REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
E DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité :Génie Informatique
Par :

MORSLI Kadda
LANTRI Naima

Sur le thème

Classifieur de tweets haineux

Soutenu le 02/12 /2020 à Tiaret devant le jury composé de :

Mr Talbi Omar	MCB	Université Ibn-Khaldoun Tiaret	Président
Mme LAKHDARI Aicha	MAA	Université Ibn-Khaldoun Tiaret	Encadreur
MrMeghazi Hadj	MAA	Université Ibn-Khaldoun Tiaret	Examinateur

Année Universitaire : 2019 /2020

Remerciements

Nous remercions tout d'abord Dieu tout puissant de nous avoir donné le courage, la force et la patience d'achever ce travail.

Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à notre encadreur Madame LAKHDARI Aicha, pour la confiance qu'elle nous a accordé en acceptant d'encadrer ce mémoire avec un grand intérêt et une grande compétence, pour ses disponibilités, ses soutiens, ses conseils, pour la qualité de son suivi durant toute la période de ce travail, et pour les encouragements qui nous ont permis de mener à bien ce travail.

Nous remercions également les membres de jury Monsieur TALBI Omar et *MEGHAZI hadj d'avoir pris le temps d'évaluer notre travail.*

Ces remerciements vont tout au corps professoral et administratif de la Faculté des Mathématiques et informatiques, spécialement ceux du département Informatique pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée

Nous adressons mes sincères remerciements Monsieur OUARED Abdelkader pour la qualité de son enseignement, ses conseils et son intérêt incontestable qu'il porte à tous les étudiants. Je voudrais exprimer notre reconnaissance envers les amis et collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Merci à tous ceux qui ont participé de près ou de loin pour la réalisation de ce travail.

Dédicaces

Je dédie ce travail :

A ma chère mère

Qui m'a soutenu et encouragé durant ces années d'études. Sa prière et sa bénédiction m'ont été d'un grand secours pour mener à bien mes études. Qu'elle trouve ici le témoignage de ma profonde reconnaissance.

A mon cher père

Qui a été à mes côtés pour me soutenir et m'encourager. Il a su m'inculquer le sens de la responsabilité, de l'optimisme et de la confiance en soi face aux difficultés de la vie. Ses conseils ont toujours guidé mes pas vers la réussite. Que ce travail traduit ma gratitude et mon affection.

A mes frères, mes grands-parents et ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

A tous mes amis Qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

A tous ceux que j'aime.

MORSLI Kadda

Je dédie ce travail :

A mes chères parents

Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez. Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

A mon très cher frère sofiane

Mon cher frère qui m'est le père et la mère, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour vous. Mon ange gardien et mon fidèle compagnon dans les moments les plus délicats de cette vie mystérieuse. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.

LANTRI Naima

Table des matières

Remerciements
Dédicaces
Liste des figures
Liste des tableaux
Liste des acronymes et abréviations
Introduction générale……1

CHAPITRE I : ANALYSE DES SENTIMENTS

1.1 Introduction 4
1.2 Analyse des sentiments....4
1.2.1 Caractéristiques de l'analyse des sentiments....5
1.3 Les applications de l'analyse du sentiment 7
1.4 Algorithmes d'analyse des sentiments 8
1.4.1 L'approche automatique....8
1.4.2 Approche à base de règles (Rule-based)....10
1.4.3 Approche hybride 11
1.5 L'analyse des sentiments et Twitter 11
1.5.1 Twitter et tweet....11
1.5.2 Caractéristiques d'un tweet 11
1.5.3 L'analyse des sentiments avec Twitter....12
1.5.4 Analyse qualitative Twitter 13
1.6 Outils d'analyse de sentiments....13
1.6.1 SentiWordNet....13
1.6.2 Sentiment140....15
1.6.3 Tweetfeel 15
1.6.4 Twitrratr....15
1.6.5 Tweet Sentiments Analyses....16
1.7 Mesures de performance de la classification des sentiments....16
1.7.1 La matrice de confusion 16
1.7.2 Accuracy, precision, recall and F-measure....17
1.7.3 La courbe ROC et AUC....17
1.8 Conclusion....19

CHAPITRE II : DEEP LEARNING

2.1 Introduction 21
2.2 Histoire du deep learning....21
2.3 Apprentissage en profondeur....22
2.4 Les applications de l'apprentissage en profondeur 23
2.4.1 La reconnaissance faciale 23
2.4.2 La détection d'objets 24
2.4.3 Le traitement du langage natural 24
2.5 Le réseau neuronal....25

2.5.1 Fonctionnement du neurone artificiel 26
2.5.2 Avantages et inconvénients des réseaux neurones 26
2.6 Architectures de réseaux de neurones profonds 27
2.6.1 Les réseaux neuronaux récurrents (RNN) : 27
2.6.2 LSTM (long short term memory) 28
2.6.3 Architecture LSTM: 29
2.7 La Promesse de l'apprentissage en profondeur 29
2.8 Conclusion 30

CHAPITRE III : Conception & Implémentation

3.1 Introduction : 32
3.2 Les outils d'expérimentation utilisés 32
3.3 Architecture générale du système 34
3.4 Description du dataset 35
3.5 Les étapes du prétraitement 35
3.5.1 Prétraitement 35
3.5.2 Annotation via une ressource lexicale 36
3.6 Base d'apprentissage : 41
3.6.1 Analyse de la base d'apprentissage : 41
3.7 Extraction et Présentation des descripteurs 44
3.7.1 Tokenisation 45
3.7.2 StopWords 45
3.7.3 LaLemmatisation 46
3.7.4 Partie du marquage de la parole (POST) 46
3.7.5 Représentation vectorielle du texte 47
3.7.6 La transformation des caractéristiques 48
3.8 La classification des tweets 49
3.8.1 Train et test 50
3.8.2 Quelquesalgorithmesde ML 50
3.9 Evaluation des résultats 52
3.9.1 Matrice de Confusion 52
3.9.2 Résultats de l'apprentissage automatique via sentiwordnet 53
3.9.3 Résultats de l'apprentissage automatique via le dictionnaire donné 54
3.10 Conclusion 56

Conclusion générale 58
Bibliographie 60
Résumé

Listes des figures

CHAPITRE I : ANALYSE DES SENTIMENTS

Figure 1- 1 : Analyse des sentiments [2]. 5
Figure 1- 2 : Flux des tâches de l'analyse des sentiments. 5
Figure 1- 3 : Les niveaux d'analyse [1]. 6
Figure 1- 4: Algorithmes d'analyse des sentiments [3]. .... **Erreur ! Signet non défini.**
Figure 1- 6 : l'approche automatique [3]. 8
Figure 1- 7 : Arbre de decision [8]. 10
Figure 1- 8 : Approche à base de règle [4]. 10
Figure 1- 9 : Diagramme montrant la structure de SentiWordNet [19]. 15
Figure 1- 10 : Matrice de confusion [23]. 16
Figure 1- 11 : La courbe ROC [24]. 18
Figure 1- 12 : AUC (aire sous la courbe ROC [24]. 18

CHAPITRE II : DEEP LEARNING

Figure 2 - 1 : La différence de performance entre le DL et la plupart des algorithmes de ML en fonction de la quantité de données [28] 23
Figure 2 - 2 : Réseaux de neurones[32]. 26
Figure 2 - 3 : Représentation mathématique/informatique d'un neurone biologique [33]. 26
Figure 2 - 4 : RNN Architecture [35]. 28
Figure 2 - 5 : Représentation simplifiée d'une cellule LSTM [36]. 29

CHAPITRE III : Conception & Implémentation

Figure 3- 1 : Architecture générale du système. 34
Figure 3- 2 : Extrait du dataset 35
Figure 3- 3 : Ecran des tweets et leurs polarités (Approche sémantique). 41
Figure 3- 4 : Analyse exploratoire des données pour le traitement du langage naturel. 42
Figure 3- 5 : Analyse statistique. 42
Figure 3- 6 : Exploration de Ngram (ngram, bingram). 43
Figure 3- 7 : Dataset du mot nuage. 43
Figure 3- 8 : Annotation via Sentiwordnet : (Positive, Négative). 44
Figure 3- 9 : Annotation via le dictionnaire donné : (Positive, Neutre, Négative). 44
Figure 3- 10 : Extraction et présentation des descripteurs 45
Figure 3- 11: Tokenisation 45
Figure 3- 12 : Exemple de stopWord. 46
Figure 3- 13 : Exemple de lemmatisation. 46
Figure 3- 14: Exemple global de prétraitement du texte 47
Figure 3- 15 : Résultats moyens de performance sentiwordnet des classifieur RNN-Lstm. 54
Figure 3- 16 : Résultats moyens de performance dictioner des classifieur RNN-Lstm. 56

Listes des tableaux

CHAPITRE II : DEEP LEARNING

Table 2- 1 : Les étapes majeurs du Deep Learning [27]. 22
Table 2- 2 : Différences entre ML & DL [28]. 23
Table 2- 3 : Algorithmes d'apprentissage et leurs utilisation avec le NLP [31]. 25

CHAPITRE III : Conception & Implémentation

Table 3- 1 : Exemple de tweets avant et après le prétraitement. 36
Table 3- 2 : Exemples de tweets étiquetés. 39
Table 3- 3 : Exemples de tweets étiquetés. 40
Table 3- 4 : Exemple de BOW 48
Table 3- 5 : Résultats des classifieurs via la matrice de confusion. 53
Table 3- 6 : Résultats des classifieurs classiques+ TF-IDF sur les tweets 53
Table 3- 7 : Résultats des classifieurs DL + Word Embedding sur les tweets 53
Table 3- 8 : Résultats des classifieurs classiques+ TF-IDF sur les tweets. 55
Table 3- 9 : Résultats des classifieurs DL + Word Embedding sur les tweets. 55

Liste des Acronymes et Abréviations

IA: Intelligence Artificielle

SVM: Machine à vecteurs de support

NB : Naïve Bayes

NLP : Natural Language Processing

POST: Part Of speech Tagging

AUC: Area Under The Curve

CNN : Convolutional Neural Network

RNN: Recurrent Neural Network

LSTM: Long short terme memory

AUC: **Area Under The Curve**

ML: Machine Learning

DL: Deep Learning

ANN: Artificial neural network

KNN: k-nearest neighbors

NLTK: Natural Language Toolkit

BOW : Bag of Words

TF-IDF: Term Frequency-Inverse Document Frequency

Introduction générale

Introduction générale :

L'analyse des sentiments est le processus qui consiste à déterminer « par ordinateur » si une écriture est positive, négative ou neutre. Il est également connu sous le nom de mine d'opinion, dérivant de l'opinion ou de l'attitude d'un orateur.

La classification est l'action de ranger par classes, par catégories des objets avec des propriétés communes. Il existe deux catégories de classification : classification supervisée et classification non supervisée. Dans la première catégorie, les méthodes consistent à classer les objets à partir d'une base de données dite d'apprentissage, tandis que dans l'autre les méthodes classent les objets sans besoin de cette base de données.

Le Deep Learning ou « apprentissage profond » est une famille d'algorithmes d'apprentissage (machine learning) pour entraîner des réseaux de neurones composés de plusieurs couches internes et potentiellement un grand nombre. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage.

La violence verbale est considérée comme étant « l'expression/l'explicitation d'une intention de communiquer quelque chose de valeur axiologique négative est le trait le plus saillant de la signification comme dans le cas de menacer, insulter, harceler, injurier, invectiver, maudire, accuser, reprocher, critiquer…etc »[1]. Ainsi, l'utilisation de la violence verbale lors des échanges communicationnels se fait d'une manière explicite ou bien directement envers le récepteur du message, et ce phénomène langagier peut prendre plusieurs formes : paroles blessantes, injures, insultes, harcèlement, moqueries, menaces, humiliations…etc., elles entrainent des conflits flagrants entre les personnes impliquées, mais les conséquences ne sont pas si graves. La victime ne ressentira pas des coups physiques mais des blessures à l'intérieur.

Cependant, le harcèlement par Internet, aussi appelé « cyberintimidation ou cyber harcèlement », consiste en l'utilisation du réseau Internet pour harceler, menacer ou embarrasser quelqu'un de façon malicieuse. L'un des types de harcèlement peut se manifester sous la forme de transmission de messages négatifs directement à la victime.

Sur les principaux réseaux sociaux, pas un jour ne se suit sans qu'une polémique absurde ne naisse, et Twitter est certainement l'un des outils les plus utilisés pour déverser sa haine ou pour faire du cyber-harcèlement. Cependant, nous essaierons de répondre à notre problématique : *Arriverons nous à nettoyer le Net en général et Twitter en particulier en développant un détecteur textuel de harcèlement par la réalisation d'un classifieur de sentiments ?.*

Introduction générale

L'analyse des sentiments (Sentiment Analyses ou Opinion Mining) forme une méthode de traitement automatique du langage naturel (Naturel Langage Processing) qui tente de repérer la présence des sentiments (ou d'émotions) malicieux ou violents exprimés dans un tweet.

En effet, notre travail vise à conduire une étude pilote pour construire un détecteur de sentiments haineux en s'appuyant sur des techniques de classification automatique du texte à savoir: machines à support vectoriel (SVM), Naïve baise , K neighbors classifier, Decision tree, mémoire à court et long terme (LSTM), et la combinaison (RNN-LSTM), ainsi que l'évaluation de leur pertinence à l'aide de différentes stratégies de prétraitement telles que Stop Words Removing, Stemming, N-grammes et différents schémas de pondération et représentation (TF-IDF, Word Embedding), ainsi que les techniques de l'approche lexicale, pour l'analyse automatique des tweets écrits en anglais. Cette analyse sera appliquée à des tweets préalablement collectés et liés à la violence verbale "exemple le harcèlement".

Afin de parvenir à notre objectif, le présent document est divisé en deux volets. Un volet théorique ainsi qu'un volet pratique. Le volet théorique (chapitre 1 & 2) explore respectivement l'essence de l'analyse des sentiments et le deep learning. Le second volet, chapitre 3, débute par la conception du détecteur textuel de tweets de harcèlements et ceux de non harcèlements et se termine par une présentation détaillée des résultats.

Ce document se termine par une conclusion générale et des perspectives pour le futur.

Chapitre I

Analyse des sentiments

CHAPITRE I : ANALYSE DES SENTIMENTS

1.1 Introduction

L'analyse du sentiment se concentre sur la réalisation de classification du texte . Il existe de nombreux autres noms apparentés se référant légèrement à différentes tâches telles que l'extraction d'opinion, le sentiment mining, l'analyse de subjectivité, l'analyse d'affect, l'analyse d'émotion, les avis mining, ...etc. Ils font maintenant tous partie de l'analyse des sentiments ou de l'opinion mining.

Dans l'industrie, le terme d'analyse du sentiment est couramment le plus utilisé, mais dans le milieu universitaire, on utilise fréquemment à la fois l'analyse du sentiment et l'opinion mining [2]. L'importance croissante de l'analyse des sentiments coïncide avec la croissance des médias sociaux tels que les forums de discussion, les blogs, les micro-blogs et les réseaux sociaux. Ce sujet est un domaine en développement qui suscite l'intérêt des recherches et a pris beaucoup d'intérêt ces dernières années ce qui pousse à suggérer plusieurs travaux de recherches avec diverses méthodes utilisant des approches multiples.

Actuellement, l'un des meilleurs exemples de réseaux sociaux permettant d'observer l'évolution de ces opinions est Twitter.

1.2 Analyse des sentiments

Un domaine utilisant les techniques de IR (Recherche d'Informations), TC (Catégorisation de Texte), ML (Apprentissage Automatique) ou Fouille de Texte est notamment le domaine de *l'Analyse des Sentiments*, connu sur le nom de (ang : *Opinion Mining*). Les recherches dans ce domaine couvrent plusieurs sujets, notamment l'apprentissage de l'orientation sémantique des mots ou des expressions, l'analyse sentimentale de documents et l'analyse des opinions et attitudes à l'égard de certains sujets ou produits. Cependant, L'analyse de sentiments est l'étude computationnelle et sémantique des parties de textes en fonction des opinions, des sentiments et des émotions exprimés dans le texte [3].

Généralement, l'expression « analyse des sentiments » concerne le traitement d'un texte d'opinion pour extraire et catégoriser les opinions d'un certain document. La polarité des sentiments est généralement exprimée en termes d'opinion positive ou négative (*classification binaire*). Cependant, il peut s'agir d'une *classification multiple*, où le sentiment peut avoir une étiquette neutre (Figure 1-1) ou même une étiquette différente comme très positive, positive, neutre, négative, très négative, les étiquettes peuvent aussi être associées à des émotions comme la tristesse, la colère, le bonheur, ...etc.

Donc, Le but de l'analyse des sentiments est de définir des outils automatiques capables d'extraire des informations subjectives à partir du texte en langage naturel, telles que les opinions et

les sentiments, afin de créer des connaissances structurées et utilisables par un système d'aide à la décision ou par un décideur [2].

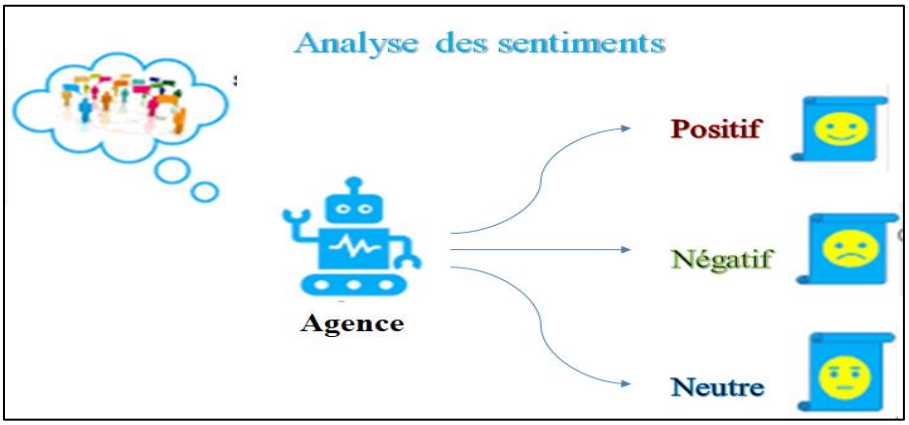

Figure 1- 1 : Analyse des sentiments [2].

1.2.1 Caractéristiques de l'analyse des sentiments

L'analyse des sentiments est un domaine de recherche vaste et complexe. Dans ce qui suit, les principales caractéristiques qui constituent le processus d'analyse des sentiments sont décrites et discutées en détail [4]:

1.2.1.1 Categorization des sentiments

Les phrases sont objectives ou subjectives. Lorsqu'une phrase est objective, aucune autre tâche fondamentale n'est requise. Lorsqu'une phrase est subjective, ses polarités (positive, négative ou neutre) doivent être estimées [4].

	Classification subjective	*Classification de polarité*
Phrase	*Subjective*	*Positive*
		Négative
		Neutre
	Objective	

Figure 1- 2 : Flux des tâches de l'analyse des sentiments.

- ✓ **Phrase subjective :** C'est une phrase dans laquelle l'écrivain exprime ses sentiments envers les entités, les événements et leurs propriétés.
- ✓ **Phrase objective :** une phrase objective indique des faits et des informations connues propos du monde

La classification de subjectivité (Subjectivité classification) est la tâche qui distingue les phrases exprimant des informations objectives des phrases exprimant des vues et opinions subjectives.[4].

L'iPhone est un smartphone – phrase objective

L'iPhone est génial – phrase subjective

1.2.1.2 Niveaux d'analyses

Comme mentionné précédemment, le but de l'analyse des sentiments est de " définir des outils automatiques capables d'extraire des informations subjectives à partir des textes en langage naturel ". Le premier choix lorsqu'on applique l'analyse des sentiments est de définir ce que signifie le texte (c'est-à-dire l'objet analysé) dans le cas d'étude considéré [2].

En général, l'analyse des sentiments dans les réseaux sociaux peut être étudiée principalement à trois niveaux (Figure1-3):

✓ ***Niveau de texte :***L'objectif est de détecter la polarité d'un texte d'opinion. Par exemple, dans le cadre d'une revue de produit, le système détermine si le texte exprime une opinion globale positive, négative ou neutre au sujet du produit. L'hypothèse est que l'ensemble du texte n'exprime qu'une seule opinion sur une seule entité (un seul produit) [2].

✓ ***Niveau de phrase :*** Le but est de déterminer la polarité de chaque phrase contenue dans un texte. L'hypothèse est que chaque phrase, dans un texte donné, désigne une seule opinion sur une seule entité [1].

✓ ***Niveau d'entité et d'aspect :*** Effectue une analyse plus fine que le niveau du texte et de la phrase. Elle repose sur l'idée qu'une opinion est constituée d'un sentiment et d'une cible (d'opinion). Par exemple, la phrase "L'iPhone est très bien, mais ils ont encore besoin de travailler sur la durée de vie de la batterie et la sécurité" évalue trois aspects : iPhone (neutre), la durée de vie de la batterie (négatif) et la sécurité (négatif) [2].

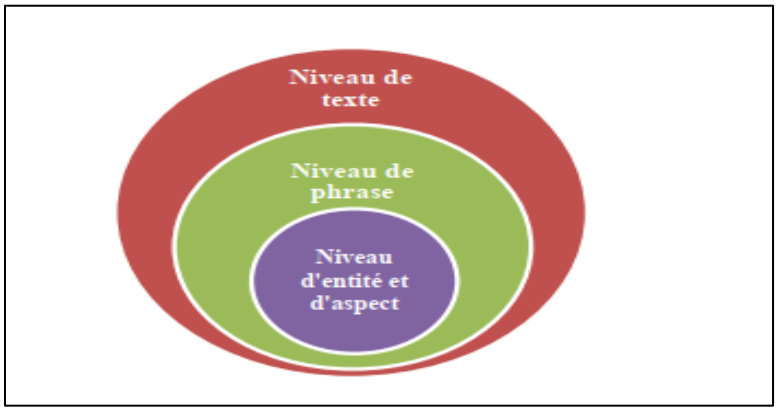

Figure 1- 3 : Les niveaux d'analyse [1].

1.3 Les applications de l'analyse du sentiment

L'utilisation de l'analyse du sentiment peut être trouvée dans de multiples domaines et a été utilisée par plusieurs petites ou grandes entreprises qui ont l'extraction d'opinion et l'analyse du sentiment comme un élément de leurs systèmes. L'utilisation peut être trouvée sous multiples formes [3] :

✓ ***En tant que sous-partie des systèmes*** : Les systèmes d'analyse du sentiment et d'analyse de l'opinion jouent un rôle important en tant qu'amélioration par rapport à d'autres systèmes comme les systèmes de recommandation, où l'on ne recommande pas des éléments qui reçoivent beaucoup de commentaires négatifs. De même, en ce qui concerne les systèmes d'extraction de l'information, il a été démontré que nous pouvons les améliorer et les rendre plus performants en coupant l'information dans des phrases subjectives. Les systèmes de réponse aux questions sont un autre domaine où l'analyse du sentiment peut être bénéfique. Le résumé du document peut faire mieux et donner des bons résultats dans le cas où nous avons intérêt à prendre en compte plusieurs points de vue, comme indiqué dans [3].

✓ ***L'utilisation par les industries et les gouvernements*** : L'analyse des sentiments peut être très utile pour des domaines spécifiques, par les gouvernements, où nous trouvons que certains d'entre eux essaient de comprendre l'opinion du public sur les décisions ou quelque chose de particulier comme les élections. L'utilité des médias sociaux, en particulier Twitter, il a été très utile de communiquer avec le public et de comprendre leurs opinions et de se concentrer sur la façon dont les électeurs pensent. De même, lorsqu'il s'agit d'industries où l'analyse du sentiment fait partie de leurs stratégies déplanification pour la prise de décision, elles ont construit ou sont en train de construire leurs propres systèmes d'analyse du sentiment, des entreprises comme Google, Microsoft, Hewlett-Packard, Amazon, Oracle, Adobe et Bloomberg. Ce sujet peut être utilisé dans le marketing par exemple lorsqu'il aide à comprendre le succès ou l'échec du lancement de nouveaux produits ou services, ou encore à déterminer quelle version d'un produit. Les grandes entreprises peuvent identifier leurs stratégies futures à partir des données relatives aux services ou produits déjà analysés [3].

✓ ***Autres domaines d'application*** : Les applications d'analyse du sentiment se sont étendues à presque tous les domaines possibles, des produits de consommation, des soins de santé, du tourisme, de l'hôtellerie, des services financiers et des événements sociaux. En plus des intérêts commerciaux, de multiples projets réels tentent d'implémenter des modèles d'analyse du sentiment comme les restaurants, les hôtels, les cosmétiques, les films, l'automobile, les compagnies de téléphonie mobile, etc. [3]

1.4 Algorithmes d'analyse des sentiments

Dans la littérature, il existe de nombreuses méthodes et algorithmes pour mettre en œuvre des systèmes d'analyse des sentiments, que l'on peut classer comme suit[3] :

- ✓ Approche automatique
- ✓ Approche à base de règles
- ✓ Approche hybride

1.4.1 L'approche automatique

Les approches automatiques reposent sur des techniques d'apprentissage automatique (Machine learning). La tâche d'analyse des sentiments est généralement modélisée comme un problème de classification dans lequel un classificateur est alimenté avec un texte et renvoie la catégorie correspondante, par ex. positif, négatif ou neutre (en cas d'analyse de polarité) [4](Figure 1- 4).

Il existe différentes classifications de méthodes d'apprentissage supervisé utilisées pour analyser les sentiments, certaines d'entre elles sont les suivantes:

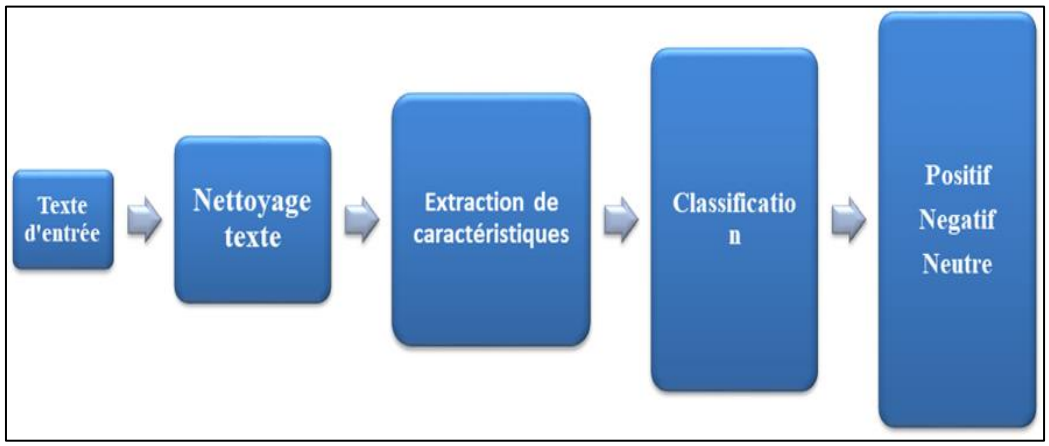

Figure 1- 5 : l'approche automatique[3].

1.4.1.1 Machine à vecteurs de support

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM). est un algorithme d'apprentissage automatique supervisé relativement simple utilisé pour la classification et / ou la régression. Elle est plus préférée pour la classification, mais elle est parfois très utile pour la régression également. Fondamentalement, SVM trouve un hyperplan qui crée une frontière entre les types de données. Dans l'espace à 2 dimensions, cet hyper-plan n'est qu'une ligne.

Dans SVM, nous représentons chaque élément de données de l'ensemble de données dans un espace à N dimensions, où N est le nombre d'entités / d'attributs dans les données. Ensuite, trouvez l'hyperplan optimal pour séparer les données. Donc, par cela, vous devez avoir compris que, par

nature, SVM ne peut effectuer qu'une classification binaire (c'est-à-dire choisir entre deux classes). Cependant, il existe différentes techniques à utiliser pour les problèmes multi-classes [5].

1.4.1.2 Naïve Bayes classificateur :

La classificateur Naïve Bayes est une technique d'apprentissage probabiliste supervisé qui est régulièrement utilisée à des fins de classification. Il est basé sur le théorème de Bayes où la probabilité conditionnelle qu'un événement X se produise, pour un indice Y, est déterminée par la règle de Bayes:

P(X/Y) = P(X) P(Y/X) / P(Y)

Ainsi, pour trouver le sentiment d'une phrase d'entrée, elle sera transformée en ce quisuit [6]:

P(Sentiment/Phrase) = P(Sentiment)P(Phrase/Sentiment) /P(Phrase)

Qui est formulé par ce qui suit :

Nombre du token dans la classe + 1 / (Nombre de tous les tokens de cette classe+ Nombredetous les tokens)

Le 1 supplémentaire est le nombre de tous les tokens s'appelle « l'ajout d'un » ou lissage deLaplace.

1.4.1.3 La régression logistique

C'est une méthode de classement qui a été remise en valeur et qui est devenu une méthode populaire grâce à ses bonnes performances dans le classement automatique. Elle permet de calculer la probabilité d'appartenance à la catégorie k, comme suit:

$$P_r(Y = k|X = x) = \frac{e^{\beta x}}{1 + e^{\beta x}}$$

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 :

$$\hat{y} = \underset{k \in \{1,\dots,k\}}{argmax}\, P(Y = k|\, x)$$

La régression logistique a bénéficié de beaucoup de travaux de recherches et est devenue une méthode pratique dans de nombreux systèmes commerciaux à grande échelle, en particulier aux grandes sociétés d'Internet comme Google et Yahoo qui l'emploient pour apprendre de grands ensembles de données [7].

Ce modèle, en plus de pouvoir être utilisé seul, constitue en outre le bloc fondamental des réseaux neuronaux

1.4.1.4 Arbre de décision

L'arbre de décision est l'outil le plus puissant et le plus populaire pour la classification et la prédiction. Un arbre de décision est un organigramme semblable à une arborescence, où chaque nœud interne indique un test sur un attribut, chaque branche représente un résultat du test et chaque nœud feuille (nœud terminal) détient une étiquette de classe [8].

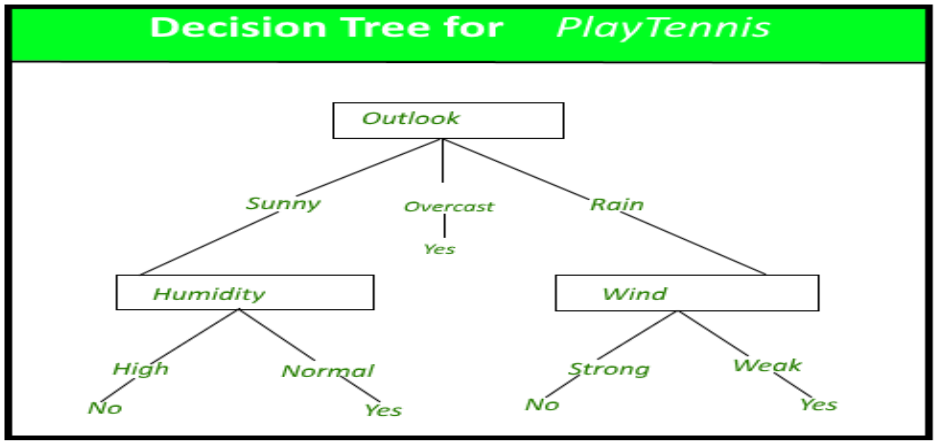

Figure 1- 6 : Arbre de decision[8].

1.4.2 Approche à base de règles (Rule-based)

L'approche à base de règles (ou l'approche 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 [4] :

- ✓ Techniques classiques de NLP, telles que la racinisation, tokenisation, POS–tagging et Chunking.
- ✓ Autre opération basée sur le lexique, ils utilisent le dictionnaire des sentiments avec des mots d'opinion et les faire correspondre avec les données pour déterminer la polarité.

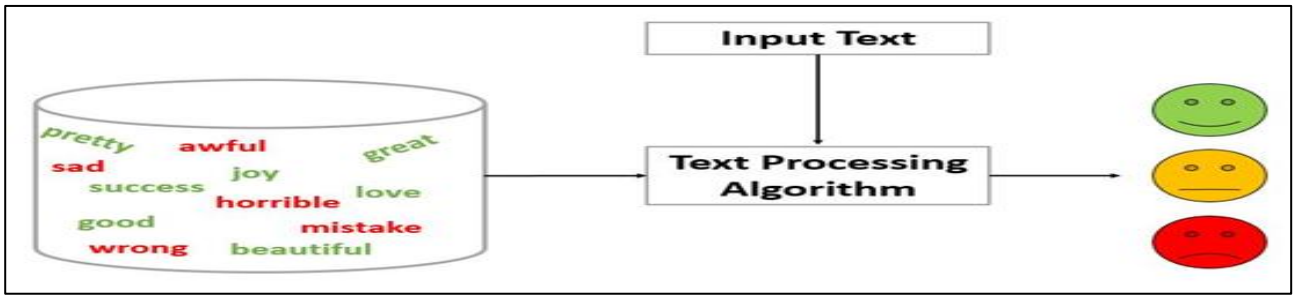

Figure 1- 7 : Approche à base de règle [4].

1.4.3 Approche hybride

Le concept de méthodes hybrides est très intuitif : combinez simplement le meilleur des deux approches, celui basé sur des règles et celui automatique. Généralement, en combinant les deux approches, les méthodes peuvent améliorer la précision [4].

1.5 L'analyse des sentiments et Twitter

En Mars 2006, Twitter a vu le jour par le développeur Jack Dorsey, comme outil de communication, pour rester en contact avec les amis. Twitter est un service Web qui permet aux utilisateurs d'envoyer et de lire un message court [9].

1.5.1 Twitter et tweet

Twitter est un réseau social et un microblog qui permet aux utilisateurs de publier des messages en temps réel, appelés tweets. Les tweets sont des messages courts, limités à 280 caractères. En raison de la nature de ce service de microblogging (messages rapides et courts), les gens utilisent des acronymes, commettent des erreurs d'orthographe, utilisent des émoticônes et d'autres symboles qui expriment des significations particulières [10].

Twitter est actuellement l'une des plates-formes de micro-blogage les plus populaires. Son premier slogan était : « Que faites-vous ? » néanmoins, son utilisation a pris une autre piste où les utilisateurs s'échangent des avis et des informations, le slogan devient « Quoi de neuf ? ».

Plusieurs célébrités utilisent Twitter, on y trouve même des chefs d'Etat.

Selon les derniers chiffres 2 :

- ✓ Twitter a plus que 645 millions d'utilisateurs inscrits.
- ✓ 58 millions de tweets envoyés chaque jour.

Dans le cadre de l'analyse des sentiments, la taille minime du message (280 caractères) formule l'hypothèse que ce dernier ne renferme, à priori, plus d'une seule idée, ce qui facilite l'identification de l'opinion exprimée. Certains tweets apparaissent comme des messages codés à cause de l'usage des hashtags, abréviations en tout genre, argot, et émoticons.

Les termes à connaître pour bien utiliser Twitter [11] :

- ✓ Followers : les personnes qui vous suivent.
- ✓ Followings : les personnes que vous suivez.
- ✓ Friends : les personnes que vous suivez et qui vous suivent.

1.5.2 Caractéristiques d'un tweet

On peut se sentir un peu perdu dans le vocabulaire utilisé dans les tweets, notamment, à cause du langage et des symboles spécifiques à l'utilisation de Twitter. A quoi sert le @ et # ?

C'est quoi RT ? Toutes ces abréviations peuvent paraître un peu floues. Dans une perspective de classement, un petit lexique des principaux mots et signes Twitter est présenté [12] [13] :

- ✓ Mention @ : se présente sous la forme @NomUtilisateur. Il cible un utilisateur de Twitter dans le tweet posté. Exemple : salut à vous de la part de @Hanae et @Moncef.
- ✓ Hashtag # : se présente sous la forme #mot-clé. Il identifie le mot-clé en question comme important et peut en faire un sujet populaire. Exemple : #gouvernement.
- ✓ RT (ReTweet) : se présente sous la forme RT Nom_Utilisatuer. Il permet de partager le tweet d'un utilisateur. Exemple : RT Hanae Excellente.
- ✓ URL (Lien) : se présente sous la forme https :// ou http ://www. Twitter, permet à l'utilisateur de rejoindre les liens dans son tweet. Exemple : https://www.fstf.com.
- ✓ VIA : s'utilise pour mentionner votre source d'information, dans votre tweet.

Example: Via YouTube, Via Facebook.

1.5.3 L'analyse des sentiments avec Twitter

Parmi les réseaux sociaux, Twitter est l'un des plus importants services de microblogging. L'architecture de Twitter fait de la question « Que se passe-t-il ?» La pierre angulaire de l'échange d'informations. Cela a inspiré l'idée d'utiliser les utilisateurs de Twitter en tant que capteurs distribués [14].

Les plates-formes des médias sociaux telles que Twitter sont de plus en plus courantes et fournissent des informations précieuses (générées par les utilisateurs via la publication et le partage de contenus) [15].

Twitter est une plate-forme de communication basée sur le Web, qui permet à ses abonnés de diffuser des messages appelés « tweets » de 280 caractères maximum, leur permettant de partager des pensées, des liens ou des images. Par conséquent, Twitter est une source riche de données pour l'exploration d'opinion et l'analyse de sentiment. La simplicité d'utilisation et les services offerts par la plate-forme Twitter lui permettent d'être largement utilisée dans le monde entier et en particulier en Algerie. Cette popularité nous donne accès à une mine riche d'informations qui peuvent servir comme base de données à l'analyse des tweets, qui nous fournissent des informations précieuses [16].

Twitter est une plate-forme multimédia qui permet de partager facilement des opinions en utilisant diverses formes de contenu, notamment du texte, des images et des liens contrairement à de nombreuses autres plates-formes de médias sociaux (tel que Instagram).

De plus, en fournissant un accès en temps quasi réel aux publications publiques via l'API, Twitter est une plate-forme appropriée pour l'exploration d'opinion à grande échelle en temps quasi réel.

1.5.4 Analyse qualitative Twitter

L'objectif du Sentiment Score est de calculer à quel point un rapport est positif ou négatif engénéral. Il est important de prendre en compte le nombre d'impacts (ou d'impressions) Twitter et d'utilisateurs qui ont tweeté de manière positive, négative ou neutre. Par conséquent, le Sentiment Score prend en considération, en général, les variables suivantes comme nous l'avons vu précédemment [17] :

- Nombre de tweets positifs, négatifs ou neutres
- Quantité d'utilisateurs qui ont participé au rapport
- Nombre d'impressions (impacts) positives, négatives ou neutres

Pourquoi avons-nous choisi ces variables ? Parce que l'analyse des sentiments de Twitter Binder sur Twitter considère « pluspositif» que plus d'utilisateurs tweetent de manière positive plutôt qu'un seul. Cela signifie que si 500 utilisateurs ont tweeté positivement à propos de notre hashtag, c'est mieux que si seulement 1 utilisateur envoie 500 tweets. Par exemple :

- 500 tweets positifs envoyés par 1 utilisateur = mauvais (probablement un spammeur ou quelqu'un associé à la campagne ou à l'événement Twitter)
- 500 tweets positifs envoyés par 100 utilisateurs = mieux (cela signifie que beaucoup de gens conviennent que l'hashtag est génial)

Même chose arrive avec des tweets négatifs. Plusieurs fois, les trolls tenteront de saboter un événement ou une campagne. Ces utilisateurs envoient généralement des tonnes de tweets mais ils ne sont pas un grand groupe d'utilisateurs. Cela signifie qu'une petite quantité d'utilisateurs enverra une grande quantité de tweets (et probablement ils ont un petit nombre d'abonnés). Ce n'est pas très négatif pour le rapport que ce petit nombre d'utilisateurs envoie des tonnes de tweets plutôt que s'ils provenaient d'une énorme quantité de comptes. Par exemple :

- 500 tweets négatifs envoyés par 500 utilisateurs = mauvais.[17]

1.6 Outils d'analyse de sentiments

1.6.1 SentiWordNet

SentiWordNet [18]a été généré à l'aide du dictionnaire WordNet[1]. Chaque synset (c'est-à-dire un groupe de termes synonymes sur une signification particulière) dans WordNet est associé à

1https://wordnet.princeton.edu/wordnet/

trois scores numériques indiquant le degré d'association du synset avec du texte positif, négatif et objectif. En générant le lexique, les synsets initiaux (positifs et négatifs) ont été étendus en exploitant les relations de synonymie et d'antonymie dans WordNet, grâce à quoi la synonymie se conserve tandis que l'antonymie inverse la polarité avec un synset donné. Puisqu'il n'y a pas de relation synonyme directe entre les synsets dans WordNet, les relations : see_also, similar_to, pertains_to, derived_from et attribute ont été utilisées pour représenter la relation de synonymie tandis que la relation antonyme directe a été utilisée pour l'antonymie. Des glosses (c.-à-d. Les définitions textuelles des ensembles élargis de synsets ainsi que celle d'un autre ensemble supposé être composé de synsets objectifs) ont été utilisés pour former huit classificateurs ternaires. Les classificateurs sont utilisés pour classer chaque synset et la proportion de classification pour chaque classe (positive, négative et objective) ont été considérés comme les scores du synset. Dans une version améliorée du lexique (SentiWordNet 3.0), les scores ont été optimisés par une marche aléatoire en utilisant l'approche PageRank. Avec les synsets sélectionnés manuellement, puis propage la polarité des sentiments (positive ou négative) à un synset cible en évaluant les synsets qui se connectent au synset cible à travers l'apparence de leurs termes dans le brillant du synset cible.[19]

SentiWordNet peut être vu comme ayant une structure arborescente comme indiqué sur la (figure 1-8). Le nœud racine de l'arbre est un terme dont les nœuds enfants sont les quatre balises PoS de base dans WordNet (c.-à-d. nom, verbe, adjectif et adverbe). Chaque POS peut avoir plusieurs sens du mot en tant que nœuds enfants. Les scores de sentiment illustrés par un point dans l'espace triangulaire du diagramme sont attachés aux sens des mots. La subjectivité augmente (tandis que l'objectivité diminue) de bas en haut, et la positivité augmente (tandis que la négativité diminue) de droite à gauche du triangle.

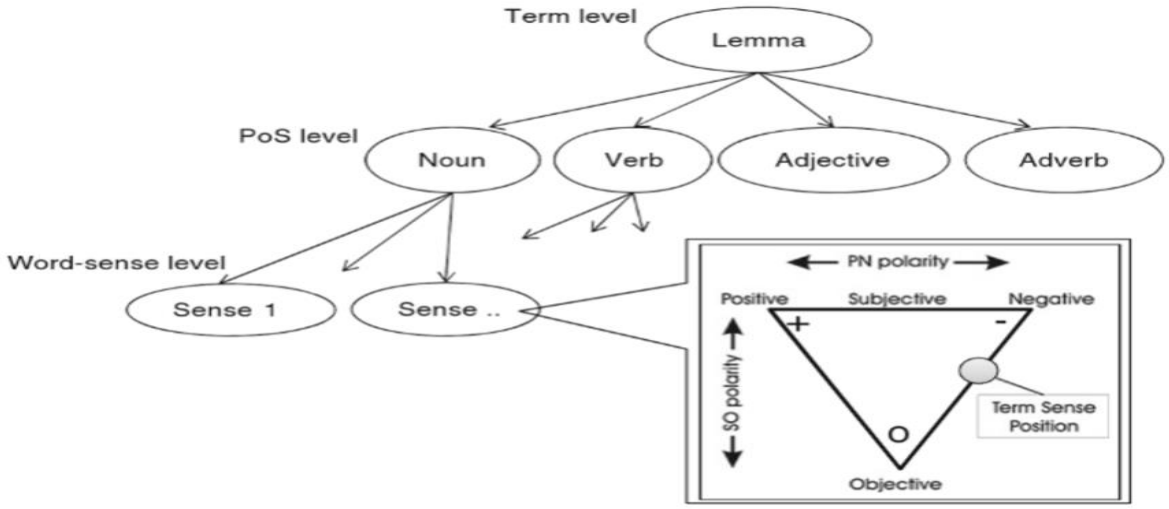

Figure 1- 8 : Diagramme montrant la structure de SentiWordNet.

1.6.2 Sentiment140

Sentiment140[2] (anciennement connu sous le nom "Twitter sentiment") est un outil en ligne gratuit qui a été créé par trois étudiants en computer science de Stanford, donc c'est un projet académique. Cet outil, contrairement à la plupart des autres sites d'analyse de sentiments, n'utilise pas de listes de mots positifs ou négatifs mais est fondé sur les algorithmes d'apprentissage automatique. Sentiment140 permet de découvrir des sentiments des tweets d'une marque, un produit ou un sujet sur Twitter.

1.6.3 Tweetfeel

Le service Tweetfeel[3] est un outil en ligne d'analyse du sentiment sur Twitter. Il propose une version gratuite et une version payante. Il s'appuie sur les capacités temps réels de Twitter qui donne des sentiments positifs et négatifs des tweets sur des choses. L'évaluation de TweetFeel se fait sur la base de présence de mots clés précis dans les tweets tels que Good, Bad, etc. (Uniquement en anglais pour l'instant).Ensuite un pourcentage est calculé selon le nombre de tweets positifs ou négatifs.

1.6.4 Twitrratr

Twitrratr[4] est un outil en ligne gratuit, qui a émergé à partir d'un projet Startup Weekend. Twitrratr fonctionne à partir d'une liste de mots positifs et d'une liste de mots négatifs. Cet outil classe une opinion sur le mot clé de la requête s'il est capable de le croiser avec un mot d'une des deux listes. Les mots positifs et négatifs qui servent à classer les tweets sont surlignés dans l'interface.

[2] http://www.sentiment140.com/
[3] http://www.tweetfeel.com
[4] http://twitrratr.com/

1.6.5 Tweet Sentiments Analyses

Tweet Sentiments Analyses est un outil en ligne gratuit et open source d'analyse du sentiment sur Twitter. Il peut donner des sentiments positifs, négatifs et neutres des tweets sur le mot clé lancé dans la requête. Il peut travailler sur 12 langues. Il donne les résultats sous forme graphique. [19]

1.7 Mesures de performance de la classification des sentiments

Mesurer les performances d'un algorithme de Machine Learning est une étape indispensable et extrêmement importante pour produire un algorithme de qualité et qui répond aux attentes métiers.

En règle générale, pour mesurer les performances de la classification des sentiments, Les normes prédéfinies suivantes sont utilisés [20]:

1.7.1 La matrice de confusion

C'est un tableau croisé entre les valeurs réelles et les prédictions (figure 1-9). Cette matrice permet d'identifier 4 catégories de résultats:

- ✓ **Vrais positifs (TP) :**les vrais positifs sont les cas où la classe réelle du point de données était 1 (Vrai) et la prédiction est également 1 (Vrai).
- ✓ **Vrais négatifs (TN):**les vrais négatifs sont les cas où la classe réelle du point de données était 0 (faux) et la prédiction est également 0 (faux).
- ✓ **Faux positifs (FP):**Les faux positifs sont les cas où la classe réelle du point de données était 0 (False) et la prédiction est 1 (Vrai). False est dû au fait que le modèle a prédit de manière incorrecte et positive parce que la classe prédite était positive. (1)
- ✓ **Faux négatifs (FN):** les faux négatifs sont les cas où la classe réelle du point de données était 1 (Vrai) et la prédite est 0 (Faux). False est dû au fait que le modèle a prédit de manière incorrecte et négative parce que la classe prédite était négative. (0)

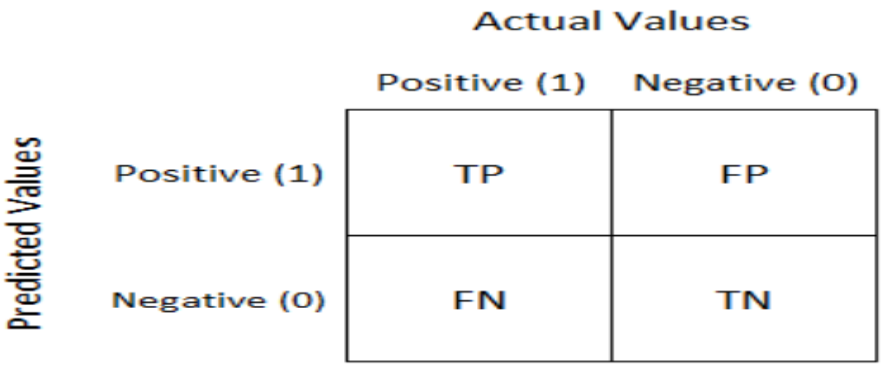

Figure 1- 9 : Matrice de confusion [20].

1.7.2 Accuracy, precision, recall and F-measure

À partir de cette table de confusion, on peut calculer différentes mesures pour évaluer le model [20][22]:

- ✓ **Accuracy**

Accuracy est utilisée pour trouver et évaluer une matrice afin de voir l'efficacité de l'algorithme de classificateur pour calculer la précision que la formule est utilisée est:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

- ✓ **Precision**

C'est le nombre de résultats positifs corrects divisé par le nombre de résultats positifs prédits par le classificateur.

$$Precision = \frac{TP}{TP + FP}$$

- ✓ **Recall**

Il s'agit du nombre de résultats positifs corrects divisé par le nombre de tous les échantillons pertinents (tous les échantillons qui auraient dû être identifiés comme positifs).

$$Recall = \frac{TP}{TP + FN}$$

- ✓ ***F*-measure**

Les mesures F sont une combinaison de précision et de rappel, elles produiront une seule matrice de précision et de rappel. Pour calculer les mesures F,l'équationsuivante est utilisé:

$$F - measures = 2 * \frac{Precision * Recall}{Precision + Recall}$$

1.7.3 La courbe ROC et AUC

- ✓ ***A.* La courbeROC**

Une **courbe ROC** (**receiver operating characteristic**) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs (figure1-10) [22]:

- ✓ **TPR (True Positive Rate) / Rappel /Sensitivité**

La sensibilité indique la probabilité d'un individu d'appartenir à la classe sachant qu'il devrait y appartenir.

$$TPR/Recall/Sensitivity = \frac{TP}{TP + FN}$$

✓ **Spécificité**

La spécificité indique, quant à elle, la probabilité qu'un individu n'appartienne pas à la classe à juste titre.

$$Specificity = \frac{TN}{TN + FP}$$

✓ **FPR (false positive rate)**

$$FPR = 1 - Specificity = \frac{FP}{TN + FP}$$

Voilà à quoi ressemblent des courbes de ROC :

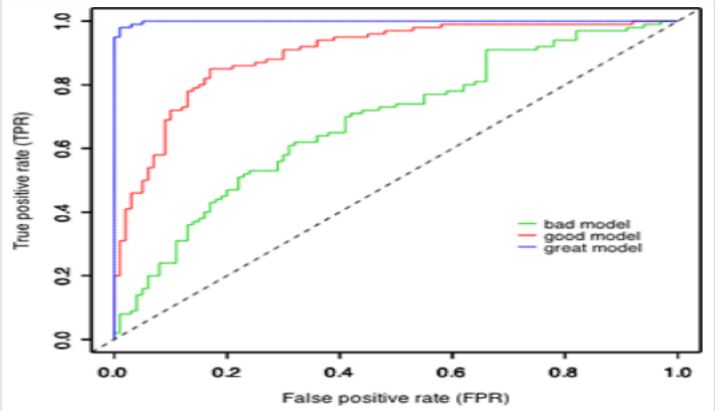

Figure 1- 10 : La courbe ROC [21].

B. L'AUC

AUC signifie "aire sous la courbe ROC". Cette valeur mesure l'intégralité de l'aire à deux dimensions situées sous l'ensemble de la courbe ROC (par calculs d'intégrales) de (0,0) à (1,1) (figure 1-11). L'AUC fournit une mesure agrégée des performances pour tous les seuils de classification possibles[25].

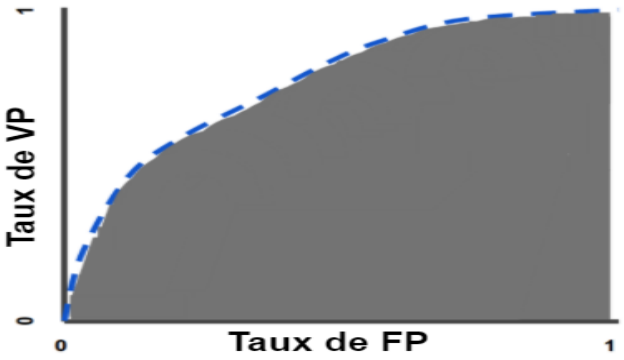

Figure 1- 11 : AUC (aire sous la courbe ROC [21].

1.8 Conclusion

Dans ce chapitre, nous avons présenté la revue de littérature de l'analyse des sentiments. Ceci comprend un survole sur les principaux fondements, méthodes et techniques d'analyse et de classification des sentiments. Le chapitre suivant sera dédié à la deuxième partie du volet théorique: l'apprentissage profond ou le deep learning.

Chapitre II

Deep Learning

CHAPITRE II : DEEP LEARNING

2.1 Introduction

L'apprentissage en profondeur, un type particulier de mécanisme d'apprentissage automatique, est caractérisé par l'effort de créer un modèle d'apprentissage à plusieurs niveaux, dans lequel les niveaux les plus profonds prennent en compte les résultats des niveaux précédents, les transformant et en faisant toujours plus d'abstraction. Cet aperçu des niveaux d'apprentissage est inspiré par la façon dont le cerveau traite l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond, par hypothèse, à l'une des différentes zones qui composent le cortex cérébral.

Actuellement, Le Deep Learning intervient dans le domaine du traitement du langage naturel (TLN) et son architectures offrent d'énormes avantages pour la classification de textes, car elles fonctionnent avec une précision extrêmement élevée avec une ingénierie et des calculs de niveau inférieur.

Les deux principales architectures d'apprentissage en profondeur pour la classification de texte sont les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN).

2.2 Histoire du deep learning

Année	Contributeur	Contribution
300 AC	Aristotle	Introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	Introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch and Pitts	Introduction du McCulloch–Pitts (MCP) modèle considéré comme l'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	Considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes
1958	Frank Rosenblatt	introduction du premier perceptron
1974	Paul Werbos	Introduction de la retro propagation
1980	TeuvoKohonen	Introduction des cartes auto organisatrices
1980	Kunihiko Fukushima	Introduction du Neocognitron, qui a inspiré les réseaux de neurones convolutif
1982	John Hopfield	Introduction des réseaux de Hopfield
1985	Hilton and Sejnowski	Introduction des machines de Boltzmann

1986	Paul Smolensky	Introduction de Harmonium,qui sera connu plus tard comme machines de Boltzmann restreintes
1986	Michael I. Jordan	Définition et introduction desréseaux de neurones récurrent
1990	Yann LeCun	Introduction de LeNet et montra la capacité des réseaux de neurones profond
1997	Schuster and Paliwal	Introduction des réseaux de neurones récurrent bidirectionnelles
1997	Hochreiter and Schmidhuber	Introduction de LSTM, qui ont résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	Introduction des Deep belief Network
2009	Salakhutdinov and Hinton	Introduction des Deep Boltzmann Machines
2012	Alex Krizhevsky	Introduction de AlexNet qui remporta le challenge ImageNet

Table 2- 1 : Les étapes majeurs du Deep Learning [22].

2.3 Apprentissage en profondeur

Les algorithmes classiques de ML fonctionnent bien pour une grande variété de problèmes. Cependant ils ont échoues à résoudre quelques problèmes majeurs de l'IA telle que le traitement de grandes quantités de données (Big Data). Mais, ce n'est qu'après que de plus grandes quantités de données ne soit disponibles grâce notamment au Big Data et aux objets connectés et que les machines de calcul soient devenues plus puissantes que la communauté scientifique a pu comprendre le potentiel réel du Deep Learning (table 2-2).

Cependant, Le développement du deep learning fut motivé en partie par l'échec des algorithmes traditionnels dans de telle tâche de l'IA.

Une des grandes différences entre le Deep Learning et les algorithmes de ML traditionnelles est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme de Deep Learning sont meilleurs (Figure 2-1). Contrairement à plusieurs algorithmes de ML classiques qui possèdent une borne supérieure à la quantité de données qu' ils peuvent recevoir des fois appelée "plateau de performance", les modèles de Deep Learning n'ont pas de telles limitations (théoriquement) et ils sont même allés jusqu'à dépasser la performance humaine dans des domaines comme text processing (Table 2-1).

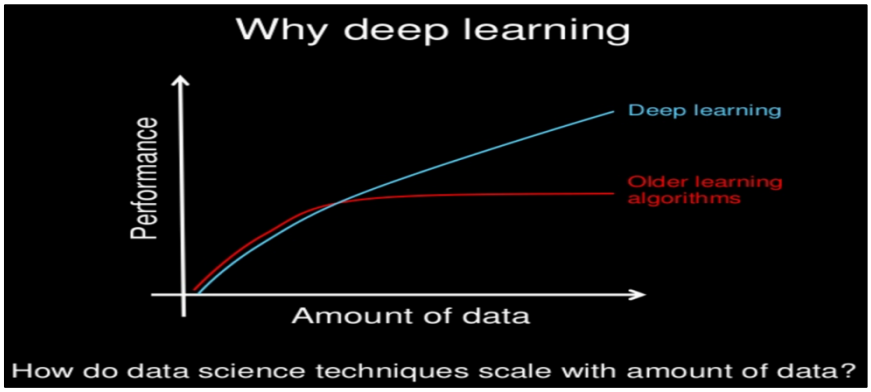

Figure 2 - 1 : La différence de performance entre le DL et la plupart des algorithmes de ML en fonction de la quantité de données [23].

Autre différence entre les algorithmes de ML traditionnelles et les algorithmes de Deep Learning est l'étape de l'extraction de caractéristiques. Dans les algorithmes de ML traditionnelles l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme[23].

Machine Learning (ML)	***Deep Learning (DL)***
✓ Fonctionne sur une petite quantité de jeu de données pour plus de précision. ✓ Il repose sur une machine bas de gamme. ✓ Divise les tâches en sous-tâches, les résout individuellement et enfin combine les résultats. ✓ Prend moins de temps pour s'entraîner. ✓ Le temps de test peut augmenter.	✓ Fonctionne sur une grande quantité de données. ✓ Très dépendant de la machine haut de gamme. ✓ Résout le problème de bout en bout. ✓ Prend plus de temps pour s'entraîner. ✓ Moins de temps pour tester les données.

Table 2- 2 : Différences entre ML & DL [23].

2.4 Les applications de l'apprentissage en profondeur

2.4.1 La reconnaissance faciale

Les yeux, le nez, la bouche, tout autant de caractéristiques qu'un algorithme de Deep Learning va apprendre à détecter sur une photo. Il va s'agir en premier lieu de donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image [24].

2.4.2 La détection d'objets

Sur une image complexe où il y a plusieurs éléments, les algorithmes de détection d'objets vont être maintenant capables d'identifier et de localiser au pixel près un élément ou une personne. 800 millions d'images sont uploadées chaque jour sur Facebook : son algorithme Deep Learning est effectivement capable d'identifier telle ou telle personne sur une photo dès lors qu'elle est uploadée [25].

2.4.3 Le traitement du langage natural

Le Natural Language Processing est une autre application du Deep Learning. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments. L'algorithme va par exemple comprendre ce qui est dit dans un avis Google, ou va communiquer avec des personnes via des chat bots. La lecture et l'analyse automatique de textes est aussi un des champs d'application du Deep Learning avec le Topic Modeling: tel texte aborde tel sujet [25].

2.4.3.1 La promesse du Deep Learning pour le NLP

- ✓ Les méthodes d'apprentissage en profondeur sont populaires pour le langage naturel, principalement parce qu'elles tiennent leurs promesses.
- ✓ Certaines des premières grandes démonstrations de la puissance de l'apprentissage profond concernaient le traitement du langage naturel, en particulier la reconnaissance vocale. Plus récemment en traduction automatique.
- ✓ Les 3 promesses clés de l'apprentissage profond pour le traitement du langage naturel sont les suivantes :
  - ➢ La promesse de l'apprentissage des fonctionnalités. Autrement dit, les méthodes d'apprentissage en profondeur peuvent apprendre les fonctionnalités du langage naturel requises par le modèle, plutôt que d'exiger que les fonctionnalités soient spécifiées et extraites par un expert.
  - ➢ La promesse d'une amélioration continue. Autrement dit, les performances de l'apprentissage profond dans le traitement du langage naturel sont basées sur des résultats réels et que les améliorations semblent se poursuivre et peut-être s'accélérer.
  - ➢ La promesse des modèles de bout en bout. Autrement dit, les grands modèles d'apprentissage profond de bout en bout peuvent être adaptés aux problèmes de langage naturel en offrant une approche plus générale et plus performante.
- ✓ Le traitement du langage naturel n'est pas « résolu », mais un apprentissage en profondeur est nécessaire pour vous amener à l'état de l'art sur de nombreux problèmes difficiles dans le domaine [25].

2.4.3.2 Analyse de texte et NLP :

La communication textuelle est l'une des formes les plus populaires de conversation quotidienne. on discute, envoie des messages, tweete, partage le statut, écris des blogs et partage des opinions et des commentaires dans notre routine quotidienne. Toutes ces activités génèrent du texte en quantité significative, qui n'est pas structurée par nature. Dans ce domaine du marché en ligne et des médias sociaux, il est essentiel d'analyser de grandes quantités de données, de comprendre l'opinion des gens. La NLP permet à l'ordinateur d'interagir avec les humains de manière naturelle. Il aide l'ordinateur à comprendre le langage humain et à en tirer un sens. La NLP est applicable dans plusieurs problématiques de la reconnaissance vocale, la traduction de la langue, la classification des documents à l'extraction d'informations. L'analyse de la critique des films est l'un des exemples classiques pour démontrer un modèle simple de sac de mots NLP, sur des critiques de films [26].

2.4.3.3 Les algorithmes de l'apprentissage profond avec le NLP :

Cette table ci-dessous représente des quelques algorithmes et leurs utilisations avec le NLP.

Algorithme	*Utilisation du NLP*
Réseau neuronalNN (feed)	• Balisage d'une partie du discours • Tokenisation • Reconnaissance des entités nommées • Extraction d'intention
Réseaux de neurones récurrents - RNN	• Traduction automatique • Système de réponse aux questions • Sous-titrage d'image
Réseaux de neurones récursifs	• Analyse des phrases • Analyse des sentiments • Détection paraphrase • Classification des relations • Détection d'objets
Réseau neuronal convolutif -CNN	• Classification des phrases / textes • Extraction et classification des relations • Détection de spam • Catégorisation des requêtes de recherche • Extraction de relations sémantiques

Table 2- 3 : Algorithmes d'apprentissage et leurs utilisation avec le NLP [27].

2.5 Le réseau neuronal

La pratique, de tous les algorithmes de DL sont des réseaux neuronaux. Les réseaux neuronaux, aussi appelés ANN, sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique. C'est similaire à la façon dont le cerveau manipule l'information au niveau du fonctionnement. Tous les réseaux neuronaux sont constitués de neurones inter connectés qui sont organisés en couches [28](figure 2 - 2).

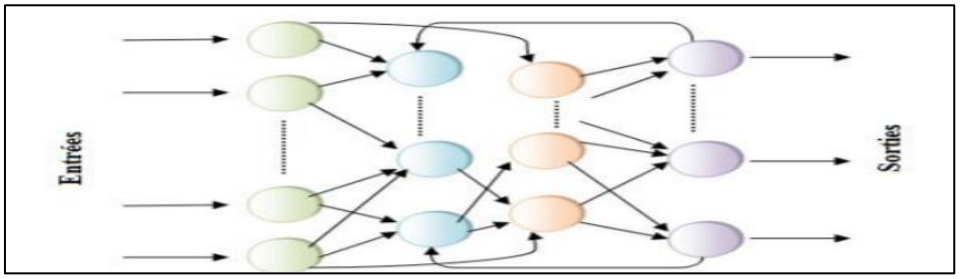

Figure 2 - 2 : Réseaux de neurones[28].

2.5.1 Fonctionnement du neurone artificiel

Le neurone artificiel va recevoir plusieurs entrées d'informations, plusieurs valeurs, qui vont être attaché à un poids qui peut être ajusté. Ces entrées correspondent aux dendrites, et les poids qui leurs sont associés, correspondent aux actions excitatrices ou inhibitrices des synapses, ils vont pouvoir amplifier ou minimiser un signal d'entrée. Le neurone dans sa forme basique, va effectuer une somme de l'ensemble de ces variables en fonction de leurs poids, correspondant au soma. Cette valeur passe ensuite par une fonction d'activation, qui en sera l'unique sortie. Celle-ci correspond au point de départ de l'axone qui est le cône d'émergence.

Le principe de ces réseaux va donc être d'assembler de grande quantité de neurone entre eux pour former des couches.[29]

Il existe une grande variété de type de neurone, qui engendre selon leur agencement, différents types d'architecture.

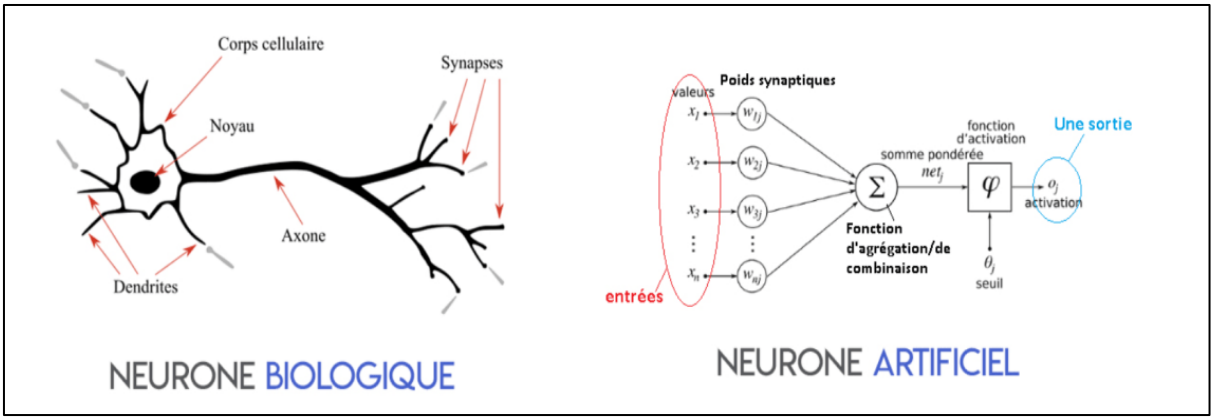

Figure 2 - 3 : Représentation mathématique/informatique d'un neurone biologique [29].

2.5.2 Avantages et inconvénients des réseaux neurones

Nous avons vu un aperçu des caractéristiques de base d'un réseau de neurones et compris comment cela fonctionne essentiellement. Nous allons donc souligner quels sont leurs principaux avantages et inconvénients dans le domaine de l'innovation [30]:

2.5.2.1 Avantages de l'utilisation d'un réseau neurone

Ils conviennent aux problèmes qui ne nécessitent pas de réponses précises, mais des réponses approximatives avec un degré d'erreur ou de variation:

- ✓ Leur capacité à généraliser : ils peuvent produire de bonnes réponses même avec une contribution qui n'a pas été prise en compte lors de leur création et de leur entrainement.
- ✓ Ils sont faciles à implémenter, vous pouvez simplement définir un neurone, puis le copier et créer des connexions entre les neurones.
- ✓ Il fournit des opérations rapides car il fonctionne en parallèle, chaque neurone utilise seulement son entrée.
- ✓ La stabilité de la sortie par rapport aux valeurs d'entrée : les valeurs d'entrée peuvent être incomplètes, bruyantes, mal connues ou accepter un degré d'erreur ou de changement
- ✓ Ils peuvent déterminer le résultat en tenant compte de toutes les entrées en même temps [30].

2.5.2.2 Inconvénients de l'utilisation des réseaux neuronaux:

- ✓ Neural Network (RN) fonctionne comme une boîte noire : généralement, vous ne serez pas en mesure de comprendre pourquoi il a produit ce résultat spécifique.
- ✓ Les connaissances mémorisées ne peuvent être décrites et localisées dans le réseau
- ✓ Ils impliquent souvent des techniques de formation sophistiquées qui prennent beaucoup de temps pour les calculs.
- ✓ Il n'y a pas toujours un réseau capable de résoudre un problème spécifique, car il n'y a pas toujours un algorithme d'apprentissage qui converge en donnant une sortie réseau à faible erreur.
- ✓ Les valeurs de sortie ne sont pas précises, mais ont une marge qui peut varier.
- ✓ Nous avons besoin d'une très grande série d'exemples pour avoir un bon processus d'apprentissage et une faible erreur de rendement. [30]

2.6 Architectures de réseaux de neurones profonds

Il existe un grand nombre de variables d'architectures profondes. La plupart d'entre eux sont dérivés de certaines architectures originales. Nous allons choisir les réseaux de neurones récurrents (RNNs)[34] :

2.6.1 Les réseaux neuronaux récurrents (RNN) :

Sont un type de réseau neuronal dans lequel la sortie de l'étape précédente est fournie comme entrée à l'étape actuelle. Dans les réseaux de neurones traditionnels, toutes les entrées et sorties sont indépendantes les unes des autres, mais dans des cas comme lorsqu'il est nécessaire de prédire le mot suivant d'une phrase, les mots précédents sont nécessaires et il est donc nécessaire de se souvenir des mots précédents. C'est ainsi que RNN a vu le jour, ce qui a résolu ce problème à l'aide d'une couche cachée. La caractéristique principale et la plus importante de RNN est l'état caché, qui mémorise certaines informations sur une séquence.

Les RNN ont une « mémoire » qui mémorise toutes les informations sur ce qui a été calculé. Il utilise les mêmes paramètres pour chaque entrée car il effectue la même tâche sur toutes les entrées ou couches masquées pour produire la sortie**.** Cela réduit la complexité des paramètres, contrairement aux autres réseaux de neurones [31].

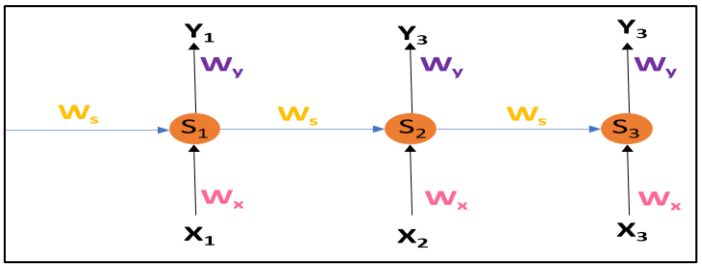

Figure 2 - 4 : RNN Architecture [31].

S2, S3 sont les états cachés ou les unités de mémoire au temps t1, t2, t3 respectivement, et Ws est la matrice de poids qui lui est associée. X1, X2, X3 sont les entrées aux temps t1, t2, t3 respectivement, et Wx est la matrice de poids qui lui est associée. Y1, Y2, Y3 sont les sorties aux temps t1, t2, t3 respectivement, et Wy est la matrice de poids qui lui est associée. Pour tout instant, t, nous avons les deux équations suivantes:

\ begin {équation *} S_ {t} = g_ {1} (W_ {x} x_ {t} + W_ {s} S_ {t-1}) \ end {équation *}

\ begin {équation *} Y_ { t} = g_ {2} (W_ {Y} S_ {t}) \end {équation *}

2.6.2 LSTM (long short term memory)

Plusieurs variantes aux RNN standards ont vu le jour pour remédier aux problèmes évoqués précédemment. Nous allons ici décrire les LSTM, pour Long Short-Term Memory. Ce type de RNN est très utilisé en traitement du langage naturel.

L'architecture de réseaux de neurones est de diviser le signal entre ce qui est important à court terme à travers le **hidden state** (analogue à la sortie d'une cellule de RNN simple), et ce qui l'est à long terme, à travers le **cell state**, qui sera explicité plus bas. Ainsi, le fonctionnement global d'un LSTM peut se résumer en 3 étapes :

1. Détecter les informations pertinentes venant du passé, piochées dans le cell state à travers la **forget gate** ;
2. Choisir, à partir de l'entrée courante, celles qui seront pertinentes *à long terme*, via l'**input gate**. Celles-ci seront ajoutées au cell state qui fait office de mémoire longue ;
3. Piocher dans le nouveau cell state les informations importantes *à court terme* pour générer le hidden state suivant à travers l'**output gate**.

Pour un autre exemple illustrant l'intuition derrière les LSTM, voir cette discussion sur Quora. Regardons cela de plus près. En respectant la même convention que pour le schéma simplifié de la cellule RNN, on peut représenter une cellule LSTM de la façon suivante :

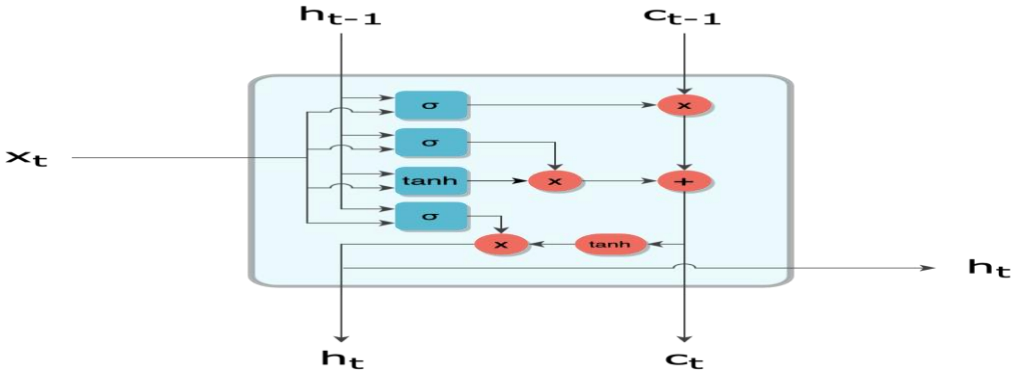

Figure 2 - 5 : Représentation simplifiée d'une cellule LSTM [32].

Comme le RNN, le LSTM définit donc une relation de récurrence, mais utilise une variable supplémentaire qui est le **cell state c** :

$h_t , c_t = f(x_t, h_{t-1}, c_{t-1})$

L'information transite d'une cellule à la suivante par deux canaux, **h** et **c**. À l'instant **t**, ces deux canaux se mettent à jour par l'interaction entre leurs valeurs précédentes $\mathbf{h_{t-1}}$ et $\mathbf{c_{t-1}}$ ainsi que l'élément courant de la séquence $\mathbf{x_t}$.[32]

2.6.3 Architecture LSTM:

La différence fondamentale entre les architectures des RNN et des LSTM : est que la couche cachée de LSTM est une unité fermée ou une cellule fermée. Il se compose de quatre couches qui interagissent les unes avec les autres de manière à produire la sortie de cette cellule avec l'état de la cellule. Ces deux choses sont ensuite transmises à la couche cachée suivante. Contrairement aux RNN qui ont la seule couche de réseau neuronal unique de tanh, les LSTM se composent de trois portes sigmoïdes logistiques et d'une couche de tanh. Des portes ont été introduites afin de limiter les informations qui transitent par la cellule. Ils déterminent quelle partie de l'information sera requise par la cellule suivante et quelle partie doit être supprimée. La sortie est généralement comprise entre 0 et 1, où «0» signifie «*tout rejeter*» et «1» signifie «*tout inclure*» [33].

2.7 La Promesse de l'apprentissage en profondeur

Les méthodes d'apprentissage en profondeur sont très populaires, principalement parce qu'elles tiennent leurs promesses [26].

Cela ne veut pas dire qu'il n'y a pas de battage médiatique au sujet de la technologie, mais ce battage médiatique est basé sur des résultats très réels démontrés sur une foule de problèmes d'IA très difficiles de la vision par ordinateur et du traitement du langage naturel [26].

Certaines des premières grandes démonstrations de la puissance de l'apprentissage en profondeur concernaient le traitement du langage naturel, en particulier la reconnaissance vocale, récemment en traduction automatique.

Dans le domaine du traitement du langage naturel, cinq promesses spécifiques de méthodes d'apprentissage en profondeur ont été examinées. Les promesses faites récemment par les chercheurs et les praticiens du domaine, des gens qui peuvent être plus modérés que la moyenne rapporte ce que peuvent être les promesses [25].

En bref, ils sont :

- ✓ La promesse de modèles de remplacement instantanés. Autrement dit, les méthodes d'apprentissage en profondeur peuvent être intégrées aux systèmes de langage naturel existants en tant que modèles de remplacement pouvant atteindre des performances proportionnées ou meilleures.
- ✓ La promesse de nouveaux modèles NLP. Autrement dit, les méthodes d'apprentissage en profondeur offrent l'opportunité de nouvelles approches de modélisation pour résoudre des problèmes de langage naturel comme la prédiction séquence à séquence.
- ✓ La promesse de l'apprentissage des fonctionnalités. Autrement dit, les méthodes d'apprentissage en profondeur peuvent apprendre les fonctionnalités du langage naturel requises par le modèle, plutôt que d'exiger que les fonctionnalités soient spécifiées et extraites par un expert.
- ✓ La promesse d'une amélioration continue. Autrement dit, les performances de l'apprentissage profond dans le traitement du langage naturel sont basées sur des résultats réels et que les améliorations semblent se poursuivre et peut-être s'accélérer.
- ✓ La promesse des modèles de bout en bout. Autrement dit, les grands modèles d'apprentissage profond de bout en bout peuvent être adaptés aux problèmes de langage naturel en offrant une approche plus générale et plus performante.

2.8 Conclusion

L'apprentissage profond est le domaine le plus émergeant de l'apprentissage automatique. Il a apporté une contribution importante dans divers domaines de recherche, ce qui est très prometteur et constitue un succès. Dans ce chapitre, nous avons exposé ce sous-ensemble des méthodologies et techniques de ML qui utilisent le réseau neuronal artificiel et son utilisation avec le traitement du langage naturel.

Le prochain chapitre, traite les détails de la conception, ainsi que la méthode et les outils utilisés pour la réalisation de notre application.

Chapitre III

Expérimentations & Résultats

CHAPITRE III : Conception & Implémentation

3.1 Introduction :

La collecte des données d'un big-social data, comme Twitter, n'est pas une tâche très simple. Nous devons en tenir compte des obstacles rencontrés. Cependant, un ensemble de données préalablement collecté (dataset) est utilisé pour entraîner et tester deux modèles d'apprentissage: algorithmes traditionnels et le Deep Learning.

Notre solution d'analyse du sentiment se base sur une ressource lexicale externe (un dictionnaire donné ou SentiWordNet) au moment du prétraitement. Cette ressource linguistique aide à donner de l'information sur la polarité des mots afin de calculer le sentiment de chaque tweet. Les tweets reflètent les statuts postés sur le réseau social Twitter.

Ce chapitre étudie la manière dont les données seront traitées, analysées et principalement comment les classifier ? Avant de passer à cette chose, parlons des outils utilisé et de l'architecture proposée.

3.2 Les outils d'expérimentation utilisés

A. ***Anaconda*** est une distribution gratuite et open source conditionnelle des langages de programmation Python et R pour le calcul scientifique, qui vise à simplifier la gestion et le déploiement des packages. La distribution comprend des packages de science des données adaptésà Windows, Linux et macOS.[34]

B. ***Google Colab*** **(**Colaboratory) est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. Cool, n'est ce pas ? Avant de présenter ce magnifique service, nous rappellerons ce qu'est un Jupyter Notebook.[35]

C. ***Jupyter Notebook*** est une application Web Open Source permettant de créer et de partager des documents contenant du code (exécutable directement dans le document), des équations, des images et du texte. Avec cette application il est possible de faire du traitement de données, de la modélisation statistique, de la visualisation de données, du Machine Learning, etc. Elle est disponible par défaut dans la distribution Anaconda (suivre ce lien pour savoir comment l'installer) [35].

D.

NLTK est une plate-forme de premier plan pour la création de programmes Python pour travailler avec des données en langage humain. Il fournit des interfaces faciles à utiliser vers plus de 50 corpus et lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, la recherche de racines, le marquage, l'analyse et le raisonnement sémantique, des wrappers pour les bibliothèques de PNL de niveau industriel, et un forum de discussion actif.[36]

E.

Scikit-learnest un module python intégrant les algorithmes classiques de l'apprentissage automatique. Il vise à fournir des solutions simples et efficaces pour les problèmes d'apprentissage, accessible à tous et réutilisables dans divers contextes.Il existe quelques ensembles de données classiques intégrés avec Scikit-learn, par exemple "le dataset de fleurs des iris de Fisher" et "digits" qui est un dataset d'images de chiffres manuscrits.[36]

F.

Pandasest une bibliothèque créée pour aider les développeurs à travailler intuitivement avec des données "étiquetées" et "relationnelles". Elle est basée sur deux structures de données principales : "Série" (unidimensionnelle, comme une liste Python) et "Dataframe" (bidimensionnelle, comme un tableau à plusieurs colonnes). Pandas permet de convertir des structures de données en objets DataFrame, de gérer les données manquantes et d'ajouter/supprimer des colonnes de DataFrame, d'imputer les fichiers manquants et de tracer les données avec un histogramme ou une boîte à moustache. C'est un outil indispensable pour la manipulation et la visualisation des données.[37]

G.

Kerasest une excellente bibliothèque pour la construction de réseaux de neurones et la modélisation. Elle est très simple à utiliser et offre aux développeurs un bon degré d'extensibilité. La bibliothèque tire parti d'autres paquets (Theano ou TensorFlow) comme terminaux. De plus, Microsoft a intégré CNTK (Microsoft Cognitive Toolkit) pour servir d'autre backend. C'est un excellent choix si vous voulez expérimenter rapidement en utilisant des systèmes compacts – l'approche minimaliste de la conception est vraiment top !.[37]

3.3 Architecture générale du système

Nous avons mené une série d'expérimentations sur un data-set préalablement collecté de twitter et stocké dans un fichier .xlsx. Pour atteindre notre objectif et obtenir la meilleure performance possible, nous proposons l'architecture suivante :

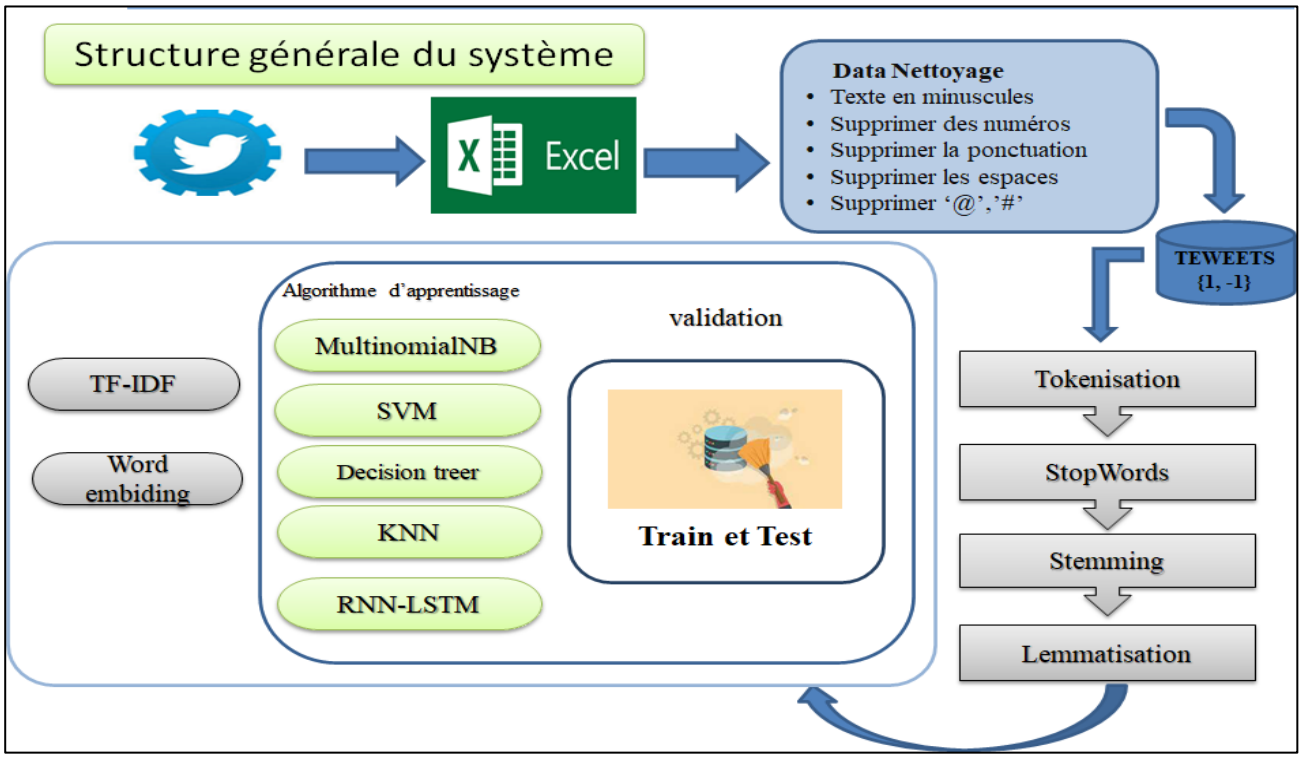

Figure 3- 1 : Architecture générale du système.

Les étapes à suivre:

1. Une étape de prétraitement passe par le data set (.xlsx) via une ressource lexicale (Sentiwordnet) ou un dictionnaire donné (.tsv)). Le dictionnaire (.tsv) contient des mots avec leur polarité correspondante, qui est essentielle pour calculer le sentiment de chaque texte du tweet. Chaque mot extrait du texte est comparé aux mots de la ressource lexicale et un score est donné.
2. Une fois le prétraitement terminé. La sortie est un ensemble de données traitées (.xlsx) contenant le texte et le sentiment correspondant(classification binaire/multiple)avec des libellés positifs, négatifs ou neutres séparés en colonnes.
3. Pour le sous-processus de training et prédiction, notre code passe le fichier .xlsx composé d'un bloc de training et d'un autre pour le test. Lorsque tout est terminé, nous devrions avoir tous les résultats à évaluer des différents classificateurs.

3.4 Description du dataset

Notre dataset original(figure3-2) se compose de 9472 tweets en anglais :

```
 0   tweets  9472 non-null   object
dtypes: object(1)
memory usage: 74.1+ KB
```

	tweets
0	Go \ until jurong point, crazy.. Available onl...
1	Ok lar... Joking wif u oni...
2	Free entry in 2 a wkly comp to win FA Cup fina...
3	U dun say so early hor... U c already then say...
4	Nah I don't think he goes to usf, he lives aro...
5	FreeMsg Hey there darling it's been 3 week's n...
6	Even my brother is not like to speak with me. ...
7	As per your request 'Melle Melle (Oru Minnamin...
8	WINNER!! As a valued network customer you have...
9	Had your mobile 11 months or more? U R entitle...

Figure 3- 2 : Extrait du dataset

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

- Tweet texte : il contient le texte du tweet publié par l'utilisateur.

3.5 Les étapes du prétraitement

3.5.1 Prétraitement

Étant donné que les données recueillies à partir des tweets sont biaisées et bruyante, le nettoyage des tweets est notre première tâche. Ce prétraitement consiste à structurer et à faciliter l'utilisation des données textuelles originales.

Le prétraitement des données est le processus de nettoyage et de préparation des tweets pour la classification. Cette première étape est la reconnaissance des termes, des ponctuations, des fins de paragraphes et des phrases. Nous devons aussi unifier l'écriture des lettres en minuscule afin de faciliter la mise en correspondance.

Les tweets 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 les tweets 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 les tweets est traité comme une dimension (tableau).

Dans ce qui suit nous allons présenter la procédure détaillée du prétraitement suivie, dont le but est de nettoyer les tweets et les rendre le plus proche possible d'un langage formel.

Nous avons procédé à un prétraitement qui suit les étapes suivantes

a) Texte en minuscules : unifier l'écriture des lettres en minuscule afin de faciliter la mise en correspondance.

b) Supprimer les chiffres @user #, les liens web : il faut les supprimer car ils ont aucun impact sur le classement

c) Supprimer des numéros : Nous pouvons soit supprimer des nombres, soit convertir les nombres en leurs représentations textuelles. Nous pouvons utiliser des expressions régulières pour supprimer les nombres.

d) Supprimer la ponctuation : Nous supprimons les ponctuations afin de ne pas avoir différentes formes du même mot. Si nous ne supprimons pas la ponctuation, elles seront traitées séparément[!"#$%&'()*+,./:;<=>?@[\]^_`{|}~].

e) Supprimer les espaces : Nous pouvons utiliser la fonction join et split pour supprimer tous les espaces blancs d'une chaîne.

Le Tableau suivant donne quelques exemples de tweets avant et après le prétraitement

Tweets avant pré-traitement	*Tweets après pré-traitement*
U dun say so early hor... U c already then say...	u dun say so early hor u c already then say
Great! I hope you like your man well endowed. I am <#> inches...	great i hope you like your man well endowed i am lt gt inches
A gram usually runs like <#> , a half eighth is smarter though and gets you almost a whole second gram for <#>	a gram usually runs like lt gt a half eighth is smarter though and gets you almost a whole second gram for lt gt

Table 3- 1 : Exemple de tweets avant et après le prétraitement.

3.5.2 Annotation via une ressource lexicale

Partant d'un dataset purement textuel, nous avons décidé de l'annoter c-à-d déterminer la polarité de ses tweets. Pour cela, nous exploitons deux ressources lexicographiques (Wordnet & un dictionnaire donné) :

3.5.2.1 Annotation via Wordnet

A. Extraction des scores de polarité

Dans SentiWordNet, chaque mot contient un score de positivité et un score de négativité. Tout d'abord, Le texte d'entrée est transmis au composant du prétraitement. À partir duquel les

jetons (les mots) sont déterminés. Pour chaque mot qui n'est ni un mot vide ni une marque de ponctuation, nous cherchons le mot dans WordNet en utilisant son étiquetage morpho-syntaxique (PoStagging) et son lemme pour trouver les mots les plus proches du sens de mot recherché. Le mot sera rejeté dans les trois cas suivants :

- ✓ S'il ne s'agit pas d'un nom, d'un adjectif, d'un adverbe ou d'un verbe.
- ✓ Si le mot de texte n'est pas réduit à l'entité "Lemme".
- ✓ Si le mot n'obtient pas la liste de tous les synsets.

Si tous les mots du texte sont rejetés, le texte suivant sera traité. La fonction sentie_synsets retourne automatiquement un positive_score, et un negative_score pour le premier synset le plus courant.

Deux scores différents pour la polarité du texte soit calculés. Un score contient l'addition du score des mots trouvés positifs (noté *score_positive*), l'autre contient l'addition du score des motstrouvés négatifs (noté $score_negative$). Et puisque les scores de chaque mot sont des valeurs absolues, nous calculons le score du mot avec formule suivante:

$$Word_Sentiment = score_positive - score_negative.$$

Le score de sentiment pour un post p est déterminé en utilisant l'équation suivante :

$$score(p)_{dim} = \frac{\sum_{w_j \in p} score(w_j)_{dim}}{|n|}$$

Où $score(p)_{dim}$ est le score du post p dans la dimension sentiment de dim (dim est positif ou négatif) tandis que $score(w_j)_{dim}$ est le score de sentiment du terme w_j en p extrait de SentiWordNet, et $|n|$ est le nombre de termes porteurs de sentiments en$p$.

L'utilisation de senti_wordnet se résume par l'algorithme1 suivant :

Algorithme 1 :Calculer le score d'un texte de tweet

```
Get_sentiment_NLTK_sentiword
    if tag.startswith('J'):
        return wn.ADJ
    elif tag.startswith('N'):
        return wn.NOUN
    elif tag.startswith('R'):
        return wn.ADV
    elif tag.startswith('V'):
        return wn.VERB
    return None

def wordenet_sementique(text):
    print("==========================")
    print("Processing please wait...")
    print("==========================\n\n")
    start_time = time.time()
```

```
    counter = 0

    for data in text:
        words = nltk.sent_tokenize(data)
        tokens_count = 0
        sentiment = 0

        for words in words:
            tagged_sentence = pos_tag(word_tokenize(words))
            for word, tag in tagged_sentence:
                wn_tag = penn_to_wn(tag)
                if wn_tag not in (wn.NOUN, wn.ADJ, wn.ADV, wn.VERB):
                    continue
                lemma = lemmatizer.lemmatize(word, pos=wn_tag)
                if not lemma:
                    continue

                synsets = wn.synsets(lemma, pos=wn_tag)
                if not synsets:
                    continue
                synset = synsets[0]
                swn_synset = swn.senti_synset(synset.name())
                sentiment += swn_synset.pos_score()- swn_synset.neg_score()
                tokens_count += 1
        if tokens_count == 0:
            counter += 1
            continue
        sentiment = sentiment/tokens_count
```

B.Catégorisation de texte

Quand l'algorithme finit de parcourir tous les mots du tweet en entrée, le sentiment du message sera :

- Si la polarité est < 0 : le message exprime une opinion négative.
- Si la polarité est >= 0 : le message exprime une opinion positive.

Le pseudo code montré dans l'algorithme 2 est appliqué pour classifier les tweets :

Algorithme 2: Mesure de polarité
if sentiment_score>= 0: *Return "Possitive"* *else:* *Return "Negative"*

Le tableau suivant montre un exemple de tweets étiquetés :

Tweet	Type	Classe
sorr i ll call later in meeting	Positive	1
yeah do don û t stand to close tho you û ll catch something	Négative	-1

Table 3- 2 : Exemples de tweets étiquetés.

En fin de traitement, nous obtenons un datatset composé de **9423** tweets annotés dont**7091**tweets positifs et **2332** négatifs.

3.5.2.2 Annotation via un dictionnaire donné

Le dictionnaire choisi est constitué de mots subjectifs, considérés comme des références universelles. Dans ce dictionnaire, une polarité est associée 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'opinions en fonction de la présence de mots issus de ce dictionnaire dans le texte.

Ce dictionnaire est constitué d'un lexique de plus de 8221 mots.

A. Extraction des polarités et calcul du score :

A partir de l'étape de prétraitement, les jetons (les mots) du texte d'entrée sont obtenus, et Chaque mot est recherché dans le dictionnaire, si le mot n'est pas trouvé, il est rejeté et le mot suivant est recherché. Si le mot se trouve dans le dictionnaire, la polarité de ce mot est tirée du dictionnaire (positive, négative ou neutre).Le score de sentiment du terme est calculé comme suit :

- ✓ Si la polarité du mot est positive, retourne un 1.
- ✓ Si la polarité du mot est négative, retourne un -1.
- ✓ Si la polarité du mot est neutre, retourne un 0.

Le score du sentiment pour un texte (t) est déterminé en utilisant l'équation suivante :

$$score(t)_{dim} = \frac{\sum_{w_i \in p} score(w_i)_{dim}}{|m|}$$

Où $score(t)_{dim}$ est le score du texte$t$ dans la dimension sentiment de dim (dim est positif, négatif ou neutre) tandis que $score(w_i)_{dim}$ est le score de sentiment du terme w_i en t(mentionné au-dessus), et $|m|$est le nombre de termes en $t$trouvés dans le dictionnaire.

B. Catégorisation de texte

Une fois que nous avons le score de sentiment textuel, la polarité du texte sera:

- ✓ Si le score est >= 0.2 : le tweet exprime une opinion positive (non harcèlement).
- ✓ Si le score est < 0.2 et le score est > -0.5: le tweet exprime une opinion neutre (état neutre).
- ✓ Si le score est<= -0.5 : le tweet exprime une opinion négative (harcèlement).

L'utilisation de sent_dictionner se résume par l'algorithme2 suinant :

```
def senti_dictionare(text):
  start_time = time.time()
  counter = 0
  print ("============================")
  print("Processing please wait...")
  print("============================\n\n")
  for j in dataset:
    tweet_token = j
    #print(tweet_token)
    token = word_tokenize(tweet_token)
    sumnum = 0
    sum_word = 0
    for t in token:
        for d in y:
            if t == d[0]:
                sentiment = d[1]
                if sentiment == "positive":
                    sumnum += 1
                    sum_word += 1
                elif sentiment == "negative":
                    sumnum += -1
                    sum_word += 1
                else:
                    sumnum += 0
                    sum_word += 1
                break
    if sum_word != 0.0:
        sum_more = sumnum / sum_word
        if sum_more >= 0.2:
            sum_more = 1
        elif (sum_more < 0.2) and (sum_more > -0.5):
            sum_more = 0
        elif sum_more <= -0.5:
             sum_more = -1
        else:
            print("****")
```

Le tableau suivant montre un exemple de tweets étiquetés :

Tweet	Polarité
sitting in mu waiting for everyone to get out of my suite so I can take a shower	negative
improving the flexibility and co ordination of the health service to better match services and needs	positive
sitting in mu waiting for everyone to get out of my suite so i can take a shower ignite	neutral

Table 3- 3 : Exemples de tweets étiquetés.

En fin de traitement, nous obtenons un datatset composé de 9469 tweets annotés dont 5711 tweets positfs, 1220 negatifs et 2538 neutres.

3.6 Base d'apprentissage :

On considère D = f (Xi ; Yi), N représentent un ensemble de taille N, où Xi (1, ……, n) est un vecteur de caractéristiques (figure 3-3):

- Xi [words] : les tweets prétraités.
- Yi [sentiment] : les tweets annotés (*sentiment 1*: tweet positif, *sentiment -1*: tweet négatif et *sentiment 0*: tweet neutre) .

Out[2]:

	words	sentiment
0	go until jurong point crazy available only in ...	1
1	ok lar joking wif u oni	1
2	free entry in 2 a wkly comp to win fa cup fina...	1
3	u dun say so early hor u c already then say	1
4	nah i don t think he goes to usf he lives arou...	1
5	freemsg hey there darling it s been 3 week s n...	1
6	even my brother is not like to speak with me t...	-1
7	as per your request melle melle oru minnaminun...	1
8	winner as a valued network customer you have b...	1
9	had your mobile 11 months or more u r entitled...	1

Figure 3- 3 : Ecran des tweets et leurs polarités (Approche sémantique).

3.6.1 Analyse de la base d'apprentissage :

L'analyse exploratoire des données est l'une des parties les plus importantes de tout flux de travail d'apprentissage automatique et le traitement du langage naturel n'est pas différent. Mais quels outils choisir pour explorer et visualiser efficacement les données textuelles ?

Dans ce mémoire, nous discuterons et implémenterons quelques algorithmes d'apprentissage automatique :

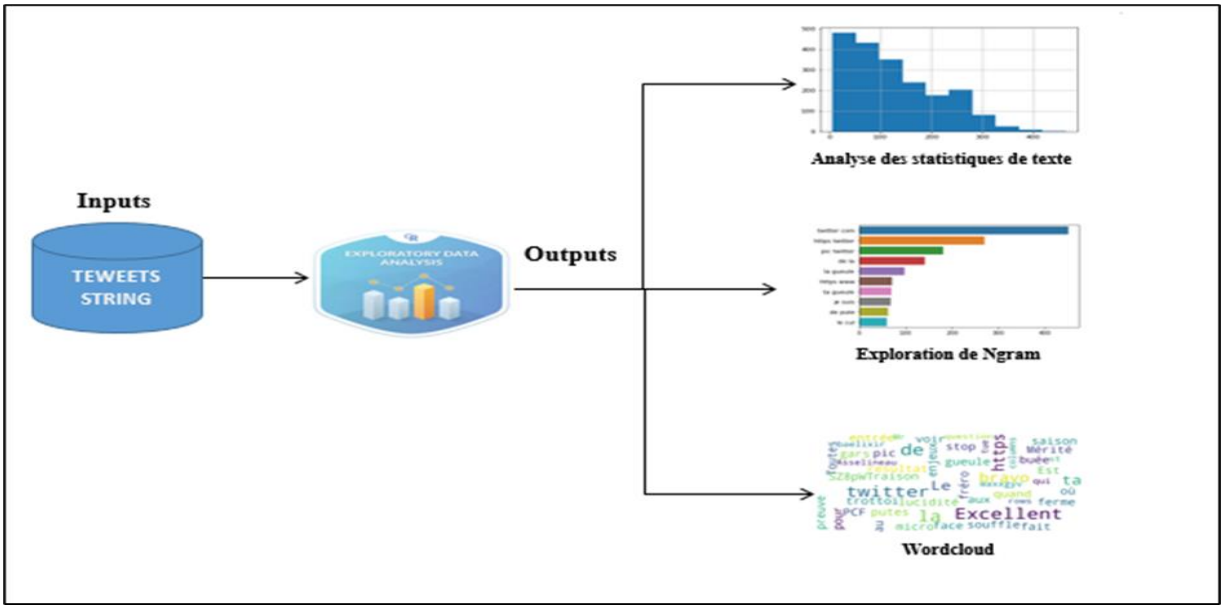

Figure 3- 4 : Analyse exploratoire des données pour le traitement du langage naturel.

3.6.1.1 Analyse des statistiques de texte

Les visualisations de statistiques de texte sont des techniques simples mais très perspicaces. Ils incluent :

- ✓ Analyse de la fréquence des mots,
- ✓ Analyse de la longueur des phrases,
- ✓ Analyse de la longueur moyenne des mots,etc.

Pour ce faire, nous utiliserons des histogrammes (données continues) et des histogrammes (données catégorielles).

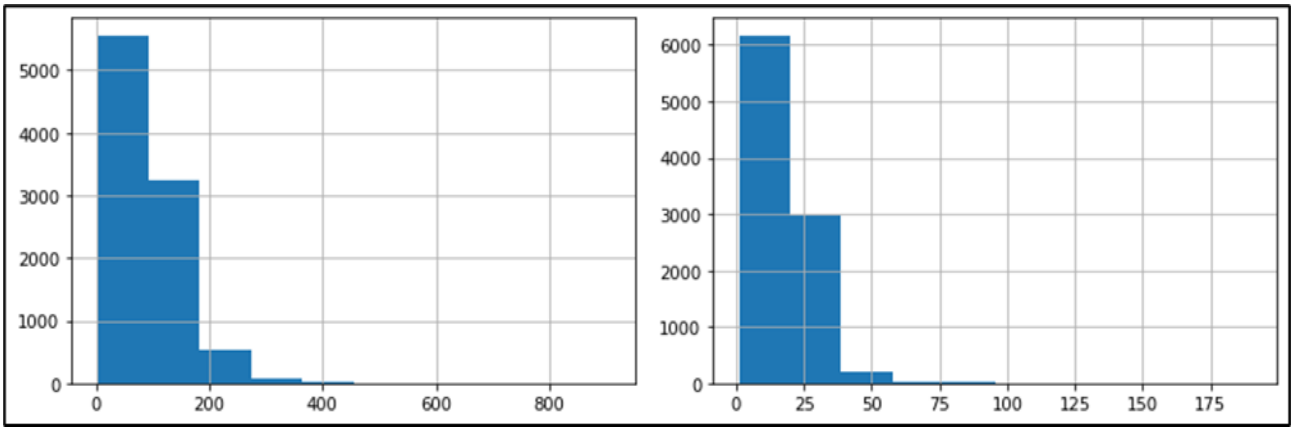

Figure 3- 5 : Analyse statistique.

3.6.1.2 Exploration de Ngram

Les ngrams sont simplement des séquences contiguës de n mots. Par exemple, «rive du fleuve», «Trois mousquetaires», etc., si le nombre de mots est de deux mots, il est appelé bigramme. Pour 3 mots, il s'appelle Trigram et ainsi de suite. Examiner les n-grammes plus fréquemment peut vous donner une meilleure compréhension du contexte dans lequel le mot est utilisé.

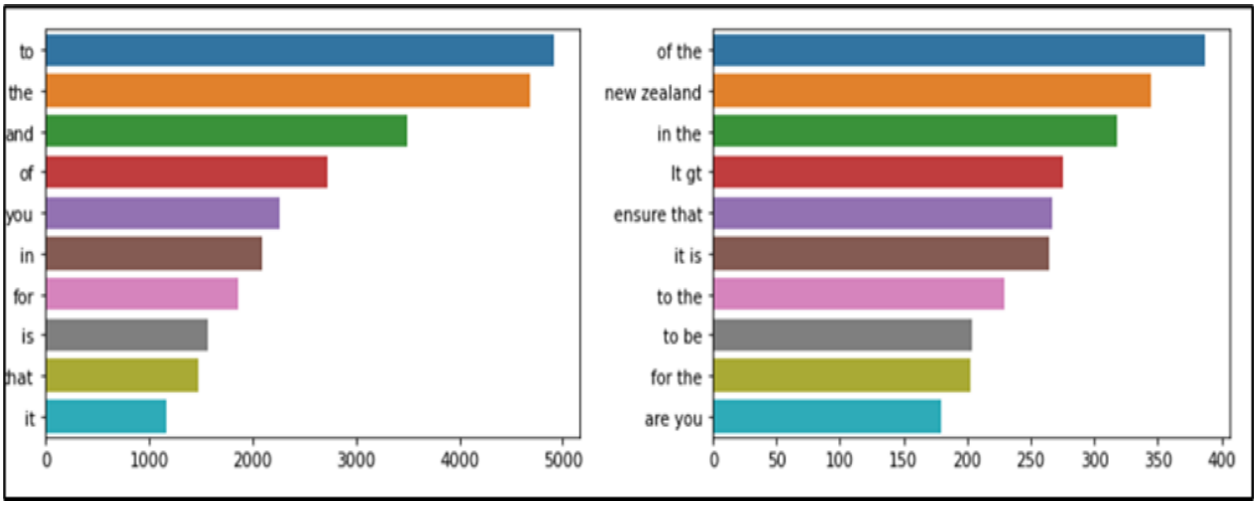

Figure 3- 6 : Exploration de Ngram (ngram, bingram).

3.6.1.3 Mot nuage (Wordcloud)

Keyword Cloud (rarement un Keyword Cloud ou Tag Cloud, Tag Cloud, Word Cloud est une ou keyword cloud en anglais) est une représentation visuelle des mots-clés (balises) les plus souvent utilisés dans un document ou un groupe. Généralement des mots sont affichés avec des tailles et des poids de police vérifiés qui sont plus simples que de les utiliser en pluriels. Voici des représentations de mots de ce dataset:

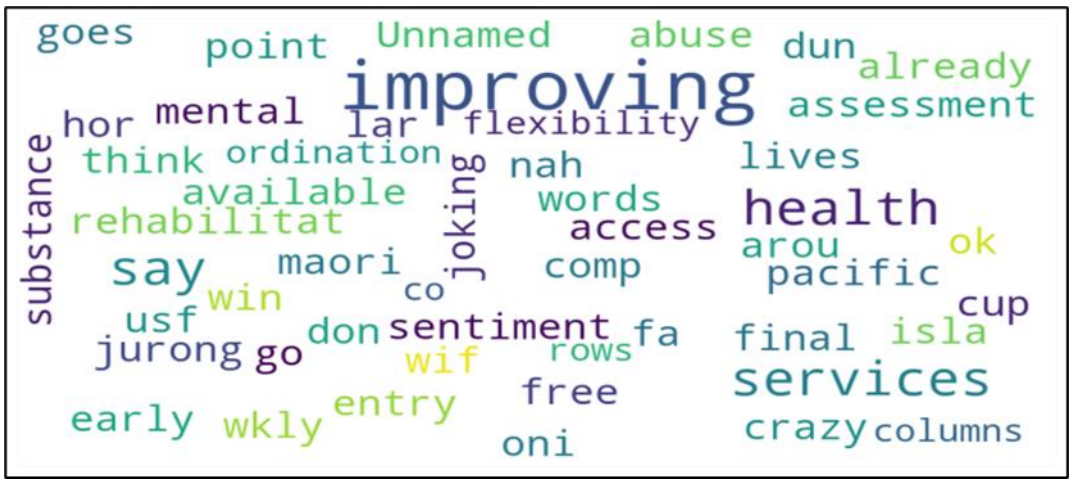

Figure 3- 7 : Dataset du mot nuage.

3.6.1.4 Analyse de sentiments des fichiers .xlsx

Nous avons prétraité les contenus textuels anglais pour l'analyse du sentiment. Ce prétraitement permet d'extraire la polarité des opinions qui s'exprime en négatif et positif d'une classification binaire et en négatif, neutre et positif dans le cas d'une classification multiple(figure 3-8 & figure 3-9). Les données de sortie que nous avons récoltées sont des tweets accompagnés de leurs polarités. Ces données vont être entraîner et tester par la suite à travers nos modèles d'apprentissage automatique (algorithmes traditionnels & Deep Learning).

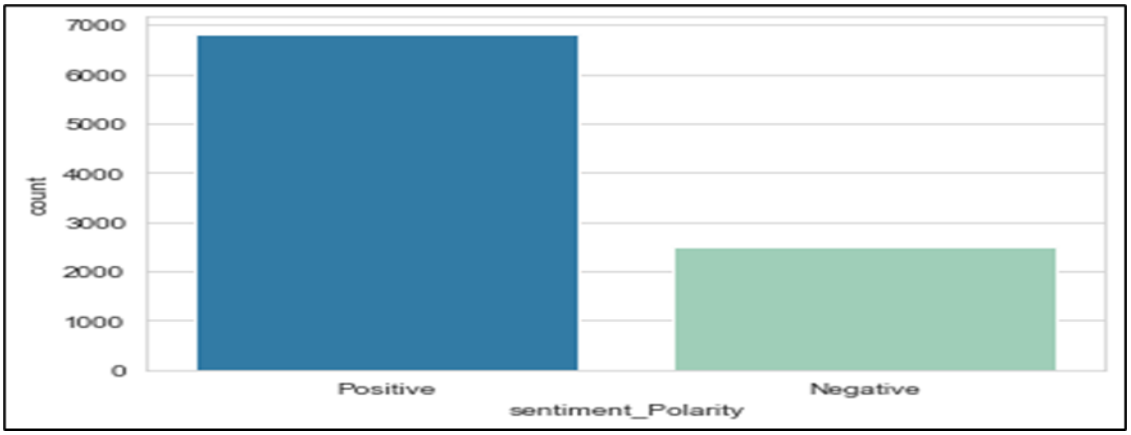

Figure 3- 8 : Annotation via Sentiwordnet : (Positive, Négative).

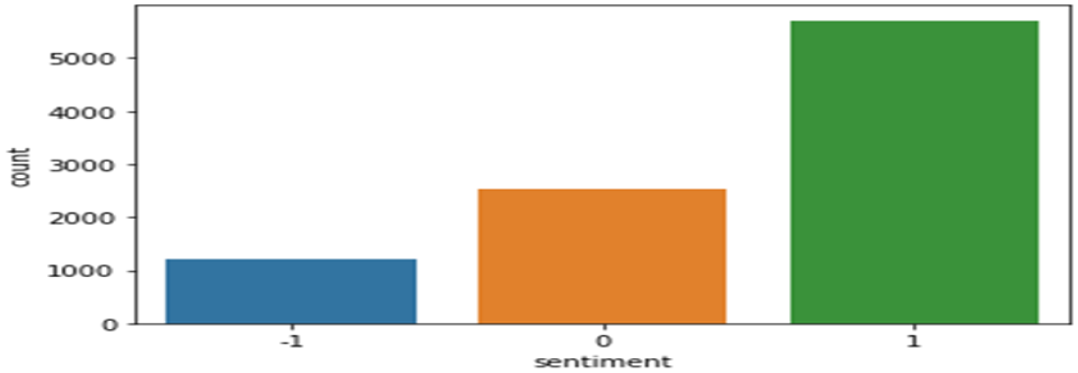

Figure 3- 9 : Annotation via le dictionnaire donné : (Positive, Neutre, Négative).

3.7 Extraction et Présentation des descripteurs

Le processus de classification du texte par des modèles d'apprentissage automatique est essentiellement le même que celui utilisé pour la classification d'un autre type de données. La principale différence, est constituée par le processus de transformation de données pour que celles-ci puissent être passées à l'algorithme de classification comme une représentation vectorielle numérique.

Dans cette transformation, il est nécessaire de faire passer les données du texte pur à une représentation dans laquelle les documents de texte sont numériquement représentés dans une matrice que le classifieur peut interpréter. On s'est basé sur la description de tâches du processus de classification illustré par la figure 3-10, on explique ci-dessous les différentes étapes de prétraitement du texte

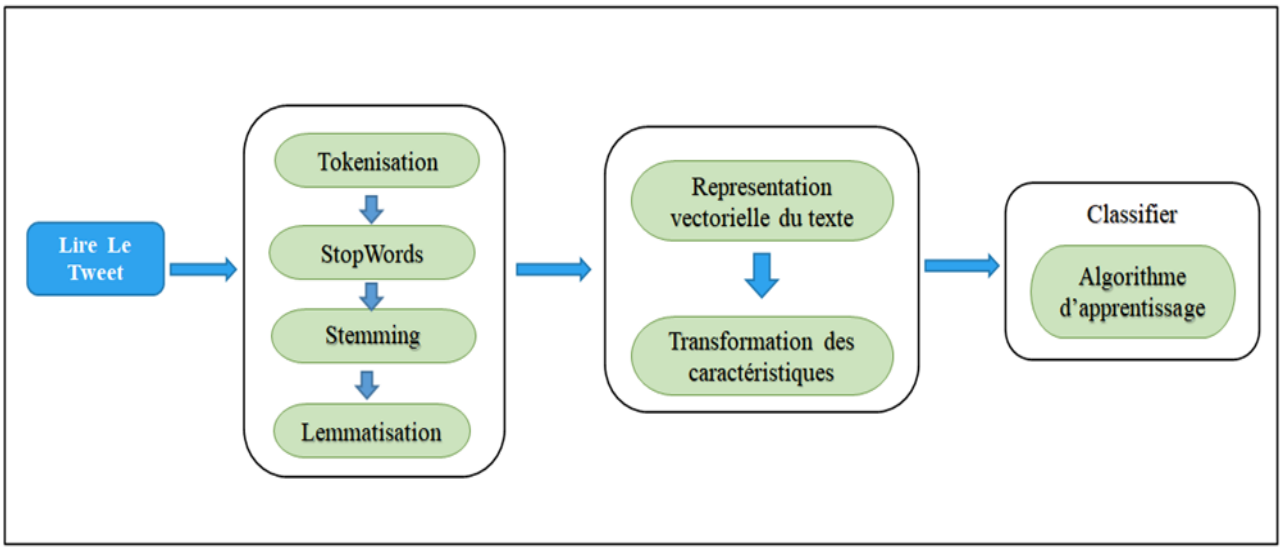

Figure 3- 10 : Extraction et présentation des descripteurs

3.7.1 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). 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 joue un rôle important dans le processus d'analyse lexicale.

Exemple :

```
In [24]: word_tokens = word_tokenize(dataset[6])
         print(word_tokens)

         ['over', 'a', 'decade', 'for', 'before', 'it', 'was', 'a', 'hashtag', 'earlier', 'today', 'i', 'was', 'on', 'alberta', 's', 'le
         gislature', 'ground', 'as', 'edmontonians', 'spoke', 'out', 'at', 'the', 'rally']
```

Figure 3- 11: Tokenisation

3.7.2 StopWords

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. Ces mots sont :

- Les conjunctions de coordination (for, and, nor, but, or, yet, so).
- Les determinants (a/an, the, this, that, these, those).
- Les prepositions (at, in, to).

Pour cibler et éliminer les mots vides, nous avons importé une liste des mots les plus fréquemment utilisés à partir de *NLToolkit* au début avec de *nltk.corpus import stopwords*. Après exécuter *stopwords.word (insérer la langue)* pour obtenir une liste complète pour chaque langue.

Exemple:

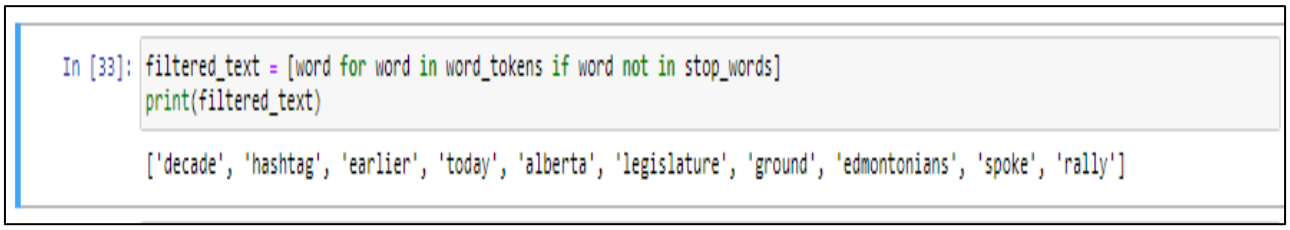

Figure 3- 12 : Exemple de stopWord.

3.7.3 La Lemmatisation

La lemmatisation est l'étape qui désigne l'analyse lexicale chargée de faire regrouper les mots d'une même famille qui partagent le même suffixe lexical. Chacun des mots du texte se trouve ainsi réduit en une entité appelée « Lemme ». Ce lemme désigne la forme canonique des mots. La lemmatisation regroupe les différentes formes que peut avoir un mot.

Par exemple, un nom en pluriel va être réduit au singulier, un verbe à son infinitif, etc. La lemmatisation est étroitement liée à la racinisation. La différence est qu'un stemmer fonctionne sur un seul mot sans connaissance du contexte, et ne peut donc pas faire la distinction entre des mots qui ont des significations différentes selon la partie du discours. Cependant, les stemmers sont généralement plus faciles à implémenter et à exécuter plus rapidement, et la précision réduite peut ne pas avoir d'importance pour certaines applications. La normalisation des verbes est une forme de la lemmatisation.

Examples of lemmatization:

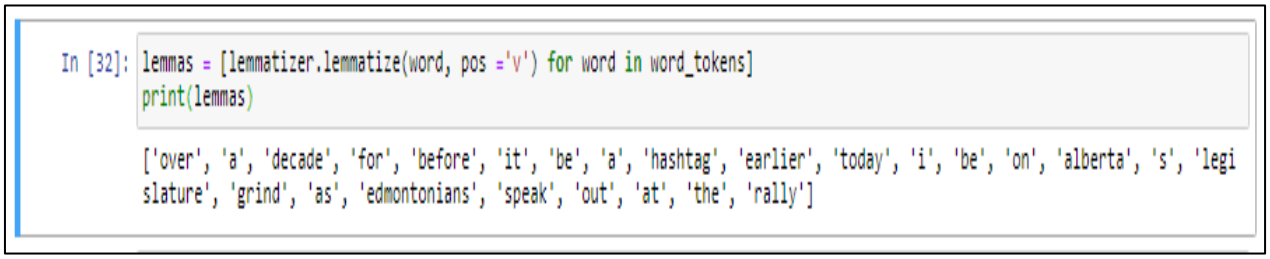

Figure 3- 13 : Exemple de lemmatisation.

3.7.4 Partie du marquage de la parole (POST)

Chaque mot du tweet à son rôle syntaxique qui définit comment le mot est utilisé. Ces rôles syntaxiques font partie du discours lié à ce mot. En anglais, il y a huit parties du discours qui sont connues comme le verbe, le nom, le pronom, l'adjectif, l'adverbe, la préposition, la conjonction et l'interjection.

Le processus consistant à classer les mots dans leurs parties du discours et à les étiqueter, en conséquence est connu sous le nom de marquage de partie de discours. On a utilisé des bibliothèques externes pour la segmentation des mots.

Dans la figure suivante, le prétraitement d'un exemple de texte :

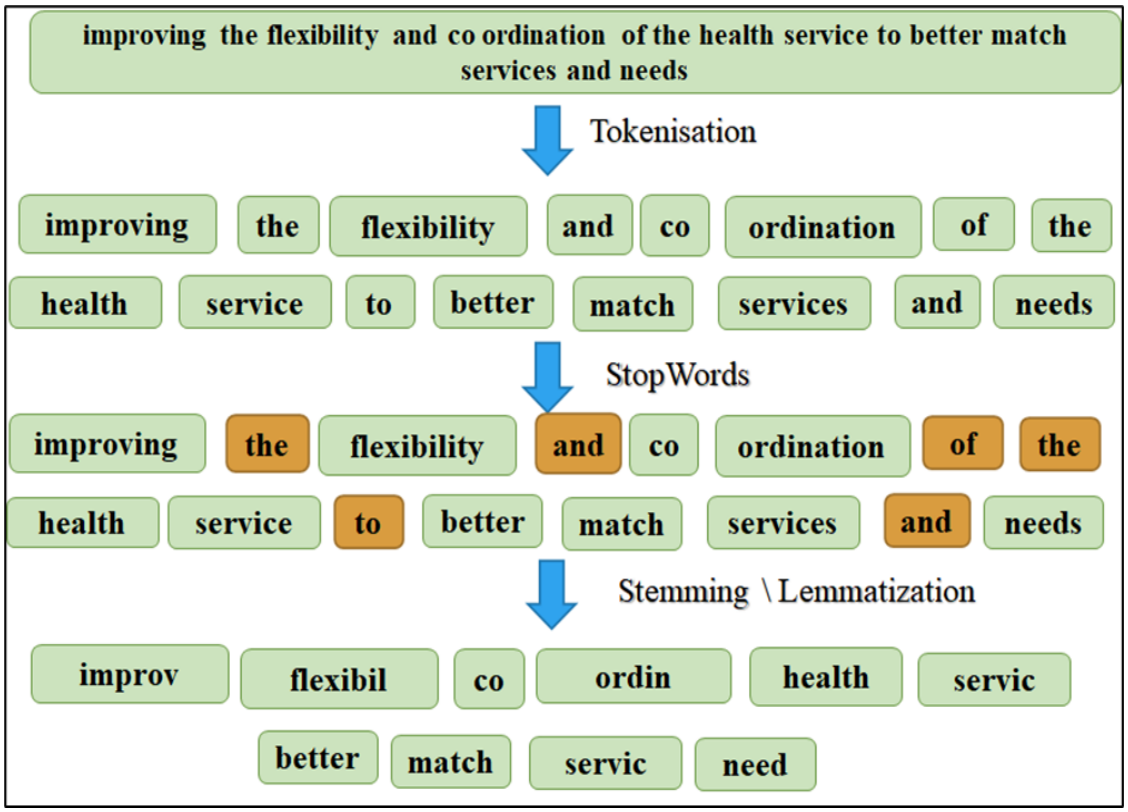

Figure 3- 14: Exemple global de prétraitement du texte

3.7.5 Représentation vectorielle du texte

Le texte original peut être vu comme une séquence de mots. Ce type de représentation est actuellement incompréhensible pour les algorithmes d'apprentissage automatique qui ont besoin de recevoir des représentations vectorielles numériques des entités à classer. La représentation vectorielle consiste à transformer chaque document en une séquence de nombres, dans laquelle chaque nombre correspond à un mot du vocabulaire de l'ensemble des documents ou corpus.

Pour transformer les documents de texte en vecteurs, on produit d'abord un vocabulaire avec tous les mots contenus dans les textes de l'ensemble d'entraînement. On produit en suite une matrice numérique dans laquelle chaque ligne correspond à un des documents de texte et chaque colonne correspond à un mot du vocabulaire du corpus. Si le mot n'apparaît pas dans le document, on lui assigne le nombre 0. Par contre, s'il apparaît, on peut lui assigner le nombre1, ou celui correspondant au total de fois que le mot apparaît dans le document. Cette dernière matrice s'appelle la matrice de fréquences.

La matrice numérique résultante peut être passée alors à l'algorithme de classementqui sera capable de l'interpréter et de travailler avec elle. Cette représentation est aussiappelée le sac de mots (bag of words). Le tableau illustre ce processus:

Document / Word	improving	mental	health	services	national	priorities	have	been	set
improving mental health services	1	1	1	1	0	0	0	0	0
national health priorities have been set	0	0	1	0	1	1	1	1	1

Table 3- 4 : Exemple de BOW

3.7.6 La transformation des caractéristiques

D'une part, on peut penser qu'il pourrait être judicieux de faire une représentationnumérique qui accorde plus d'importance aux mots dont la fréquence est haute dans lacatégorie à laquelle ils appartiennent et basse dans les autres catégories, en vue de pondérerla valeur numérique de chaque mot selon l'information qu'elle apporte pour le classement.Aussi, c'est l'effet produit par la pondération TF-IDF, introduite par [38], qu'on utilisera dansla partie pratique de notre recherche. Il y a par ailleurs d'autres pondérations possibles commele $\mathcal{X}2$, le $\mathcal{X}p2$, le gini index, et le gain d'information, expliquées par [39], qui permettent ausside capturer cette sorte de relations entre mots et documents.

Dans cette étape, nous considérons les termes restant après l'étape de prétraitementcomme descripteurs. Ces descripteurs ont un rôle important pour le classement dessentiments. Pour réaliser l'opération d'apprentissage, nous avons proposé deuxreprésentations : Le modèle de représentation Word Embedding et le modèle de pondérationTF-IDF.

3.7.6.1 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).

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 parl'équation suivante

$$IDF_i = \log(\frac{D}{d_j : t_i \epsilon d_j})$$

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 :

$$TF_i - IDF_{i,j} = TF_{i,j} * IDF_i$$

3.7.6.2 Incorporation de mot

L'incorporation de mot, ou "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 le Topic Modeling ou le Sentiment Analysis.

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

3.8 La classification des tweets

Ils existent plusieurs méthodes de classification supervisée et beaucoup d'entre elles ont été testés pour la catégorisation des sentiments. On peut citer les réseaux de neurones, la régression logistique, les arbres de décision, les machines à support de vecteurs, les réseaux de neurones convolutifs, les réseaux de neurones récurrents ainsi que des méthodes combinant différents classifieurs.

3.8.1 Train et test

Le fractionnement train-test est une technique permettant d'évaluer les performances d'un algorithme d'apprentissage automatique.

- Ensemble de données de train : utilisé pour s'adapter au modèle d'apprentissage automatique.
- Jeu de données de test: utilisé pour évaluer le modèle d'apprentissage automatique d'ajustement

Vous devez choisir un pourcentage fractionné qui répond aux objectifs de votre projet en tenant compte des éléments suivants:

- Coût de calcul de la formation du modèle.
- Coût de calcul lors de l'évaluation du modèle.
- Représentativité de l'ensemble de formation.
- Représentativité de l'ensemble de test.

Les pourcentages de fractionnement courants comprennent:

- Train: 80%, Test: 20%
- Train: 67%, Test: 33%
- Train: 50%, Test: 50%

Pour applique le pourcentage de fraction courante de classement Train : 80%, Test: 20% pour les code et résultat parmi les model bag of word comme suit :

```
X_train, X_test, y_train, y_test = train_test_split(df['words'], df['sent
                  iment'], test_size=0.2, random_state=1)
```

	Dataset	Train set	Test set	Positive	Negative
Shape	9423	(7538,)	(1885,)	7078	2345

3.8.2 Quelques algorithmesde ML

✓ Naive baise (MultinomialNB)

```
#nbTrain
from sklearn.metrics import average_precision_score
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
start_timenb = time.time()
tf=TfidfVectorizer()
train = tf.fit_transform(X_train)
test = tf.transform(X_test).toarray()
actual = y_test
nb = MultinomialNB()
```

- ✓ Machine à vecteurs de support

```
#Tsvm
from sklearn.svm import SVC
start_timesvm = time.time()

train_2 = tf.fit_transform(X_train)
actual2 = y_test
test_2 = tf.transform(X_test).toarray()
svc = SVC()
svc = svc.fit(train_2, y_train)
```

- ✓ Decision tree :

```
#datree
from sklearn import tree
start_timedt = time.time()

train_1 = tf.fit_transform(X_train)
actual1 = y_test
test_1 = tf.transform(X_test).toarray()
dtree = tree.DecisionTreeClassifier()
dtree = dtree.fit(train_1, y_train)
prediction1 = dtree.predict(test_1)
#Accuracy
```

- ✓ K neighbors classifier

```
#knN
from sklearn.neighbors import KNeighborsClassifier
start_timekn = time.time()

train_3 = tf.fit_transform(X_train)
actual3 = y_test
test_3 = tf.transform(X_test).toarray()
kn = KNeighborsClassifier(n_neighbors=1)
kn = kn.fit(train_3, y_train)
prediction3 = kn.predict(test_3)
```

✓ RNN-Lstm de deep learning

```
def RNN():
    inputs = Input(name='inputs',shape=[max_len])
    layer = Embedding(max_words,50,input_length=max_len)(inputs)
    layer = LSTM(64)(layer)
    layer = Dense(256,name='FC1')(layer)
    layer = Activation('relu')(layer)
    layer = Dropout(0.5)(layer)
    layer = Dense(1,name='out_layer')(layer)
    layer = Activation('sigmoid')(layer)
    model = Model(inputs=inputs,outputs=layer)
    return model
```

3.9 Evaluation des résultats

3.9.1 Matrice de Confusion

Dans cette partie, nous présentons les résultats des algorithmes classiques de ML:

Total des lignes de test est de 1885.

Modèles classiques	Features		TP	TN	FP	FN
MultinomialNB	**TF_IDF**	**n_gram**	1438	15	0	437
		Bi_gram	1390	73	43	379
		Tri_gram	1411	30	22	422
Decision tree		**n_gram**	279	279	169	173
		Bi_gram	1254	189	179	263
		Tri_gram	1356	93	77	359
SVM		**n_gram**	1409	192	14	250
		Bi_gram	1392	106	41	346
		Tri_gram	1398	62	35	390

KNN	**n_gram**	1408	111	25	341
	Bi_gram	1296	176	137	276
	Tri_gram	1365	80	68	372

Table 3- 5 : Résultats des classifieurs via la matrice de confusion.

3.9.2 Résultats de l'apprentissage automatique via sentiwordnet

Malgré la répartition assez inégale des sentiments (positive pour la grande majorité), le système parvient néanmoins à réaliser des résultats plutôt convaincants.

Afin d'appuyer les propos ci-dessus, les tableaux suivants contiennent les différents résultats avec les classifieurs classiques avec TF-IDF :

Modèles classiques	**Features**		**Accuracy**	**Precision**	**Recall**	**F-measures**
MultinomialNB	**TF_IDF**	**n_gram**	76.87	76.63	100	86.76
		Bi_gram	77.61	82.66	96.99	86.82
		Tri_gram	76.44	76.97	98.46	86.40
Decision tree		**n_gram**	83.34	87.47	89.67	88.55
		Bi_gram	76.55	82.66	87.50	85.01
		Tri_gram	76.78	79.06	94.62	86.14
SVM		**n_gram**	85.35	84.51	99.02	91.19
		Bi_gram	79.46	80.09	97.13	87.79
		Tri_gram	77.45	78.18	97.55	86.80
KNN		**n_gram**	80.58	80.50	98.25	88.49
		Bi_gram	78.09	82.44	90.43	86.25
		Tri_gram	76.58	78.58	95.25	86.11

Table 3- 6 : Résultats des classifieurs classiques+ TF-IDF sur les tweets

ModèlesDeepLearning	**Features**	**Accuracy**
LSTM (RNN)	**Word Embedding**	82.40%

Table 3- 7 : Résultats des classifieurs DL + Word Embedding sur les tweets

1) Le classifieur Multinomial NB utilisant les différents d-grammes (Uni-, Bi- et Tri_grams) atteint un très bon niveau de Accuracy de 77.61% et un très bon niveau de Précision 82.66% avec utilisant des bigrammes, un très bon niveau de Recall 100% utilisant des n_ngram, un très bon niveau de score F1 86.82% utilisant des Tri_grams.

2) Le classifieur Decision tree atteint un très bon niveau de Accuracy de 82.38% et un très bon niveau de Précision 87.47%, et un très bon niveau de score 88.55% tout utilisant des n_ngram, un très bon niveau de Recall 94.62% utilisant des Tri_grams.

3) Le classifieur SVM atteint un très bon niveau de Accuracy de 85.46% et un très bon niveau de Précision 84.51% et un très bon niveau de Recall 99.02% et un très bon niveau de score F1 91.19% utilisant des n_ngram.

4) Le classifieur KNN atteint un très bon niveau de Accuracy de 80.58% et un très bon niveau de Précision 82.44% utilisant des bigrammes, un très bon niveau de 98.25% etun très bon niveau de score F1 88.49% utilisant des Unigrammes.

5) Le classificateur LSTM-RNN sans élimination des mots fonctionnels, avec le Word Embedding a obtenu le meilleur résultat de Accuracy 82.40% surperformant le résultat des autres classifieurs.

Nous avons pensé à tester les performances de notre système sur corpus, pour cela on a lancé l'apprentissage sur tout le système.

En générale, la fonction Unigrammes et l'inclusion des mots fonctionnels ont obtenu les meilleurs résultats de précision dans tous les classifieurs.

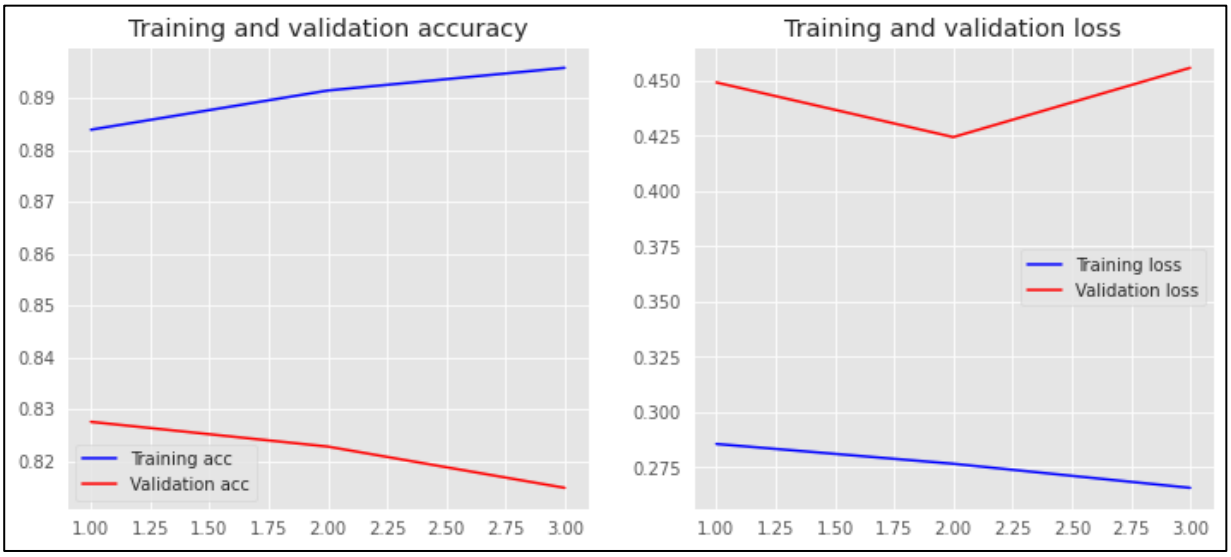

Figure 3- 15 : Résultats moyens de performance sentiwordnet des classifieur RNN-Lstm.

3.9.3 Résultats de l'apprentissage automatique via le dictionnaire donné

Le tableau suivant contient les résultats de l'analyse des corpus proche lexicale :

Modèles classiques	Features	Accuracy	Precision	Recall	F-measures
MultinomialNB	n_gram	63.30	72.20	36.66	32.06
	Bi_gram	63.93	64.74	41.07	40.43

	TF_IDF	**Tri_gram**	62.24	61.55	37.34	34.43
Decision tree		**n_gram**	65.78	57.19	55.11	56.0
		Bi_gram	61.77	52.09	47.94	49.19
		Tri_gram	60.82	48.44	38.80	37.78
SVM		**n_gram**	72.22	76.14	53.16	55.78
		Bi_gram	65.99	67.36	44.93	46.30
		Tri_gram	62.77	58.30	38.45	36.35
KNN		**n_gram**	65.57	63.31	44.64	45.64
		Bi_gram	51.63	50.10	48.20	45.69
		Tri_gram	42.92	52.45	43.19	36.34

Table 3- 8 : Résultats des classifieurs classiques+ TF-IDF sur les tweets.

ModèlesDeepLearning	**Features**	**Accuracy**
LSTM (RNN)	**Word Embedding**	62.2%

Table 3- 9 : Résultats des classifieurs DL + Word Embedding sur les tweets.

1) Le classifieur MultinomialNB utilisant les différents d-grammes (Uni-, Bi- et Tri_grams) atteint un très bon niveau de Accuracy de 63.93% et un très bon niveau deRecall41.07% et très bon niveau de score F1 86.82%avec utilisant des bigrammes,un très bon niveau de Précision82.66%utilisant des n_ngram.

2) Le classifieurDécision treeatteint un très bon niveau de Accuracy de 65.78% et un très bon niveau de Précision57.19%,et un très bon niveau de score 55.11% et un très bon niveau de Recall 56%tout utilisant des n_ngram.

3) Le classifieur SVMatteint un trèsbon niveau de Accuracy de 72.22% et un très bon niveau de Précision76.14% et un très bon niveau de Recall53.16% et un très bon niveau de score F1 55.78% utilisant desn_ngram.

4) Le classifieur KNNatteint un très bon niveau de Accuracy de 65.57% et un très bon niveau de Précision63.31%utilisant des Unigrammes,un très bon niveau de 48.20% etun très bon niveau de score F1 45.69%utilisant des bigrammes.

5) Le classificateur LSTM-RNN sans élimination des mots fonctionnels, avec le Word Embedding a obtenu le meilleur résultat de Accuracy 62.2% surperformant le résultat des autres classifieurs.

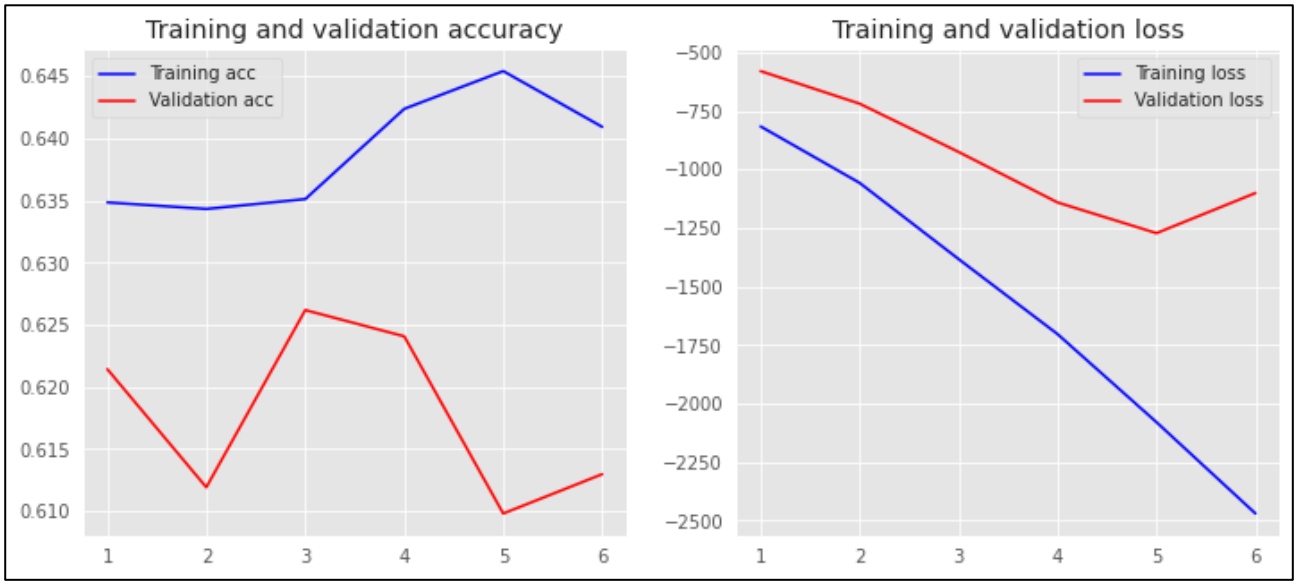

Figure 3- 16 : Résultats moyens de performance dictioner des classifieur RNN-Lstm.

Selon les résultats obtenus, et grâce à accuracy, nous concluons que les algorithmes ML pourl'annotation sémantique via sentiwordnet obtiennent les meilleures performances par rapport à cequ'ils ont réalisé dans l'annotation via dictionnaire donné, ce qui nous prouve la précision,l'exactitude et l'efficacité de l'utilisation de l'approche sémantique pour analyser une grande quantitéde données afin de déduire les sentiments positfs qui ont été exprimés.

Nous constatons également que le modèle de DL le RNN et SVM surpassent les autres algorithmes. Comme nous pouvons l'observer dans la table 3-7, pour notre dataset, ces deux classificateursdonnent de bonnes performances et ilssont très précis et prédientt très bien parmi tous.

Cela nous montre l'avantage de l'application d'une approche DL dans ce type de tâches d'analyse de sentiments et la bonne performance que nous obtenons par rapport aux algorithmes ML traditionnels.

Les résultats que nous avons obtenus sont très intéressants et très prometteurs et montrent que notre système est très performant et très compétitif dans son domaine.

3.10 Conclusion

Dans ce chapitre, nous avons présenté les méthodes utilisées pour traiter les données préalablement extraites de Twitter, et les expériences effectuées sur les données analysées. Nous avons également vu les différentes étapes pour faire une représentation vectorielle du document textuel (.xls) (tokenisation, stemming, élimination des mots vides, vectorisation et Transformation), et discuter les résultats de l'analyse de ce dataset. Nous avons constaté que les algorithmes d'apprentissage en profondeur et le SVM dépassent les autres algorithmes classiques et donnent des résultats très intéressants.

Conclusion générale

Conclusion générale

Dans ce mémoire, nous avons présenté en détail les aspects théoriques et pratiques liés à l'implémentation de notre application d'analyse de sentiments pour les tweets violants. Il s'agit donc de la classification automatique des textes de tweets en trois classes : positive, négative ou neutre via les algorithmes traditionnels et en profondeur de ML en passant par une ressource lexicale externe (dictionnaire donné ou SentiWordnet) pour la phase de prétraitement des données originales.

Le potentiel de l'utilisation du big-social data "Twitter" comme outil de mesure et de prédiction de la violence verbale a été soigneusement démontré et les expérimentations menées ont permis de valider notre application. Les expériences faites ont montré que plus le niveau d'analyse est profond plus les résultats sont spécifiques.

Nous venons donc de réaliser un outil permettant de classifier si un texte est injurieux ou non. Cela peut pourrait être utile sur les plateformes d'échanges de tweets, de commentaires, ... des grands réseaux sociaux pour lutter contre le harcèlement moral qui peut se présenter sous diverses formes.

Ce travail nous a permis de mettre plus en pratique nos connaissances théoriques sur les algorithmes traditionnelles de ML et les réseaux de neurones ainsi que de les enrichir, et le plus important est que nous fassions le premier pas vers l'apprentissage profond, un des champs les plus importants de l'Intelligence Artificielle.

Parmi les orientations futures:

- Assemblage (hybridation/combinaison) des modèles de ML doit être une forme de technique d'algorithmes de méta-apprentissage dans laquelle nous proposons de combiner différents classificateurs afin d'améliorer la précision de la prédiction.
- Tester l'outil implémenté sur d'autres jeux de données (emojis, images, videos...) et sur d'autres plateformes comme facebook, youtube ou autres.
- Application d'autres classificateurs et utilisation d'autres fonctionnalités.
- Création d'un lexique orienté violence verbale.

Bibliographie

Bibliographie :

[1]Bellachhab Abdelhadi et galatanu Olga. La violence verbale Disponible sur Internet : http://www.revue-signes.info/document.php?id=2893. ISSN 1308-8378

[2] https://biblio.univ-annaba.dz/wp-content/uploads/2020/01/These-Ziani-Amel.pdf[3] Study of Different Levels for Sentiment Analysis. Seema Kolkur, Gayatri Dantal and Reena Mahe. International Journal of Current Engineering and Technology. 2015

[4] https://medium.com/@mehdihadji/analyse-des-sentiments, consulté (juillet 2020) https://medium.com/@mehdihadji/analyse-des-sentiments-g%C3%A9n%C3%A9ralit%C3%A9s-99ab87503a5e

[5] https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/, consulté (Aout 2020)

[6] Sentiment Analysis of Micro blogs using Opinion Mining Classification Algorithm.A.Tamilselvi1, M. ParveenTaj. International Journal of Science and Research (IJSR). 2014

[7] https://www.meltwater.com/fr/blog/analyse-sentiment, consulté (juillet 2020)

[8] https://www.geeksforgeeks.org/decision-tree/?ref=leftbar-rightbar, consulté (Aout 2020)

[9]Matthew Eric Glassman, Jacob R. Straus and Colleen J. Shogan, "Social Net- workingand Constituent Communications: Members Use of Twitter and Facebook Dur- ing aTwo-Month Period in the 112th Congress,"Congressional Research Service, 2009.

[10] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passon- neau, "Sentiment analysis of Twitter data," LSM 11 Proceedings of the Workshop on Languagesin Social Media,2011.

[11] Laurent Dijoux, "Boostez votre business avec Twitter,"Almabic, 2009.

[12] Fred Colantonio, "Communication professionnelle en ligne: comprendre et exploiterles médias et réseaux sociaux,"Edipro, 2011.

[13] Tim O'Reilly and Sarah Milstein," The Twitter Book," Angham B3 2PB, UK, 2

[14]Kumar Ela, "Natural Language Processing," India,I.K.International Publishing HousePvt. Ltd 2011.

[15] Jean Véronis, "Natural Language Processing", http://sites.univ- provence.fr/veronis, 2001.

[16] Meena Rambocas and Joo Gama, "The Role of Sentiment Analysis,"FEP Economics and Managment, 2013

[17]Soumia Elyakoute HERMA et Khadidja SAIFIA, "Analyse des sentiments cas Twitter",Université de Ghardaia, 2015

[18] « The International Conferenceon Advanced Machine Learning Technologies and ... - Google Livres ». https://books.google.dz/books?id=N0-

NDwAAQBAJ&pg=PA891&lpg=PA891&dq=wordnet+preprocessing&source=bl&ots=4DR_HowZj3&sig=ACfU3U0KzkIeiJtWORZF7PDF3qxl_XCoOw&hl=fr&sa=X&ved=2ahUKEwjn9YTW4dfpAhVFC2MBHV3fBg0Q6AEwDnoECAcQAQ#v=onepage&q=wordnet%20preprocessing&f=false.p. 891.

[19] « streamcrab». http:// smm.streamcrab.com (consulté le juin. 26, 2020).

[20] S. Narkhede, « Understanding Confusion Matrix », *Medium*, août 29, 2019. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62 (consulté le juillet. 29, 2020).

[21] « Classification : ROC et AUC | Cours d'initiation au machine learning », *Google Developers*. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr (consulté le juillet. 29, 2020).

[22] ZACCONE Giancarlo, MD REZAUL Karim, MENSHAWY Ahmed. Deep learning with tensorflow. 2017

[23]Thème: Deep Learning pour la classification des images //Université Abou Bakr Belkaid– Tlemcen //Année universitaire : 2016-2017 //MoualekDjaloul Youcef

[24] https://medium.com/jedha/les-applications-du-deep-learning-d674483d44af#:~:text=Dans pratique langages,nombre neurones couches, consulté (juillet 2020)

[25] https://machinelearningmastery.com/promise-deep-learning-natural-language-processing/, consulté (septembre 2020)

[26] http://www.fullai.org/short-history-artificial-intelligence/, consulté (septembre 2020)

[27]Python Deep Learning. Ivan Vasilev, Daniel Slater, GianmarioSpacagna, Peter Roelants, Valentino Zocca. 2019

[28] Thème par Colorlib Propulsé par WordPress https://deeplylearning.fr/cours-theoriques-deep-learning/fonctionnement-du-neurone-artificiel/

[29] NAKAMOTO Pat. Neural networks and deep learning : deep learning explained to your granny a visual introduction for beginners who want to make their own deep learning neuralnetwork. 2017

[30] https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/, consulté (Novembre 2020)

[31] https://blog.octo.com/les-reseaux-de-neurones-recurrents-des-rnn-simples-aux-lstm/, consulté(Novembre 2020)

[32] https://www.geeksforgee ks.org/understanding-of-lstm-networks, consulté(Novembre 2020)

[33] https://en.wikipedia.org/wiki/Anaconda_(Python_distribution),consulté(Novembre 2020)

[34] https://ledatascientist.com/google-colab-le-guide-ultime/, consulté(Octobre 2020)

[35] https://ichi.pro/fr/post/111402284763203, consulté(Octobre 2020)

[36] https://www.vneuron.com/2018/01/23/introduction-a-lapprentissage-automatique-avec-scikit-learn/, consulté(Octobre 2020)

[37] https://moncoachdata.com/blog/python-pour-la-science-des-donnees/, consulté(Octobre 2020)

[38]Hiroshi Ogura, Hiromi Amano, and Masato Kondo. “Distinctive characteristics of ametric using deviations from poisson for feature selection”. Expert Systems withApplications, 37(3) :2273-2281, 2010.

[39]Gerard Salton, Anita Wong, and Chung-Shu Yang. “A vector space model for automaticindexing”. Communications of the ACM, 18(11) :613-620, 1975S

Résumé :

Notre projet de fin d'études porte sur une analyse textuelle d'une collection de tweets préalablement collectés et contenant de la violence verbale et plus particulièrement le harcèlement. Donc, notre objectif à travers ce travail est de déceler automatiquement les caractéristiques du discours violent analysant notre corpus en se basant sur deux domaines de recherches: l'analyse des sentiments et leurs techniques de classification. D'après notre analyse du corpus, nous avons constaté que la violence verbale est pertinente dans l'ensemble des tweets, autrement-dit, les utilisateurs du big social data Twitter interagissent violemment entre eux. En tant qu'une solution à cette problématique, nous abordons le problème de l'analyse des sentiments sur un ensemble de tweets en passant par une ressource lexicale linguistique externe (dictionnaire donné ou SentiWordnet) pour la phase de prétraitement et en utilisant un certain nombre de modèles d'apprentissage automatique traditionnel et en profondeur pour la classification. Il s'agit donc de la classification automatique de tweets en tweets harcelés et tweets non harcelés. Le Deep Learning, approche incontournable en Machine Learning, s'avère particulièrement utile et performant pour cette classification.

Mots-clés: Extraction de texte, Analyse des sentiments, polarité, apprentissage automatique, Twitter.

Abstract:

Our end of studies project concerns a textual analysis of a collection of tweets previously collected and containing verbal violence and more particularly harassment. Therefore, our objective through this work is to automatically detect the characteristics of the violent speech analyzing our corpus based on two areas of research: the analysis of feelings and their classification techniques. Based on our analysis of the corpus, we found that verbal abuse is relevant across all tweets – that is, users of Twitter big social data interact violently with each other. As a solution to this problem, we tackle the problem of analyzing feelings on a set of tweets by going through an external linguistic lexical resource (given dictionary or SentiWordnet) for the preprocessing phase and using a number traditional and deep machine learning models for classification. This is therefore the automatic classification of tweets as harassed tweets and non-harassed tweets. Deep Learning, an essential approach in Machine Learning, is particularly useful and efficient for this classification.

Keywords: Text mining, Sentiment analysis, polarity, machine learning, Twitter.

ملخص:

يتعلق مشروع نهايةالدراساتالخاصبنابتحليلنصيلمجموعةمنالتغريداتالتيتمجمعهامسبقًاوالتيتحتويعلىعنف لفظي وتحرش بشكل خاص. لذلك ، فإن هدفنامنخلال هذاالعمل هوالكشفالتلقائيعنخصائصالكلامالعنيفالذييحلل مجموعتنا بناً ء على مجالين من مجالات البحث: تحليل المشاعر وتقنيات تصنيفها. بناًءعلىتحليلناللمحتوىالأساسي،وجدنا أن الإساءة اللفظية وثيقة الصلة بجميع التغريدات – أي أن مستخدمي البيانات الاجتماعية الكبيرة على تويتر يتفاعلون بعنفمع بعضهمالبعض. كحلل هذهالمشكلة،نعالج مشكلة تحليل المشاعر في مجموعة من التغريدات من خلال المرور بمورد لغويخارجي (قاموس معين أو SentiWordnet)لمرحلة المعالجة المسبقة واستخدام رقم نماذج التعلم الآلي التقليدية والعميقةللتصنيف. وبالتالي فإن هذاهوالتصنيفالتلقائيللتغريداتعلىأنهاتغريداتمضايقة وتغريدات غير مضايقة. التعلم العميق،وهونهجأساسيفيالتعلمالآلي، مفيد وفعال بشكل خاص لهذاالتصنيف.

الكلمات المفتاحية: التنقيب عن النص ، تحليل المشاعر ، القطبية ، التعلم الآلي ، تويتر...