance est oscillatoire selon la méthode et l'ensemble de donnée utilisé.

Nous pouvons également remarquer que les résultats que nous avons obtenus sont très intéressants et montrent que nos classifieurs sont performants et très compétitifs dans le domaine de l'analyse des sentiments.

7 Conclusion

Nous nous sommes concentrés principalement dans ce chapitre sur l'implémentation et la discussion des résultats obtenus. Nous avons utilisé une combinaison de différentes méthodes de prétraitement pour faire la représentation vectorielle des données (Tokenisation, Stemming, lemmatisation, élimination des mots vides), réduire le bruit dans le texte, nous avons choisi les plus communs pour avoir une meilleure et plus efficace analyse.

Après avoir prétraité les tweets, l'étape qui suit serait la catégorisation qui est essentiellement le classement des tweets en fonction du sentiment exprimé: positif, négatif.

L'évaluation du classifieur est une étape très critique et importante pour tous projet de classification, et pour ce faire nous aurons besoin des critères de mesure de performance cité plus haut.

Nous avons également présenté les expériences effectuées sur l'ensemble de données pour y arriver à une meilleure précision de 76.52% pour le clissifieur Régression Logistique, 76.68% pour le classifieur Naïve Bayes et 76.75% pour la machine à vecteur de support.

Afin de valider le niveau d'exactitude et de robustesse de notre système nous l'avons évalué et comparé par rapport aux autres travaux similaires dans l'état de

l’art, et les résultats obtenus confirment l’efficacité de notre système de classification de sentiments.

Conclusions et perspectives

Ce mémoire de fin d'études aborde l'analyse des sentiments des réseaux sociaux (Twitter) dans un contexte Big Data en utilisant l'approche basé sur l'apprentissage automatique avec trois méthodes différentes Logistique Régression, Naïve Bayes et Machine à vecteur de support.

Conclusions

L'utilisation des médias sociaux est devenue une opération quotidienne de tout le monde dans l'époque numérique d'aujourd'hui. Par conséquent, d'énormes quantités de données en temps réel sont générées chaque seconde à travers le monde, principalement sous formes de messages textuelles non structurés. Le traitement de ce type de données qualifiées de Big Social Data, cette analyse peut fournir des informations cruciales pour la prise de décision. Pour cela nous avons introduit dans ce mémoire un système de classification de sentiments selon deux axes (positif, négatif).

En ce qui concerne les données utilisées nous avons utilisé un très grand ensemble de données qui contient 1 600 000 tweets partitionné en deux 50% positif, 50% négatif, pareillement aux études antérieurs qui reposent principalement sur des classifications binaires (négative ou positive).

Pour répondre à la problématique nous avons travaillé dans un environnement Big Data en utilisant ses outils de traitement tels que Spark et Hadoop et vu que la source de donnée est le réseau social Twitter cela nous mène au contexte de Big Social Data.

Afin de développer un système automatique d'analyse des sentiments, nous avons implémenté une approche Machine Learning avec trois diverses méthodes à

savoir Logistique Régression, Naïve Bayes et Machine à vecteur de support. Après plusieurs tests nous avons constaté que l'algorithme SVM à surclasser les modèles Naïve Bayes et Régression Logistique.

Afin d'optimiser notre système nous avons effectué plusieurs scénarios en utilisant plusieurs paramètres dans l'étape de prétraitement à savoir la suppression des mots vide, éliminations des url, la lemmatisation… etc. Et nous avons conclus que cette étape est très décisive dans le processus de l'analyse des sentiments.

Finalement, en vue de vérifier l'efficacité de nos trois modèles nous avons songé à les comparer avec les différents travaux similaires existant dans la littérature, nous avons constaté que les résultats que notre système a pu achever sont très compétitifs et intéressants.

Perspectives

Le modèle que nous avons proposé comporte trois méthodes de l'apprentissage automatique, il fonctionne bien pour classer les sentiments dans les tweets. Nous croyons que la précision pourrait être améliorée. Donc notre étude ne s'arrête pas là.

Vu que la technique de l'apprentissage profond est plus développée que l'apprentissage automatique, nous pensons que l'inclusion d'un classifieur basé sur l'apprentissage profond tels que les réseaux de neurones convolutifs et récurrents CNN, RNN peut améliorer la performance de notre système.

Nous voudrons aussi améliorer l'étape du prétraitement et ajouter d'autres langues plus délicates comme l'arabe par exemple.

Références bibliographiques

[1] Pang, B. et Lee, L. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval,Vol 2.

[2] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Paper presented at: 42nd Annual Meeting on Association for Computational Linguistics; 2004;

[3] B. Pang and L. Lee, “Opinion mining and Sentiment analysis”, Foundations and trends in information retrieval, Vol. 2, No. 1–2. Now Publishers, 2008,

[4] Rahnama, A.H.A : Distributed real-time sentiment analysis for big data social streams. In : 2014 International Conference on Control, Decision and Information Technologies (CoDIT), Nov 2014 (pp. 789-794). IEEE.

[5] Sariman, M. F., Yusof, A. M., & Tasir, Z. CHALLENGES OF RESEARCH UNIVERSITIES : LITERATURE ANALYSIS. *Sharing Visions and Solutions for Better Future*, 425.

[6] Mihalcea, R., Banea, C. et Wiebe, J. (2007). Learning Multilingual Subjective Language viaCross-LingualProjections. In Proceeding of the 45th Annual Meeting of the Association of Computational Linguistics, ACL ’07, (pp. 976-983)

[7] Big Data Technologies. (s. d.). Science Directe (Data-intensive applications, challenges, techniques and technologies : A survey on Big Data). Consulté le 8 juillet 2020, à l’adresse https://www.google.com/url

[8] Ounis, I., Macdonald, C., & Soboroff, I. (2008). Overview of the trec-2008 blog track. GLASGOW UNIV (UNITED KINGDOM).

[9] Hatzivassiloglou, V., & Wiebe, J. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.

[10] Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 105-112).

[11] Waldhör, K., & Rind, A. (2008). etBlogAnalysis—Mining virtual communities using statistical and linguistic methods for quality control in tourism. In *Information and communication technologies in tourism 2008* (pp. 453-462). Springer, Vienna.

[12] Claster, W., Pardo, P., Cooper, M., & Tajeddini, K. (2013). Tourism, travel and tweets: algorithmic text analysis methodologies in tourism. *Middle East Journal of Management*, *1*(1), 81-99.

[13] Schmunk, S., W. Höpken, M. Fuchs, and M. Lexhagen. 2014. “Sentiment Analysis: Extracting Decision-Relevant Knowledge from UGC.” In Information and Communication Technologies in Tourism 2014, edited by P. Xiang and I. Tussyadiah, 253-65. New York: Springer.

[14] Pappas, N., and A. Popescu-Belis. 2013. “Sentiment Analysis of User Comments for One-Class Collaborative Filtering over TED Talks.” In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, 773-76. New York: ACM.

[15] Gautam, G., & Yadav, D. (2014, August). Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In *2014 Seventh International Conference on Contemporary Computing (IC3)* (pp. 437-442). IEEE.

[16] Menner, T., W. Höpken, M. Fuchs, and M. Lexhagen, 2016. “Topic Detection— Identifying Relevant Topics in Tourism Reviews.” In Information and Communication Technologies in Tourism 2016, edited by A. Inversini and R. Schegg, 411-23. New York: Springer.

[17] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up ? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*..

[18] Wan, Y., & Gao, Q. (2015, November). An ensemble sentiment classification system of twitter data for airline services analysis. In *2015 IEEE international conference on data mining workshop (ICDMW)* (pp. 1318-1325). IEEE.

[19] Akaichi, J. (2013, September). Social networks' Facebook'statutes updates mining for sentiment classification. In *2013 International Conference on Social Computing* (pp. 886-891). IEEE.

[20] Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine learning-based sentiment analysis for twitter accounts. Mathematical and Computational Applications, 23(1), 11.

[21] Lai,S.;Xu,L.;Liu,K.;Zhao,J. Recurrent convolutional neural networks for texte classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, 25–30 January 2015.

[22] *Naive_Bayes*. (s. d.). https://scikit-learn.org/stable/modules/naive_bayes.html. Consulté le 15 mai 2020, à l'adresse https://scikit-learn.org/stable/modules/naive_bayes.html

[23] Naives_Bayes_Model. (s. d.). https://www.aclweb.org/anthology/P14-1146. Consulté le 2 mai 2020, à l'adresse https://www.aclweb.org/anthology/P14-1146.

[24] Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014, June). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1555-1565).

[25] *Modèles thématiques pour la découverte non supervisée*. (s. d.).

https://www.google.com/search. Consulté le 22 avril 2020, à l'adresse https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiF2Z7OqNrtAhWYQhUIHWX5DIcQFjAAegQIARAC&url=https%3A%2F%2Ftel.archives-ouvertes.fr%2Ftel-01655278v2%2Fdocument&usg=AOvVaw0Tz58UiVwC6oWXUFRCraUK

[26] *Word-Embedding*. (s. d.). https://dataanalyticspost.com/Lexique/word-embedding/. Consulté le 3 juin 2020, à l'adresse https://dataanalyticspost.com/Lexique/word-embedding/

[27] Kessler, J. S., & Nicolov, N. (2009, March). Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Third international AAAI conference on weblogs and social media*.

[28] Opinion Mining and Sentiment Analysis », *Foundations and Trends in Information Retrieval*, vol. 2, n^{o} 1-2, p. 1-135

[29] Bagheri, A., Saraee, M., & De Jong, F. (2013). Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews. *Knowledge-Based Systems*, *52*, 201-213.