N° d'ordre : 226/2019

UNIVERSITÉ SULTAN MOULAY SLIMANE
Faculté des Sciences et Techniques
Béni Mellal

Centre d'Etudes Doctorales : Sciences et Techniques

Formation doctorale : Mathématiques et Physique Appliquées

THÈSE

Présentée par

MADANI Youness

Pour l'obtention du grade de

Docteur

Spécialité : Informatique

Contribution à l'analyse des réseaux sociaux dans un environnement Big Data.

Soutenue le 01/11/2019 devant le jury :

Pr **Abdellatif HAIR**	Professeur à la FST-Béni Mellal	Président
Pr **Françoise SAILHAN**	Maitre de Conférences-CNAM-Paris FRANCE	Rapporteur
Pr **Mohamed BAHAJ**	Professeur à la FST-Settat	Rapporteur
Pr **Rachid EL AYACHI**	Professeur à la FST-Béni Mellal	Rapporteur
Pr **Najla IDRISSI**	Professeur à la FST-Béni Mellal	Examinatrice
Pr **Jamaa BENGOURRAM**	Professeur à la FST-Béni Mellal	Encadrant
Pr **Mohammed ERRITALI**	Professeur à la FST-Béni Mellal	Co-encadrant

À ***ma grande mère***
À ***ma mère***
À ***Mon père***
À ***ma chère sœur***
À ***mes frères***
À ***ma grande famille***
À ***mes amis.***

Avant-Propos

Je ne saurais commencer ce manuscrit sans remercier ceux qui ont, de manière plus ou moins directe, contribuent à la réussite de ces trois années de travail. Durant les trois dernières années j'ai beaucoup appris sur moi-même, sur le métier de chercheur et sur la nature humaine. J'ai l'occasion ici d'exprimer ma gratitude envers chacune des personnes qui de près ou de loin ont contribué à faire de cette thèse un succès.

En premier lieu, je voudrais exprimer ma profonde gratitude à mes encadrants, les professeurs **Mohammed ERRITALI** et **Jamaa BENGOURRAM**, pour leur soutien, leurs conseils et leurs encouragements tout au long de ces trois années. Ils m'ont aidé à faire passer les idées, à formuler clairement les concepts présentés ici et à me donner l'occasion de découvrir et de développer de nombreux intérêts. Je tiens donc à leur exprimer toute ma reconnaissance et saluer leurs grandes compétences scientifiques, pédagogiques et leurs qualités personnelles.

Mes remerciements vont également aux Messieurs **Mohamed Fakir - Mustapha Mabrouki** les directeurs de mes laboratoires d'accueil : *le laboratoire de traitement de l'information et aide à la décision* et *le laboratoire de génie industriel*, et à Monsieur le directeur du centre d'études doctorales de la faculté des sciences et techniques de Béni Mellal.

Je tiens également à remercier Madame **Françoise SAILHAN**, Monsieur **Mohamed BAHAJ** et Monsieur **Rachid EL AYACHI** pour l'intérêt qu'ils ont porté à ce travail en acceptant d'en être rapporteurs, pour le temps qu'ils ont investi et leurs précieux commentaires. Un immense merci également à Monsieur **Abdellatif HAIR** pour m'avoir fait l'honneur d'accepter de présider le jury de ma thèse. Je remercie également Madame **Najla IDRISSI** d'avoir accepté de participer au jury de ma thèse, c'est un honneur.

Je voudrais remercier ma chère sœur, mes frères et particulièrement mes parents, qui m'ont toujours encouragé et soutenu pour tout ce que j'ai accompli. Ils m'ont inculqué un amour d'apprendre qui a grandi au fil des ans. Merci à tous mes amis, mes proches pour leur aide en diverses occasions et de leur encouragement continu. Un profond merci à mon cher ami **Ismail ELOUARGUI** qui m'a soutenu et encouragé pendant ces années.

Je vous remercie enfin, vous, cher lecteur, pour l'intérêt que vous portez à nos contributions. En espérant que la lecture de ce manuscrit pourra vous apporter ce que vous y cherchez.

Liste des Publications et des Communications

Liste des Publications

- **Youness MADANI**, Mohammed ERRITALI and Jamaa BENGOURRAM, "Sentiment analysis using semantic similarity and Hadoop MapReduce", Knowledge and Information Systems, volume 59, issue 2, pp 413–436, May 2019.
- **Youness MADANI**, Mohammed ERRITALI and Jamaa BENGOURRAM, "A Hybrid Multilingual Sentiment Classification Using Fuzzy Logic and Semantic Similarity". In : Ezziyyani M. (eds) Advanced Intelligent Systems for Sustainable Development (AI2SD'2018). AI2SD 2018. Advances in Intelligent Systems and Computing, vol 915. Springer, Cham, 2019.
- **Youness MADANI**, Mohammed ERRITALI and Jamaa BENGOURRAM, "Classification of Social Network Data Using a Dictionary-Based Approach", In : Zbakh M., Essaaidi M., Manneback P., Rong C. (eds) Cloud Computing and Big Data : Technologies, Applications and Security. CloudTech 2017. Lecture Notes in Networks and Systems, vol 49. Springer, Cham.
- **Youness MADANI**, Mohammed ERRITALI and Jamaa BENGOURRAM, "Arabic stemmer based big data", Journal of Electronic Commerce in Organizations (JECO), Volume 16, Number 1, pages 17-28, 2018.
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa, "Twitter Data Classification Using Big Data Technologies", In Proceedings of the 2018 International Conference on Internet and e-Business, pp 124-129, ACM, April 2018, DOI : 10.1145/3230348.3230368.
- **Youness MADANI**, Jamaa BENGOURRAM, Mohammed ERRITALI, Badr HSSINA and Marouane Birjali, "Adaptive e-learning using Genetic Algorithm and Sentiments Analysis in a Big Data System", International Journal of Advanced Computer Science and Applications(IJACSA), volume 8, issue 8, 2017,
  DOI : http ://dx.doi.org/10.14569/IJACSA.2017.080851.
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa, "A parallel semantic sentiment analysis", In 3rd international conference of cloud computing

technologies and applications (CloudTech), Singapore, Singapore, pp 1-6, IEEE, 2017, DOI : 10.1109/CloudTech.2017.8284714.

- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa, "Social login and data storage in the big data file system HDFS", In proceedings of the International Conference on Compute and Data Analysis, Lakeland, FL, USA, pp 91-97, ACM, DOI : 10.1145/3093241.3093265.

Liste des Communications

- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The **international Conference on Advanced Intelligent Systems for Sustainable Development** AI2SD'2019, 08-11 juillet 2019. Participation avec une Communication oral intitulée : "**Social Collaborative Filtering Approach for Recommending Courses in an E-learning Platform**".
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The 2019 **5th international conference on Business Intelligence** (CBI'19), 17-19 Avril 2019 à Béni Mellal, Maroc. Participation avec une Communication oral intitulée : "**A Recommender approach for an e-learning platform based on social network analysis and collaborative filtering**".
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The 2018 **6th international conference on networked systems** (NETYS 2018), 9-11 May 2018 à Essaouira, Maroc. Participation avec une Communication affichée(Poster) intitulée : "**A Hybrid Multilingual Sentiment Classification Using Fuzzy Logic and Semantic Similarity**".
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The 2018 **9th international conference on computer technologies and development** (ICCTD 2018), 24-26 Mars 2018 à Istanbul, Turquie. Participation avec une Communication oral intitulée : "**Classification of Tweets based on Big Data**".
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The 2018 4th **international conference on Business Intelligence** (CBI'18) ,25-27 Avril 2018 à Béni Mellal, Maroc. Participation avec une Communication oral intitulée : "**A hybrid Multilingual Sentiment Analysis Based on Fuzzy Logic and SentiWordNet**".
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The **3rd international Conference on Cloud Computing Technologies and Applications** – CloudTech'17, 24-26 Octobre, 2017, à Rabat, Maroc. Participation avec une

Communication oral intitulée : "**A parallel Semantic sentiment analysis**".

- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The 2017 **3th international conference on Business Intelligence** (CBI'17) ,29-31 Mars 2017 à Béni Mellal, Maroc. Participation avec une Communication oral intitulée : "**Arabic Stemmer Based Big Data**".
- **MADANI Youness**, Erritali Mohammed and Bengourram Jamaa. The 2017 **8th international conference on computer technologies and development** (ICCTD 2017), 20-22 Mars 2017 à Paris, France. Participation avec une Communication oral intitulée : "**Social Login and Data Storage in the Big Data File System HDFS**".

Résumé

Les réseaux sociaux connaissent une augmentation spectaculaire ces dernières années, les utilisateurs de ces plates-formes peuvent publier le contenu de leur choix sans entrave, ce qui génère une énorme quantité de données sous forme de tweets, commentaires, Likes(j'aime), publications sur Facebook, etc. Ces données peuvent être utilisées pour extraire des informations utiles telles que des sentiments, des opinions ou des motivations. L'analyse de sentiment (Sentiment Analysis SA) parfois appelée extraction d'opinions ou en particulier l'analyse des réseaux sociaux (Social Networks Analysis SNA) est un domaine qui consiste à analyser et à extraire des émotions ou des opinions à partir des critiques de produits, des critiques de films, des tweets, des publications sur Facebook ; ou à les classer en classes telles que positive, négative et neutre. Il était devenu un domaine de recherche scientifique très actif ces dernières années, notamment avec le développement des réseaux sociaux.

Dans cette thèse, nous utilisons les données provenant des réseaux sociaux (tweets de Twitter et publications de Facebook) pour les classer et les analyser. Nous décrivons comment extraire, stocker et analyser ces données à l'aide des technologies Big Data, telles que le système de fichiers distribué Hadoop **HDFS**, Apache **Flume**, Apache **Sqoop** et le modèle de programmation **MapReduce**. Nous présentons également comment préparer les données pour la classification en appliquant les méthodes de traitement automatique du langage naturel(NLP). L'objectif de notre travail est de proposer des nouvelles approches optimales pour analyser et classifier les données des réseaux sociaux en classes(positive, négative et neutre) d'une manière automatique. Les approches proposées reposent sur la similarité sémantique, les dictionnaires (AFINN, WordNet et SentiWordNet) et les concepts de la logique floue. Nous décrivons également comment nous pouvons utiliser nos approches pour proposer de nouvelles plateformes d'apprentissage en ligne. Notre travail est développé d'une manière parallèle en répartissant le travail entre plusieurs machines (cluster Hadoop) à l'aide du **HDFS** et **MapReduce**. Les approches proposées montrent leur efficacité par rapport à quelques méthodes de la littérature et aux méthodes basées sur l'apprentissage automatique.

Mots-clés–*Analyse des Réseau Sociaux, Analyse des Sentiments, Big Data, Traitement du langage Naturel, similarité sémantique, Apprentissage en ligne adaptatif, Systèmes de recommandation, Logique floue.*

Abstract

Social networks are experiencing a dramatic increase in recent years, users of these platforms can publish the content they want without hindrance, which generates a huge amount of data in the form of tweets, comments, likes, Facebook posts, etc. This data can be used in extracting useful information such as sentiments, opinions or motivations. Sentiment Analysis (SA), sometimes referred to as opinion mining or social network analysis (SNA) is a domain that analyses people's opinions, sentiments, evaluations, attitudes, and emotions from a written language ; or classifies them into classes such as positive, negative or neutral. It had become a very active area of scientific research in recent years, especially with the development of social networks.

In this thesis, we use data coming from social networks(tweets from Twitter and Facebook posts) for classifying and analyzing them. We describe how to extract, store and analyse this data using Big data technologies such as the Hadoop Distributed File System HDFS, Apache Flume, Apache Sqoop, and MapReduce programming model. We present also how to prepare data for the classification by applying the natural language processing methods. Our work aims to propose new optimal approaches for analyzing and classifying social networks data into classes(positive, negative or neutral) in an automatic way. The proposed approaches are based on semantic similarité, dictionaries(AFINN, WordNet and SentiWordNet) and fuzzy logic concepts. We describe also how we can use the proposed approaches for proposing new adaptive e-learning platforms. Our work is developed in a parallel way by distributing the work between several machines (Hadoop cluster) using the Hadoop Distributed File System HDFS and MapReduce programming model. The proposed approaches show their effectiveness compared to some methods of the literature and methods based on machine learning.

Keywords–*Social Networks Analysis, Sentiment Analysis, Big Data, Natural Language Processing NLP, Semantic Similarity, Adaptive e-learning, Recommendation Systems, Fuzzy Logic.*

Liste des acronymes

- **HDFS** : Hadoop Distributed File System
- **NLP** : Natural Language Processing
- **TP** : Text Preprocessing
- **POS** : Part of Speech Tagging
- **SA** : Sentiment Analysis
- **ML** : Machine Learning
- **SVM** : Support Vector Machine
- **NB** : Naive Bayes
- **API** : Application Programming Interface
- **CR** : Classification Rate
- **ER** : Error Rate
- **IRS** : Information Retrieval System
- **AG** : Algorithme génétique
- **PA** : Période d'activité
- **SRI** : Système de Recherche d'information
- **SR** : Systèmes de Recommandation
- **FS** : Filtrage Social
- **FC** : Filtrage collaboratif
- **OP** : Objectif Pédagogique
- **MC** : classe majoritaire
- **KPP** : K-Plus Proches Voisins
- **FLS** : Fuzzy Logic System
- **MF** : Membership Function
- **FA** : Fonction d'appartenance
- **COG** : Center of Gravity
- **CVO** : Crisp Value of the Output

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Contexte du travail de recherche

Dans les dernières années et avec l'apparition des réseaux sociaux(Facebook, Twitter, Linkedin, etc.), beaucoup des données sont générées d'une manière explosive, cette énorme masse de données peut être utilisée dans plusieurs domaines, tels que la recommandation des produits et films, l'analyse des sentiments, ou même dans des systèmes d'apprentissage en ligne(e-learning). Le but de ce travail est de trouver une solution pour les problèmes liés à l'analyse des sentiments des données issues des réseaux sociaux.

Le domaine de recherche présenté dans cette thèse est **l'analyse des sentiments** (**Sentiment Analysis SA en anglais**). SA est un domaine qui consiste à extraire des sentiments ou opinions à partir d'un texte écrit en utilisant des techniques du traitement automatique du langage naturel(**Natural Language Processing NLP en anglais**). Récemment ce domaine est appliqué sur les données des réseaux sociaux en raison de la grande masse de données publiées chaque jour.

SA a connu un développement rapide ces dernières années, il est utilisé par des entreprises pour analyser les besoins de leurs clients, et aussi pour augmenter la qualité de leurs produits(la surveillance des opinions pour améliorer les produits ou l'étude de marché.), de même un consommateur peut par exemple utiliser SA pour analyser les opinions et les critiques des autres consommateurs avant d'acheter un produit. SA peut être utilisé aussi pour la prédiction des résultats d'une élection(explication des sondages des suffrages aux élections), pour la détection de spam, ou même pour recommander des cours dans une plateforme d'e-learning.

L'application des notions du SA sur les réseaux sociaux est connue sous le nom : "Social Networks Analysis SNA", analyse des réseaux sociaux en français. SA tire partie de la grande quantité des données publiées régulièrement dans ces réseaux sociaux et qui sont pleines d'information afin de l'utiliser dans des problèmes réels tels que : l'éducation, la recommandation, la détection des spams, les problèmes de classification, etc.

Défis et contributions de la thèse

Cette thèse propose des nouvelles approches et techniques pour l'analyse des sentiments sur les données issues des réseaux sociaux. Les approches proposées consistent à classifier des données comme les tweets de Twitter et les publications Facebook en classes telles que : positive, négative et neutre. La question qui se pose est comment nous pouvons classifier ces données d'une manière automatique et optimale afin d'augmenter le taux de classification et diminuer le taux d'erreur. Beaucoup des approches ont été proposée pour classifier les publications Facebook ou les tweets selon des classes(positive, négative, neutre), ces approches sont soit basées sur des algorithmes d'apprentissage automatique(Machine learning), soit basées sur des lexiques, ou sur des méthodes hybrides(qui combinent les deux méthodes précédentes).

Dans des réseaux sociaux comme Twitter ou Facebook, les gens peuvent exprimer et publier ce qu'ils veulent d'une manière libre sans entrave, ce qui produit une grande masse de données, donc la nécessité d'un volume important pour stocker ces données, sans oublier le temps de calcul nécessaire pour avoir le résultat d'analyse. Une des solutions possible pour remédier à ce défi, est la distribution et le partage du stockage des données entre un nombre de machines en utilisant les technologies Big Data(Hadoop, Flume, Sqoop), avec le système de fichiers Hadoop distribué HDFS et le modèle de programmation MapReduce.

Les données issues des réseaux sociaux sont non structurées, elles contiennent des données non nécessaires(des bruits) pour l'analyse des sentiments. Des données comme les mots vides, les URLs, les chiffres et les entités des réseaux sociaux(hashtags) affectent négativement l'analyse et la classification des données des réseaux sociaux, ce qui nécessite un pré-traitement pour les rendre prêtes à l'analyse. Afin de traiter ce problème, nous utilisons les différentes méthodes du traitement automatique du langage naturel avec les techniques de pré-traitement du texte telles que : la suppression de mots vides, la suppression des URLs, stemming, étiquetage morpho-syntaxique, etc.

Si nous examinons de près le contenu des approches existantes qui traitent l'analyse des réseaux sociaux, nous constatons que la majorité d'entre elles ont des problèmes liés à la gestion du langage naturel (langage écrit) et de ses complications telles que l'ambiguïté des mots, la négation, la complexité, etc.

L'objectif principal de notre thèse est d'utiliser les données extraites à partir du Twitter et Facebook pour extraire des informations utiles comme la motivation des apprenants d'une plateforme d'e-learning, les opinions des clients sur un produit, ou les sentiments et attitudes des gens dans une période donnée. Notre travail consiste à améliorer les techniques d'analyse des données des réseaux sociaux en proposant des nouvelles ap-

proches basées sur les technologies Big Data, les techniques de pré-traitement du texte, les dictionnaires(AFINN, WordNet, SentiWordNet), les notions des systèmes de recherche d'informations, la similarité sémantique, et les concepts de la logique floue.

Les contributions de cette thèse concernent principalement les points suivants :

- La façon d'extraire les données à partir des réseaux sociaux d'une manière optimale.
- La parallélisation de nos approches en distribuant/partageant le stockage des données extraites, le travail d'analyse et de la classification entre un nombre de machines(Cluster Hadoop) afin d'optimiser le volume de stockage et le temps de calcul.
- L'étude de l'effet du pré-traitement des données des réseaux sociaux sur la qualité de la classification. Nous utilisons dans ce point les méthodes NLP avec les différentes méthodes du pré-traitement du texte.
- L'implémentation d'une nouvelle approche de l'analyse des sentiments basée sur un dictionnaire, et son application pour la proposition d'une nouvelle plateforme d'apprentissage en ligne en utilisant un algorithme d'optimisation.
- La proposition de deux approches basées sur la similarité sémantique, et l'implémentation d'un nouveau système de recommandation pour recommander les cours dans une plateforme d'e-learning.
- L'utilisation de la logique floue dans l'analyse des sentiments pour proposer deux nouvelles approches hybrides.

Organisation de la thèse

Notre thèse traite l'analyse des données des réseaux sociaux dans un environnement Big Data(parallélisation). Elle contient deux parties :

La ***première partie*** décrit le contexte général de notre thèse et comment nous pouvons régler les problèmes liés à la complexité et le langage naturel. Elle présente également une étude bibliographique des méthodes existantes. Elle est organisée comme suit :

- Le **premier chapitre** introduit les concepts des technologies Big Data : le framework Hadoop, le système de fichiers distribué Hadoop(Hadoop Distributed File System HDFS), Apache Flume, Apache Sqoop et le modèle de programmation MapReduce. Nous présentons la manière d'extraire les données à partir du Twitter et Facebook, et comment nous les stockons d'une manière distribuée(distribué le stockage entre plusieurs machines Hadoop) à l'aide du HDFS, Flume et Sqoop.
- Le **deuxième chapitre** présente les différentes méthodes du traitement automatique du langage naturel (Natural Language Processing NLP), et comment les utiliser pour rendre les données des réseaux sociaux(données non structurées) prêtes

pour la classification et l'analyse.

- Le **troisième chapitre** donne une présentation générale du domaine d'analyse des sentiments, ses types et les différentes approches existantes. Nous présentons aussi comment nous pouvons rendre nos approches multilingues.
- Dans le **quatrième chapitre**, nous réalisons une étude bibliographique des travaux existants dans la littérature et qui traitent le sujet de l'analyse des sentiments. Pour cela, dans un premier niveau nous présentons les travaux qui ont utilisé les technologies Big Data dans SA, dans un deuxième niveau nous présentons les travaux qui étudient le rôle du NLP dans SA, et finalement nous présentons les approches SA existantes selon trois catégories(approches basées sur le lexique, approches basées sur l'apprentissage automatique et approches hybrides).

La ***deuxième partie*** présente nos contributions, avec les discussions et les résultats obtenus. Elle est organisée comme suit :

- Le **premier chapitre** commence la présentation de nos contributions. Nous proposons une nouvelle approche basée sur le lexique pour classifier les tweets en trois classes(positive, négative et neutre), en utilisant les dictionnaires AFINN et WordNet enrichis par la sémantique. Nous appliquons l'approche proposée pour implémenter une nouvelle plateforme d'e-learning basée sur un algorithme génétique.
- Le **deuxième chapitre** traite la deuxième contribution de notre travail, où nous proposons deux approches. La première approche consiste à classifier les tweets en 3 classes(positive, négative ou neutre), elle est basée sur le calcul de la similarité sémantique entre le tweet à classer et trois documents d'opinion. La deuxième approche classifie les tweets en deux classes (positive et négative) par le calcul de la similarité sémantique entre le tweet et les mots "**positive**" et "**négative**". Les deux approches sont développées d'une manière parallèle en utilisant HDFS et MapReduce. La première approche est utilisée pour la proposition d'une nouvelle méthode de recommandation pour recommander des cours dans une plateforme d'e-learning.
- Dans **le troisième chapitre**, nous proposons deux nouvelles approches hybrides en se basant sur les concepts de la logique floue, les notions des systèmes de recherche d'informations, la similarité sémantique et le dictionnaire **SentiWordNet**.

Enfin, nous finalisons cette thèse par une conclusion générale, où nous reprenons l'ensemble des contributions de ce travail avec les résultats obtenus. De plus, nous identifions quelques perspectives.

Première partie

Préliminaires et état de l'art

Chapitre 1

Big Data : Extraction et stockage des données.

Comme présenté dans le titre de notre thèse, notre travail est basé sur les données issues des réseaux sociaux. Une étape très importante consiste à extraire les données sur laquelle on va travailler, et aussi trouver la manière optimale de les stocker. Ce chapitre présentera un aperçu sur les technologies Big Data utilisées dans notre travail, les différentes étapes suivies pour construire notre base de données(les données des réseaux sociaux comme Facebook et Twitter), et comment nous stockons les données extraites d'une manière distribuée dans le système de fichiers Hadoop distribué(Hadoop Distributed File System HDFS).

1.1 Big Data

1.1.1 Définition

Le terme Big data est utilisé pour désigner une collection des données qui est large, volumineux et complexe. Cette grande quantité de données devient difficile pour la stocker et l'analyser en utilisant les bases de données traditionnelles (MySQL, Oracle...). Le challenge autour du Big data inclue des problèmes liés au stockage d'une grande masse de données, et aussi comment analyser et visualiser ces données.

1.1.2 Les caractéristiques Du Big Data :

Big Data possède 5 caractéristiques que les experts ont appelées 5V : volume, vélocité, variété, véracité et valeur.

- **Volume :** le volume représente la quantité des données qui augmente chaque jour et d'une manière très rapide. Par exemple, la quantité des données générées par les êtres humains, les machines et leurs interactions dans les réseaux sociaux est massive. Les chercheurs ont estimé que 40 zettabytes des données vont générer dans 2020, ce qui représente une augmentation de 300 fois que 2005.

- **Vélocité :** la vélocité est définie comme la vitesse avec laquelle les données générées et se déplacées chaque jour, ce flux de données est massif et continu. Pensez juste aux nombres des messages et tweets publiés chaque seconde dans les réseaux sociaux, les transactions bancaires et aussi le nombre d'utilisateurs des ordinateurs et de smartphones qui génèrent une grande masse de données avec l'utilisation de l'Internet. Big data donne aux entreprises la possibilité d'analyser les données dès lors création(analyse en temps réel), ce qui n'est pas le cas pour les technologies traditionnelles(bases de données relationnelles).
- **Variété :** comme il existe beaucoup de sources qui ont contribué dans le monde de big data, les types de données générées sont différents, ils peuvent être structurés, semi-structurés ou non structurés, et aussi sous différents formats(images, fichiers texte, vidéo, etc). Ces types de données requièrent un pré-traitement supplémentaire pour dégager les informations voulues.
- **Véracité :** avec la diversité du format des données de big data, la qualité et la précision deviennent difficile à contrôler comme les messages de Twitter avec hashtags, abréviations, fautes de frappe et discours familier. Le volume est souvent la raison du manque de qualité et de précision des données. La véracité fait référence à l'incertitude des données disponibles en raison de leur incohérence et de leur caractère incomplet.
- **Valeur :** après que nous discutions le volume, la vélocité, la variété et la véracité, il y a un autre V que nous devons pris en compte quand nous parlons du Big data, c'est la valeur. Cette V fait référence à la manière dont les organismes (entreprises, chercheurs) peuvent utiliser des grandes masses de données pour extraire des informations utiles. C'est très bien d'avoir accès au Big Data, mais à moins que nous puissions le transformer en valeur, il est inutile.

1.1.3 Les types de Big Data :

Big Data peut avoir trois formes :

- **Structurée :** les données qui peuvent être stockées et traitées dans un format fixe sont appelées **données structurées**. Les données stockées dans les systèmes de gestion des bases de données relationnelles (SGBDR) est un exemple des données structurées, le langage de requête structuré (Structured Query Language SQL) est souvent utilisé pour gérer ce type de données.
- **Semi-structurée :** les données semi-structurées sont un type de données qui n'a pas de structure formelle d'un modèle de données. On peut définir les données semi-structurées comme un type de données contenant des balises sémantiques, mais non

conformes à la structure associée à des bases de données relationnelles classiques. Bien que les entités semi-structurées appartiennent à la même classe, elles peuvent avoir des attributs différents. Exemples : messagerie électronique, XML et autres langages de balisage.

– **Non structurée :** les données dont la forme est inconnue et qui ne peuvent pas être stockées dans le SGBDR et ne peuvent être analysées que si elles sont transformées en un format structuré sont appelées données non structurées. Les fichiers texte et les contenus multimédias tels qu'images, audios et vidéos sont des exemples de données non structurées. ces données croissent plus rapidement que d'autres, les experts disent que 80% des données d'une organisation ne sont pas structurées.

1.1.4 Technologies Big Data :

Après que nous définissions Big data, ses caractéristiques et ses types. Nous présenterons dans ce qui suit les technologies que nous avons utilisées dans notre travail pour extraire, stocker et analyser la grande quantité des données issues des réseaux sociaux.

Le monde du big data parle son propre langage. Pour stocker et analyser des données non structurées, nous devons utiliser les systèmes de bases de données NoSQL. Les bases de données NoSQL sont spécialisées dans le stockage de données non structurées, et ils offrent des performances rapides. Les bases de données NoSQL les plus populaires incluent MongoDB, Redis, Cassandra, Couchbase et bien d'autres.

Cloud Computing est un autre type de technologies pour faire face au Big Data. il s'agit de manipuler, configurer et accéder à distance aux ressources matérielles et logicielles. Il offre un stockage de données en ligne, une infrastructure et des applications.

Dans notre thèse, nous avons utilisé l'écosystème **Hadoop**, pour extraire, stocker et analyser les données des réseaux sociaux.

1.2 Hadoop Ecosystem

Hadoop est la principale plateforme du Big Data utilisée pour le stockage et le traitement d'immenses volumes de données.

L'écosystème Hadoop fait référence aux divers composants de la bibliothèque de logiciels Apache Hadoop, ainsi qu'aux accessoires et outils fournis par "Apache Software Foundation" pour ces types de projets logiciels et à la manière dont ils fonctionnent ensemble. Hadoop est un framework java extrêmement populaire pour la gestion et l'analyse de grands ensembles de données(Big Data).

Hadoop est un framework logiciel open source, c'est une structure qui permet le traitement distribué de grands ensembles de données sur des grappes d'ordinateurs à l'aide des modèles de programmation simples. Il est conçu pour passer de serveurs uniques à des milliers de machines, chacune offrant un calcul et un stockage locaux.

A la base, le framework Hadoop comporte deux principales couches, à savoir :

- Couche de traitement/calcul basée sur le modèle de programmation MapReduce.
- Couche de stockage basée sur le système de fichiers distribué Hadoop(HDFS).

1.2.1 Système de Fichiers Distribué Hadoop : HDFS

HDFS est un système de fichiers distribué destiné pour le stockage des masses de données volumineuses en partageant le stockage entre une grappe des machines (cluster Hadoop). HDFS et la couche de stockage principale pour l'écosystème Hadoop.

Le système de fichiers distribué Hadoop a été développé à l'aide d'une conception du système de fichiers distribuée. Contrairement aux autres systèmes distribués, HDFS est hautement tolérant aux pannes et conçu avec un matériel à faible coût. Pour stocker des données aussi volumineuses, les fichiers sont stockés sur plusieurs ordinateurs. Ces fichiers sont stockés d'une manière redondante pour empêcher le système de perdre des données en cas de panne. HDFS rend également les applications disponibles pour le traitement parallèle.

Caractéristiques de HDFS

- **Tolérance aux Pannes :** HDFS est hautement **tolérant aux pannes**. Le système de fichiers réplique ou copie chaque donnée à plusieurs reprises et distribue les copies sur des nœuds individuels (les machines du cluster hadoop). Par conséquent, les données sur les nœuds qui tombent en panne peuvent être trouvées, cela garantit que le traitement peut continuer pendant la récupération des données.
- Il convient au stockage et au traitement distribué.
- HDFS fournit des autorisations de fichiers et d'authentification.
- **Grande fiabilité :** HDFS stocke les données d'une manière fiable sur un cluster, il divise les données en blocs. Le framework Hadoop stocke ces blocs sur les nœuds présents dans le cluster. HDFS réplique chaque bloc présent dans le cluster, par conséquent, il fournit une fonction de tolérance aux pannes. Si le nœud du cluster contenant des données tombe en panne, un utilisateur peut facilement accéder à ces données à partir des autres nœuds, pour cela, HDFS crée par défaut 3 répliques de chaque bloc contenant des données présentes dans les nœuds. Ainsi, les données

sont rapidement disponibles pour les utilisateurs, ce qui permet de les aider à éviter le problème de la perte de données.

- **Réplication** : la réplication de données est une fonctionnalité unique de HDFS. La réplication résout le problème de la perte de données dans des conditions défavorables, telles qu'une panne matérielle, un crash de nœuds, etc. HDFS maintient le processus de réplication à un intervalle de temps régulier. Il continue également à créer des répliques de données utilisateur sur différentes machines présentes dans le cluster. Ainsi, lorsqu'un nœud tombe en panne, l'utilisateur peut accéder aux données d'autres machines, en évitant la possibilité de perdre des données.
- HDFS suit l'architecture maître-esclave et comporte les éléments suivants :
  - **Namenode :** le namenode est le matériel standard qui contient le système d'exploitation GNU/Linux et le logiciel namenode. C'est un logiciel qui peut être exécuté sur du matériel standard. Le namenode est également appelé le maître. Il est tellement essentiel pour HDFS et lorsqu'il est désactivé, le cluster HDFS/Hadoop est inaccessible et considéré comme inactif. Le maître ne stocke pas les données réelles(l'ensemble de données à analyser). Les données elles-mêmes sont réellement stockées dans les datanodes. Il connaît la liste des blocs et son emplacement pour tout fichier donné dans HDFS. Avec ces informations, le namenode sait comment construire le fichier à partir des blocs.
  - **Datanode :** le datanode est un matériel standard doté du système d'exploitation GNU/Linux et du logiciel datanode. Le datanode est également appelé l'esclave. Pour chaque nœud (matériel/système) dans un cluster, il y aura un noeud de données(datanode). Ces nœuds gèrent le stockage des données, et effectuent les opérations demandées par le namenode. Lorsqu'un datanode est en panne, cela n'affecte pas la disponibilité des données ni du cluster, c'est le namenode qui organisera la réplication des blocs gérés par le datanode non disponible.

1.2.2 Modèle de Programmation MapReduce

MapReduce est une technique de traitement et un modèle de programmation pour l'analyse distribuée basée sur Java. MapReduce est un composant central du framework logiciel Apache Hadoop, qui permet de traiter de grands volumes de données en parallèle en divisant le travail en un ensemble de tâches indépendantes.

MapReduce divise le travail en petites parties, chacune pouvant être effectuée en parallèle sur le cluster de serveurs. Un problème est divisé en un grand nombre de problèmes plus petits, chacun étant traité pour donner des sorties individuelles. Ces sorties individuelles sont ensuite traitées pour donner une sortie finale. Hadoop MapReduce est évolutif

et peut également être utilisé sur de nombreux ordinateurs. De nombreuses petites machines peuvent être utilisées pour traiter des tâches qui ne pourraient pas être traitées par une grosse machine.

L'algorithme MapReduce contient deux tâches importantes, à savoir **Map** et **Reduce** :

- **Map** : le travail de la fonction mapper consiste à traiter les données d'entrée. Généralement, les données d'entrée se présentent sous la forme d'un fichier ou d'un répertoire et sont stockées dans le système de fichiers Hadoop (HDFS). Le fichier d'entrée est transmis ligne par ligne à la fonction mapper. La fonction mapper traite les données et crée plusieurs petits morceaux de données. Elle prend un ensemble de données et le convertit en un autre ensemble de données, dans lequel les éléments individuels sont décomposés en paires clé-valeur.
- **Reduce** : cette phase est la combinaison de la phase de shuffle et de la phase de reducer. Le travail du reducer consiste à traiter les données provenant du mapper. Après le traitement, il génère un nouvel ensemble de sorties qui seront stockées dans le HDFS. La tâche Reducer prend la sortie du mapper en tant qu'entrée et combine les données (paires clé-valeur) en un ensemble plus petit.

Notre travail est basé sur le Framework Hadoop, pour distribuer le stockage des données issues des réseaux sociaux en se basant sur HDFS, et pour paralléliser la classification et l'analyse avec le modèle de programmation MapReduce. La question qui se pose après toutes ces définitions est : comment nous pouvons extraire les données et les stocker en HDFS pour possible analyse(extraction des informations utiles) ?

1.3 Notre travail : Extraction de données et construction de notre base de données

Comme nous avons présenté précédemment, notre travail est basé sur les données issues des réseaux sociaux, à savoir : les publications et les commentaires Facebook, tweets, etc. La première étape consiste à extraire ces données pour la construction de notre base de données du travail.

Dans cette étape, nous utilisons deux API java : restFB et Twitter4J, et apache Flume.

1.3.1 RestFB API

RestFB est un simple client Facebook Graph API écrit en Java, c'est un logiciel open source. Cet API permet de récupérer et de modifier des informations Facebook, telles que le téléchargement de photos, la rédaction de commentaires, l'obtention d'une liste

d'amis. Elle est basée sur HTTP et permet à tous les langages informatiques ayant une bibliothèque HTTP de fonctionner avec, y compris Java [1].

Nous utilisons cette API pour extraire des publications et des commentaires à partir du Facebook. RestFb nous donne la possibilité de récupérer des publications/commentaires par mot-clé et dans une date prédéfinie. Par exemple, récupérer toutes les publications ou commentaires liés au mot "Iphone" et publiés entre le premier janvier 2017 et le premier janvier 2019.

1.3.2 Twitter4J API

Twitter4J [2] est une bibliothèque Java open source, qui fournit une API pratique pour accéder à Twitter. avec Twitter4J nous pouvons : poster un tweet, obtenir la chronologie d'un utilisateur, avec une liste des derniers tweets, envoyer et recevoir des messages directs, rechercher des tweets et bien plus encore.

Dans notre travail, Twitter4J nous aide pour extraire les tweets qui nous intéressent dans la période voulue. Par exemple, si nous voulons classifier des données en relation avec un produit pour avoir une idée sur les avis des consommateurs, cet API nous permet de récupérer des tweets à base du nom de produit et dans une date prédéfinie(ex : récupérer tous les tweets sur le produit "Samsung" qui sont publiés entre le premier janvier 2017 et le premier janvier 2019).

1.3.3 Apache Flume

Flume [3] est un outil hautement distribué, fiable et configurable. Apache Flume est un outil de stockage des données dans HDFS. Il collecte, agrège et transporte une grande quantité de données en continu, telles que des fichiers journaux, des événements provenant de diverses sources, comme le trafic réseau, les réseaux sociaux, les courriers électroniques ; vers HDFS. Flume est une solution extrêmement fiable et distribuée. Il est robuste et tolérant aux pannes avec des mécanismes de fiabilité ajustables et de nombreux mécanismes de basculement et de récupération. Il utilise un modèle de données extensible simple qui permet une application analytique en ligne.

Dans notre travail, Flume nous serve pour collecter des tweets en temps réel et les transformer directement vers HDFS. Avec Flume nous pouvons extraire une masse de données volumineuses et la stocker d'une manière distribuée dans Hadoop HDFS.

1. https ://restfb.com/
2. http ://twitter4j.org/
3. https ://flume.apache.org/

Dans les trois cas d'extraction soit avec RestFb, Twitter4J ou apache Flume, nous avons besoin de créer une application qui nous donne la permission d'interagir avec les réseaux sociaux Facebook et Twitter.

Pour extraire les publications et les commentaires Facebook, nous devons créer une application dans l'espace de développeurs Facebook (Facebook for developers [4]). Après la création de l'application, c'est l'étape de configuration dans laquelle nous créons des paramètres comme : **App ID** et **Secret Key**. Ces paramètres sont utilisés par RestFB API pour interconnecter avec Facebook et récupérer les données voulues.

Comme pour RestFB, l'utilisation de Twitter4J API et Flume se basent sur la création d'une application Twitter dans l'espace du développement [5] et ses 4 paramètres : Consumer Key, Consumer Secret, Access Token and Access Token Secret. Après cette configuration, Twitter4J et Flume peuvent connecter à Twitter pour récupérer les tweets.

1.4 Stockage des données

Le but principal de notre travail c'est de distribuer le stockage des données extraites entre plusieurs machines(cluster Hadoop), en utilisant le système de fichiers distribué Hadoop (HDFS). L'idée est de stocker les données récupérer dans la phase d'extraction en HDFS. Ce travail est divisé en deux cas.

1.4.1 Premier cas :

Si on utilise RestFB ou Twitter4J APIs dans l'étape d'extraction, les données sont stockées dans une simple base de données relationnelle(MySQL par exemple). Pour les transformer en HDFS nous utilisons Apache SQOOP.

Apache Sqoop [6] est un outil de l'écosystème Hadoop conçu pour transférer des données entre HDFS (stockage Hadoop) et des serveurs de bases de données relationnelles telles que MySQL, Oracle, SQLite, Teradata, Netezza, Postgres, etc. Apache Sqoop importe des données des bases de données relationnelles vers HDFS, et exporte les données de HDFS vers des bases de données relationnelles. Il transfère efficacement des données en vrac entre Hadoop et des magasins de données externes tels que des entrepôts de données d'entreprise, des bases de données relationnelles, etc.

Après que nous stockons les données issues des réseaux sociaux (publications/commentaires Facebook, tweets...) dans une base de données MySql, nous utilisons apache Sqoop pour

4. https ://developers.facebook.com/
5. https ://apps.twitter.com/
6. https ://sqoop.apache.org/

les transférer à en HDFS, afin de rendre le stockage distribué entre les différentes machines de notre cluster Hadoop.

1.4.2 Deuxième cas :

L'utilisation d'apache flume pour récupérer les tweets de Twitter ne nécessite pas apache Sqoop pour les transformer en HDFS, car flume stocke les données extraites directement d'une manière distribuée dans HDFS.

Figure I.1 présente les différentes étapes pour construire notre base de données de travail (extraction, transfert et stockage).

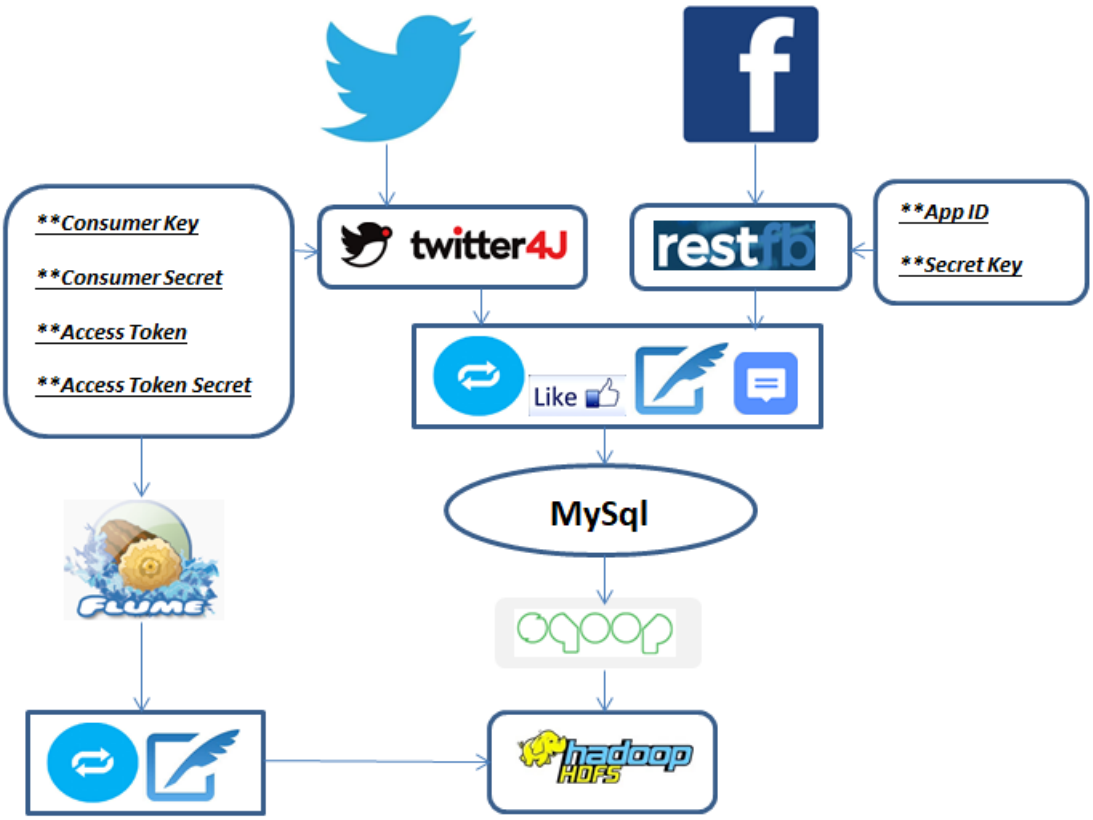


FIGURE I.1 – Étapes de Construction de notre base de données.

La partie de construction de notre base de données est une étape cruciale dans notre thèse. Comme présenté dans la figue I.1, nous utilisons des réseaux sociaux (Facebook, Twitter) pour extraire des données (publications/commentaires Facebook, tweets...etc), en utilisant des APIs java (RestFB et Twitter4J) ou la technologie Big data : *Flume*. Les données extraites sont transférées à HDFS pour les stocker d'une manière distribuée. Pour rendre les données extraites prêtes à l'analyse, nous devons les traiter en supprimant les données inutiles. Ce travail est fait par l'application des méthodes du traitement automatique du langage naturel (Natural Langage Processing NLP).

Chapitre 2

Traitement Automatique du Langage Naturel : Pré-traitement des données

Les données issues des réseaux sociaux sont non structurées, elles contiennent des données inutiles pour l'analyse comme des chiffres, des mots vides, des caractères spéciaux (#, @), le référencement à d'autres utilisateurs, etc. Pour rendre les données extraites utiles et prêtes pour l'analyse, nous appliquons des méthodes de pré-traitement du texte basées sur le domaine de "Natural Language Processing (NLP)" autrement appelé en français traitement automatique du langage naturel qui est une branche très importante du Machine Learning(apprentissage automatique) et donc de l'intelligence artificielle.

Afin de produire des informations significatives et exploitables à partir des données textuelles, il est important de se familiariser avec les techniques et les principes du traitement automatique du langage naturel.

2.1 Traitement Automatique du Langage Naturel (NLP)

Le langage naturel fait référence à un langage parlé et écrit par des personnes, et le traitement automatique du langage naturel ("Natural Language Processing (NLP)") tente d'extraire des informations du mot parlé et écrit en utilisant des algorithmes.

Natural Language Processing NLP est un sous-domaine de l'intelligence artificielle qui vise à permettre aux ordinateurs de comprendre et de traiter les langages humains. Les ordinateurs n'ont pas encore la même compréhension intuitive du langage naturel que les humains. Ils ne peuvent pas vraiment comprendre ce que la langue essaie vraiment de dire. En résumé, un ordinateur ne peut pas lire entre les lignes. Le traitement automatique du langage naturel est la technique utilisée pour aider les ordinateurs à comprendre le langage naturel humain.

NLP est un moyen pour les ordinateurs d'analyser, de comprendre et de tirer du sens à partir du langage humain d'une manière intelligente et utile. En utilisant NLP, les développeurs peuvent organiser et structurer leurs connaissances pour effectuer des tâches telles que le résumé automatique, la traduction, la reconnaissance d'entités nommées,

l'extraction de relations, l'analyse de sentiments et la reconnaissance vocale.

Le traitement automatique du langage naturel est caractérisé comme un problème difficile en informatique. Le langage humain est rarement précis ou clairement parlé. Comprendre le langage humain, c'est comprendre non seulement les mots, mais aussi les concepts et la manière dont ils sont reliés pour créer un sens.

2.2 Tâches importantes du NLP

2.2.1 Classification du texte

La classification du texte est l'un des problèmes classiques du NLP. Les exemples notoires comprennent : l'identification du courrier électronique indésirable, la classification des sentiments et l'organisation des pages Web par les moteurs de recherche.

La classification de texte, en termes courants, est définie comme une technique permettant de classifier systématiquement un objet de texte (document ou phrase) dans l'une des catégories fixe. Cela s'avère très utile lorsque la quantité de données est trop importante, notamment pour l'organisation, le filtrage des informations et le stockage.

Ce sous-domaine du NLP consiste par exemple à extraire des sentiments à partir d'un document(fichier texte). Dans la majorité des fois, il base sur les algorithmes d'apprentissage automatique (Machine Learning) afin de classifier le texte en classes(positive, négative...).

2.2.2 Similarité/Text Matching

l'objectif de *Text matching* ou similarité est de calculer le degré de similarité entre deux morceaux de texte en sens ou en proximité de surface. Le premier est appelé similarité sémantique et le second, similarité lexicale.

Le traitement automatique du langage naturel est utilisé pour indexer les morceaux du texte à comparer et les rendre prêts pour l'utilisation. Le calcul de la similarité devient dans les dernières années un domaine de recherche très actif, il est utilisé dans des nombreux domaines, à savoir :

- Les systèmes de recherche d'informations (SRI) : les SRIs consistent à calculer la similarité entre une requête utilisateur et un corpus des documents textuels, l'objectif est de trouver les documents pertinents à la requête. Un exemple de ces systèmes est les moteurs de recherche.
- Les systèmes de détection du plagiat (SDP) : la détection du plagiat est un domaine qui permet de trouver les documents suspects à partir d'un document source. Elle

est utilisée fréquemment dans le monde de la recherche scientifique par les journaux et les revues.

2.2.3 Résolution de la Coréférence

La résolution de la coréférence est la tâche de déterminer les expressions linguistiques faisant référence à la même entité du monde réel en langage naturel[63].

La résolution de la coréférence est un processus de recherche de liens relationnels entre les mots dans les phrases. Prenons un exemple de phrase : "Donald se rendit au bureau de John pour voir la nouvelle table. Il l'a regardé pendant une heure.

Les humains peuvent rapidement comprendre que "il" désigne Donald (et non John) et que "l'" désigne la table (et non le bureau de John). La résolution de la coréférence est la composante du NLP qui effectue ce travail automatiquement. Il est utilisé dans la synthèse des documents, la réponse aux questions et l'extraction d'informations.

2.2.4 Autres tâches du NLP

- **Résumé du texte (Text Summarization) :** le résumé du texte consiste à raccourcir un document texte afin de créer un résumé des principaux éléments du document original. L'idée principale du résumé est de trouver un sous-ensemble de données contenant les informations de l'ensemble complet. De telles techniques sont largement utilisées dans l'industrie aujourd'hui. Les moteurs de recherche sont un exemple ; d'autres incluent la synthèse de documents. La synthèse de documents tente de créer un résumé représentatif ou un résumé de l'ensemble du document en recherchant les phrases les plus informatives.
- **Traduction automatique (Machine translation MT) :** traduisez automatiquement le texte d'un langage humain à un autre en prenant soin de la grammaire, de la sémantique et de l'information sur le monde réel. MT est la traduction de texte par un ordinateur, sans implication humaine. La traduction automatique (MT) désigne un logiciel entièrement automatisé capable de traduire le contenu source dans les langues cibles. Les humains peuvent utiliser MT pour les aider à rendre le texte et la parole dans une autre langue.
- **Génération et compréhension du langage naturel (Natural Language Generation and Understanding) :** convertir des informations provenant des bases de données informatiques ou d'intentions sémantiques en langage humain lisible est appelé génération du langage naturel. La conversion de fragments de texte en structures plus logiques, plus faciles à manipuler pour les programmes informatiques, est

appelée compréhension du langage naturel.

- **Document à l'information(Document to Information) :** Il s'agit de transformer les données textuelles présentes dans des documents tels que : des sites Web, des fichiers PDF ou images, en un format propre et analysable.

2.3 Pré-traitement du texte (Text Preprocessing TP)

Le texte est la forme la plus non structurée de toutes les données disponibles, pour l'analyser d'une manière fiable et le rendre prêt à être exploité, il est nécessaire d'y appliquer un traitement préalable pour supprimer les différents types de bruit y sont présents. Ce processus complet de nettoyage et de normalisation, est appelé **Text Preprocessing TP** autrement appelé en français **pré-traitement du texte**.

TP est une tâche importante et une étape critique dans NLP, il permet d'appliquer un nombre de traitement sur un texte afin de le préparer pour une possible analyse. Les données issues des réseaux sociaux sont non structurées, ils contiennent beaucoup de données inutiles et qui influencent négativement les résultats d'analyse. L'application du pré-traitement du texte sur ces données est une étape fondamentale dans notre travail. Les lignes ci-dessous présentent les pré-traitements du texte appliqués :

2.3.1 Tokenization

Tokenization signifie diviser votre texte en unités significatives. C'est une étape obligatoire avant tout type de traitement.

La tokenization est un processus qui consiste à diviser le texte donné en unités appelées jetons (ou tokens). Les jetons peuvent être des mots, un nombre ou un signe de ponctuation. La tokenization effectue cette tâche en localisant les limites des mots. Le point final d'un mot et le début du mot suivant est appelé les limites des mots. Ce type de pré-traitement est également appelé segmentation de mots.

Les jetons résultants sont ensuite utilisés comme entrée pour d'autres types d'analyses ou de tâches, telles que l'analyse syntaxique (marquage automatique de la relation syntaxique entre les mots).

Ci-dessous un exemple qui présente comment appliquer la tokenization sur une phrase.

Phrase 1 : "Bonjour les amis, Bienvenue dans le monde du traitement automatique du langage naturel".

l'application de la toknezation sur la phrase 1 donne :

'Bonjour' 'les' 'amis' ',' 'Bienvenue' 'dans' 'le' 'monde' 'du' 'traitement' 'automatique' 'du' 'langage' 'naturel'

Le nombre total des jetons (tokens) est : 14.

2.3.2 Suppression du bruit

La suppression du bruit(Noise Removal) consiste à supprimer tout morceau de texte qui n'est pas pertinent, ou qui ne s'utilise pas dans l'étape d'analyse (n'affecte pas l'analyse). Il est basé sur plusieurs pré-traitements, à savoir :

Suppression des mots vides

L'une des principales formes de pré-traitement consiste à filtrer les données inutiles. Dans le traitement du langage naturel, les mots inutiles sont appelés des mots vides.

Les mots vides sont des mots qui sont filtrés lors de l'étape de pré-traitement. Ces mots sont, par exemple, les pronoms, les articles, etc. Il est important d'éviter d'avoir ces mots dans le modèle de classification, car ils peuvent conduire à une classification moins précise.

La raison pour laquelle les mots vides doivent être supprimés d'un texte est qu'ils le rendent plus lourd et moins important pour les analystes. La suppression des mots vides réduit la dimensionnalité de l'espace des termes. Les mots les plus courants dans les documents texte sont : les articles, les prépositions, les pronoms, etc, qui ne donnent pas la signification des documents, ces mots sont traités comme des mots vides. Les mots vides sont supprimés des documents car ils ne sont pas mesurés en tant que mots-clés dans les applications d'exploration du texte.

La liste ci-dessous représente des exemples des mots vides de la langue française.

['nous-mêmes', 'entre', 'vous', 'mais', 'encore', 'là', 'environ', 'dehors',' avec ','ils','posséder ','un','être','pour','son','votre','tel','dans' ,'ou', 'qui', 'comme', 'de', 'lui','chacun','le','jusqu'à','ci-dessous','nous','ceux-ci ',' votre ','par','ni', 'moi', 'elle', 'plus', 'lui-même', 'ceci','leur', 'alors', 'ci-dessus','le', 'à','elle', 'tous', 'non', 'quand', 'à', 'n'importe', 'avant' , 'eux', 'mêmes', 'et','sur','alors','que','parce que', 'quoi','ne',' pas ','maintenant','sous' , 'il', 'vous','a', 'juste', 'où', 'aussi', 'seulement','qui', 'ceux','après','si','contre', 'comment','ici']

suppression des chiffres

Les chiffres sont un autre type du bruit qui peut exister dans les données des réseaux sociaux. Il est important de les éliminer du texte à analyser, ce qui améliore la qualité d'analyse et de classification.

suppression des URLs

Dans la majorité des cas, les publications Facebook ou tweets peuvent contenir des URLs vers des pages web, ces URLs n'influencent pas la qualité d'analyse. Nous ne suivrons pas le contenu des liens Web, donc l'URL sera supprimé.

Suppression des entités des réseaux sociaux

Les tweets et les publications Facebook répondent à un ensemble de règles spécifiques au service des réseaux sociaux, avant de pouvoir l'analyser, nous allons nettoyer ces données de certaines de ces spécificités (#, @,. . .).

le symbole '@' est utilisé lorsque nous voulons baliser un autre utilisateur dans une publication. '@' et le nom de l'utilisateur ne sont pas nécessaires pour la classification ; nous décidons donc de les supprimer de la publication.

Le mot commençant par '#' est un hashtag. Un hashtag est différent des autres mots, il donne une étiquette ou un sujet sur la publication. Habituellement, le hashtag spécifie le sujet sur lequel les gens parlent dans la publication, et non sur les attitudes des gens. Ce mot peut fournir des informations mais ils ne sont pas importants. Nous avons donc décidé de ne pas supprimer le mot entier, mais simplement de supprimer le symbole '#' et de traiter le hashtag comme un mot normal dans la publication.

Suppression de la Ponctuation

Cette méthode de pré-traitement du texte permet de supprimer les caractères de ponctuation à partir des publications, afin de les rendre propres.

2.3.3 méthodes de normalisation lexique

La normalisation est un ensemble de règles visant à réduire toutes les occurrences d'une classe d'équivalence lexicale à leur forme canonique.

La normalisation est un processus qui convertit une liste des mots en une séquence plus uniforme. Ceci est utile pour préparer un texte pour un traitement ultérieur, en transformant les mots en un format standard. En général, elle signifie convertir un texte à une forme standard plus pratique.

Normaliser le texte peut signifier effectuer un certain nombre de tâches. Dans notre cadre de travail, nous aborderons la normalisation en 2 étapes distinctes : (1) la racinisation (Stemming) , (2) la lemmatisation (Lemmatization).

la racinisation et la lemmatisation sont les méthodes de traitement de texte de base. Le but de ces deux pré-traitements est de réduire les formes flexionnelles et les formes

parfois dérivées d'un mot à une forme de base commune.

Stemming

Stemming ou en français la racinisation est le processus de production des variantes morphologiques d'un mot, elle correspond à la partie du mot restante une fois que l'on a supprimé son (ses) préfixe(s) et suffixe(s), à savoir son radical. Un algorithme de racinisation pour la langue anglaise, réduit par exemple les mots "chocolates", "chocolatey", "choco" à la racine "chocolate" ; et "retrieval", "retrieved", "retrieves" à la racine "retrieve".

il existe différents algorithmes qui peuvent être utilisés dans le processus de stemming ou racinisation, mais le plus commun est **Porter stemmer**.

Les algorithmes de stemming fonctionnent en coupant la fin ou le début du mot, en prenant en compte une liste de préfixes et de suffixes communs pouvant être trouvés dans un mot infléchi. Cette coupe sans discernement peut être efficace dans certaines circonstances, mais pas toujours, et c'est pourquoi nous affirmons que cette approche présente certaines limites.

Lemmatisation

La lemmatisation, par contre, prend en compte l'analyse morphologique des mots. Pour ce faire, il est nécessaire de disposer des dictionnaires détaillés que l'algorithme peut consulter pour relier le mot à son lemme.

La lemmatisation est le processus de conversion des mots d'une phrase à sa forme de dictionnaire. Par exemple, étant donné les mots anglais : "amusement", "amusing", et "amused", le lemme de chacun serait "amuse".

Contrairement à la racinisation, la lemmatisation dépend de l'identification correcte de la partie voulue du langage et du sens d'un mot dans une phrase, ainsi que du contexte plus large entourant cette phrase, comme des phrases voisines ou même un document entier.

2.3.4 Normalisation d'objets (Object Standarization)

Si le texte contient des mots qui ne figurent dans aucun dictionnaire lexical classique, il va nous proposer un problème lors de l'analyse du texte. Les données des réseaux sociaux contiennent des mots spéciaux comme RT qui signifie Retweet en anglais, ces deux lettres sont souvent utilisées dans Twitter. Un autre exemple est awsm qui signifie awesome en anglais. La solution proposée est de préparer manuellement un dictionnaire qui contient deux éléments : le mot incorrect (RT) avec son équivalent (Retweet).

2.3.5 Étiquetage morpho-syntaxique

L'étiquetage morpho-syntaxique (aussi appelé étiquetage grammatical, POS tagging (part-of-speech tagging) en anglais) explique comment un mot est utilisé dans une phrase. Il y a huit parties principales de la partie du discours(POS tags) : noms, pronoms, adjectifs, verbes, adverbes, prépositions, conjonctions et interjections. L'étiquetage morpho-syntaxique est le processus d'attribuer un POS tags à un mot donné.

Comprendre les parties du discours (POS tags) peut faire la différence dans la détermination du sens d'une phrase. Une partie du discours (POS) nécessite souvent de regarder les mots précédents et suivants, et la combinée avec une méthode soit basées sur des règles soit stochastique. Elle peut aussi être combinée avec d'autres processus afin d'optimiser les résultats d'étiquetage.

Ci-dessous des exemples de POS tagging :

- Jouer ⇒ Tag : **Verbe, VRB**.
- Automatiquement ⇒ Tag : **Adverbe, RB**.
- Voiture ⇒ Tag : **Nom, NN**.
- Bleu ⇒ Tag : **Adjective, JJ**.
- Pour ⇒ Tag : **Preposition, IN**.

2.3.6 Caractéristiques statistiques

Un autre type de pré-traitement du texte fréquemment utilisé dans NLP est basé sur l'extraction des caractéristiques statistiques à partir du texte à analyser. Un exemple de ce type de traitement est la représentation vectorielle du texte en calculant le nombre d'apparition des mots(fréquence). La méthode la plus utilisée et basée sur l'utilisation du **TF-IDF** (de l'anglais term frequency-inverse document frequency), qui permet d'évaluer le degré d'importance d'un mot dans un document texte[103].

2.4 NLP et son utilisation dans notre Travail

Comme nous avons présenté précédemment, les données extraites dans la partie de construction de la base de données sont pleines de bruits(des données inutiles pour l'analyse). Tout ça rend la partie du traitement automatique du langage naturel très importante dans notre travail.

Après l'extraction et le stockage des données (publications/commentaires Facebook et tweets) dans HDFS, nous y appliquons les différentes méthodes de pré-traitement de texte, pour les rendre propres et prêtes pour l'analyse (extraction des informations utiles).

Le but de notre travail est de transformer les données non structurées extraites lors de la première étape en une forme compréhensible par les ordinateurs, en exploitant les avantages des techniques de la NLP, et de l'application des différentes méthodes de pré-traitement du texte. Notre travail est basé sur Apache OpenNLP library [1].

Apache OpenNLP est une bibliothèque Java open source. Il s'agit d'une boîte à outils d'apprentissage automatique (machine learning) pour le traitement de texte en langage naturel, qui prend en charge les tâches de NLP les plus courantes. Vous pouvez créer un service de traitement de texte efficace en utilisant cette bibliothèque. OpenNLP fournit des services tels que la tokenisation, la segmentation de phrases, POS tags, l'extraction d'entités nommées, l'analyse, la résolution de coréférence, etc.

2.4.1 Étapes du pré-traitement de texte dans notre travail

L'objectif derrière l'utilisation des réseaux sociaux, est la quantité des données publiées chaque jour, et aussi la richesse d'informations qu'ils contiennent. Afin de rendre les données extraites exploitables, nous y appliquons des traitements basés sur les tâches NLP.

Nous parcourons tous les publications/commentaires et tweets stockés dans HDFS, le premier traitement à faire est la tokénisation afin d'extraire les différents termes, en éliminant les espaces. Après, en utilisant un dictionnaire d'acronymes, nous élargissons les acronymes afin d'extraire les mots ou les phrases équivalentes, par exemple, l'acronyme "aws" devient "awesome". L'étape suivante est la suppression du bruit en appliquons les pré-traitements de : suppression des mots vides, suppression des chiffres, suppression des URLs, suppression des entités des réseaux sociaux et la suppression de la ponctuation.

Les mots résultants seront les entrées d'un algorithme de racinisation(stemming), pour extraire les racines des mots. L'algorithme utilisé est Porter stemmer[73].

L'étape suivante est l'application de l'étiquetage morpho-syntaxique(POS tagging) pour définir les types grammaticaux (verbe, nom, adjective, adverbe) des mots. Dans cette étape, nous utilisons l'hypothèse qui dit que seuls les verbes, les adverbes, les adjectifs et les noms peuvent exprimer des opinions et des sentiments et peuvent être des porteurs d'informations. À partir de cette hypothèse, nous supprimons tous les mots qui ne sont pas des verbes, des adverbes, des adjectifs ou des noms.

Figure I.2 présente les différentes étapes appliquées sur nos données pour les préparer à l'étape d'analyse.

1. https ://opennlp.apache.org/

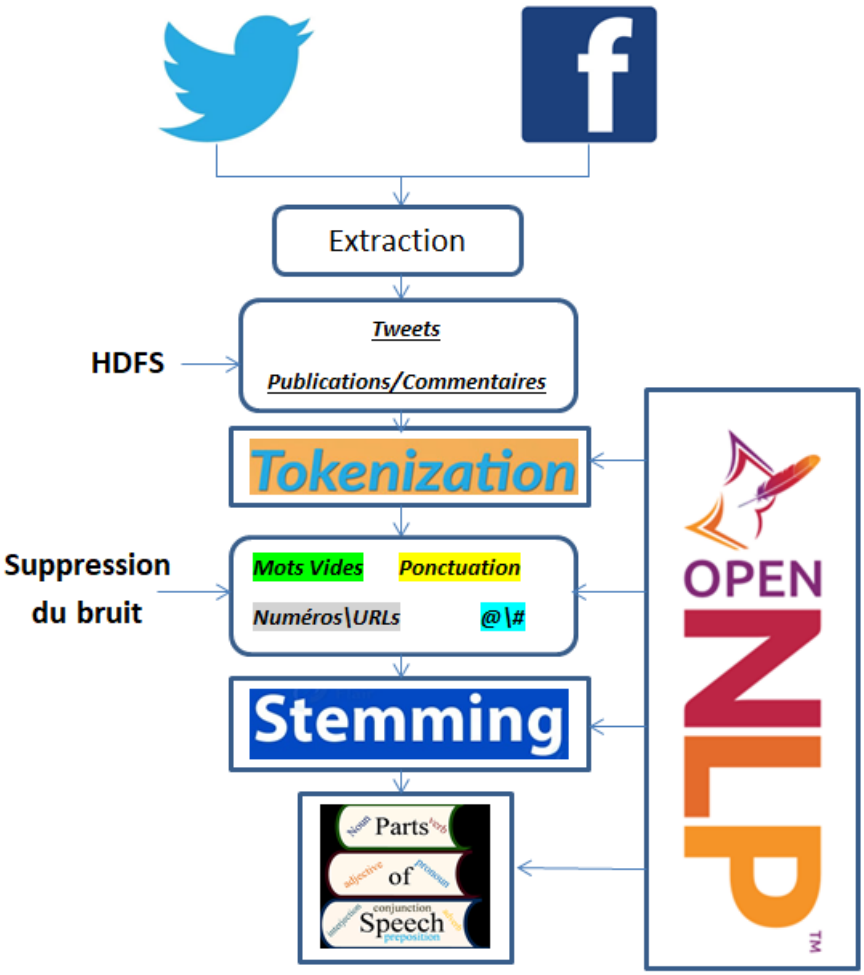


FIGURE I.2 – Étapes du pré-traitement du texte.

Après que les données extraites sont stockées d'une manière distribuée dans HDFS, et après l'application des différentes méthodes de pré-traitement de texte(tokénisation, suppression du bruit, stemming, POS tagging...), les données seront prêtes pour l'étape de l'analyse et de la classification. Notre thèse est basée sur la classification de ces données afin d'extraire des informations utiles comme les opinions, les sentiments et les attitudes des gens qui ont publié ces données dans les réseaux sociaux Facebook et Twitter. Ce type de classification est connu sous le nom **analyse des sentiments** ou **Sentiment Analysis** en anglais.

Chapitre 3

Analyse des sentiments et classification des données

Analyser les données issues des réseaux sociaux signifie les classifier (classer) en classes comme : positive, négative ou neutre. Cette classification sert à extraire des informations utiles comme les sentiments, les opinions ou la motivation. Ce type de classification a connu ces dernières années un grand développement dans la recherche scientifique et aussi dans l'industrie, surtout avec l'avènement du web 2.0 et les réseaux sociaux, ce qui génère des grandes masses de données. Cette disponibilité immédiate du texte a créé un nouveau domaine d'analyse de texte, c'est **Sentiment Analysis (SA)**, en français analyse des sentiments.

3.1 Qu'est-ce que l'analyse des sentiments ?

L'analyse des sentiments, également appelée exploration ou extraction d'opinion, est le domaine qui analyse les opinions, sentiments, évaluations, appréciations, attitudes et émotions des gens envers des entités telles que des produits, des services, organisations, individus, problèmes, événements, sujets et leurs attributs. Il existe également de nombreux noms et tâches légèrement différentes, par exemple, analyse de sentiment, extraction d'opinion, extraction de sentiment, analyse de subjectivité, analyse des effets, analyse des émotions.

L'analyse des sentiments est le processus permettant de déterminer si un écrit est positif, négatif ou neutre. Un système d'analyse des sentiments pour l'analyse du texte combine des techniques de traitement automatique du langage naturel (NLP) et d'apprentissage automatique afin d'attribuer des scores des sentiments pondérés aux entités, sujets, thèmes et catégories d'une phrase.

Depuis le début des années 2000, l'analyse des sentiments est devenue l'un des domaines de recherche les plus actifs en traitement automatique du langage naturel. Elle est aussi largement étudiée en data mining, web mining et text mining. En fait, elle s'est propagée depuis le domaine d'informatique aux sciences de gestion et aux sciences sociales

en raison de son importance pour les entreprises et la société dans son ensemble.

Avec la croissance explosive des médias sociaux (avis, discussions de forums, blogs, microblogs, Twitter, commentaires et publications sur des sites de réseaux sociaux) sur le Web, les particuliers et les organisations utilisent de plus en plus le contenu de ces médias pour la prise de décision. De nos jours, si on veut acheter un produit de consommation, on n'est plus obligé de demander les opinions aux amis et à la famille car il existe de nombreux avis d'utilisateurs et discussions sur des forums publics sur le web à propos du produit. Pour une organisation, il peut ne plus être nécessaire de mener des enquêtes, des sondages d'opinion, et des groupes de discussion afin de recueillir l'opinion publique parce qu'il y a une abondance de telles informations disponible publiquement.

De même, dans les débats politiques, par exemple, nous pourrions analyser les opinions des personnes sur certains candidats aux élections ou partis politiques et prévoir les résultats des élections à des postes politiques.

Nous devons toujours garder à l'esprit que l'analyse des sentiments est un problème du traitement du langage naturel (NLP). De nombreux problèmes de la NLP sont également des problèmes qui doivent être résolus lorsqu'on traite les problèmes d'analyse des sentiments.

3.1.1 Types d'analyse des sentiments

Il existe de nombreux types d'analyse des sentiments et d'outils, allant des systèmes qui se concentrent sur la polarité (positive, négative, neutre) aux systèmes permettant de détecter les sentiments et les émotions (en colère, heureux, triste, etc.) ou d'identifier les intentions (intéressés, non intéressés).

Parfois, vous pouvez aussi être intéressé à être plus précis sur le niveau de polarité de l'opinion. Par conséquent, au lieu de simplement parler d'opinions positives, neutres ou négatives, vous pouvez envisager les catégories suivantes : très positive, positive, neutre, négative, très négative. Cela pourrait être mappé sur une note de 5 étoiles dans un avis, par exemple : très positive = 5 étoiles et très négative = 1 étoile.

D'autre part, la détection des émotions vise à détecter les sentiments tels que le bonheur, la frustration, la colère, la tristesse, etc. De nombreux systèmes de détection d'émotions ont recours à des lexiques (c'est-à-dire des listes de mots et des émotions qu'ils véhiculent) ou à des algorithmes d'apprentissage automatique complexes.

3.1.2 Pourquoi l'analyse des sentiments est importante ?

On estime que 80% des données mondiales sont non structurées et non organisées de manière prédéfinie. La plupart de ces informations proviennent de données textuelles, telles que des e-mails, des tickets d'assistance, des chats, des médias sociaux, des enquêtes, des articles et des documents. Ces textes sont généralement difficiles, longs et coûteux à analyser, à comprendre et à trier.

Les systèmes d'analyse des sentiments permettent aux entreprises de comprendre cet océan de texte non structuré en automatisant les processus opérationnels, en obtenant des informations exploitables et en économisant des heures de traitement manuel des données, autrement dit en rendant le travail d'analyse plus efficace.

Le problème de l'analyse des sentiments comme pour tout problème scientifique, avant de le résoudre, nous devons le définir ou le formaliser. La formulation introduira les définitions de base, les concepts de base et problèmes, sous problèmes et objectifs visés.

3.2 Formulation du problème de l'analyse des sentiments

3.2.1 C'est quoi une opinion

Avant d'entrer dans les détails, donnons d'abord une définition de l'opinion. Les informations textuelles peuvent être classées en deux types principaux : les faits et les opinions. Les faits sont des expressions objectives sur quelque chose. Les opinions sont généralement des expressions subjectives qui décrivent les sentiments, les appréciations des gens à l'égard d'un sujet ou d'un produit.

Une opinion est un quintuple, $(E_i, A_{ij}, S_{ijkl}, H_k, T_l)$, où E_i est le nom d'une entité, A_{ij} est un aspect de E_i, S_{ijkl} est le sentiment sur l'aspect A_{ij} de l'entité E_i, H_k est le détenteur de l'opinion , et T_l est l'heure à laquelle l'opinion est exprimée par H_k. Le sentiment S_{ijkl} est positif, négatif ou neutre, ou est exprimé avec différents niveaux de force/intensité, par exemple 1 à 5 étoiles, comme utilisé par la plupart des avis sur le web. Lorsqu'un avis porte sur l'entité elle-même dans son ensemble, c'est l'aspect spécial GENERAL qui est utilisé pour le désigner. Ici, E_i et A_{ij} représentent ensemble la cible de l'opinion. Par exemple, dans la phrase "la qualité de l'appel d'IPhone est bonne, mais l'autonomie de la batterie est courte", évalue deux aspects : la qualité de l'appel et l'autonomie de la batterie, de l'IPhone (entité). Le sentiment sur la qualité d'appel de l'iPhone est positif, mais le sentiment sur l'autonomie de sa batterie est négatif.

Les opinions sont au cœur de presque toutes les activités humaines, car ce sont elles qui influencent le plus nos comportements. Chaque fois que nous voulons prendre une décision, nous devons connaître l'opinion des autres. Dans le monde réel, les entreprises et les organisations veulent toujours connaître l'opinion des consommateurs ou du public sur leurs produits et prestations de services. Les consommateurs individuels veulent aussi connaître les opinions existantes des utilisateurs sur un produit avant l'achat, et aussi les opinions des autres sur les candidats politiques avant de prendre une décision de vote dans une élection politique.

3.2.2 opinion régulière (directe) et comparative

Pour rendre les choses encore plus intéressantes et stimulantes, il existe deux types d'opinions, à savoir les opinions régulières et les opinions comparatives. Une opinion régulière n'exprime un sentiment que sur une entité en particulier ou sur un aspect de celle-ci[41], par exemple "le jus d'orange a très bon goût", ce qui exprime un sentiment positif sur l'aspect goût du jus d'orange. Une opinion comparative compare plusieurs entités en fonction de certains de leurs aspects communs, par exemple, "le jus d'orange a meilleur goût que le jus de pomme", qui compare le jus d'orange et de pomme en fonction de leurs goûts (un aspect) et exprime une préférence pour le jus d'orange.

Une phrase avec opinion est une phrase qui exprime des opinions positives ou négatives, explicites ou implicites. Ce peut être une phrase subjective ou objective.

3.2.3 subjectivité et objectivité dans l'analyse des sentiments

Pour effectuer une analyse des sentiments, nous devons d'abord identifier le texte subjectif et objectif. Une phrase objective présente des informations factuelles sur le monde. Un exemple d'une phrase objective est : "IPhone est un produit d'Apple". tandis qu'une phrase subjective exprime des sentiments, des points de vue ou des croyances personnels. Un exemple d'une phrase subjective est "J'aime IPhone.".

Les expressions subjectives se présentent sous de nombreuses formes, par exemple des opinions, des allégations, des désirs, des convictions, des suspicions et des spéculations.

Le fait de classifier une phrase comme subjective ou objective (exprime des informations factuelles ou des sentiments), est appelé **classification de subjectivité**. Les phrases subjectives peuvent être classées comme exprimant une opinion positive, négative ou neutre, ce type de classification est connu sous le nom de **classification de polarité**.

3.2.4 opinion explicite et implicite

Une opinion **explicite** est une opinion explicitement exprimée dans une phrase subjective. C'est une affirmation subjective qui donne une opinion régulière ou comparative[22]. Un exemple de ce type d'opinion est cité dans la phrase suivante : "Je n'aime pas cette caméra", dans cette phrase le sentiment est exprimé d'une manière explicite "je n'aime pas".

Une opinion **implicite** est impliquée dans une phrase objective. Une opinion implicite est une déclaration objective qui implique une opinion régulière ou comparative. Une telle déclaration objective exprime généralement un fait souhaitable ou indésirable. Un exemple est exprimé dans les deux phrases suivantes : "J'ai acheté un matelas il y a une semaine et une vallée s'est formée.", "Cette caméra est chère". Dans la deuxième phrase "chère" exprime une opinion négative dans une phrase qui est objective.

3.3 Différents niveaux d'analyse

Les chercheurs ont étudié l'extraction d'opinions à trois niveaux de granularité, à savoir le niveau de document, le niveau de phrase et le niveau d'aspect.

3.3.1 Niveau du document

SA au niveau du document(Document level) vise à classer un document d'opinion comme exprimant une opinion ou un sentiment positif ou négatif. Il considère le document entier comme une unité d'information de base (parlant d'un seul sujet). Dans ce niveau d'analyse, on estime que le document parle d'un seul sujet ou produit.

3.3.2 Niveau de la phrase

SA au niveau de la phrase(sentence level) vise à classer le sentiment exprimé dans chaque phrase. La première étape consiste à déterminer si la phrase est subjective ou objective. Si la phrase est subjective, SA déterminera si la phrase exprime des opinions positives ou négatives. Cependant, il n'y a pas de différence fondamentale entre la classification par niveau de document et par phrase, car les phrases ne sont que des documents courts.

Soit une phrase P, deux sous-tâches sont effectuées :

– Classification de subjectivité : déterminer si P est une phrase subjective ou une phrase objective.

– Classification du sentiment au niveau de la phrase : Si P est subjective, déterminez s'elle exprime une opinion positive ou négative.

3.3.3 Niveau d'entité et d'aspect

Les analyses au niveau du document et au niveau de la phrase ne découvrent pas exactement ce que les gens ont aimé ou n'ont pas aimé [109]. Un texte d'évaluation positive sur une entité particulière ne signifie pas que l'auteur a des opinions positives sur tous les aspects de l'entité. De même, un texte d'évaluation négative pour une entité ne signifie pas que l'auteur n'aime pas tout ce qui concerne l'entité. Par exemple, dans une évaluation de produit, le réviseur écrit généralement les aspects positifs et négatifs du produit.

L'analyse des sentiments basée sur les aspects-parfois appelée analyse des sentiments basée sur les fonctionnalités-vise à classer le sentiment par rapport aux aspects spécifiques des entités. La première étape consiste à identifier les entités et leurs aspects.

Les détenteurs d'opinion peuvent donner des opinions différentes sur différents aspects de la même entité, comme le dit la phrase suivante : "la qualité de la voix de ce téléphone n'est pas bonne, mais la durée de vie de la batterie est longue", cette phrase présente une opinion négative sur l'aspect "la qualité de la voix" et une opinion positive sur l'aspect "la durée de vie de la batterie" pour la même entité "le téléphone".

Ce niveau de classification (entity and aspect level) s'est composé de trois sous-tâches principales :

– identifier et extraire les entités des textes à évaluer (à classifier).
– identifier et extraire les aspects des entités.
– déterminer les polarités de sentiment sur les entités et les aspects des entités.

Par exemple, dans la phrase "J'ai acheté un appareil photo hier, et sa qualité d'image est excellente", l'entité est "l'appareil photo", et "la qualité d'image" représente l'aspect de l'entité. L'auteur de cette phrase exprime une opinion positive sur l'aspect.

3.4 Types d'approches d'analyse des sentiments

Dans la littérature beaucoup des approches d'analyse des sentiments ont été publiées, chaque approche vise à proposer une nouvelle méthode pour améliorer les résultats d'analyse et de classification [1]. ces approches peuvent être divisées en trois catégories.

3.4.1 Approche basée sur le lexique

ce type d'approche s'appuie sur un lexique de sentiments(Lexicon Based Approach), un ensemble des termes de sentiments connus et précompilés. Le lexique des sentiments est un ensemble de mots (ou d'expressions) qui transmet des sentiments. Chaque entrée du lexique est associée à son orientation de sentiment. Les entrées du lexique peuvent être divisées en trois catégories en fonction de l'orientation de leurs sentiments : positif, négatif ou neutre[47].

L'approche basée sur le lexique calcule la valeur finale du sentiment exprimé dans un avis ou une critique en notant la polarité de chaque mot ou expression dans un avis donné[90].

L'atout le plus important de cette technique est qu'elle ne nécessite aucune donnée d'apprentissage, tandis que son point le plus faible est qu'un grand nombre de mots et d'expressions ne sont pas inclus dans les lexiques de sentiment. Cette technique est divisé en une **approche basée sur les dictionnaires** et une **approche basée sur les corpus**.

- **Approche basée sur le dictionnaire(Dictionary based approach) :** la méthode basée sur le dictionnaire, trouve d'abord le mot d'opinion à partir du texte d'avis, puis trouve leurs synonymes et antonymes du dictionnaire. Le dictionnaire utilisé peut-être WordNet[62] ou SentiWordNet[4] ou autres.

  L'une des techniques simples de cette approche consiste à utiliser une petite série de mots d'opinion initiales et un dictionnaire en ligne, par exemple WordNet. La stratégie consiste d'abord à collecter manuellement un petit groupe de mots d'opinion avec des orientations connues, puis à l'agrandir en recherchant leurs synonymes et antonymes dans WordNet. Les mots récemment trouvés sont ajoutés à la liste, et la prochaine itération commence. Le processus itératif s'arrête quand on ne trouve plus de nouveaux mots[32].
- Approche basée sur le corpus (Corpus-based approach) : l'approche basée sur le corpus commence par une liste de départ de mots d'opinion pour trouver d'autres mots d'opinion dans un grand corpus, ce qui aide à trouver des mots avec des orientations spécifiques au contexte. Cela pourrait être fait en utilisant des méthodes statistiques ou sémantiques[29].

3.4.2 Approches d'apprentissage automatique ou systèmes automatiques

Les méthodes automatiques(Machine learning (ML)), contrairement aux systèmes basés sur les dictionnaires ou les corpus, ne reposent pas sur des règles conçues manuellement,

mais sur des techniques d'apprentissage automatique. La tâche d'analyse des sentiments est généralement modélisée comme un problème de classification dans lequel un classifieur est alimenté avec un texte et renvoie la catégorie correspondante, par ex. positif, négatif ou neutre. Cette approche utilise une technique d'apprentissage automatique et diverses fonctionnalités pour construire un classifieur capable d'identifier le texte exprimant un sentiment.

L'apprentissage automatique pour l'analyse des sentiments commence par la collecte d'un ensemble de données étiquetées. Ces données peuvent être bruyantes et doivent donc être pré-traitées en utilisant diverses techniques de traitement du langage naturel (NLP). Ensuite, il faut extraire les caractéristiques pertinentes pour l'analyse des sentiments et, enfin, former et tester le classifieur sur des données invisibles.

L'approche basée sur l'apprentissage automatique (ML) utilise plusieurs algorithmes d'apprentissage (algorithmes supervisés ou non supervisés)[70][8], tels que support vector machines, naïve bayésienne, arbre de décision(ID3, C4.5, CART), etc.

3.4.3 Approches hybrides

La combinaison d'apprentissage automatique et d'approches basées sur le lexique pour traiter l'analyse des sentiments est appelée hybride. Bien que peu utilisée, cette méthode produite généralement des résultats plus prometteurs que les approches mentionnées ci-dessus[65].

Afin d'améliorer les performances de la classification des sentiments, peu de techniques de recherche suggèrent d'utiliser une combinaison d'approches basées sur le lexique et d'apprentissage automatique. Le principal avantage de cette approche hybride est que nous pouvons atteindre le meilleur des deux mondes[74]. La combinaison lexique/apprentissage automatique s'est avérée améliorer la précision.

3.5 Notre travail : Analyse des réseaux sociaux

Après que nous avons fait un tour sur les points importants du domaine de l'analyse des sentiments, nous parlerons dans cette section sur l'application de ce domaine dans notre thèse. Comme présenté précédemment, notre travail consiste à analyser et classifier les données issues des réseaux sociaux (Facebook et Twitter) comme les publications Facebook, les commentaires, les tweets, etc. Ce type d'analyse est appelé analyse des réseaux sociaux, ce qui signifie l'application des différents aspects de l'analyse des sentiments sur les données des réseaux sociaux.

Les deux étapes qui précèdent l'étape d'analyse vont générer une base de données

des réseaux sociaux (étape d'extraction) qui est prête pour l'analyse et la classification (étape de pré-traitement du texte). Cette étape de classification (social networks analysis ou sentiment analysis) va nous aider à extraire des informations utiles comme la motivation en classifiant les données selon deux classes (motivé, démotivé), ou en analysant les sentiments et les opinions en calculant la polarité de chaque donnée (positive, négative ou neutre).

Avec l'analyse des données des réseaux sociaux, nous pouvons par exemple classifier les tweets ou/et les publications Facebook sur une période donnée pour connaître le sentiment d'un public particulier à propos d'un produit particulier (en analysant par exemple tous les tweets publiés sur un modèle de la caméra canon entre janvier 2017 et janvier 2019). Nous utilisons SA également pour prédire la motivation des apprenants dans une plateforme d'E-learning afin d'améliorer le processus d'apprentissage.

La question qui se pose après toutes ces informations et données que nous avons, est comment nous utiliserons le domaine d'analyse des sentiments afin d'extraire des informations utiles comme la motivation et les sentiments à partir des données textuelles(tweets-publications/commentaires Facebook). Nous verrons tous ça en détails dans les chapitres de la deuxième partie.

3.5.1 Le rôle de l'étiquetage morpho-syntaxique dans notre travail

Avant même de pouvoir analyser un tweet ou une publication pour extraire une information, vous devez toutefois comprendre les éléments qui la composent. L'étiquetage morpho-syntaxique (part of speech tagging) comme présenté dans le chapitre 2 permet de spécifier pour chaque mot d'une phrase son type grammatical(adjectif, adverbe, verbe, nom...). il a été démontré que les adjectifs sont des indicateurs d'opinion importants. Ainsi, certains chercheurs[100] ont traité les adjectifs comme des caractéristiques spéciales.

Dans nos travaux, nous nous utilisons une hypothèse qui dit que seules les adjectives, les adverbes et les verbes peuvent exprimer des opinions ou les avis des gens dans une unité de texte. Nous appelons ces trois types grammaticaux des **mots d'opinion (opinion words)**.

Les mots d'opinion sont des mots couramment utilisés pour exprimer des sentiments positifs ou négatifs. Par exemple, *belle*, *merveilleuse*, *bonne* et *étonnante* sont des mots d'opinion positive ; *mauvaise*, *pauvre* et *terrible* sont des mots d'opinion négative. Bien que de nombreux mots d'opinion soient des adjectifs et des adverbes ; les verbes (par exemple détester et aimer) peuvent également indiquer des opinions.

Afin d'extraire des mots d'opinion, nous avons utilisé la bibliothèque Apache OpenNLP

qui nous aide à définir le type de chaque mot(POS tagging) dans un tweet ou une publication/commentaire Facebook. notre idée est de ne conserver que les mots qui sont soit un verbe, un adverbe ou un adjectif.

3.5.2 Négation

Une autre tâche qui joue un rôle important dans la phase de classification, est l'étape de pré-traitement de négation. Faire face à la négation (comme "pas bon") est une étape critique de l'analyse des sentiments. Un mot de négation peut influencer le ton de tous les mots qui l'entourent, et ignorer les négations est l'une des principales causes de mauvaise classification. Il est clair que les mots de négation sont importants car leur apparence change souvent l'orientation de l'opinion. Par exemple, la phrase "Je n'aime pas cet appareil photo" est négative.

En linguistique, la négation est un moyen d'inverser la polarité des mots ou des phrases. Les chercheurs utilisent différentes règles linguistiques pour déterminer si la négation se produit, mais il est également important de déterminer la gamme de mots affectés par les mots de négation.

pour faire face à cette issue, nous utilisons dans notre travail une liste de mots exprimant la négation, tels que : not, do not, will not, never, cannot, does not, etc. Après la classification en utilisant les mots d'opinion, nous vérifions si le tweet ou la publication/commentaire contient un mot de négation. L'idée est que si le tweet, par exemple, est positif mais contient un mot de négation, il sera négatif et inversement.

3.5.3 Analyse du sentiment multilingue

Tout au long de nos travaux réalisés dans cette thèse, toutes les approches et les méthodes que nous avons proposées pour l'analyse des sentiments et la classification des données sont basées sur la langue anglaise. La raison de baser sur la langue anglaise est que la plupart des données publiées dans les réseaux sociaux sont en langue anglaise (les tweets, les avis dans des sites d'e-commerce comme Amazon, les avis sur les films sur Netflix ou des publications/commentaires Facebook).

La majorité des systèmes actuels d'analyse des sentiments s'adressent à une seule langue, généralement l'anglais ; cependant, avec la croissance d'Internet, en particulier celle des réseaux sociaux, les utilisateurs peuvent écrire des tweets, des publications sur Facebook ou des commentaires dans différentes langues. L'analyse des sentiments dans une seule langue augmente le risque de manquer des informations essentielles dans des textes écrits dans d'autres langues. Afin d'analyser les données dans différentes langues,

des techniques d'analyse de sentiments multilingues ont été développées.

Afin de diminuer le risque de manquer des informations essentielles lors de l'analyse de nos données, nous prenons en considération toutes les langues du monde, car sur Twitter, nous trouvons par exemple que les tweets sont écrits dans plus de 40 langues (français, espagnol, anglais, etc.).

L'analyse des sentiments multilingue peut être une tâche difficile. Habituellement, un pré-traitement important est nécessaire et ce pré-traitement utilise un certain nombre de ressources. La plupart de ces ressources sont disponibles en ligne (par exemple, des lexiques de sentiments), mais de nombreux autres doivent être créées (par exemple, des corpus traduits ou des algorithmes de détection de bruit).

Notre idée de créer un système d'analyse de sentiment multilingue consiste à traduire chaque tweet ou publication Facebook écrit dans une autre langue (que l'anglais) de cette langue à l'anglais. Pour la traduction, nous utilisons une API Java appelée WhatsMate API [1], cette API permet de traduire des mots ou des phrases d'une langue à une autre. Nous utilisons la traduction afin de rendre toutes les données collectées en anglais, ce qui nous aide dans la partie de classification, car par exemple nous utilisons dans la phase de pré-traitement de text des algorithmes qui sont destinés à l'anglais (Porter Stemming).

Donc, la première étape à effectuer après la collecte d'un tweet ou d'une publication Facebook liée à un sujet (par exemple, un événement, un produit, etc.) consiste à détecter sa langue(en utilisant l'API WhatsMate). Si la langue détectée est différente de l'anglais, nous la traduisons, afin de rendre tous nos données écrites en anglais et aussi pour rendre nos approches d'analyse des sentiments multilingues.

3.5.4 Métriques et évaluation de nos approches proposées

Il y a plusieurs façons dans lequel vous pouvez obtenir des mesures de performance pour évaluer un classifieur et comprendre la précision d'un modèle d'analyse des sentiments. Après l'analyse et la classification d'un texte(extraction des sentiments ou la classification en classes), il est important d'évaluer les résultats obtenus afin de comprendre la qualité du modèle de l'analyse de sentiment adopté[93].

Selon le nombre de classes, il existe deux types de problèmes de classification, à savoir, la classification binaire où il n'y a que deux classes et la classification multi-classes où le nombre de classes est supérieur à deux. Supposons que nous avons deux classes, ce qui signifie une classification binaire, P pour la classe positive et N pour la classe négative. C'est-à-dire que nous avons un système d'analyse de sentiment qui vise à classifier un texte en deux classes, soit le texte représente un avis positif ou négatif.

1. http ://api.whatsmate.net/

Dans notre travail pour évaluer les approches proposées nous basons sur les métriques suivantes :

- **taux de classification(Accuracy) :**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

  Où P et N indiquent le nombre d'échantillons positifs et négatifs, respectivement.
  - **TP (true positive) :** le nombre d'échantillons positifs correctement prédits, ce qui signifie que la valeur de la classe de l'échantillon est positive et la valeur prévue après le classement est aussi positive.
  - **TN (true negative) :** le nombre d'échantillons négatifs correctement prédits, ce qui signifie que la valeur de la classe de l'échantillon est négative et la valeur prévue après la classification est négative.
  - **FP (false positive) :** lorsque l'échantillon est négatif mais la classe prévue après la classification est positive.
  - **FN (false negative) :** lorsque l'échantillon est positif mais la classe prévue après la classification est négative.
- **Taux d'erreur :** Le complément de la métrique du taux de classification est le taux d'erreur (Error rate ER) ou taux de classification erronée. Cette métrique représente le nombre d'échantillons mal classés des classes positives et négatives, et est calculée comme suit :

$$ERR = 1 - Accuracy = \frac{FP + FN}{TP + TN + FP + FN} \tag{2}$$

- **rappel :** Sensibilité, True positive rate(TPR), taux de succès, ou le rappel d'un classifieur représente les échantillons positifs correctement classés au nombre total d'échantillons positifs, il est présenté dans l'équation 3 :

$$rappel = \frac{TP}{TP + FN} \tag{3}$$

- **Précision :** Positive prediction value(PPV), ou la précision représente la proportion d'échantillons positifs correctement classés au nombre total d'échantillons prédits positifs, comme indiqué dans l'équation 4.

$$Précision = \frac{TP}{FP + TP} \tag{4}$$

- **F1-score :** s'appelle également F1-measure, il représente la moyenne harmonique de précision et de rappel comme dans l'équation 5. La valeur de F1-score est compris

entre zéro et un. Des valeurs élevées de F1-score indiquent une performance de classification élevée.

$$F1 - score = \frac{2 * précision * rappel}{précision + rappel} \tag{5}$$

3.5.5 Type de nos systèmes d'analyse des sentiments

Le but de notre travail est de classifier les données issues des réseaux sociaux en classes(positive, négative ou neutre), ces données sont dans la majorité des cas sous forme des phrases (par exemple dans Twitter les tweets sont limités à 280 caractères). Donc les approches de l'analyse des sentiments que nous avons proposées se situent dans le niveau de phrase (sentence level), où on cherche la classe de chaque phrase (tweet ou publication/commentaire Facebook).

Les méthodes proposées peuvent être classées en trois catégories : **des approches basées sur le lexique** où on a utilisé un dictionnaire qui contient des mots avec leurs degrés de polarité équivalents, des approches basées sur des méthodes de classification automatique où nous avons utilisé des approches de l'intelligence artificielle, et enfin des approches hybrides qui combines les deux types précédents.

Dans la phase de classification, nous classifions les données selon deux catégories : les données exprimant des sentiments (phrases subjectives) et les données exprimant des faits ou les phrases neutres (classe neutre/phrases objectives), ce qui signifie une classification de subjectivité. Après, nous classifions les données subjectives selon deux classes (positive ou négative), c'est-à-dire une classification de polarité.

Figure I.3 présente les différentes étapes appliquées sur nos données pour les analyser et les classifier.

Pour faire une classification, il y a trois grandes tâches à réaliser, premièrement c'est la phase de construction de la base de données à analyser (tweets, publications et commentaires Facebook). Après nous arrivons à la phase de pré-traitement qui est une étape primordiale dans l'analyse des données textuelles (en appliquant les méthodes de traitement automatique de la langue naturelle). Et enfin quand les données sont prêtes pour l'analyse, on applique nos méthodes afin de classifier les données en classes(extraire les sentiments), en prenant en compte la phase d'extraction des mots d'opinion et la négation. Nos approches d'analyse des sentiments sont multilingues, elles prennent en compte toutes les langages, cela est fait par la traduction des données qui ne sont pas écrites en anglais.

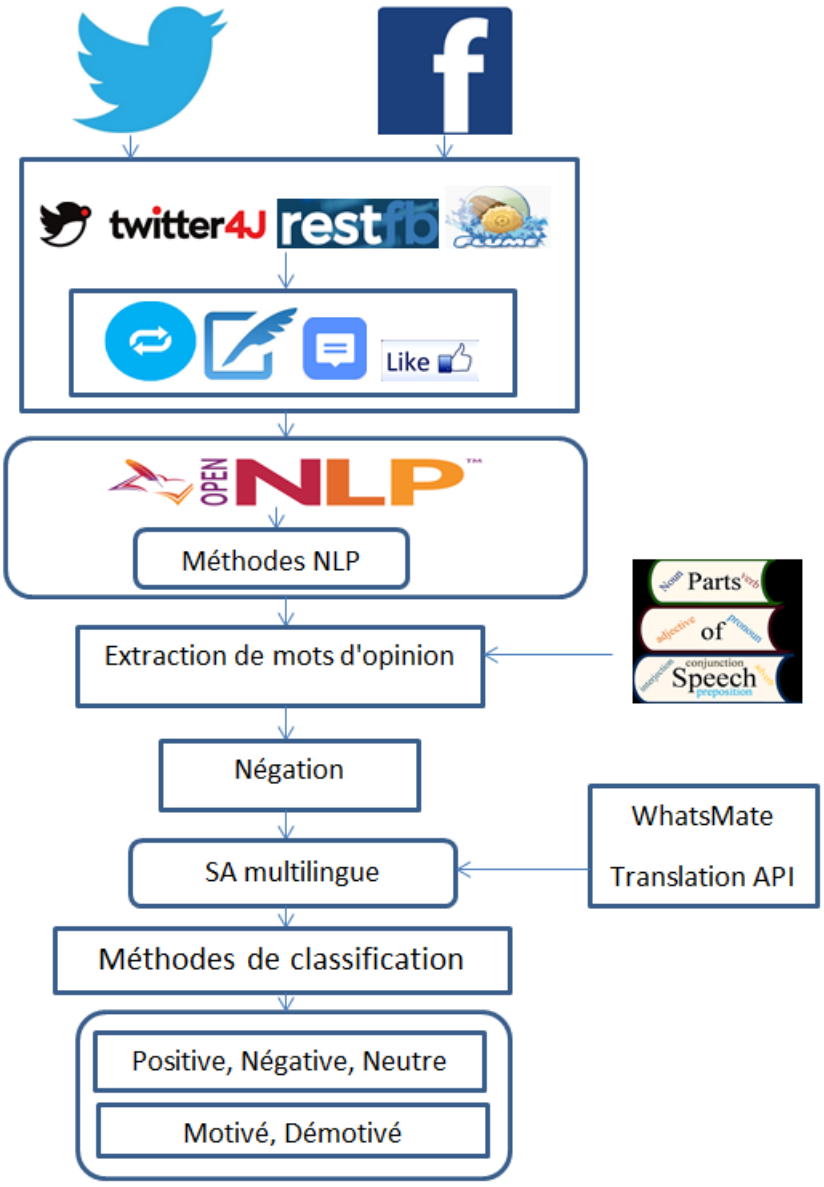


FIGURE I.3 – Étapes de classification des données.

Dans ce chapitre, nous avons présenté les notions principales du domaine de l'analyse des sentiments, et comment on va l'utiliser dans notre travail. Avant de commencer la partie de présentation de nos contributions, le chapitre suivant va être consacré pour l'état de l'art.

Chapitre 4

Outils et approches d'analyse des sentiments

Après que nous avons présenté les grands axes de notre travail, à savoir la façon d'extraire les données à partir des réseaux sociaux, et la méthode adoptée afin de les stocker d'une manière distribuée dans le système de fichiers HDFS, la partie de prétraitement du texte afin d'éliminer le bruit des données et les rendre prêtes pour l'analyse, et enfin, la présentation de notre domaine d'application(sentiment analysis) et la façon avec laquelle nous analysons(classifions) les données extraites. Ce chapitre est consacré à la partie de l'état de l'art, en faisant une étude bibliographique des différents travaux réalisés, soit au niveau de la partie de l'extraction et du stockage des données(Big Data), soit dans la partie du traitement automatique de langage naturel (NLP) et son rôle dans l'analyse des sentiments, ou au niveau des approches et méthodes publiées pour l'analyse et la classification des données textuelles afin d'extraire les sentiments et la motivation.

4.1 Big Data dans l'analyse des sentiments

Avec l'apparition des réseaux sociaux, la quantité de données publiées en ligne augmente de façon explosive, ce qui rend l'utilisation des technologies Big Data dans l'analyse des sentiments une tâche fondamentale. Au cours des dernières années, les chercheurs ont publié de nouveaux travaux utilisant le Big Data pour améliorer les résultats de la classification des sentiments.

Les auteurs de[44] décrivent le processus d'élaboration d'un framework Big Data permettant d'analyser les sentiments des données des réseaux sociaux. Ce framework consiste en une collecte de données en temps réel suivie des méthodes d'apprentissage automatique permettant d'analyser les sentiments sur une grande masse de données. Pour obtenir les données et effectuer une analyse des sentiments, les auteurs ont utilisé Twitter. Pour les vitesses de traitement et la capacité de stockage, une infrastructure de calcul à grande échelle, telle que Spark et Map Reduce, a été utilisée, en raison de son efficacité à gérer les données et sa rapidité. Les auteurs utilisent deux API pour la collecte des tweets :

Streaming API et REST API. Pour l'analyse, ils ont utilisé une architecture Big Data basée sur le modèle Map Reduce et l'algorithme Naïve bayésienne.

Dans l'article[10], les auteurs proposent une approche généralisée de l'analyse des sentiments dans un environnement Big Data. Le modèle proposé servirait à incorporer différentes approches supervisées et non supervisées d'extraction, de classification et de notation d'opinions et des mots de sentiments. L'approche proposée vise à fournir la classification et l'analyse des sentiments au niveau de la phrase pour les applications Big Data. Les auteurs extraient des données de Twitter, Facebook et d'autres sites de micro-blogging. Ils ont utilisé l'analyse des données volumineuses pour un stockage et une récupération efficaces des données non structurées. Après l'extraction et le stockage des données de manière distribuée, les auteurs impliquent une analyse de texte intégrant des approches de traitement du langage naturel pour détecter la subjectivité du texte, c'est-à-dire la classification et l'extraction des sentiments.

Dans les travaux de Nandimath et al.[42], les auteurs utilisent le Big data pour extraire des données de réseaux sociaux et les stocker ensuite dans mongoDB, une base de données NoSQL. Les auteurs ont utilisé MapReduce pour analyser les données et les classer en classes.

La méthode présentée dans[15] est une approche pour organiser des données complexes non structurées et pour récupérer les informations nécessaires. L'article vise à trouver un moyen efficace de stocker des données non structurées et une approche appropriée pour les extraire. Les données non structurées ciblées dans ce travail sont les tweets publics de Twitter. Le travail consiste à construire une application Big Data qui reçoit le flux de tweets, et qui est ensuite stocké dans HBase [1] à l'aide d'un cluster Hadoop et suivi d'une analyse des données.

Liu et al.[53] ont utilisé HDFS et le modèle de programmation MapReduce pour développer un système d'analyse des sentiments. Pour analyser les données, ils ont utilisé deux bases de données : Cornell University [2] et Stanford SNAP Amazon [3]. Dans cet article, les auteurs ont pour objectif d'évaluer l'évolutivité du classifieur Naïve bayésienne (NBC) dans des grands ensembles de données. Ils ont mis en œuvre NBC pour obtenir un contrôle précis de la procédure d'analyse. Un système d'analyse Big Data est également conçu pour cette étude. Le résultat est encourageant dans la mesure où la précision de NBC est améliorée et approche les 82% lorsque la taille du jeu de données augmente.

Les chercheurs de [34] ont proposé une nouvelle approche non supervisée basée sur

1. *https ://hbase.apache.org/*, Apache HBase est la base de données Hadoop, un magasin Big Data distribué et évolutif.
2. http ://www.cs.cornell.edu/people/pabo/movie-review-data/
3. http ://snap.stanford.edu/data/web-Amazon.html

l'algorithme support vecteur machine (SVM), permettant de classer un grand volume de données (Big Data) provenant de différentes sources telles que les réseaux sociaux, les communautés Web, les blogs, Wikipedia et d'autres formes de médias collaboratifs en ligne. Les résultats ont montré une amélioration significative à la fois de la reconnaissance des émotions et de la détection de polarité, en ouvrant ainsi la voie à des approches d'apprentissage semi-supervisé pour l'analyse de données sociales volumineuses.

Dans le travail présenté dans[61], les auteurs tentent de classer les tweets postés sur Twitter en trois classes : positive, négative ou neutre. Ils utilisaient Apache Hadoop pour générer et analyser des données volumineuses sous forme de tweets. Et après, ils ont utilisé deux dictionnaires pour la classification : le premier est un dictionnaire annoté à la main pour les émoticônes, ce dictionnaire aide à mesurer la polarité des émoticônes. Et le second est un dictionnaire d'acronyme.

Dans[79], les auteurs analysent les performances des différents types de systèmes de stockage tels que HDFS et Hbase (solutions Big Data) pour le stockage de contenu social. Ils suggèrent que la grande quantité de données publiées en ligne rend impossible le stockage et l'analyse des données avec une seule machine ou même un seul cluster. Au lieu de cela, le stockage et le traitement doivent être distribués.

Yu et al.[104] ont collecté des données volumineuses en temps réel de Twitter. Les auteurs ont récupéré les tweets via l'API de recherche de Twitter lors de la coupe du monde FIFA 2014. L'API de recherche Twitter fournit un accès programmé pour lire et écrire des données Twitter et fournit environ 1 à 2% d'un échantillon aléatoire de tous les tweets. Ils ont conçu un robot d'exploration Web(web crawler) pour collecter et analyser les tweets anglais en temps réel à l'aide d'une liste d'hashtags prédéfinie, tels que #FIFA, #Football, #Worldcup ou #Soccer. Pour chaque tweet ou retweet, ils ont analysé plusieurs propriétés clés, telles que le nom d'utilisateur, l'identifiant de l'utilisateur, l'heure de publication, la fréquence de retweet ainsi que le contenu.

4.2 Traitement automatique du langage naturel pour l'analyse des sentiments

Comme présenté dans le chapitre 1, les données textuelles surtout les données des réseaux sociaux sont non structurées, et pour les analyser il est très important de les préparer avant la classification en appliquant les différents pré-traitements du texte(les méthodes du NLP). Dans la littérature, nous trouvons beaucoup des travaux qui ont démontré le rôle du NLP dans le domaine de la classification des données textuelles(analyse des sentiments).

Shiliang et al.[89] présentent une revue des techniques de traitement automatique du langage naturel (NLP) pour l'extraction d'opinion. Les auteurs donnent une présentation générale des approches d'analyse des sentiments et démontrent le rôle des techniques de pré-traitement du texte dans l'amélioration du domaine de l'extraction d'opinion.

Le papier de [40] a examiné les effets des méthodes de pré-traitement du texte sur les performances de la classification des sentiments selon deux types de tâches de classification : une classification binaire en deux classes négative et positive, et une classification à 3 voies en 3 classes positive, négative ou neutre. Ils ont résumé les performances de classification de six méthodes de pré-traitement en utilisant quatre classifieurs supervisés sur cinq jeux de données Twitter(The Stanford Twitter Sentiment Test, SemEval2014, The Stanford Twitter Sentiment Gold, The Sentiment Strength Twitter Dataset, The Sentiment Evaluation Dataset). Les résultats expérimentaux indiquent que la suppression des URLs, des mots vides et des nombres n'affectent que très peu les performances des classifieurs ; en outre, le remplacement des négations et le développement des acronymes peuvent améliorer la précision de la classification. Par conséquent, la suppression des mots vides, des nombres et des URLs sont appropriés pour réduire le bruit mais n'affectent pas les performances. Remplacer la négation est efficace pour l'analyse des sentiments.

Dans[14] les auteurs utilisent des méthodes de pré-traitement de texte pour traiter et présenter les tweets dans un format organisé, et augmenter la compréhension de la machine sur le texte. Les techniques NLP utilisées incluent la suppression des URLs et des hashtags, le remplacement spécial des symboles, la suppression des caractères répétés, l'extension des abréviations et des acronymes et la capitalisation du sujet.

Saif et al.[80] ont étudié l'effet de différentes méthodes de suppression des mots vides pour la classification de polarité des tweets, et ils ont déterminé si la suppression des mots vides avait une incidence sur les performances des classifieurs de sentiment de Twitter. Ils ont appliqué six différentes méthodes d'identification de mots vides à six différents jeux de données Twitter, et ils ont observé en quoi la suppression de mots vides affecte deux méthodes de classification supervisées des sentiments. Ils ont évalué l'impact de la suppression des mots vides en observant les fluctuations du niveau de fragmentation des données, de la taille de l'espace de traitement du classifieur et de ses performances de classification. L'utilisation des listes de mots vides précompilés a eu un impact négatif sur les performances des approches de classification des sentiments de Twitter.

Zhao dans[39] a évalué la précision des URLs, des mots vides, des lettres répétées, de la négation, de l'acronyme et des nombres dans la tâche de classification des sentiments binaires de Twitter. Les expériences montrent que la précision de la classification des sentiments augmente après l'extension de l'acronyme et le remplacement de la négation,

bien qu'elle ne change pratiquement pas lorsque la suppression des URLs, la suppression des numéros et des mots vides sont appliquées.

Dans le travail de[88] les auteurs disent que dans les médias sociaux, le langage utilisé par les utilisateurs est très informel. Les utilisateurs créent leurs propres mots, raccourcis orthographiques, signes de ponctuation, fautes d'orthographe, mots d'argot, mots nouveaux, URLs et abréviations. Dans cet article, Singh et al. essaient d'analyser l'impact du pré-traitement et de la normalisation sur les messages courts tels que les tweets qui regorgent d'informations, de bruit, de symboles, d'abréviations, et de mots non identifiés.

Dans le travail de Gamon[24] les auteurs ont proposé une approche permettant d'analyser les sentiments des commentaires des clients, et ils ont utilisé des techniques de traitement automatique de langage naturel pour éviter les problèmes liés aux données bruitées.

Khader et al.[46] ont proposé une méthode d'analyse des sentiments pour classifier un ensemble de données issues de Twitter. La méthode utilise l'algorithme Naïf Bayes pour classifier le texte en polarité positive et négative. Plusieurs techniques linguistiques et de pré-traitement NLP ont été appliquées à l'ensemble des données(tokenization, part of speech tagging, et lemmatisation). Le but de ces techniques de pré-traitement est d'étudier leurs effets sur la qualité de la classification. Les techniques de pré-traitement appliquées ont permis d'améliorer la précision de la classification de l'algorithme Naïve Bayes. Les expériences démontrent que la performance de l'analyse des sentiments est améliorée de 5% à l'aide de NLP et du traitement linguistique, ce qui permet d'obtenir une précision de 73% sur l'ensemble des données utilisées.

Haddi et al.[28] ont exploré le rôle du pré-traitement de texte dans l'analyse des sentiments. Ils ont commencé par l'application d'une transformation aux données, qui inclut le nettoyage des balises HTML, le développement des abréviations, la suppression des mots vides, la gestion de la négation et la racinisation(stemming), dans lesquels ils utilisent des techniques de traitement de langage naturel pour les exécuter. Les résultats expérimentaux montrent que la précision de la classification des sentiments peut être considérablement améliorée grâce à l'utilisation de fonctions et d'une représentation appropriées après le pré-traitement du texte.

Les auteurs dans[7] ont étudié l'impact des méthodes de pré-traitement du texte sur la classification des sentiments du réseau social Twitter. Pour ce faire, ils utilisent 5 méthodes de pré-traitement du texte (effets des URLs, négation, lettres répétées, stemming et la lemmatisation) pour étudier leurs effets. Les résultats expérimentaux montrent que l'utilisation de la fonctionnalité de suppression d'URL, de transformation de négation, de répétition de lettres et de normalisation accroît la précision de la classification des

sentiments sur le jeu de données : "Stanford Twitter Sentiment Dataset", d'autre part la précision descend lorsque la racinisation et la lemmatisation sont appliquées.

Dans[85], les auteurs analysent l'effet de la négation dans l'analyse des sentiments, ils proposent une nouvelle approche pour traiter la négation dans les tweets afin d'améliorer la précision de la classification. Cette approche est utile pour calculer la négation sans les mots de négations (not, no, n't, never...etc). Les résultats expérimentaux de ce travail ont montré que si la négation est ignorée dans un avis, la précision est de 79%, et avec la négation la précision est de 84,87%. Leur approche de négation modifiée améliore également le rappel et la précision de 84,81% et 91,8% par rapport à la classification sans négation.

4.3 Étude bibliographique des approches de l'analyse des sentiments

Comme présenté dans le chapitre 3, après la collecte des données et l'application des différentes méthodes de pré-traitement du texte, c'est l'étape de l'analyse des sentiments afin d'extraire des sentiments et des opinions en classifiant le texte en classes(positive, négative ou neutre).

dans cette section, nous présenterons une étude bibliographique des différents types d'approches existantes dans la littérature.

4.3.1 Approches basées sur le lexique pour l'analyse des sentiments

Les auteurs de[72] ont utilisé Twitter API pour collecter un corpus de tweets jouant le rôle du jeu de données comprenant trois classes : sentiments positifs, sentiments négatifs et un ensemble de textes objectifs (aucun sentiment ou sentiments neutres). Les auteurs ont utilisé Twitter pour collecter des données sous forme des tweets, puis effectuer une analyse des sentiments. Ils construisent des corpus à l'aide d'émoticônes telles que « :-)» et « :-(» pour former des données d'entraînement permettant de classer les sentiments afin d'obtenir des échantillons "positifs" et "négatifs". Des évaluations expérimentales montrent que la technique proposée est efficace et fonctionne mieux que les méthodes proposées précédemment.

Dans le travail de Geli et al.[20], les auteurs ont proposé une approche basée sur un dictionnaire qui complète l'approche basée sur un corpus et peut résoudre le problème de l'identification des entités et des aspects d'une phrase d'une manière implicite. par exemple, dans la phrase "le prix de ce vélo est très élevé", l'aspect prix est exprimé expli-

citement, mais dans "ce vélo est cher", cher indique l'aspect prix du vélo d'une manière implicite. L'approche proposée est basée sur le dictionnaire Wordnet pour extraire "glosses" et certaines relations sémantiques telles que la synonymie et l'antonymie. Les résultats expérimentaux montrent que l'approche proposée est efficace et produite des résultats bien meilleurs que les approches basées sur la classification supervisée traditionnelle.

Dans[6] les auteurs ont exploré le rôle d'un corpus d'émotions Twitter pour extraire un lexique de sentiments, qui peut être utilisé pour analyser les sentiments des tweets. Ils ont fait une comparaison qualitative entre les lexiques de sentiments standard et les lexiques de sentiments proposés. Leurs expériences confirment que les lexiques de sentiments proposés apportent des améliorations significatives par rapport aux lexiques standard dans les tâches de classification et de prévision de l'intensité des sentiments.

Moreno-Ortiz et al.[64] ont proposé une nouvelle approche basée sur un corpus pour identifier la polarité dans les textes financiers pour l'analyse des sentiments. Dans cet article, les auteurs étudient d'une manière approfondie le problème de l'extraction des caractéristiques émotionnelles en se basant sur une approche basée sur le lexique pour classifier un texte.

Dans l'article de Dibakar[76], une étude préliminaire sur l'analyse des sentiments Twitter basée sur le lexique est présentée. Les auteurs ont utilisé un corpus Twitter dépourvu d'émoticônes, et pour la classification des données Twitter, une recherche basée sur SentiWordNet est effectuée afin de déterminer la polarité au niveau des mots. Les polarités au niveau des mots sont ensuite combinées avec une méthode basée sur des règles pour calculer la polarité globale.

Taboada et al.[91] utilisent une approche basée sur le lexique utilisant l'orientation sémantique des mots, pour extraire le sentiment des textes et pour classer les documents. Ils ont conclu que les méthodes d'analyse basées sur le lexique sont robustes.

Le but de l'article présenté dans[60] est de développer un nouveau système appelé **SentiView**, une approche basée sur le lexique pour l'analyse des sentiments sur des données Twitter. Le système proposé repose sur une approche lexicale permettant de générer un score unique pour chaque tweet. Le score est généré par le nombre de mots positifs ou négatifs présents dans le tweet. Chaque résultat de mot positif est augmenté de 1 et chaque résultat de mot négatif est diminué de 1.

Le travail de[16] présente une méthodologie permettant de déterminer la polarité d'un texte dans un framework multilingue. La méthode exploite le dictionnaire lexical SentiWordNet pour l'analyse des sentiments disponibles en anglais. Tout d'abord, un document dans une langue différente de l'anglais est traduit en anglais à l'aide d'un logiciel de traduction standard. Ensuite, le document traduit est classé selon son sentiment dans l'une

des deux classes "positive" ou "négative". Pour la classification des sentiments, un document est recherché pour des mots contenant des sentiments tels que des adjectifs. Au moyen de SentiWordNet, les scores de positivité et de négativité sont déterminés pour ces mots. La méthode est testée pour les critiques de films allemands sélectionnés sur Amazon et comparée à un classifieur statistique de polarité basé sur n-grammes. Les résultats montrent que l'utilisation de la technologie standard et des approches existantes d'analyse du sentiment constitue une approche viable de l'analyse du sentiment dans un cadre multilingue.

4.3.2 Approches basées sur l'apprentissage automatique (Machine Learning)

L'article[9] présente des expériences d'analyses de sentiment dans des textes de blog, des comptes rendus et des forums trouvés sur le World Wide Web et rédigés en anglais, néerlandais et français. Les auteurs s'intéressent aux sentiments que les gens expriment à l'égard de certains produits de consommation. Ils apprennent plusieurs modèles de classification à partir d'un ensemble d'exemples annotés manuellement, plus spécifiquement à partir de phrases annotées comme positives, négatives ou neutres à l'égard d'une certaine entité d'intérêt. Les auteurs utilisent une machine à vecteurs de support (SVM), une multinomiale Naive Bayes (MNB) et une entropie maximale(ME) pour la classification des données.

L'objectif de l'étude de Catal et al.[11] est d'étudier les avantages potentiels du concept de systèmes de classifieurs multiples sur le problème de classification des sentiments turcs et de proposer une nouvelle technique de classification. L'algorithme de vote a été utilisé avec trois classifieurs, à savoir Naif Bayes, Support Vector Machine (SVM) et Bagging. Les paramètres du SVM ont été optimisés lorsqu'il était utilisé en tant que classifieur individuel. Les résultats expérimentaux ont montré que les systèmes de classifieurs multiples améliorent les performances des classifieurs individuels sur les jeux de données de classification des sentiments turcs, et que les méta-classifieurs contribuent à la puissance de ces systèmes de classifieurs multiples.

Le travail cité dans[43] étudie les critiques de films en ligne en utilisant des approches d'analyse de sentiment. Dans cette étude, les techniques de classification des sentiments ont été appliquées aux critiques de films. Plus précisément, les auteurs ont comparé trois approches d'apprentissage automatique supervisé : SVM, Naive Bayes et KNN. Les résultats empiriques indiquent que l'approche SVM a surpassé les approches Naive Bayes et KNN, et qu'elle a atteint une précision d'au moins 80%.

Dans le document[94] quatre algorithmes d'apprentissage automatique : Naive Bayes

(NB), Maximum Entropy (ME), Stochastic Gradient Descente (SGD) et Support Vector Machine(SVM) ont été pris en compte pour la classification des sentiments humains. Ces algorithmes sont ensuite appliqués en utilisant l'approche n-gramme au jeu de données **acl Internet Movie Database (IMDb)**. La précision de différentes méthodes est examinée de manière critique afin d'accéder à leurs performances en fonction de paramètres tels que la précision, le rappel, et f-mesure.

Les auteurs de[25] proposent une méthode d'apprentissage automatique supervisée pour détecter la polarité des tweets financiers. La méthode utilise un ensemble de caractéristiques lexico-morphologiques et sémantiques extraites avec l'outil UMTextStats [4]. Un corpus de tweets financiers espagnols a été mis en point pour réaliser une véritable expérience. Les résultats obtenus dans l'expérience sont comparés à d'autres méthodes pour obtenir de bons résultats. En outre, les auteurs ont comparé les performances de trois algorithmes de classification : J48, Bayesian network et Sequential Minimal Optimization (SMO). Les résultats ont montré que SMO fournit de meilleurs résultats que les algorithmes réseau bayésien(Bayesian network) et J48 avec une F-mesure de 73,2%.

Neethu et al.[70] ont essayé d'analyser les tweets sur des produits électroniques tels que les mobiles et les ordinateurs portables en utilisant une approche d'apprentissage automatique. Ils présentent un nouveau vecteur de fonctionnalités permettant de classer les tweets comme positifs ou négatifs, et d'extraire l'opinion des gens sur les produits. Les auteurs de ce travail ont utilisé les classifieurs Naive Bayes, SVM et Maximum Entropy.

Dans[30] les auteurs ont utilisé un modèle d'ensemble combinant les modèles de réseau de neurones convolutionnels(CNN) et de mémoires à court et long termes(Long Short-Term Memory LSTM) pour prédire le sentiment des tweets arabes. Le modèle proposé atteint un F1-score de 64,46%, ce qui est supérieur au F1-score du modèle d'apprentissage profond (deep learning) 53,6%, sur le jeu de données **Arabic Sentiment Tweets Dataset (ASTD).**

Dhanya et al.[18] ont effectué une analyse des sentiments sur un ensemble de données Twitter à l'aide d'approches d'apprentissage automatique. Les données Twitter du 9 novembre au 3 décembre 2016 sont considérées pour l'étape de classification. Les données sont pré-traitées pour les nettoyer et les rendre prêtes pour l'analyse. Une série finale de 5000 tweets est analysée à l'aide des techniques d'apprentissage automatique telles que SVM, Naïve Bayes et arbre de décision, puis les résultats sont comparés.

Le document[26] contribue à l'analyse des sentiments pour classer les commentaires

4. UMTextStats est un logiciel permettant d'étudier les composantes lexicales, morphologiques, sémantiques, émotionnelles et cognitives contenues dans un texte espagnol. L'entrée de cet outil est un ensemble de textes en langage naturel et le résultat est un vecteur formé par différentes caractéristiques de chaque texte.

des clients, qui se présentent sous la forme de tweets. Pour cela, les auteurs ont d'abord traité le jeu de données, puis en extrait les adjectifs qui ont une signification appelée "vecteur de caractéristiques", puis sélectionné la liste de "vecteurs de caractéristiques", et enfin appliqué des algorithmes de classification basés sur l'apprentissage automatique, à savoir : Naive Bayes, Maximum entropy(ME), et SVM avec WordNet basé sur l'orientation sémantique qui extrait les synonymes et les similarités. Ils ont mesuré les performances des classifieurs en matière de rappel, de précision et d'exactitude.

Jagdale et al.[36] utilisent un jeu de données provient d'Amazon qui contient des critiques sur des appareils photo, des ordinateurs portables, des téléphones mobiles, des tablettes, des téléviseurs et la vidéosurveillance. Après le pré-traitement, ils ont appliqué des algorithmes d'apprentissage automatique pour classifier les commentaires en deux classes : positive ou négative. Cet article conclut que les techniques d'apprentissage automatique donnent des meilleurs résultats pour classifier les revues de produits. Naïve bayes a obtenu une précision de 98,17% et SVM a obtenu une précision de 93,54% pour les commentaires sur les appareils photo.

Rezwanul Huq et al.[33] développent une classe fonctionnelle capable de classer correctement et automatiquement le sentiment d'un tweet inconnu. Ils proposent des techniques permettant de classer précisément l'étiquette de sentiment. Pour ça, ils introduisent deux méthodes : l'une des méthodes est connue sous le nom d'**algorithme de classifieur des sentiments** basée sur l'algorithme du plus proche voisin et l'autre est basée sur SVM. Les auteurs de cet article classifient les tweets en deux classes : positive et négative. Leurs paramètres clés pour évaluer les performances des classifications sont l'exactitude, le rappel et la précision sur 1000 tweets. Les expériences montrent que l'algorithme de classifieur des sentiments est plus performant que le SVM.

Dans l'article présenté dans[67], les auteurs proposent une nouvelle approche pour classer les données des critiques de films du site web **Epinions.com** sur la base de machines à vecteurs de support SVM.

Dans la recherche présentée dans[101], les techniques de classification des sentiments ont été incorporées dans le domaine des analyses des critiques tirées de blogs de voyage. Plus précisément, les auteurs ont comparé trois algorithmes d'apprentissage automatique supervisés : Naïve Bayes, SVM et le modèle basé sur les caractères(Character based N-gram model) pour la classification des sentiments des critiques sur les blogs de voyages de sept destinations de voyage populaires aux États-Unis et en Europe. Les résultats empiriques ont montré que les approches SVM et N-gramme surpassaient l'approche Naïve Bayes et que, lorsque les jeux de données comportaient un grand nombre de critiques, les trois approches atteignaient une précision d'au moins 80%.

Bac et al.[50] ont utilisé un ensemble de données d'environ 200000 tweets pour tester les classifieurs. Ils ont construit un modèle qui classe les tweets collectés à partir des API Twitter en deux classes : positive ou négative. Le modèle fonctionne en trois étapes : un classifieur catégorise les tweets comme étant objectifs ou subjectifs, un autre classifieur organise les tweets subjectifs en positifs ou en négatifs et enfin, le système résume les tweets en un graphe virtuel. Pour les tests, ils ont appliqué trois types de fonctionnalités : Unigramme, Bigram et Orienté objet. Les classifieurs intégrés dans un modèle pour une classification efficace sont Naive Bayes (NB) et Support Vector Machine (SVM).

Dans [27], les auteurs présentent les résultats d'algorithmes d'apprentissage automatique permettant de classer le sentiment des messages Twitter en utilisant des classifieurs supervisés. Ils montrent que les algorithmes d'apprentissage automatique (Naive Bayes, Maximum Entropy et SVM) ont une précision supérieure à 80% pour les émoticônes. L'idée principale de cet article est de savoir comment les auteurs peuvent utiliser des émoticônes dans la classification des tweets à l'aide d'un apprentissage supervisé.

4.3.3 Approches Hybride

En plus des approches basées sur le lexique et les dictionnaires, et les méthodes basées sur l'apprentissage automatique, nous trouvons dans la littérature des approches hybrides qui utilisent une combinaison entre deux ou plusieurs approches, par exemple l'utilisation des dictionnaires et l'apprentissage automatique, ou l'utilisation des algorithmes de l'intelligence artificielle avec des algorithmes de classification supervisée.

Les auteurs de[86] ont proposé une approche hybride combinant à la fois l'apprentissage automatique à l'aide de machines à vecteurs de support et l'approche d'orientation sémantique pour améliorer les mesures de performance de l'analyse des sentiments au niveau des phrases en dialecte égyptien. Le but est de classer une phrase que ce soit un blog, une critique, un tweet, comme détenant un sentiment globalement positif, négatif ou neutre. Les résultats obtenus montrent des améliorations significatives en matière d'exactitude, de précision, de rappel et de F-mesure, indiquant que l'approche proposée est efficace pour la classification des sentiments au niveau de la phrase.

Xia et al.[21] ont proposé une approche hybride du problème d'analyse de sentiment pour un microblog chinois. Cette approche hybride combine les techniques de base du traitement du langage naturel (NLP) et de l'apprentissage automatique pour déterminer l'orientation sémantique du microblog chinois. La méthode hybride est testée sur deux ensembles de données publiques et les résultats montrent que cette méthode est efficace.

L'article de[68] combine des approches basées sur le lexique et sur l'apprentissage automatique. À cette fin, les auteurs choisissent d'utiliser une machine à vecteurs de

support (SVM) et un dictionnaire lexical anglais : AFINN. L'architecture hybride a une plus grande précision que la méthode lexicale pure et fournit plus de structures et une redondance accrue par rapport à l'approche d'apprentissage automatique.

Dans le travail de D. V. Nagarjuna et al.[17], les auteurs ont proposé une approche hybride basée sur une approche d'apprentissage automatique et une approche basée sur le lexique. D'abord, ils classent les avis de chaque domaine à l'aide des algorithmes naïve Bayes et SVM, puis déterminent la polarité au niveau du document à l'aide de l'algorithme de HARN, qui relève de l'approche basée sur le lexique. Les résultats montrent que l'approche proposée a obtenu une précision d'environ 80 à 85% par rapport à l'algorithme de HARN proposé dans l'approche basée sur le lexique.

Dans [66], les auteurs ont présenté un nouveau système d'analyse des sentiments appelé pSenti, combinant des approches basées sur le lexique et sur l'apprentissage. Comparé aux systèmes basés sur le lexique pur, pSenti atteint une précision nettement supérieure dans la classification de polarité des sentiments ainsi que dans la détection de la force des sentiments. Comparé aux systèmes purement basés sur l'apprentissage, il offre des résultats plus structurés et lisibles avec une explication et une justification orientées aspect.

Le but de la recherche de Nurulhuda et al.[108] est de proposer une nouvelle approche hybride de classification des sentiments utilisant les attributs de Twitter comme caractéristiques permettant d'améliorer les performances d'analyse des sentiments de Twitter. La classification de sentiment hybride contient une règle et une sélection de caractéristiques pour identifier les mots de sentiment, ainsi que l'algorithme machine à vecteurs de support(SVM) pour la classification des sentiments. la classification hybride des sentiments a été validée à l'aide de jeux de données Twitter pour représenter différents domaines, et les implémentations ont montré que la nouvelle méthode a pu améliorer les performances de précision.

Nurulhuda et al.[107] proposent une approche hybride pour analyser les sentiments basés sur les aspects à l'égard des tweets. le but de ce travail est d'identifier les aspects explicites et implicites qui sont cruciaux pour l'analyse de sentiment basée sur les aspects. L'approche hybride entre l'exploration de règles d'association, l'analyse de dépendance et SentiWordNet est appliquée pour résoudre ce problème d'analyse de sentiment basé sur des aspects.

L'article de[75] combine une classification basée sur des règles, l'apprentissage supervisé et l'apprentissage automatique dans une nouvelle méthode combinée. Les auteurs présentent les résultats empiriques d'une étude comparative qui évalue l'efficacité de différents classificateurs et montre que l'utilisation de plusieurs classificateurs de manière hybride peut améliorer l'efficacité de l'analyse des sentiments. Cette méthode est testée

sur des critiques de films, des critiques de produits et des commentaires sur MySpace. Les résultats montrent qu'une classification hybride peut améliorer l'efficacité de la classification en matière de valeurs F1-mesure.

Les approches proposées au cours de cette thèse peuvent être aussi divisées en trois types : une approche basée sur le dictionnaire, une approche basée sur une classification automatique, et une approche hybride. Les approches basées sur le dictionnaire sont basées sur des mots avec un degré de polarité(par exemple entre 5 et -5 comme le cas du dictionnaire AFINN[71]), parfois un mot qui exprime le sentiment d'un écrit à classifier n'existe pas dans ces dictionnaires ce qui affecte négativement la classification, notre travail permet de remédier à ce problème en enrichissant les dictionnaires avec des relations sémantiques.

La majorité des approches basées sur l'apprentissage automatique utilisent des méthodes d'apprentissage supervisé qui sont basées sur des données étiquetées pour classifier les données, parfois ces données ne sont pas étiquetées d'une manière précise ce qui diminue la précision et l'exactitude de la classification, notre idée consiste à proposer une nouvelle approche de classification automatique basée sur la similarité sémantique afin d'automatiser le processus d'extraction d'opinion sans utiliser des données étiquetées.

A partir des travaux présentés précédemment, nous avons réalisé que la majorité des approches sont basées soit sur méthodes statiques soit sur des méthodes d'apprentissage automatique sans prendre en considération le flou et l'imprécision des sentiments. Afin d'augmenter la qualité de la classification, nous avons proposé une approche basée sur un système à logique floue, en combinant l'utilisation de la similarité sémantique et un dictionnaire avec la logique floue.

Les approches existantes dans la littérature travaillent avec une seule langue(généralement l'anglais) pour classifier les données des réseaux sociaux, mais travailler avec une seule langue affecte l'analyse des données en manquant des informations importantes publiées en d'autres langues. A partir de cela, les approches proposées dans cette thèse sont multilingues, elles prennent en considération toutes les langues du monde.

Avec la présentation de quelques approches hybrides pour l'analyse et la classification des données, Nous avons terminé le chapitre de l'état de l'art, où nous avons fait le tour sur les travaux existants soit pour la partie d'extraction des données, soit dans la phase de pré-traitement des données pour les rendre prêtes pour l'analyse, ou dans la partie d'analyse et d'extraction des sentiments. Nous avons présenté également comment nous pouvons remédier aux limites des méthodes existantes en proposant des nouvelles approches. Dans la partie suivante, nous allons présenter les contributions que nous avons réalisées au cours de cette thèse.

Deuxième partie

Contributions et Discussions

Chapitre 1

Classification des données des réseaux sociaux à l'aide d'une approche basée sur un dictionnaire

La classification des données des réseaux sociaux devient au cours des dernières années un domaine actif. Elle tente d'étudier les opinions et les émotions des gens, et à classer les données des réseaux sociaux(tweets, publications Facebook) en classes telles que : positive, négative ou neutre. Dans ce travail, nous proposons une nouvelle méthode pour classer les tweets en trois classes : positive, négative ou neutre ; de manière sémantique à l'aide des dictionnaires WordNet [1] et AFINN [2], et de manière parallèle en utilisant le framework Hadoop avec le système de fichiers distribués Hadoop HDFS et le modèle de programmation MapReduce. L'objectif principal de ce travail est la proposition d'une nouvelle approche d'analyse des sentiments en combinant plusieurs domaines tels que la recherche d'informations, l'exploration d'opinion ou l'analyse de sentiment et le Big Data.

1.1 Analyse des données des réseaux sociaux

De nos jours, les réseaux sociaux deviennent un environnement important dans lequel les gens expriment leurs sentiments, leurs émotions et leurs attitudes face aux produits, marques, films et même aux phénomènes politiques ; d'une manière libre, ce qui facilite la recherche de l'opinion des personnes sur un produit avant de l'acheter.

Les réseaux sociaux tels que Google+, Facebook et Twitter, fournissent une grande quantité de données pouvant être utilisées pour analyser la personnalité d'un utilisateur ou ses opinions sur le produit d'une entreprise. Cette énorme quantité de données contient

1. WordNet est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton depuis une vingtaine d'années. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Des versions de WordNet pour d'autres langues existent, mais la version anglaise est cependant la plus complète à ce jour.
2. AFINN est un dictionnaire contenant des mots avec un poids compris entre -5 et 5 qui exprime le degré sentimental

des informations cruciales relatives aux opinions qui peuvent être utiles aux entreprises et aux autres aspects des industries commerciales et scientifiques. L'extraction d'opinions, de sentiments ou d'attitudes à partir de cette grande quantité de données est impossible avec des méthodes manuelles, car il est important d'utiliser des approches basées sur l'intelligence artificielle et des algorithmes d'apprentissage automatique pour classifier les données des réseaux sociaux de manière automatique.

Parmi les différents réseaux sociaux, Twitter est le plus utilisé dans le domaine de l'analyse des sentiments. C'est une application populaire de microblogging avec plus de 302 millions d'utilisateurs actifs par mois et environ 500 millions de tweets par jour. Sur Twitter, les messages sont limités à 280 caractères. Ils sont connus sous le nom de tweets et peuvent inclure du texte, des URLs, des mentions d'autres utilisateurs et des métadonnées d'hashtag. Actuellement, de nombreux chercheurs utilisent Twitter comme une excellente source de données dans de nombreux domaines [35].

Sur Twitter, les utilisateurs peuvent exprimer leurs opinions de manière libre et sans entrave, ce qui permet d'agréger les commentaires sans intervention. Les utilisateurs peuvent utiliser l'analyse des sentiments pour rechercher des produits ou des services avant de procéder à un achat. Les marchands peuvent l'utiliser pour trouver les opinions publiques de leurs entreprises et sur leurs produits ou services, ou pour analyser la satisfaction de la clientèle. Les organisations peuvent également s'en servir pour recueillir des commentaires critiques sur les problèmes des produits récemment publiés.

L'analyse des sentiments gagne la popularité en raison de l'abondance de données provenant des réseaux sociaux, notamment ceux fournis par Twitter. L'exploration d'opinions a pour objectif d'analyser une grande quantité de données afin d'en déduire différents sentiments qui y sont exprimés. Les sentiments extraits peuvent alors faire l'objet des statistiques sur le sentiment général d'une communauté.

La taille, la variété et la complexité des données des réseaux sociaux augmentent rapidement, tout en restant généralement dans un format non structuré, l'analyse d'un réseau social donné est un processus très coûteux et fastidieux. Pour résoudre ce problème de performance des réseaux à grande échelle (ensemble de données), les chercheurs ont commencé à utiliser des plates-formes de traitement parallèle. Par exemple, Google a développé le modèle de programmation MapReduce pour le traitement de données dans des problèmes de graphes à grande échelle. Hadoop Framework est une alternative open source aux solutions Google pour l'algorithme MapReduce[49].

Dans ce travail, nous proposons une nouvelle méthode pour analyser les tweets et les classer en trois classes : positive, négative ou neutre, en utilisant les notions de la recherche d'informations, plus précisément, nous proposons d'enrichir le dictionnaire AFINN avec la

sémantique. La relation sémantique utilisée est la synonymie en utilisant le dictionnaire sémantique WordNet. La méthode proposée est une approche multilingue, c'est-à-dire qu'elle prend en compte toutes les langues et pas seulement la langue anglaise.

Pour analyser un vaste ensemble de données de tweets sans avoir le problème du temps d'exécution, nous décidons de paralléliser la méthode proposée (classification sémantique des tweets) en travaillant avec les technologies Big Data, plus précisément avec le projet open source Hadoop, en stockant les tweets à analyser, et les résultats de l'analyse dans HDFS. Le développement de l'analyse est effectué à l'aide du modèle de programmation MapReduce.

1.2 Motivation

L'analyse des sentiments ou sentiment analysis(SA) en anglais, est un domaine de recherche scientifique très actif et connaît un développement rapide ces dernières années, en particulier après l'apparition des réseaux sociaux. L'application de l'analyse des sentiments ou des méthodes d'extraction d'opinions aux réseaux sociaux a suscité beaucoup d'intérêt. L'utilisation des réseaux sociaux pour exprimer des opinions sur les produits plutôt que sur les événements politiques ou sociaux a considérablement augmenté au cours de la dernière décennie.

L'analyse de sentiment a plus d'importance dans la business intelligence. En affaires, il est utile pour trouver des réponses aux questions telles que "pourquoi la vente d'un produit est faible ?", "Les besoins des utilisateurs sont satisfaits par nos services" ou "nos utilisateurs sont-ils satisfaits ou veulent-ils plus". Pour obtenir une réponse à ces questions, nous utilisons l'analyse des sentiments. C'est-à-dire qu'au lieu de créer un formulaire manuel pour avoir une idée sur notre produit, nous pouvons facilement classifier les tweets qu'y sont liés à l'aide des approches SA. En bref, nous pouvons dire que l'analyse des sentiments nous aide à trouver une mine d'informations à partir des commentaires des clients.

Nous trouvons dans la littérature plusieurs travaux traitant ce sujet de recherche afin de proposer des nouvelles méthodes pour bien analyser les données provenant des réseaux sociaux. Des données de microblogs telles que Twitter, sur lesquelles les utilisateurs publient des commentaires en temps réel, présentent de nouveaux défis. Récemment, Twitter est également utilisé dans les systèmes d'apprentissage en ligne(e-learning systems)[56] et aussi dans les systèmes de recommandation.

1.3 Classification des tweets

Comme présenté précédemment, l'approche proposée consiste à classer les tweets en trois classes (positive, négative ou neutre) à l'aide du dictionnaire **AFINN**[71] en l'enrichissant avec la sémantique en se basant sur le dictionnaire **WordNet**. La relation de sémantique utilisée est la synonymie. Dans cette section, nous présenterons la méthodologie de la méthode proposée.

Notre proposition est de travailler avec une approche basée sur les dictionnaires. En plus de Wordnet, qui nous donne la possibilité de récupérer des synonymes pour un mot donné ; nous utilisons également le dictionnaire AFINN qui contient des mots en anglais avec un poids qui peut prendre une valeur comprise entre 5 et -5 (fortement positif, légèrement négatif, etc.). Par exemple, le mot "abandonné" a la valeur -2 et le mot "accept" a la valeur 1, etc. Pour rechercher le sentiment d'un tweet, l'idée est de parcourir tous les mots qu'il contient et d'additionner leurs poids à l'aide du dictionnaire AFINN et après utiliser un seuil pour le classer.

WordNet était perçu comme un réseau sémantique. Dans ce réseau sémantique, chaque nœud est un concept. Un nœud est constitué d'un ensemble de synonymes (ou de synsets). Ces termes désignent le concept représenté par le nœud. Dans WordNet, les concepts sont liés par des relations sémantiques, la relation de synonymie est la relation de base dans WordNet.

1.3.1 Pourquoi avons-nous utilisé WordNet ?

Nous utilisons WordNet dans notre travail pour rechercher les synonymes de mots, c'est-à-dire que nous enrichissons le dictionnaire AFINN avec la sémantique.

Le problème dans la classification avec AFINN est que si un mot du tweet qui exprime son sentiment ou son opinion n'existe pas dans ce dictionnaire, ce qui va diminuer donc la précision de la classification. Notre proposition pour remédier à ce problème consiste à enrichir le dictionnaire AFINN avec la relation de synonymie, c'est-à-dire que si, par exemple, un mot du tweet n'existe pas dans AFINN, nous recherchons ses synonymes à l'aide de WordNet, et après, cette fois nous cherchons les synonymes du mot dans AFINN, et par conséquent, nous enrichissons la classification avec la relation sémantique "synonymie" à l'aide de la ressource sémantique WordNet.

1.3.2 préparation des données pour la classification

En raison de la politique de confidentialité des profils Facebook, notre travail se concentre sur Twitter, où la plupart des contenus et des activités partagés en ligne sont ouverts et disponibles.

Comme présenté dans la dernière section du chapitre 1 de la première partie, nous avons utilisé Twitter4J API, Apache Flume et Apache Sqoop afin d'extraire les tweets du Twitter et les stocker dans HDFS d'une manière distribué.

Les données issues des réseaux sociaux sont non structurés et contiennent beaucoup d'informations inutiles. Comme nous avons présenté dans le chapitre 2 de la partie II et afin de rendre les tweets extraits prêts pour la phase de classification et d'analyse, nous y avons appliqué les différentes méthodes de pré-traitement de texte(tokenization, suppression des mots vides, suppression d'URL et des chiffres, enlever les ponctuations, la racinisation ou stemming, etc).

1.3.3 Approche proposée

Après la collection des données à classer, la traduction des tweets non anglais et l'application des différentes méthodes de pré-traitement du texte, en particulier l'étape part-of-speech tags, nous trouvons chaque tweet avec ses mots d'opinion.

Pour classer un tweet selon trois classes, nous utilisons le dictionnaire AFINN et la ressource sémantique WordNet. L'idée est d'enrichir AFINN avec la relation sémantique de synonymie. Afin de classer un tweet nous parcourons tous ses mots d'opinion pour calculer leurs polarités à partir du dictionnaire AFINN, et si nous ne trouvons pas un de ces mots d'opinion, nous cherchons ses synonymes dans WordNet. Nous calculons ensuite la polarité de ces synonymes en utilisant AFINN. Enfin, pour trouver la classe du tweet, nous calculons la moyenne de la somme de la polarité des mots d'opinion s'ils existent dans AFINN et de la polarité de ses synonymes s'ils n'y existent pas, et en utilisant un seuil, nous classons le tweet selon trois classes : positive, négative ou neutre.

Le calcul de la moyenne des poids (polarités) se fait selon la formule suivante :

$$Average = \frac{\sum_{i=1}^{N} P_i}{N} \tag{6}$$

Avec :

- P_i : est la polarité du terme i dans AFINN.
- N : est le nombre de mots d'opinion et leurs synonymes dans le tweet et existants dans AFINN.

Par exemple, supposons que nous voulions classer un tweet T, la première chose à faire est de détecter sa langue, s'il est différent de l'anglais, nous le traduisons. Nous appliquons ensuite les méthodes de pré-traitement du texte pour préparer le tweet T à la classification et pour extraire les mots d'opinion (verbe, adverbe, adjectif). Après la phase de pré-traitement, c'est l'étape de la classification en appliquant la méthode proposée. Nous cherchons les mots d'opinion un par un dans le dictionnaire AFINN pour calculer leurs poids, et si nous ne trouvons pas un de ces mots d'opinion dans AFINN, nous recherchons ses synonymes à l'aide de WordNet, puis nous les recherchons un par un dans AFINN afin de récupérer leurs polarités. À la fin de cette phase, nous aurons les poids (polarités) des mots d'opinion et des synonymes, et pour trouver la classe du tweet, nous calculons la moyenne de la somme de ces poids en utilisant la formule 6.

1.3.4 Nos étapes de travail

Figure II.1 illustre les différentes étapes nécessaires à la classification d'un tweet en appliquant notre proposition.

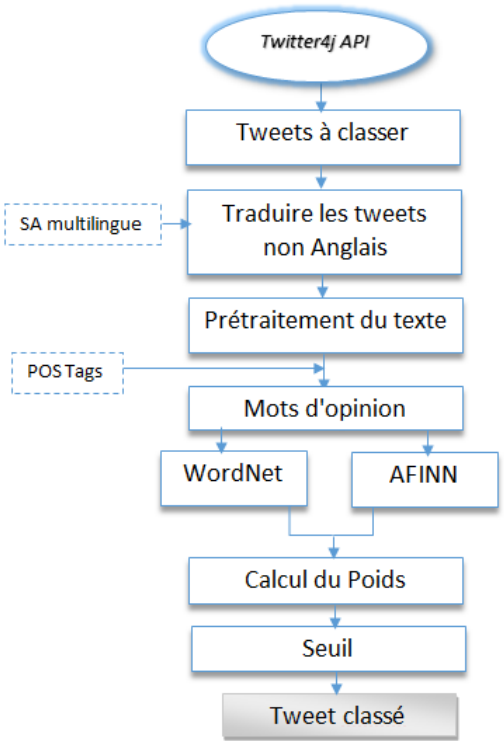


FIGURE II.1 – Étapes de notre travail

Selon la fig. II.1, la première étape est la collection des tweets à analyser à l'aide de l'API Twitter4j, et après l'application des différentes méthodes de pré-traitements du texte, nous utilisons WordNet et AFINN pour classer le tweet en trois classes.

1.4 Parallélisation de la classification et les résultats expérimentaux

Le problème qui se pose lors de la classification des tweets est le temps de calcul nécessaire pour obtenir le résultat de la classification si nous voulons procéder à la classification sur un grand jeu de données contenant des millions de tweets, ainsi que le volume de stockage nécessaire pour stocker cet énorme jeu de données.

Pour remédier à ce problème, nous avons décidé de paralléliser notre travail de classification des tweets en travaillant avec le framework open source HADOOP. L'idée consiste à partager le travail de stockage et de parallélisation entre plusieurs machines afin de réduire le temps de calcul. Pour cela, nous travaillons avec le système de fichiers distribué Hadoop (HDFS) pour stocker le jeu de données de tweets que nous souhaitons classer et le résultat de la classification après l'application de la méthode proposée. La partie programmation de la parallélisation est réalisée avec le modèle de programmation MapReduce.

Notez que notre objectif en matière de parallélisation n'est pas d'étudier un cluster Hadoop, mais simplement de décrire comment nous pouvons paralléliser notre travail en utilisant Hadoop et plus précisément Hadoop MapReduce. Pour cela, nous utilisons Hadoop 2.6.0 avec MapReduceV2 (Yarn). L'installation du cluster est effectuée dans le système d'exploitation UBUNTU 16.04, qui servira comme machine principale du cluster (master machine), et de deux autres nœuds Hadoop(machines slaves) installés dans une machine virtuelle VMWare workstation 12.

La figure II.2 montre la configuration de notre cluster (ensemble de machines Hadoop) qui contient trois machines Hadoop : une machine maître et deux machines esclaves. Notre travail sera donc partagé entre trois machines afin de réduire le temps de calcul. Les nœuds utilisés sont trois nœuds avec le système d'exploitation UBUNTU.

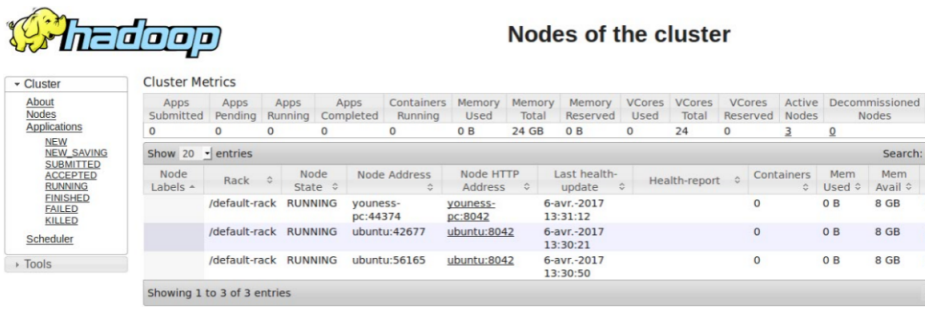


FIGURE II.2 – Configuration de notre cluster

1.4.1 Étapes de parallélisation

La Fig II.3 montre les différentes étapes pour paralléliser la méthode proposée en utilisant HDFS et MapReduce.

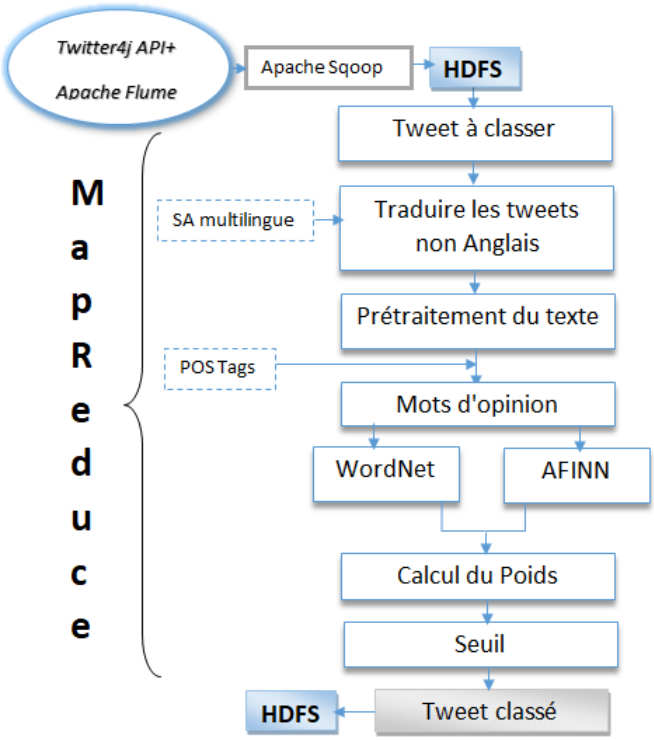


FIGURE II.3 – Étapes de parallélisation

D'après la figure II.3, la première étape de la parallélisation consiste à stocker le jeu de données à classer dans HDFS afin de partager le stockage entre plusieurs machines(machines Hadoop). Et après l'application des méthodes de pré-traitement du texte, c'est l'étape de la classification en appliquant la méthode proposée mais cette fois avec le modèle de programmation MapReduce (de manière parallèle).

L'entrée de l'opération MapReduce à chaque itération contient un tweet à classer et la sortie contient le tweet classifié. Le résultat de la classification est stocké dans HDFS. À la fin du processus de classification, nous stockons le résultat dans HDFS sous forme de deux colonnes, une pour le tweet et l'autre pour sa classe (positive, négative ou neutre).

L'**algorithme 1** montre notre algorithme MapReduce proposé pour la classification des tweets.

Algorithm 1 Modèle de Programmation MapReduce

```
Require: tweet's class
  S ← 0
  c ← 0
  if Tweet is not in English then
    Tweet ← Translate(Tweet)
  end if
  tweet ← TextPreProcesssing(tweet)
  SE[] ← Split(tweet)
  OW[] ← OpinionWord(SE)
  for all word ∈ OW do
    if word ∈ AFINN then
      c ← c+1
      S ← S+AFINN(word)
    else if word ∉ AFINN then
      w[] ← WordNet(word)
      for all syn ∈ w[] do
        if syn ∈ AFINN then
          c ← c+1
          S ← S+AFINN(syn)
        end if
      end for
    end if
  end for
  moy ← S/c
  sentiment ← threshold(moy)
  write(tweet,sentiment)
```

Où :

- Translate(Tweet) : une fonction qui traduit les tweets non anglais.
- TextPreProcesssing(tweet) : applique les différentes méthodes du pré-traitement de texte.
- OpinionWord(SE) : extrait les mots d'opinion des tweets.
- AFINN(**word**) : une fonction qui permet de retourner le poids du sentiment équivalant au mot (**word**) s'il existe dans le dictionnaire AFINN.
- Split(**tweet**) : nous permet de scinder le tweet en mots, pour faciliter son utilisation afin de calculer la similarité sémantique, parce que nous nous concentrons dans notre travail sur des mots individuels.
- WordNet(**word**) : retourne une table qui contient les synonymes du **word**.
- write(**tweet, sentiment**) : est une fonction qui permet de stocker le résultat de la classification dans HDFS sous la forme clé/valeur, le tweet à classer comme clé et

sa classe (le résultat de la classification) comme valeur.

1.4.2 Résultats expérimentaux

Dans cette sous-section, nous allons présenter les résultats expérimentaux de l'approche proposée pour montrer pourquoi nous avons choisi de travailler avec AFINN et comment la sémantique et l'application des différentes méthodes de pré-traitement du texte peuvent améliorer la qualité de la classification. Notre ensemble de données de tweets a été créé à l'aide de l'API Twitter4j et d'Apache Flume. Il est stocké dans HDFS pour permettre la classification d'une manière parallèle.

Pour démontrer pourquoi nous avons choisi de travailler avec le dictionnaire AFINN dans la phase de classification, en particulier dans le calcul de la pondération des mots, nous avons comparé le dictionnaire AFINN avec les algorithmes bien connus d'apprentissage automatique fréquemment utilisés dans l'analyse des sentiments comme présenté dans le chapitre 4 de la partie I. La figure II.4 montre le résultat obtenu pour un ensemble de données de tweets contenant 400 tweets positifs et 400 tweets négatifs.

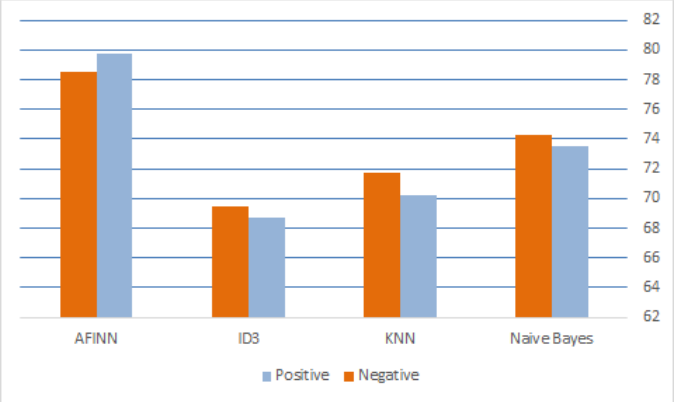


FIGURE II.4 – Taux de précision en utilisant AFINN et les algorithmes d'apprentissage automatique

D'après la figure II.4, nous remarquons que le dictionnaire AFINN surpasse les autres algorithmes d'apprentissage automatique avec une grande précision, que ce soit pour les tweets positifs ou négatifs, ce qui montre pourquoi nous avons choisi d'utiliser le dictionnaire AFINN pour la classification.

Pour montrer l'effet de l'enrichissement du dictionnaire AFINN avec la sémantique, nous calculons le taux de classification et d'erreur de la méthode proposée et du dictionnaire AFINN, en utilisant initialement uniquement AFINN et dans un second temps, avec l'application de l'approche proposée (AFINN + l'utilisation de la sémantique).

Nous présentons également l'effet du pré-traitement du texte sur la classification et le taux d'erreur, et comment l'extraction des mots d'opinion peut améliorer la qualité de notre méthode. Pour cela, nous avons réalisé une classification parallèle en utilisant HDFS et MapReduce d'un jeu de données contenant des tweets de différents sujets rassemblés par l'API Twitter4J et Apache Flume, avec deux méthodes : AFINN et l'approche proposée(AFINN+WordNet).

Le tableau II.1 indique le nombre de tweets mal classés après la classification selon trois classes : positive, négative ou neutre, avec les méthodes de pré-traitement du texte(TP) et d'extraction de mots d'opinion, et sans eux.

Méthodes	Avec les méthodes de TP	Sans les méthodes de TP
AFINN	15	26
Approche Proposée	10	15

TABLE II.1 – Effet du pré-traitement du texte et d'extraction des mots d'opinion sur la classification

Selon le tableau II.1, on remarque que l'utilisation du pré-traitement du texte et surtout l'extraction des mots d'opinion (mots ayant un effet sur la classification des tweets) ont un effet sur la classification, elles permettent de diminuer le bruit qui existe dans les tweets et le nombre de tweets mal classés, c'est-à-dire que les méthodes de pré-traitement du texte augmentent la qualité de la classification ; une autre remarque du tableau II.1 est qu'en utilisant notre approche, nous diminuons le nombre de tweets mal classés.

Suite à ces résultats, nous calculons le taux de classification (TC) et le taux d'erreur (TE) pour les deux méthodes (dictionnaire AFINN et l'approche proposée) en utilisant les deux formules suivantes :

$$TC = \frac{Nombre\ de\ tweets\ bien\ classés}{Nombre\ total\ de\ tweets} \tag{7}$$

$$TE = 1 - TC \tag{8}$$

Les tableaux II.2 et II.3 montrent respectivement les résultats obtenus pour TC et TE en utilisant AFINN et l'approche proposée, avec l'utilisant des méthodes de pré-traitement de texte et sans les utiliser.

Méthodes	Taux de Classification	Taux d'erreur
AFINN	0.55	0.45
Approche Proposée	0.74	0.26

TABLE II.2 – Taux de classification et d'erreur sans pré-traitement du texte

Méthodes	Taux de Classification	Taux d'erreur
AFINN	0.74	0.26
Approche Proposée	0.82	0.18

TABLE II.3 – Taux de classification et d'erreur avec pré-traitement du texte

D'après les tableaux II.2 et II.3, nous remarquons que la méthode proposée surpasse celle qui utilise seulement le dictionnaire AFINN sans sémantique, avec un taux de classification élevé. De plus, l'approche proposée diminue le taux d'erreur, sans oublier que l'utilisation du pré-traitement de texte augmente le taux de classification et diminue le taux d'erreur. Nous concluons donc que l'enrichissement du dictionnaire AFINN avec la sémantique améliore la qualité de la classification.

Une autre expérience de notre travail consiste à montrer l'effet de la parallélisation sur le temps de calcul de la classification, en utilisant trois nœuds Hadoop et un jeu de données contenant 45640 tweets (voir figure II.5).

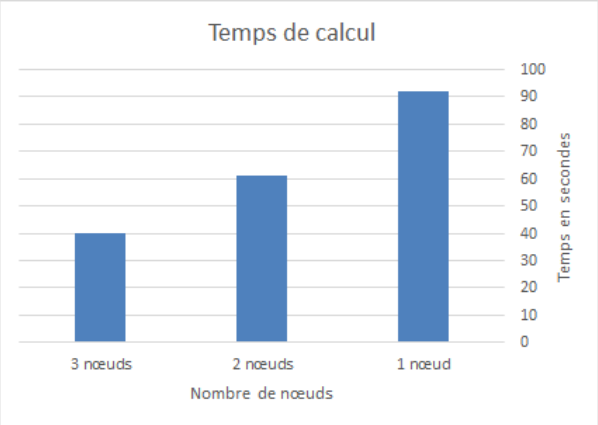


FIGURE II.5 – Temps de calcul de la classification

De la fig. II.5, nous notons que si le nombre de nœuds Hadoop augmente, le temps de calcul diminue, ce qui démontre le choix de paralléliser la classification. Ainsi, à partir de ces résultats, nous concluons que si nous avons un grand nombre de tweets, il suffit d'augmenter le nombre de nœuds Hadoop pour optimiser le temps de classification.

Dans ce travail, nous avons présenté la méthode proposée pour classer les tweets en trois classes : positive, négative ou neutre. Notre proposition consiste à enrichir le dictionnaire AFINN avec la sémantique en utilisant la base lexicale Wordnet. Nous avons également montré comment l'application des méthodes de pré-traitement du texte et d'extraction des mots d'opinion des tweets peuvent améliorer la qualité de la classification. Nous avons présenté le moyen de paralléliser la classification en utilisant Hadoop (HDFS et MapReduce) pour optimiser le temps de classification. La pratique montre que notre

méthode surpasse les algorithmes bien connus d'apprentissage automatique et que l'utilisation de la sémantique donne de bons résultats au niveau du taux de classification et du taux d'erreur. De plus, le pré-traitement du texte améliore le résultat de la classification. Nous avons également montré l'effet de la parallélisation sur le temps de calcul.

1.5 Application de l'approche proposée : E-learning adaptatif en utilisant un algorithme génétique et l'analyse des sentiments dans un système Big Data

Dans cette section, nous décrivons notre système d'E-learning adaptatif, qui permet à l'apprenant de suivre des cours adaptés à son profil et aux objectifs pédagogiques fixés par un enseignant. Nous utilisons pour l'adaptation un algorithme génétique afin de donner à l'apprenant les concepts qui doit apprendre d'une manière optimale en recherchant les objectifs les plus adaptés à son profil. Un deuxième niveau d'adaptation consiste à utiliser l'un des réseaux sociaux de l'apprenant (Twitter ou Facebook) pour extraire ses tweets et publications/commentaires Facebook, ce niveau est basé sur deux étapes, la première consiste à trouver la période d'activité qui nous donne une idée sur la période du jour dans laquelle l'apprenant est actif, et la seconde étape consiste à analyser les sentiments des publications(classer les publications) publiées au cours de la période d'activité. Pour l'analyse des sentiments, nous basons sur l'approche proposée précédemment. Notre travail consiste donc à adapter le profil de l'apprenant aux objectifs pédagogiques en utilisant l'algorithme génétique, les notions de recherche d'informations et l'analyse des sentiments, en effectuant ce travail dans un système Big Data, c'est-à-dire que nous parallélisons le problème de recherche en utilisant Hadoop avec le système de fichiers distribué Hadoop(HDFS) et le modèle de programmation MapReduce.

1.5.1 E-learning Adaptatif

Les nouvelles technologies de l'information et de la communication "NTIC" améliorent profondément nos façons de nous informer, de communiquer et de nous former. Cette émergence technologique a révélé un nouveau mode d'apprentissage appelé e-learning. Il repose sur l'accès à des formations en ligne, interactives et parfois personnalisées, diffusées via un réseau (Internet ou Intranet) ou un autre support électronique. Cet accès permet de développer les compétences des apprenants tout en rendant le processus d'apprentissage

indépendant du temps et du lieu.

Depuis les débuts de l'e-learning, les techniques d'intelligence artificielle (IA) ont été testées pour améliorer l'expérience d'apprentissage. L'utilisation des algorithmes d'intelligence artificielle dans l'e-learning donne naissance à l'**e-learning adaptatif**.

L'apprentissage en ligne adaptatif a pour objectif de donner à l'utilisateur d'une plate-forme d'apprentissage en ligne un contenu pédagogique personnalisé en fonction de son profil, grâce à l'utilisation des algorithmes comme celles de l'intelligence artificielle. L'idée est de trouver l'objectif pédagogique le plus adapté au profil de l'apprenant, car on peut trouver un cours avec plusieurs objectifs pédagogiques proposés par un enseignant.

Nous trouvons plusieurs algorithmes utilisés dans la littérature en e-learning adaptatif, tels que les algorithmes génétiques qui transforment le travail en un problème d'optimisation et donnent à l'apprenant les concepts qu'il doit apprendre.

Notre travail repose sur cet axe de recherche, un système d'apprentissage adaptatif en ligne basé sur un algorithme génétique et les notions de recherche d'informations. À titre de comparaison entre notre travail et les systèmes de recherche d'informations, le profil de l'apprenant jouera le rôle de la requête et les différents objectifs pédagogiques joueront le rôle des documents recherchés. C'est comme nous voulons calculer la similarité entre le profil de l'apprenant et les objectifs pédagogiques pour trouver ceux qui correspondent à son profil et ne garder que les objectifs pertinents qui joueront ensuite le rôle de la population initiale de l'algorithme génétique afin de trouver un objectif optimal.

En plus de rechercher un contenu pédagogique optimal pour le profil de l'apprenant, notre travail consiste également à utiliser l'une des plates-formes les plus puissantes au monde : les réseaux sociaux. Notre proposition est de donner à l'apprenant la possibilité de se connecter à notre plate-forme d'apprentissage avec un réseau social tel que Facebook ou Twitter[57]. L'idée est de calculer une mesure que nous avons appelée **période d'activité** qui nous donnera une idée sur la période de la journée où l'apprenant est très actif et nous aiderons donc à améliorer les compétences de l'apprenant. Après nous faisons une classification des sentiments des publications de cette période d'activité selon trois classes(motivé, démotivé ou neutre) pour connaître le sentiment de l'apprenant, ce qui va aider l'enseignant à définir le niveau d'apprentissage.

1.5.2 Algorithme génétique et système de recherche d'informations

Système de recherche d'informations

La recherche d'informations tente de résoudre le problème suivant : "Étant donné une très grande collection d'objets (principalement des documents), recherchez ceux qui répondent à un besoin d'informations exprimé par un utilisateur (requête)". Dans les systèmes de recherche d'informations, nous trouvons une requête et nous voulons trouver les objets (documents) qui la concernent, le moyen d'évaluer un document, qu'il soit pertinent ou non, consiste à calculer la similarité entre la requête et ce document.

Avant le calcul de la similarité, il est important d'indexer tous les documents, ainsi que la requête, pour les rendre dans une nouvelle présentation afin de faciliter son utilisation. Dans notre cas, nous utilisons la représentation vectorielle [83], où chaque élément du vecteur représente le poids (fréquence) de chaque terme ou concepts dans le document ou dans la requête.

Dans notre cas, notre corpus contient les documents qui représentent les objectifs pédagogiques de l'apprenant. La première chose à faire est d'extraire tous les termes ou concepts du corpus, et pour chaque document, nous construisons un vecteur qui le représente. Si un terme existe dans le document on calcule son poids et sinon on met 0. A la fin de cette opération on construit un vecteur pour chaque document afin de calculer la similarité entre le profil de l'apprenant et chaque objectif pédagogique.

Le calcul de la pondération des termes ou des concepts dans chaque document est calculé à l'aide des formules suivantes :

$$Poid(t_i, d_j) = TF * IDF \tag{9}$$

Où :

- $$TF = \frac{f(t_i, d_j)}{N} \tag{10}$$

  $f(t_i, d_j)$ est le nombre d'occurrences du terme t_i dans le document d_j et N est le nombre total de termes dans le document d_j.

- $$IDF = \frac{log(f(t_i, d_j))}{M} \tag{11}$$

  $f(t_i, d_j)$ est le nombre d'occurrences du terme t_i dans le document d_j et M le nombre total de documents du corpus.

La similarité utilisée dans notre travail est la similarité de Cosine. Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots). Deux objets (documents) sont similaires si leurs vecteurs sont confondus. La formule est définie par le rapport du produit scalaire des vecteurs X et Y et le produit de la norme de X et Y.

$$Sim_{cos}(X,Y) = \frac{\sum_{i=1}^{n} x.y}{\sqrt{(\sum_{i=1}^{n} x^2)}.\sqrt{(\sum_{i=1}^{n} y^2)}} \tag{12}$$

Algorithme génétique

Les algorithmes génétiques (AG) sont des algorithmes d'optimisation stochastiques basés sur les mécanismes de sélection naturelle et génétique, leur fonctionnement est extrêmement simple. Nous partons avec une population de solutions potentielles initiales (chromosomes) sélectionnées arbitrairement, nous évaluons leur performance relative (fitness). Et sur la base de ces performances, une nouvelle population de solutions potentielles est créée à l'aide d'opérateurs évolutifs simples tels que la sélection, le croisement et la mutation. Ce cycle est répété jusqu'à ce qu'une solution satisfaisante soit trouvée.

Dans notre travail, nous utilisons un AG simple, qui consiste à itérer les trois opérations suivantes : reproduction, croisement et mutation,. La population créée au cours de chaque itération s'appelle une génération et est notée P_t.

La première chose dans un algorithme génétique est la définition de la population initiale(opérateur de sélection ou d'évaluation) sur laquelle nous appliquerons le traitement. Dans notre cas, il s'agit de définir les documents (objectifs pédagogiques) pertinents pour le profil de l'apprenant en utilisant la similarité de Cosine qui jouera le rôle de la fonction de fitness, qui est un paramètre très important dans AG, car avec elle, nous pouvons décider si un individu va être sélectionné ou non. Il y a beaucoup de méthodes pour faire la sélection comme la loterie biaisée, la méthode élitiste ou la sélection par tournoi. Après avoir appliqué l'opérateur de sélection à la population initiale, c'est l'étape de reproduction avec l'application de l'opération de croisement et de l'opération de mutation.

Nous trouvons dans la littérature de nombreux travaux qui appliquent des algorithmes génétiques à la recherche d'informations. Vajitoru [95] utilise les algorithmes génétiques dans la recherche d'informations, il a proposé une nouvelle opération de croisement pour améliorer la recherche. Pour ce faire, il a comparé sa proposition à un AG classique. Les résultats montrent l'efficacité de sa proposition. Sathya et Simon [87] utilisent les algorithmes génétiques pour améliorer un système de recherche d'informations et le rendre efficace pour obtenir davantage de pages pertinentes pour la requête de l'utilisateur et optimiser le temps de recherche.

Algorithme génétique dans l'e-learning adaptatif : Plusieurs travaux d'intelligence artificielle ont utilisé les algorithmes génétiques dans l'e-learning adaptatif pour donner à l'apprenant un contenu adéquat à son profil. Chang et Ke [12] ont proposé une composition personnalisée de cours dans un système d'apprentissage adaptatif, basée sur l'algorithme génétique (AG), dans le but de spécifier les ressources d'apprentissage appropriées pour chaque apprenant.

le travail de Romero et al. [78] représente une méthodologie pour améliorer les systèmes éducatifs, utilisant une grammaire basée sur des techniques d'algorithmes génétiques et une optimisation multi-objectifs pour extraire des règles de prédiction permettant aux enseignants de sélectionner les changements les plus appropriés pour améliorer l'efficacité de la formation.

En plus des algorithmes d'intelligence artificielle(optimisation) et de l'apprentissage automatique que beaucoup des approches de la littérature ont utilisés afin de donner à l'apprenant un contenu pédagogique adapté à son profil, l'approche proposée utilise également les interactions sociales de l'apprenant avec son environnement. Contrairement aux méthodes existantes qui se basent uniquement sur les connaissances des apprenants, notre méthode fait l'adaptation en se basant aussi sur leur productivité et leur motivation.

1.5.3 Rôle des réseaux sociaux dans notre plateforme

Comme nous l'avons dit précédemment, notre système d'e-learning adaptatif utilise les réseaux sociaux pour extraire des informations sur l'apprenant, pour cela nous utilisons l'authentification sociale.

L'authentification sociale avec un réseau social est un type d'authentification qui nous permet d'utiliser les informations de connexion existantes d'un utilisateur sur un réseau social tel que Facebook ou Twitter, pour connecter l'utilisateur à un site web au lieu de créer un nouveau compte spécifiquement pour ce site. La connexion sociale est simple et efficace, elle permet aux utilisateurs de s'authentifier sur des sites web sans avoir à créer un compte supplémentaire. Il suffit de cliquer sur un bouton pour s'authentifier[57].

Le bouton d'authentification augmente l'inscription sur une plate-forme. Pourquoi ? simplement parce que le bouton d'authentification élimine la nécessité pour l'utilisateur de remplir un formulaire, choisissez un nom d'utilisateur et un mot de passe sécurisé.

Nous utilisons l'authentification sociale dans notre travail pour récupérer les publications de l'apprenant afin de les analyser, soit pour rechercher la période d'activité, soit pour analyser les sentiments des publications(la motivation).

Productivité

La productivité définit la période de la journée dans laquelle l'apprenant est plus productif(pour rechercher la période du jour où l'apprenant publie de nombreux tweets et publications dans les comptes de ses réseaux sociaux). Nous définissons pour cela trois périodes : (1) de 8h à 12h, (2) de 14h à 18h et (3) de 19h à 23h. L'idée est de calculer le nombre de messages/tweets publiés, appréciés ou commentés par l'apprenant au cours de chacune de ces trois périodes. Après, la productivité est pour la période avec un maximum de messages/tweets. les formules suivantes montrent comment calculer la productivité.

$$NPT = Max[Card(P1/T1), Card(P2/T2), Card(P3/T3)] \tag{13}$$

$$Productivité = \begin{cases} 1 & \text{si } NPT = Card(P1/T1) \\ 2 & \text{si } NPT = Card(P2/T2) \\ 3 & \text{si } NPT = Card(P3/T3) \end{cases}$$

Où :

- NPT : est le nombre des publications(Facebook)/tweets(Twitter).
- $Card(P1/T1), Card(P2/T2), Card(P3/T3)$: est le nombre de messages/tweets publiés, aimés ou commentés au cours de la période 1, 2 et 3 respectivement.
- $Max[Card(P1/T1), Card(P2/T2), Card(P3/T3)$: renvoie le nombre maximum après le calcul de la cardinalité dans chaque période.

Après tout cela, la productivité est égale à 1 si l'apprenant est plus productif pendant la période 1, égal à 2 s'il est plus productif durant la période 2 et égal à 3 s'il est plus productif durant la période 3.

Analyse des sentiments

La deuxième chose à faire après le calcul de la période d'activité est de récupérer toutes les publications publiées au cours de cette période. Après avoir utilisé l'approche proposée (AFINN + WORDNET), nous classons tous les tweets/publications en trois classes (motivé, démotivé ou neutre).

Ensuite, la motivation de l'apprenant au cours de la période d'activité est équivalente à la classe majoritaire des publications du PA. Par exemple, si nous avons 20 publications ou tweets publiés dans la période d'activité, dont 10 sont de classe motivée, 6 de classe démotivée et 4 de classe neutre, la classe majoritaire est donc la classe motivée, puis la motivation de l'apprenant est **motivée**.

La classification des publications de l'apprenant qui sont publiées dans PA en fonction

de trois classes (motivation, démotivation, neutre) donne à l'enseignant de la plate-forme une idée du niveau auquel l'apprenant doit apprendre et cela dépend de sa motivation, bien sûr, s'il est motivé nous pouvons augmenter le niveau d'étude, et l'inverse s'il est démotivé.

Dans les sous-sections ci-dessous, nous présenterons les différentes étapes de notre travail, comment nous combinons les algorithmes génétiques avec les notions de recherche d'informations et appliquons les résultats pour donner à l'apprenant un contenu pédagogique correspondant à son profil, et comment nous utilisons les publications et tweets de l'apprenant pour l'adaptation. Ce travail est réalisé à l'aide du modèle de programmation MapReduce et du système de fichiers distribué Hadoop HDFS.

1.5.4 Recherche d'informations et son utilisation dans notre travail

Comme nous avons dit précédemment, l'utilisation du domaine de la recherche d'informations est importante dans notre travail, que ce soit pour évaluer la population initiale de l'algorithme génétique (AG) ou pour la reproduction d'une nouvelle génération lors de l'application de l'AG.

Comme notre travail consiste à trouver un contenu pédagogique pour un profil d'apprenant, la première chose à faire est de sauvegarder le profil ainsi que les différents objectifs pédagogiques d'un cours en fichiers texte. L'idée est que nous allons commencer à rechercher les objectifs pédagogiques pertinents pour le profil de l'apprenant, c'est comme si nous voulions trouver les documents pertinents pour une requête exprimée par un utilisateur. Dans notre cas, le profil de l'apprenant jouera le rôle de la requête.

Pour construire les profils, chaque apprenant doit passer un quiz sous forme des questions à choix multiples. Les profils sont construits à partir des concepts que les apprenants ne maitrisent pas(à partir des questions que les apprenants y répondent incorrectement). Dans notre système, le profil d'un apprenant est représenté comme un document texte qui contient tous les concepts que l'apprenant doit apprendre.

Au début, c'est comme un système de recherche d'informations (IRS). Nous indexons le profil de l'apprenant et les objectifs pédagogiques en créant des vecteurs représentatifs à l'aide du modèle vectoriel [83], où chaque élément est le poids d'un terme ou d'un concept dans les documents (profil ou objectif pédagogique), nous calculons la similarité entre le profil de l'apprenant et chaque objectif pédagogique en utilisant la similarité de Cosine. Figure II.6 indique les différentes étapes permettant de trouver les objectifs pertinents pour le profil de l'apprenant.

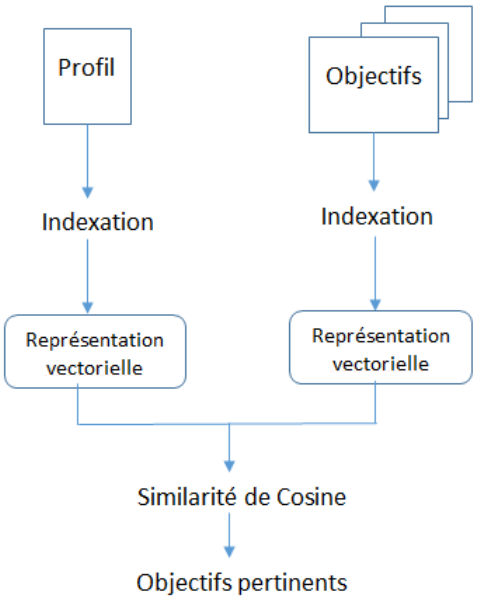


FIGURE II.6 – étape de notre système de recherche d'information

1.5.5 Recommander un contenu pédagogique avec les algorithmes génétiques

La deuxième étape de notre travail est l'application de l'algorithme génétique pour trouver un objectif pédagogique optimal pour le profil de l'apprenant en appuyant sur le résultat du système de recherche d'informations afin de trouver la population initiale.

Après avoir calculé la similarité entre le profil de l'apprenant et chaque objectif pédagogique, nous trions le résultat obtenu par ordre décroissant afin de conserver les huit premiers objectifs qui joueront le rôle de la population initiale. Cette étape est appelée l'étape de sélection des individus les plus adaptés à l'environnement de travail de l'algorithme génétique.

Dans notre problème, les individus sont les objectifs pédagogiques et la fonction de mise en forme(la fonction de fitness) pour évaluer un individu est la similarité de Cosine.

Nous utilisons pour coder le chromosome d'un individu le codage binaire, le gène de chaque chromosome présente un concept du cours par exemple pour le cours de la programmation JAVA ; les classes est un concept, l'héritage est un autre concept, etc. Chaque gène prend la valeur 1 si le concept existe dans l'objectif pédagogique ou 0 s'il n'existe pas, avec la taille de chaque chromosome (nombre de gènes) est égale au nombre de concepts présents dans tous les objectifs pédagogiques (notre corpus). Nous trouvons après le codage chaque objectif pédagogique ainsi que le profil de l'apprenant avec le chromosome équivalent où chaque gène (élément de chromosome) prend soit la valeur 0 ou 1.

Le chromosome est équivalent à la représentation vectorielle du point de vue des systèmes de recherche d'informations car après chaque génération de l'AG, nous calculons la similarité entre le profil et chaque individu (objectif) de la génération.

Les différentes étapes pour trouver l'objectif pédagogique optimal pour le profil de l'apprenant sont les suivantes :

- **Choix de la fonction de fitness :**
  Le choix de la fonction de fitness est important dans les algorithmes génétiques, car c'est celui qui nous donnera une idée sur un individu s'il peut être choisi ou non, et aussi pour trouver une solution optimale au problème étudié. Puisque nous voulons travailler avec les notions des systèmes de recherche d'informations, nous avons choisi d'utiliser la similarité de Cosine en tant que fonction de fitness.
- **Sélection de la population initiale :**
  C'est l'étape de calcul de la similarité entre le profil et chaque objectif pédagogique. Après, nous trions le résultat par ordre décroissant pour trouver les 8 premiers objectifs qui constituent la population initiale.
- **Codage des individus :**
  Le codage utilisé dans notre travail est le codage binaire où chaque élément du chromosome (gène) prend les valeurs 0 ou 1. 1 si un concept est présent dans un objectif pédagogique et 0 sinon. Chaque objectif pédagogique représente un individu de la population et chaque chromosome représente un vecteur dont la taille est égale au nombre de concepts de tous les objectifs pédagogiques.
- **Opération de croisement :**
  Une autre étape qui est également très importante est le croisement qui consiste à mélanger les caractéristiques de deux parents de la génération précédente pour construire deux autres enfants. Nous utilisons dans notre travail le croisement en un seul point.
- **Opération de mutation :**
  Cette opération consiste à modifier la valeur d'un gène pour augmenter la valeur de similarité de son chromosome afin de mieux s'adapter à l'environnement de travail.

Le critère d'arrêt de notre algorithme génétique est le nombre de générations fixé au début, chaque génération étant une itération de l'algorithme. A chaque itération, nous trouvons les individus de la population avec de nouvelles valeurs de similarité.

Notre solution est un objectif pédagogique (chromosome) existe dans la dernière génération, c'est le chromosome de la population finale avec la plus grande valeur de similarité. L'apprenant doit apprendre les concepts équivalents aux gènes qui ont la valeur 1.

La figure II.7 présente les différentes étapes de notre algorithme génétique pour trouver

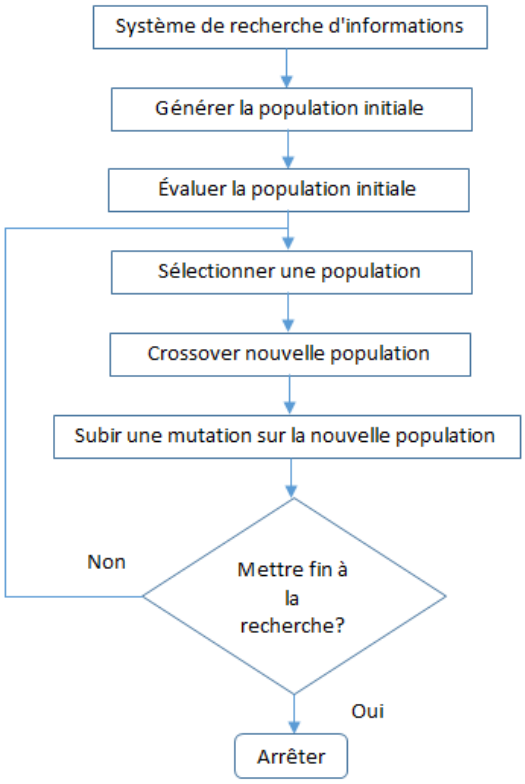


FIGURE II.7 – Le processus de notre algorithme génétique

l'objectif pédagogique optimal.

1.5.6 Réseaux sociaux et analyse des sentiments

Période d'activité : Comme nous l'avons présenté précédemment, la première chose à faire après avoir trouvé l'objectif pédagogique optimal pour le profil de l'apprenant est de calculer la période de la journée où l'apprenant doit apprendre. Pour cela, nous avons proposé d'utiliser les comptes Facebook et Twitter de l'apprenant pour connaître la période dans laquelle l'apprenant publie beaucoup, c'est donc à cette période que l'apprenant est très actif, conséquemment c'est à cette période que l'apprenant peut améliorer facilement son niveau d'apprentissage.

Pour calculer la période d'activité, nous devons extraire les données à partir des profils de l'apprenant. Pour extraire les tweets de Twitter et les publications/commentaires de Facebook, nous avons utilisé Twitter4J et RestFB. comme présentée dans le chapitre 1 de la partie I.

L'API RestFB nécessite une configuration telle que la création d'une application Facebook dans l'espace réservé aux développeurs Facebook[3], cette application contient un nom, un nom de domaine de travail qui doit être le même nom de domaine de notre plate-

3. https ://developers.facebook.com/

FIGURE II.8 – Application Facebook

forme, un identifiant d'application et un identifiant secret d'application. Les deux derniers paramètres sont ceux qui nous permettent de collecter des données à partir du compte de l'apprenant, la figure II.8 illustre une application Facebook sous le nom **madani** :

La deuxième étape nécessaire à l'utilisation de l'API RestFB consiste à installer le bouton de connexion. C'est-à-dire, établir une connexion entre l'application Facebook à l'aide de l'un des SDK proposés par Facebook, comme PHP SDK ou JavaScript SDK. Dans notre travail, nous avons utilisé JavaScript SDK avec JavaEE.

Après avoir créé l'application Facebook et le bouton de connexion, nous pouvons récupérer un paramètre utilisé par RestFB et permettant de récupérer les publications de l'apprenant afin de pouvoir l'utiliser dans la phase de recherche de la période d'activité : c'est l'**ACCESS TOKEN**.

Après toutes ces étapes (soit en utilisant l'API Twitter4J ou l'API RestFB), nous aurons les tweets/publications de l'apprenant, et en utilisant la formule 13, nous pouvons facilement trouver la période d'activité, et aussi la motivation de l'apprenant en se basant sur l'approche proposée(WordNet + AFINN).

1.5.7 Parallélisation de notre travail

Dans une plate-forme d'apprentissage en ligne, il est possible de trouver de nombreux apprenants, ainsi que de nombreux cours, et pour chaque cours, de nombreux chapitres et concepts, d'où une grande quantité de données qui rend l'apprentissage sur la plate-forme inefficace en raison du temps que nous pouvons attendre pour trouver le résultat d'adaptation. Aussi aujourd'hui, tout le monde publie dans les réseaux sociaux donc pour un apprenant, il est possible que nous trouvions beaucoup de publications sur son compte Facebook ou Twitter (énorme volume de publications), de sorte que l'analyse de ses publications et de son profil devient difficile avec les moyens classiques.

Pour remédier à ces problèmes, notre proposition consiste à paralléliser notre algo-

rithme génétique afin de trouver l'objectif pédagogique optimal pour le profil de l'apprenant, ainsi que la classification des publications permettant de trouver le sentiment de l'apprenant au cours de la période d'activité comme présenté dans la section 1.4.

Nous proposons de travailler dans un système Big Data en utilisant les solutions HADOOP : HDFS et le modèle de programmation MapReduce. L'idée est d'installer un cluster Hadoop (ensemble de machines) et de partager le travail d'analyse des publications de l'apprenant et de la recherche de l'objectif pédagogique optimal entre les machines du cluster.

Description de la parallélisation : recherche de l'objectif pédagogique optimal

Nous enregistrons le profil de l'apprenant ainsi que les objectifs pédagogiques d'un cours dans des fichiers texte et les sauvegardons dans HDFS afin de paralléliser le stockage entre les différentes machines du cluster Hadoop. Nous enregistrons également le résultat de l'analyse du profil de l'apprenant dans HDFS en tant que chromosome optimal.

Pour optimiser le volume de stockage des données dans HDFS, nous avons stocké tous les objectifs pédagogiques dans le même fichier au lieu de donner à chaque objectif un fichier, car le système de fichiers (HDFS) donnera à chaque fichier la taille du bloc qui est de 64 Mo. Par exemple si un fichier a une taille de 10 Mo, après le stockage dans HDFS sa taille devient de 64 Mo, ce qui entraîne une augmentation de la taille des données stockées. Nous plaçons donc tous les objectifs dans le même fichier pour optimiser la taille de stockage.

Après qu'un apprenant ait passé un quiz pour un tel cours, il nous donne une idée sur les concepts qu'il ne maîtrisent pas, ces concepts sont stockés dans un ficher texte, qui va jouer le rôle du profile de l'apprenant. De même, tous les objectifs pédagogiques sont stockés dans le même fichier.

Les différentes étapes de la parallélisation de notre travail sont les suivantes :

- **L'enregistrement des données à analyser :**
  Comme je l'ai déjà dit, nous enregistrons le profil de l'apprenant et les objectifs pédagogiques (même fichier texte) dans HDFS.
- **Programme Mapreduce :**
  La parallélisation de notre travail se fait avec le modèle de programmation MapReduce. Dans notre cas, la méthode Map nous permet de parcourir le fichier contenant les objectifs pédagogiques et de les extraire tous. Ensuite, nous envoyons le résultat à la méthode Reduce qui nous permet d'indexer le profil de l'apprenant et ses objectifs pédagogiques, l'application du système de recherche d'informations et l'application de l'algorithme génétique de manière parallèle.

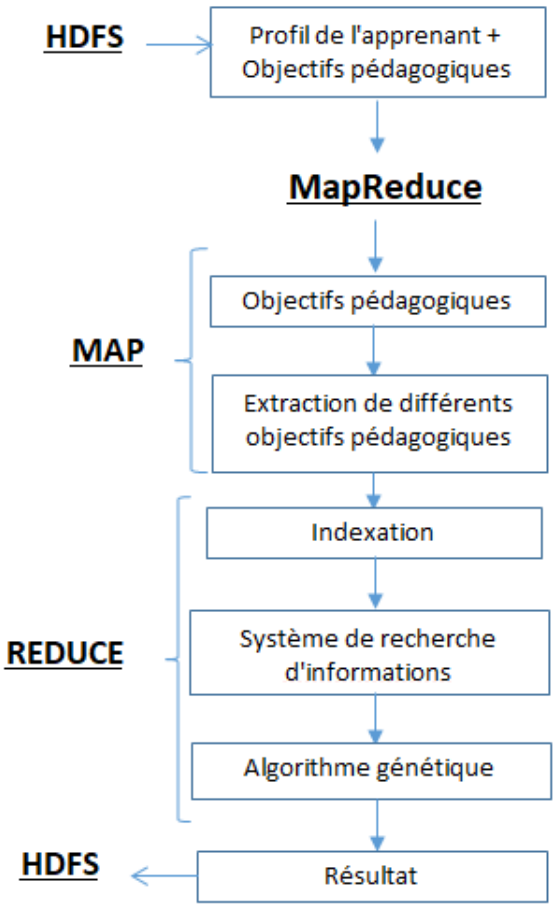


FIGURE II.9 – Étapes de parallélisation

– **Enregistrement du résultat de l'analyse :**
La dernière étape consiste à enregistrer le résultat obtenu après l'opération MapReduce dans HDFS, qui constitue l'objectif pédagogique optimal pour le profil de l'apprenant sous la forme d'un chromosome formé de gènes ayant soit la valeur 1, ce qui signifie que l'apprenant doit apprendre le concept équivalent, ou la valeur 0 sinon.

La figure II.9 indique les différentes étapes permettant de paralléliser l'algorithme d'apprentissage adaptatif.

algorithme mapreduce

Notre algorithme MapReduce suivi pour trouver l'objectif pédagogique optimal pour un apprenant est présenté dans l'algorithme 2 :

Où :

– **Index(List2 & Learner's profile) :** une fonction qui construit la représentation

Algorithm 2 Algorithme d'apprentissage adaptatif

Require: L'objectif pédagogique optimal
——————**-Class Mapper**——————
Method **Map**(Docid,FIleOfObjective)
for all line ∈ FIleOfObjective **do**
  Write(Docid,line)
end for
——————**-Class Reducer**——————
Method **Reduce**(Docid,List(line))
$S \leftarrow NULL$
for all n ∈ List(line) **do**
  $S \leftarrow S + n$
end for
$List2 \leftarrow Split(s)$
Index(List2 & Learner's profile)
$Result \leftarrow BuildingGA()$
Write(Result," ")

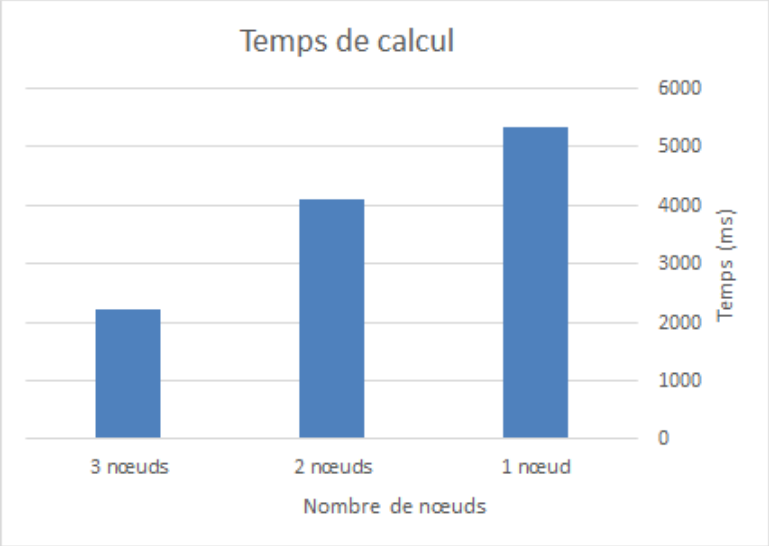


FIGURE II.10 – Temps de calcul

vectorielle de chaque objectif pédagogique et du profil de l'apprenant.

– **BuildingGA() :** l'exécution de l'algorithme génétique.

L'étape de parallélisation de l'analyse des sentiments est présentée dans la section 1.4.

Effet de la parallélisation

Sur la base de la configuration présentée dans la figure II.2, nous avons réalisé une expérience pour démontrer l'effet de la parallélisation. Figure II.10 représente le temps de calcul de notre travail en utilisant respectivement 1,2 et 3 noeuds Hadoop.

La Figure II.10 montre qu'en augmentant le nombre de nœuds dans le cluster, le temps

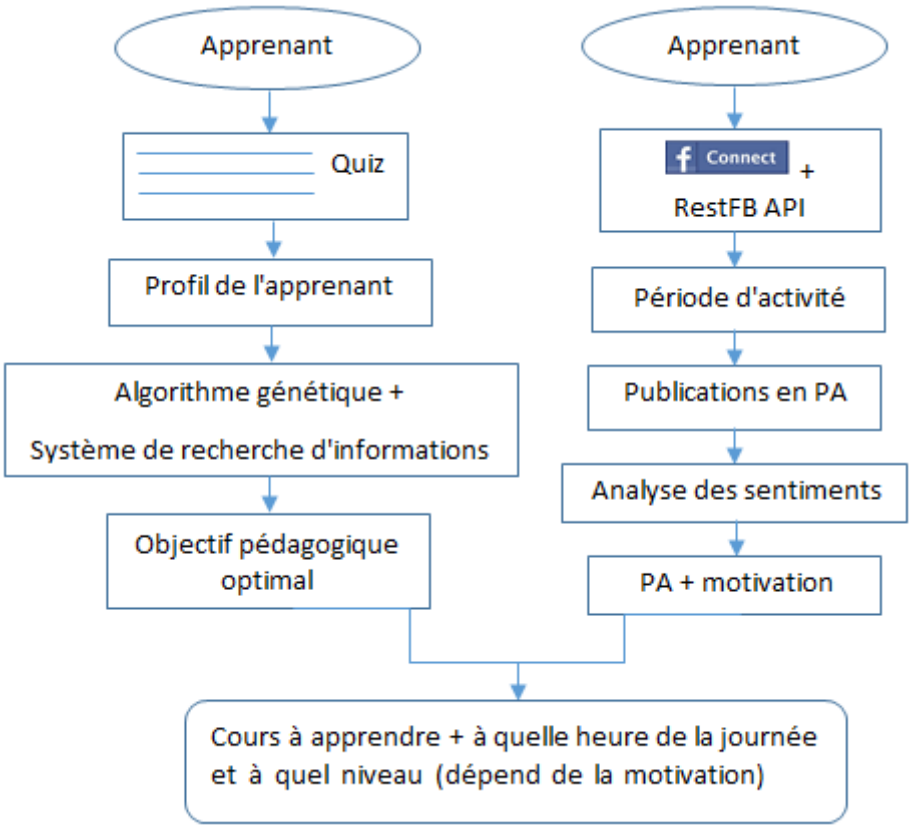


FIGURE II.11 – Différentes étapes de notre système d'apprentissage en ligne adaptatif

de calcul diminue, ce qui montre pourquoi nous avons partagé le travail entre plusieurs machines.

Une information très importante à noter est qu'un programme MapReduce qui fonctionne avec un nœud fonctionne également avec trois nœuds et peut également fonctionner avec un cluster de 100 nœuds. Par conséquent, si nous voulons réduire le temps de calcul, nous devons simplement ajouter un autre nœud au cluster qui est une opération très facile.

1.5.8 Résumé de notre système d'e-learning adaptatif

La figure II.11 présente les différentes étapes permettant d'adapter un cours à un apprenant en fonction de son profil, de la période d'activité et de sa motivation.

Selon la figure II.11, notre système repose sur deux niveaux d'adaptation, le premier sous la forme d'un problème d'optimisation permettant de trouver l'objectif pédagogique optimal pour le profil de l'apprenant, et la seconde consiste à utiliser les données de l'apprenant publiées dans ses comptes de réseaux sociaux pour trouver sa période d'activité et sa motivation.

À la fin de tous ces processus, nous trouvons les cours que l'apprenant doit apprendre,

à quelle période de la journée (période d'activité de l'apprenant) et à quel niveau (dépend de sa motivation).

Dans ce travail, nous avons présenté notre système d'apprentissage adaptatif basé sur un algorithme génétique, les notions des systèmes de recherche d'informations, les données de réseaux sociaux, l'analyse de sentiments et les technologies Big Data (Hadoop, HDFS, MapReduce) ; en recherchant le contenu pédagogique optimal pour un apprenant sur la base de son profil et d'un ensemble d'objectifs pédagogiques fixés par un enseignant, de la période d'activité (quand l'apprenant est actif) et du niveau d'apprentissage de l'apprenant (sa motivation). Nous avons également proposé une nouvelle approche pour paralléliser l'algorithme qui nous permet de rechercher l'objectif pédagogique optimal et de rechercher la motivation de l'apprenant en utilisant un cluster Hadoop de trois nœuds (machines UBUNTU) pour partager le travail entre plusieurs machines. Dans le prochain chapitre, nous présenterons la deuxième contribution de cette thèse, où nous avons proposé deux nouvelles approches basées sur la similarité sémantique afin d'automatiser la classification des tweets sans l'utilisation des méthodes d'apprentissage automatique. Nous présenterons également comment nous avons appliqué une des approches proposées pour réaliser une nouvelle plateforme d'apprentissage en ligne basée sur les notions des systèmes de recommandation.

Chapitre 2

Analyse des sentiments à l'aide de la similarité sémantique et Hadoop MapReduce.

L'automatisation du processus d'analyse des données des réseaux sociaux a connu ces dernières années un développement crucial. Dans ce chapitre, nous proposons deux nouvelles approches pour classer les tweets (recherchez le sentiment exprimé dans un tweet), la première selon trois classes : négative, positive ou neutre, et la seconde selon deux classes : négative ou positive. La première méthode proposée consiste à calculer la similarité sémantique entre le tweet à classer et trois documents, où chaque document représente une classe (contient les mots qui représentent une classe), après le calcul de la similarité, le tweet prend la classe du document qui a la plus grande valeur de la similarité sémantique avec il. La seconde méthode consiste à calculer la similarité sémantique entre chaque mot du tweet à classer et les mots "positif" et "négatif" en proposant une nouvelle formule. Nous décidons de faire l'analyse d'une manière parallèle et distribuée, en utilisant le framework Hadoop avec le système de fichiers distribué Hadoop et le modèle de programmation MapReduce pour résoudre le problème du temps de calcul de l'analyse si le jeu de données des tweets est très volumineux.

2.1 Système de recherche d'information et similarité sémantique

L'analyse des sentiments des tweets est devenue un domaine de recherche scientifique très actif ces dernières années, en particulier avec l'augmentation du nombre d'utilisateurs sur Twitter et le nombre de tweets publiés quotidiennement pour exprimer des critiques ou des opinions. Pour cela, de nombreux scientifiques cherchent à proposer des nouvelles approches pour apporter des solutions aux problèmes liés à ce domaine de recherche.

Dans ce travail, nous proposons deux nouvelles approches pour classer les tweets selon

trois classes : positive, négative ou neutre ; c'est-à-dire une classification de subjectivité, ou selon deux classes positive et négative ; qui est une classification de polarité, d'une manière parallèle en partageant le travail de classification entre un ensemble de machines (cluster Hadoop). La première approche proposée consiste à calculer la similarité sémantique entre chaque tweet à classer et trois documents, chacun contenant des mots représentant une classe, c'est-à-dire un document pour la classe positive, un pour la classe négative et le troisième pour la classe neutre. Après le calcul de la similarité sémantique, le tweet prend la classe du document qui a la haute valeur de la similarité sémantique avec il. La seconde approche consiste à proposer une nouvelle formule qui permet de calculer la soustraction entre la somme de la similarité sémantique de chaque mot du tweet et le mot "positif", et la somme de la similarité sémantique de chaque mot du tweet et le mot "négatif", après cela, si la valeur trouvée est positive, le tweet est positif et si elle est négative, le tweet est négatif.

La méthode proposée consiste donc à combiner plusieurs domaines, tels que la similarité sémantique avec l'utilisation de WordNet, les systèmes de recherche d'informations, l'analyse des sentiments ou l'extraction d'opinions et le Big Data.

A partir des principes des systèmes de recherche d'informations et comme comparaison avec notre travail, la requête est le tweet à classer, et la base de données de documents sont dans notre cas les trois documents d'opinions qui représentent les trois classes (positive, négative ou neutre). Pour le calcul de la similarité, nous utilisons la similarité sémantique en se basant sur la ressource sémantique Wordnet [1], et l'une des approches existantes dans la littérature qui calcule la similarité sémantique.

2.1.1 Similarité sémantique

Nos deux méthodes proposées pour la classification des tweets reposent toutes les deux sur le calcul de la similarité sémantique. Dans cette sous-section, nous présentons quelques approches existantes dans la littérature permettant de calculer la similarité sémantique.

La mesure de similarité de Wu et Palmer [99] repose sur le principe suivant : étant donné une ontologie W formée d'un ensemble de nœuds et d'un nœud racine R (Figure II.12). X et Y sont deux éléments de l'ontologie pour laquelle nous voulons calculer leur similarité. Le principe du calcul de similarité est basé sur les distances (N1 et N2) séparant les nœuds X et Y du nœud racine (R) et la distance entre le concept subsumant (CS) ou le concept le plus spécifique (CPS) de X et Y du noeud R.

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

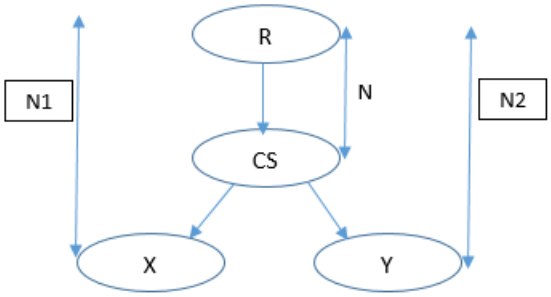


FIGURE II.12 – Exemple d'extrait d'ontologie

La mesure de Wu et Palmer est définie par la formule suivante :

$$Sim_{Wu\&Palmer} = \frac{2 * N}{N_1 + N_2} \tag{14}$$

Resnik [77] a utilisé la notion de distance sémantique de la manière suivante : deux concepts sont plus similaires si la valeur de la distance sémantique qui les sépare est faible. La similarité est définie par rapport à la longueur des chemins qui relient deux concepts dans la hiérarchie. La similarité entre deux concepts c_1 et c_2 est la suivante :

$$Sim(c_1, c_2) = 2 * D - Len(c_1, c_2) \tag{15}$$

Où D est la longueur maximale des chemins possibles qui connectent c_1 et c_2, et $Len(c_1, c_2)$ est le plus petit chemin entre c_1 et c_2.

Lin [52] a défini une mesure de similarité différente de celle de Resnik par cette formule :

$$Sim_{Lin} = \frac{2 * \log(P(CS(c_1, c_2)))}{\log(P(c_1)) + \log(P(c_1))} \tag{16}$$

Cette mesure utilise une approche hybride qui combine deux sources de connaissances différentes(thésaurus, corpus).

Jiac [38] a mis au point une nouvelle formule qui associe l'entropie (contenu informationnel) du concept spécifique à celle des concepts pour lesquels nous cherchons la similarité. Cette approche est calculée par la formule suivante :

$$Sim_{jiac}(c_1, c_2) = \frac{1}{distance(c_1, c_2)} \tag{17}$$

La distance entre c_1 et c_2 est calculée à l'aide de la formule suivante :

$$distance(c_1, c_2) = E(c_1) + E(c_2) - (2 * E(CS(c_1, c_2))) \tag{18}$$

Dans [31], les auteurs ont proposé une nouvelle mesure de la similarité sémantique.

L'idée de base de cette mesure est que si deux concepts sont reliés par un chemin très court et qu'il "ne change pas le sens", les deux concepts sont similaires. Le calcul de la similarité est basé sur le poids du plus court chemin menant d'un concept à un autre et sur le nombre de changements de direction.

$$Sim_{Hirst-st}(c_1, c_2) = T - PCC - K * D \tag{19}$$

Où T et K sont des constantes, PCC est la distance du plus court chemin en nombre d'arcs et D est le nombre de changements de direction.

Une autre méthode pour calculer la similarité sémantique est celle proposée par Leacock et Chodorow [51], qui est basée sur la longueur du plus court chemin entre deux synsets de Wordnet. Les auteurs ont limité leur attention aux liens hiérarchiques "is-a" ainsi qu'à la longueur du chemin par la profondeur globale P de la taxonomie. La formule est définie par :

$$Sim_{lc} = -\log(\frac{cd(c_1, c_2)}{2 * M}) \tag{20}$$

Où M est la longueur du plus long chemin séparant le concept racine de l'ontologie du concept le plus en bas. On note $cd(c_1, c_2)$ la longueur du plus court chemin séparant c_1 de c_2.

2.1.2 Choix de l'approche

Le choix d'une approche parmi celles proposées dans la littérature est une tâche très importante dans notre travail car elle joue un rôle fondamental dans la classification des tweets. Pour faciliter le choix d'une approche par rapport à une autre(l'approche optimale), nous essayons de calculer la similarité sémantique entre deux documents identiques et de choisir la ou les approches qui donnent de bons résultats. Le tableau II.4 illustre les résultats obtenus.

Mesure / Approches	Similarité	Temps d'exécution(ms)
Leacock et Chodorow	0.14	1016
Wu et Palmer	0.13	1297
Resnik	0.04	1360
JiangConrath	0.04	1391
Lin	0.02	1344

TABLE II.4 – Similarité et temps d'exécution en (ms) avec différentes approches

D'après ce tableau, nous remarquons que la meilleure approche est celle de Leacock et Chodorow, qui offre la plus grande similarité avec un temps d'exécution minimal. L'approche de Wu et Palmer donne également de bons résultats.

A partir de ces résultats, nous avons choisi de travailler avec l'approche **Leacock et Chodorow** à l'étape de classification.

2.2 revue de la littérature et motivation

L'utilisation des réseaux sociaux dans l'analyse des sentiments devient explosive et de nombreux chercheurs les utilisent dans différents domaines, par exemple, pour prédire les résultats d'une élection, extraire des opinions sur un produit ou une marque, ou dans un système d'e-learning comme présenté dans[56].

Ces dernières années, plusieurs auteurs ont utilisé la sémantique dans l'analyse des sentiments pour réduire les problèmes liés au traitement du langage naturel (NLP). Les résultats expérimentaux montrent que l'utilisation de la sémantique dans l'analyse des sentiments améliore les résultats de la classification des tweets. Les auteurs de[81] ont proposé une nouvelle approche qui ajoute la sémantique dans la classification en tant que caractéristiques supplémentaires à l'ensemble d'apprentissage pour l'analyse des sentiments. Les résultats montrent une augmentation moyenne du score de précision harmonique pour l'identification des sentiments négatifs et positifs d'environ 6,5% et 4,8% par rapport aux lignes de base des traits unigrammes et des parties de discours(POS tags) respectivement. Après cela, ils comparent son approche à une approche basée sur l'analyse de sujets porteurs des sentiments et ils découvrent que les caractéristiques sémantiques améliorent le rappel et F-score dans la classification des sentiments négatifs, ainsi que la précision avec un score inférieur au rappel et F-score dans la classification des sentiments positifs.

Dans [82], les chercheurs ont proposé une nouvelle approche qui capture automatiquement les modèles de mots de sémantique contextuelle et de sentiment similaire dans les tweets. Leur approche extrait des modèles de la sémantique contextuelle et des similarités des sentiments entre les mots d'un corpus d'un tweet donné.

Les auteurs de [92] ont proposé une approche sémantique pour découvrir les attitudes des utilisateurs et les informations commerciales issues des médias sociaux en Arabe standard et dialectal. Ils présentent également la première version de leur ontologie arabe (ASO) qui contient différents mots exprimant des sentiments et la force avec laquelle ils sont exprimés. Ils montrent ensuite que leur approche est utilisable pour classer différents flux Twitter sur différents sujets.

Comme présenté au chapitre 4 de la partie I, la plupart des approches existantes uti-

lisent les algorithmes bien connus d'apprentissage automatique pour classer les tweets ou les phrases en classes, ces techniques sont basées dans la plupart du temps sur des données étiquetées(apprentissage supervisé), et parfois ces données sont mal étiquetées ce qui affecte négativement la classification. Notre travail consiste à proposer deux nouvelles approches pour la classification des tweets (positif, négatif ou neutre) sans utiliser les algorithmes d'apprentissage automatique. L'idée est de tirer parti de la sémantique et des notions des systèmes de recherche d'informations pour proposer des méthodes optimales donnant de bons résultats si on les compare à celles utilisant des algorithmes d'apprentissage automatique.

2.3 Utilisation de la similarité sémantique pour analyser les réseaux sociaux

Dans cette section, nous présentons la méthodologie de notre travail avec les différentes étapes à suivre pour la classification des tweets (extraction des émotions, des attitudes ...). La classification est faite sémantiquement en utilisant Wordnet et l'approche **Leacock et Chodorow**, et d'une manière parallèle en parallélisant la classification entre plusieurs machines à l'aide d'un cluster Hadoop avec le système de fichiers Hadoop distribué (HDFS) pour stocker les tweets à classer et le résultat de la classification. Le modèle de programmation MapReduce est utilisé pour le développement et la parallélisation de nos méthodes. Nos deux approches sont basées sur la langue anglaise.

Chaque approche proposée représente un type de l'analyse des sentiments. La première approche qui classe les tweets en trois classes (positive, négative et neutre) est une **classification de subjectivité**, c'est-à-dire qu'elle classe les tweets en deux catégories ; la catégorie des tweets sentimentaux ou subjectifs (classification de polarité) qui expriment un sentiment ou une opinion sur quelque chose, ils sont classés en deux classes (positive ou négative) ; et la catégorie des tweets factuels ou des tweets objectifs (aucun sentiment) qui n'expriment aucun sentiment ou opinion, et ils sont classés en une seule classe(neutre). La seconde approche classe les tweets seulement en deux classes (positive ou négative) qui est appelée une classification de polarité.

2.3.1 Première approche proposée

La première approche consiste à classer les tweets selon trois classes (positive, négative ou neutre), l'idée proposée consiste à calculer la similarité sémantique entre le tweet et trois documents d'opinions (document positif Dp, document négatif Dn et Document

neutre Dne), chaque document représente une classe, c'est-à-dire que chaque document contient des mots qui représentent une classe. Par exemple, le document de la classe positive contient les mots : positive, good, happy...etc, et le document de la classe négative contient les mots : negative, bad, sad...etc.

Cette approche consiste à créer une méthode hybride basée sur les notions des systèmes de recherche d'informations et de la similarité sémantique en utilisant le dictionnaire WordNet et l'approche **Leacock et Chodorow**. Le principe est comme si nous voulions calculer la similarité sémantique entre un besoin d'utilisateur (requête) et un ensemble de documents permettant de trouver ceux qui sont pertinents pour la requête du point de vue des systèmes de recherche d'informations (SRI).

Pour classer un tweet, nous calculons la similarité sémantique entre celui-ci et chaque document d'opinion (Dp, Dn et Dne). Le tweet prend la classe du document qui a la plus grande valeur de la similarité sémantique avec il.

La formule suivante montre comment nous classons les tweets selon la première approche proposée.

$$CV = \max\left[Sim_{LC}(t, d_p), Sim_{LC}(t, d_n), Sim_{LC}(t, d_{ne})\right] \tag{21}$$

Où :

- CV : (Classification Value) valeur de la classification (positive, négative, neutre).
- $Sim_{LC}(t, d_p)$: est la similarité sémantique entre le tweet à classer et l'un des trois documents en utilisant l'approche **Leacock et Chodorow**.
- t : le tweet à classer.
- d_p, d_n, d_{ne} : représentent respectivement le document de la classe positive, le document de la classe négative et le document de la classe neutre.

La figure II.13 illustre les différentes étapes permettant de classer les tweets en utilisant la première approche proposée.

DP, DN et DNE représentent les trois documents qui représentent les trois classes.

Selon la fig. 3, la première étape est la collection des tweets à analyser (classifier), en utilisant l'API Twitter4J et Apache Flume. La deuxième étape après la collecte des tweets est la création des trois documents d'opinions, chaque document étant un fichier texte contenant des mots représentant une classe, c'est-à-dire que pour le document de la classe positive, nous y mettons des mots tels que : happy, good, happiness, kindness, etc. La troisième étape est l'application des méthodes de pré-traitement du texte comme présenté dans le chapitre 2 de la partie I.

Après l'étape de pré-traitement du texte, c'est l'étape de calcul de la similarité sémantique entre le tweet à classer et chaque classe (document) en utilisant Wordnet et

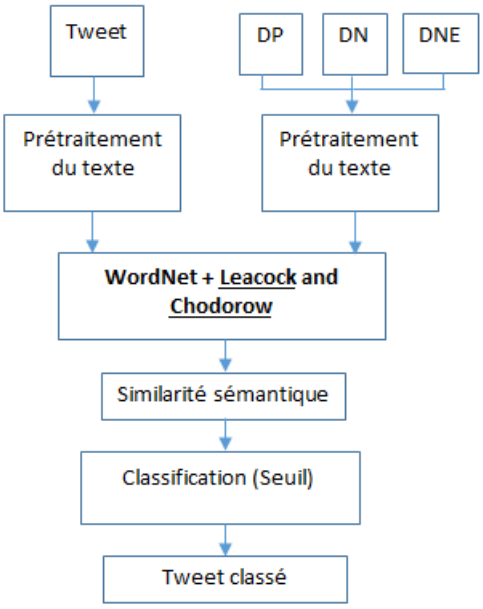


FIGURE II.13 – Étapes de classification pour la première approche

l'approche **Leacock et Chodorow**. Après le calcul de la similarité sémantique, le tweet prendra la classe du document qui a la plus grande valeur de similarité avec il.

Cette première approche tire parti des notions des systèmes de recherche d'informations et des meilleurs résultats obtenus par l'approche **Leacock et Chodorow** comme présentée précédemment, sans oublier que cette approche utilise WordNet qui donne de bons résultats pour le calcul de la similarité sémantique.

Exemple de classification

Nous supposons que nous voulons classifier le tweet qui dit : "I am happy and motivated", en utilisant la première approche proposée. Telle que présentée précédemment, la première chose à faire est le pré-traitement du texte pour éliminer le bruit du tweet, après cela, le tweet devient : "happy, motivated". La prochaine étape est la création des trois documents représentant les trois classes : D_p pour la classe positive, D_n pour la classe négative et D_{ne} pour la classe neutre. Puis, c'est l'étape du calcul de la similarité sémantique entre le tweet et chaque document.

$$\begin{cases} S_p = Sim_{LC}(tweet, D_p) \\ S_n = Sim_{LC}(tweet, D_n) \\ S_{ne} = Sim_{LC}(tweet, D_{ne}) \end{cases}$$

Après, nous cherchons le maximum de ces trois mesures, et le tweet prend le sentiment de la plus grande mesure, par exemple, si S_p est la valeur maximale, le tweet est positif.

2.3.2 Deuxième approche proposée

La seconde approche consiste à classer les tweets selon deux classes : positive et négative (classification de polarité). Nous proposons une nouvelle formule permettant de classer les tweets en fonction de la similarité sémantique avec WordNet et de la méthode **Leacock et Chodorow**. L'idée de cette nouvelle approche est que, après l'application des différentes méthodes de pré-traitement du texte, nous calculons la différence entre la somme de la similarité sémantique entre chaque mot du tweet et le mot "positif", et la somme de la similarité sémantique entre chaque mot du tweet et le mot "négatif", telle que présentée dans la formule suivante :

$$CV = \sum_{i=1}^{N} Sim_{LC}(W_i, positive) - \sum_{i=1}^{N} Sim_{LC}(W_i, negative) \tag{22}$$

Où :

- N : est le nombre de mots dans le tweet.
- CV : valeur de la classification (Classification value.).
- W_i : est le mot numéro i dans le tweet.
- $Sim_{LC}(W_i, positive)$: est la similarité sémantique utilisant l'approche **Leacock et Chodorow** entre le mot i du tweet et le mot "positif".

Pour chaque tweet à classer, la première chose à faire est le pré-traitement du texte afin de supprimer le bruit qui y est présent et d'extraire les mots d'opinions. Après l'étape de pré-traitement du texte, pour chaque mot du tweet, nous appliquons la formule proposée. Si la valeur de la formule après le calcul est supérieure à 0, le tweet est positif. Sinon, si sa valeur est inférieure à 0, le tweet est négatif, comme indiqué dans la formule suivante :

$$Sentiment = \begin{cases} "positive" & \text{si } CV > 0 \\ "negative" & \text{si } CV < 0 \end{cases}$$

L'idée principale de cette approche est de calculer la positivité et la négativité du tweet. Pour calculer la positivité, notre proposition consiste à calculer la similarité sémantique entre chaque mot du tweet et le mot "**positif**", puis calculer la somme de ces similarités sémantiques pour trouver la valeur de la positivité du tweet (la première somme de la formule 22). Pour la négativité, c'est la même procédure mais cette fois nous calculons la similarité sémantique entre chaque mot du tweet et le mot "**négatif**(la deuxième somme de la formule 22). Après, pour classer le tweet, nous calculons la différence entre la positivité et la négativité. S'il est positif (supérieur à 0), le tweet est positif et s'il est négatif (inférieur à 0), le tweet est négatif.

Cette approche par rapport à la première approche présente l'avantage de la complexité et du temps de calcul car il n'est pas nécessaire de calculer la similarité sémantique entre chaque mot du tweet et chaque mot des trois documents d'opinions(comme dans la première approche), mais seulement entre deux mots "positif" et "négatif" ce qui optimise la complexité et le temps de calcul. Cette approche tire également parti du WordNet et de l'approche de **Leacock et Chodorow**.

Figure II.14 montre les différentes étapes pour classer un tweet avec la deuxième approche proposée.

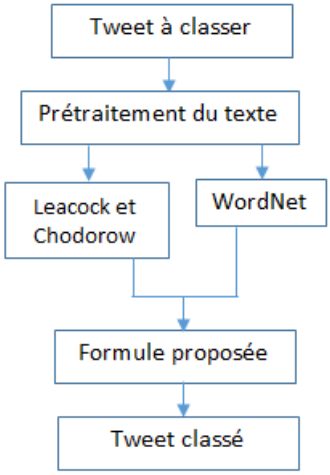


FIGURE II.14 – Étapes de classification pour la deuxième approche

Exemple de classification

Nous supposons que nous voulons classer le même tweet que précédemment : "I am happy and motivated", en utilisant la deuxième approche proposée. Comme présenté précédemment, la première chose à faire est le pré-traitement du texte pour éliminer le bruit du tweet. Ainsi, après cela, le tweet devient : "happy, motivated". C'est-à-dire que sa taille devient 2, donc N=2.

La prochaine étape est l'application de la formule proposée pour trouver le sentiment(classe) du tweet :

$$\begin{aligned} CV &= \sum_{i=1}^{N} Sim_{LC}(W_i, positive) - \sum_{i=1}^{N} Sim_{LC}(W_i, negative) \\ &= Sim_{LC}(happy, positive) + Sim_{LC}(motivated, positive) - Sim_{LC}(happy, negative) \\ &- Sim_{LC}(motivated, negative) \end{aligned}$$

Après le calcul de la valeur de classification, si CV est positif (la positivité est su-

périeure à la négativité), le tweet est **positif**, et il est **négatif** sinon (la négativité est supérieure à la positivité).

2.3.3 Parallélisation de la classification

Nous avons décidé de paralléliser le travail de classification des tweets avec nos deux méthodes proposées en travaillant dans un système Big Data, avec l'utilisation du framework Hadoop. La configuration de notre cluster Hadoop est la même que celle présentée dans la figure II.2.

Étapes de la parallélisation de la première approche

La figure II.15 montre les différentes étapes pour paralléliser la première méthode en utilisant HDFS et Hadoop MapReduce.

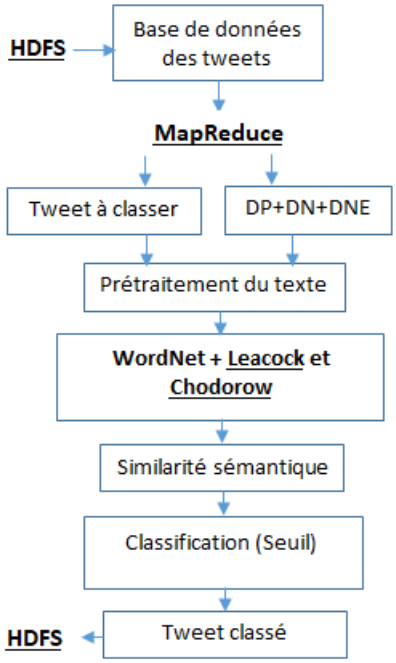


FIGURE II.15 – Étapes de parallélisation de la première approche

Selon la figure II.15, la première étape de la parallélisation consiste à stocker la base de données des tweets à classer dans HDFS afin de partager le stockage entre plusieurs machines (cluster Hadoop). Dans cette étape, nous utilisons l'API Twitter4j et Apache Flume pour collecter les tweets. Après l'étape de pré-traitement du texte, c'est l'étape de la classification en appliquant la méthode proposée mais cette fois avec le modèle de programmation MapReduce (d'une manière parallèle).

L'entrée de l'opération MapReduce à chaque itération (pour chaque tweet) contient un tweet à classer, et la sortie contient le tweet classifié. La classification est effectuée avec

l'application de la première approche proposée, le résultat de classification est stocké dans HDFS. À la fin du processus de classification, nous stockons le résultat dans HDFS sous forme de deux colonnes, une pour le tweet en tant que clé de l'algorithme MapReduce et l'autre pour la classe du tweet en tant que valeur (positive, négative ou neutre).

Notre algorithme MapReduce suivi pour la classification des tweets avec la première approche est présenté dans l'**algorithme 3** :

Algorithm 3 Approche 1

Require: tweet's class
  Cp ← 0
  Cn ← 0
  Cne ← 0
  tweet ← TextPreProcesssing(tweet)
  SE[] ← Split(tweet)
  for all word ∈ SE **do**
    for all W1 ∈ FilePositive **do**
      Cp ← Cp+$Sim_{LC}(word, W1)$
    end for
    for all W2 ∈ FileNegative **do**
      Cn ← Cn+$Sim_{LC}(word, W2)$
    end for
    for all W3 ∈ FileNeutral **do**
      Cne ← Cne+$Sim_{LC}(word, W1)$
    end for
  end for
  if max(Cp,Cn,Cne)=Cp **then**
    sentiment ← positive
  else if max(Cp,Cn,Cne)=Cn **then**
    sentiment ← negative
  else if max(Cp,Cn,Cne)=Cne **then**
    sentiment ← neutral
  end if
  write(tweet,sentiment)

Où :

- $Sim_{LC}(word, W1)$: est une fonction qui calcule la similarité sémantique entre le tweet à classer et l'un des trois documents à l'aide de l'approche de **Leacock et Chodorow**.
- max(Cp, Cn, Cne) : est une fonction qui renvoie le maximum des trois mesures.

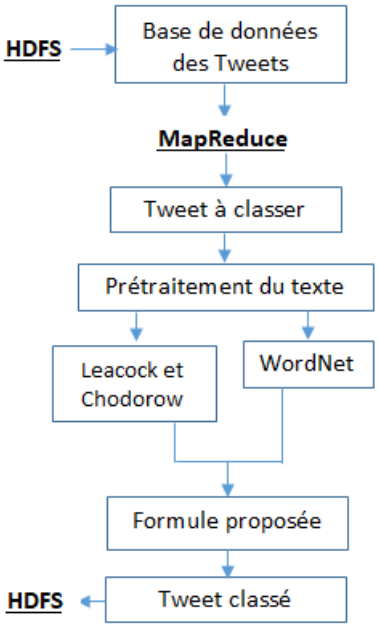


FIGURE II.16 – Étapes de parallélisation de la deuxième approche

Étapes de la parallélisation : deuxième approche

La figure II.16 montre les différentes étapes pour paralléliser la deuxième méthode en utilisant HDFS et Hadoop MapReduce.

Notre algorithme MapReduce suivi pour la classification des tweets avec la deuxième approche est présenté dans l'**algorithme 4** :

2.4 Résultats expérimentaux

Dans cette section, nous présenterons les résultats expérimentaux de nos travaux en comparant les résultats de nos deux approches avec d'autres méthodes, telles que l'utilisation du dictionnaire AFINN simple ou avec l'utilisation d'AFINN et Wordnet[58] (l'approche proposée dans le chapitre précédent).

Pour cela, nous calculons le taux de classification, le taux d'erreur, la précision, le rappel et le F1-score de la classification des tweets en utilisant les deux approches proposées et en les comparant aux approches utilisant le dictionnaire AFINN avec et sans WordNet. Nous présentons également dans cette section les effets des méthodes de pré-traitement du texte sur la classification et comment des méthodes telles que la suppression des mots vides, stemming ou l'effet de la négation peuvent améliorer la qualité des deux approches proposées.

La classification se fera d'une manière parallèle à l'aide du framework Hadoop avec HDFS et du modèle de programmation MapReduce d'un jeu de données contenant des

Algorithm 4 Approche 2

Require: tweet's class
S ← 0
$S_1 \leftarrow 0$
$S_2 \leftarrow 0$
tweet ← TextPreProcesssing(tweet)
SE[] ← Split(tweet)
for all word ∈ SE **do**
$S_1 \leftarrow S_1 + Sim_{LC}(word, positive)$
$S_2 \leftarrow S_2 + Sim_{LC}(word, negative)$
end for
S ← S_1 - S_2
if $S > 0$ **then**
sentiment ← positive
else if $S < 0$ **then**
sentiment ← negative
end if
write(tweet,sentiment)

tweets de différents sujets collectés par Twitter4j API et Apache Flume. Ces deux outils nous permettent de collecter facilement les tweets sur Twitter comme présenté précédemment, par exemple, la collecte de tweets liés à un événement, une élection, des produits ou des marques en général. Nous pouvons également collecter des tweets publiés au cours d'une période donnée, par exemple entre 2015 et 2017.

La plupart des approches d'analyse des sentiments existent dans la littérature utilisent les algorithmes d'apprentissage automatique pour classer les tweets(voir chapitre 4 de la partie I). Pour cela, et avant de comparer nos deux approches aux approches basées sur AFINN, nous faisons une expérience qui consiste à comparer les résultats obtenus avec l'utilisation des algorithmes d'apprentissage automatique bien connus et AFINN pour la classification d'une base de données de tweets contenant 400 tweets positifs et 400 tweets négatifs, ces 800 tweets ont été choisis au hasard de la base de données appelé **Sentiment140**, disponible à l'adresse http ://help.sentiment140.com/for-students. La figure II.17 montre le résultat du taux de précision après l'application des 3 algorithmes d'apprentissage automatique bien connus et du dictionnaire AFINN.

Selon la figure II.17, le dictionnaire AFINN surpasse les 3 autres algorithmes d'apprentissage automatique avec un taux de précision élevé (78,5% pour les tweets négatifs et 79,75% pour les tweets positifs). Ces résultats démontrent pourquoi nous avons décidé de comparer nos deux approches proposées avec des approches basées sur AFINN pour montrer la qualité de nos travaux.

Pour démontrer l'effet de l'application des méthodes de pré-traitement du texte sur

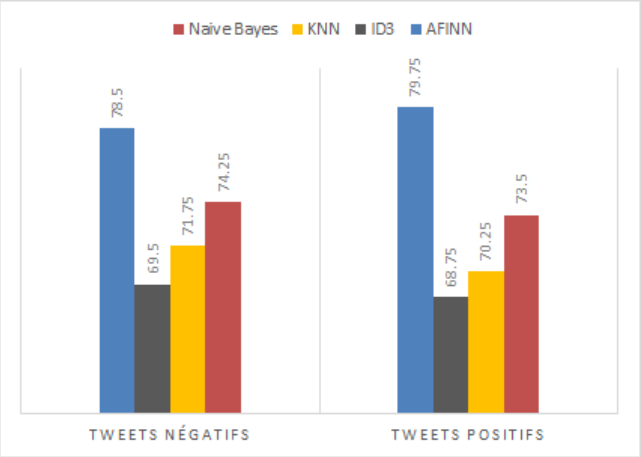


FIGURE II.17 – Taux de précision à l'aide du dictionnaire AFINN et d'algorithmes d'apprentissage automatique

la classification et comment elles peuvent en améliorer la qualité, nous avons réalisé une expérience qui consiste à calculer le nombre des tweets mal classés avec et sans l'utilisation des méthodes du prétraitement du texte ; en utilisant nos deux approches proposées et les deux approches basées sur le dictionnaire AFINN (sans et avec wordnet). Le tableau II.5 montre les résultats obtenus.

Méthodes	Avec pré-traitement de texte	Sans pré-traitement de texte
AFINN	15	26
AFINN+WordNet	10	15
Première approche proposée	5	10
deuxième approche proposée	4	9

TABLE II.5 – Effet du pré-traitement du texte sur la classification

Selon le tableau II.5, et en utilisant toutes les approches, le nombre des tweets mal classés diminue lorsque nous utilisons les méthodes de pré-traitement du texte (par exemple, en utilisant la première approche proposée, le nombre des tweets mal classés est de 10 sans pré-traitement du texte mais après son utilisation ce nombre est diminué à 5). De là, on constate que l'utilisation des méthodes de pré-traitement de texte (suppression des mots vides, stemming, effet de négation ...) a un effet sur la classification, elle permet de diminuer le bruit qui existe dans les tweets et le nombre des tweets mal classés. c'est-à-dire le pré-traitement du texte augmente la qualité de la classification.

Une autre remarque du tableau II.5 est que le nombre des tweets mal classés diminue en utilisant nos deux approches proposées par rapport à AFINN et WordNet (de 15 et 10 à 5 et 4). c'est-à-dire que nos deux méthodes surpassent les approches qui utilisent AFINN.

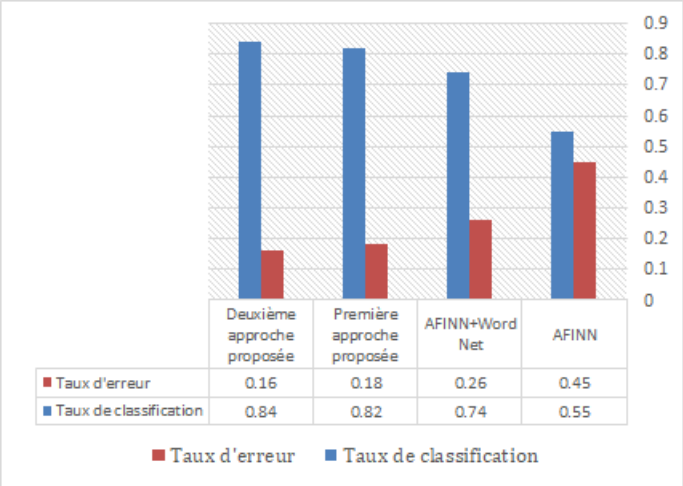


FIGURE II.18 – Classification et taux d'erreur sans pré-traitement du texte

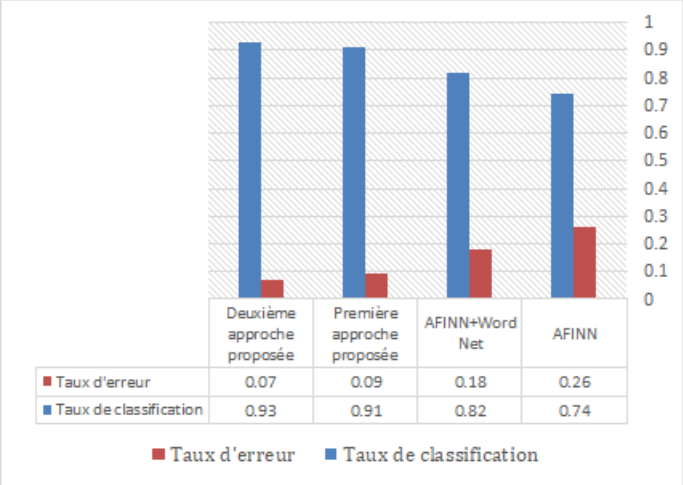


FIGURE II.19 – Taux de classification et d'erreur avec pré-traitement du texte

À partir de ces résultats, nous calculons le taux de classification (Classification rate CR) et le taux d'erreur(Error rate) pour les approches proposées en comparaison avec AFINN (avec et sans l'utilisation de WordNet).

La figure II.18 montre les résultats obtenus pour CR et ER en utilisant les approches proposées. Nous comparons les résultats obtenus avec les approches AFINN simple (sans Wordnet) et AFINN avec Wordnet, sans appliquer les méthodes de prétraitement du texte.

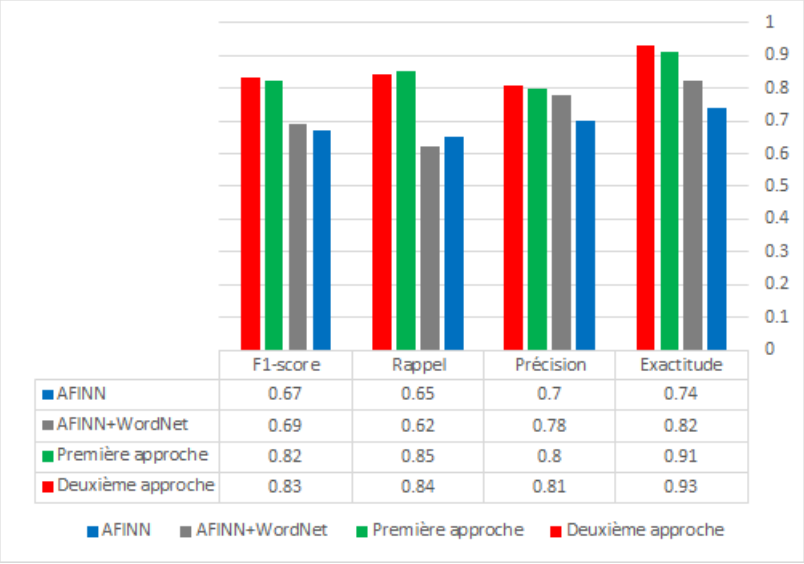
D'après la figure II.18, nous remarquons que nos méthodes surpassent la méthode qui utilise le dictionnaire AFINN avec et sans sémantique (WordNet), avec un taux de classification élevé (0.84 et 0.82) et un taux d'erreur faible (0,16 et 0,18).

Pour démontrer l'effet du pré-traitement du texte sur la classification des tweets, nous proposons de répéter la même expérience que celle illustrée à la figure II.18 mais cette fois avec l'utilisation des différentes méthodes de pré-traitement de texte. La figure II.19 montre les résultats obtenus.

À partir de la figure II.19 et à titre de comparaison avec la figure II.18, nous remarquons que les méthodes de pré-traitement du texte améliorent la qualité de la classification, c'est-à-dire qu'elles augmentent le taux de classification (0.93 et 0.91) et abaissent le taux d'erreur (0.07 et 0.09), sans oublier que nos deux méthodes proposées surpassent les approches basées sur AFINN avec et sans Wordnet avec un taux de classification égale à 93% et 91%.

Pour évaluer un système de classification des sentiments, CR et ER ne suffisent pas, les mesures les plus couramment utilisées dans l'évaluation des approches de l'analyse des sentiments sont : l'exactitude(Accuracy), le rappel, la précision et le F1-score(voir le chapitre 3 de la partie I).

Pour calculer ces 4 paramètres, nous construisons un jeu de données de tweets classés. Pour cela, nous avons utilisé deux bases de données contenant des tweets classés : la première est **Sentiment140**, disponible à l'adresse http ://help.sentiment140.com/for-students, la deuxième est fourni par Twitter Sentiment Analysis Training Corpus, qui peut être téléchargé à l'adresse http ://thinknook.com/twitter-sentiment-analysis-training-corpus dataset-2012-09-22/. Nous avons choisi au hasard 1000 tweets de chaque type (négatif, positif ou neutre) pour les utiliser dans nos expériences. La figure II.20 montre les résultats obtenus après la classification de ces tweets en utilisant nos deux approches proposées et les approches basées sur le dictionnaire AFINN.



	F1-score	Rappel	Précision	Exactitude
AFINN	0.67	0.65	0.7	0.74
AFINN+WordNet	0.69	0.62	0.78	0.82
Première approche	0.82	0.85	0.8	0.91
Deuxième approche	0.83	0.84	0.81	0.93

FIGURE II.20 – Evaluation de notre système SA

La figure II.20 montre que nos deux méthodes proposées donnent de bons résultats pour les 4 indices (exactitude, précision, rappel et F1-score) par rapport aux approches basées sur le dictionnaire AFINN. Les deux méthodes proposées ont une bonne exactitude

(93% et 91%) et améliorent également la précision à 80% et 81%, le rappel à 85% et 84% et le F1-score à 82% et 83%.

Une autre expérience de notre travail consiste à montrer l'effet de la parallélisation sur le temps de calcul de la classification en utilisant nos approches. Pour cela, nous avons utilisé trois nœuds Hadoop et un jeu de données contenant 45640 tweets. La figure II.21 montre les résultats obtenus.

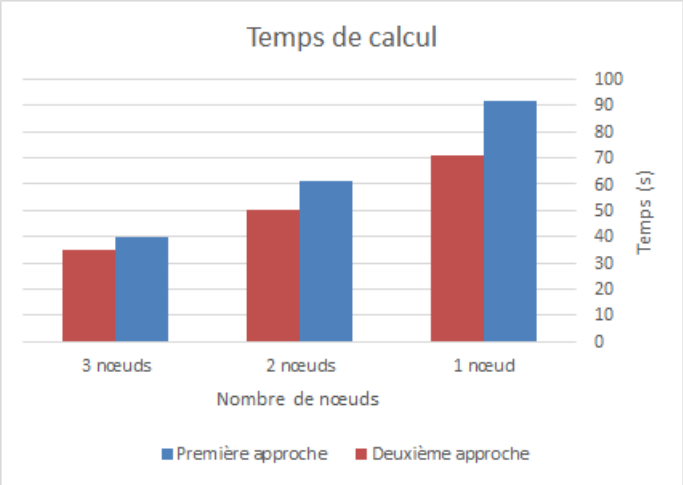


FIGURE II.21 – Temps de calcul de la classification

De la fig. 6.12, Nous notons que si le nombre de nœuds Hadoop augmente, le temps de calcul de la classification diminue soit avec la première approche proposée, ou avec la seconde approche, ce qui démontre le choix de paralléliser la classification. Sans oublier que le temps de calcul nécessaire pour obtenir le résultat de la classification avec la seconde approche est inférieur à celui requis avec la première approche, car la seconde nécessite quelques traitements, contrairement à la première qui nécessite un passage par trois documents pour obtenir les résultats de la classification.

Nous avons présenté dans ce travail nos deux approches proposées pour classer les tweets en trois classes : positive, négative et neutre (classification de subjectivité) avec la première approche, et en deux classes : positive et négative (classification de polarité) avec la seconde approche. Notre proposition consiste à proposer deux nouvelles méthodes enrichies par la sémantique en utilisant la base lexicale Wordnet. Nous avons également présenté le moyen de paralléliser la classification en utilisant Hadoop (HDFS et MapReduce) pour optimiser le temps de classification.

La pratique montre que nos méthodes donnent de bons résultats au niveau du taux de classification, du taux d'erreur, de la précision, du rappel et du F1-score. De plus, les méthodes de pré-traitement du texte améliorent le résultat de la classification. Nous avons également montré l'effet de la parallélisation sur le temps de calcul.

2.5 Application de notre travail : Approche de filtrage collaboratif social pour la recommandation des cours dans une plate-forme d'apprentissage en ligne

Au cours des dernières années, l'apprentissage en ligne à l'aide des plateformes d'e-learning est devenu indispensable dans le processus d'enseignement. Les entreprises et les chercheurs scientifiques essaient de trouver de nouvelles méthodes et approches optimales qui permettent d'améliorer l'éducation en ligne. Dans cette section, nous utilisons la première approche présentée précédemment(qui calcule la similarité sémantique entre le tweet à classer et trois documents d'opinions), pour proposer une nouvelle approche de recommandation afin de recommander des cours pertinents aux apprenants. Notre méthode est basée sur le filtrage social et le filtrage collaboratif, pour définir la meilleure façon avec laquelle l'apprenant doit apprendre et recommander les cours les plus pertinents à son profil et à son contenu social(les tweets et publications publiés dans les comptes Facebook et Twitter).

2.5.1 E-learning et systèmes de recommandation

Avec la croissance exponentielle de la population qui souhaite apprendre en ligne, les plateformes d'apprentissage en ligne doivent s'adapter et innover dans la manière dont elles proposent des cours aux apprenants. Dans la littérature, nous trouvons de nombreuses méthodes et approches qui tentent de proposer des cours optimaux à un apprenant, comme ceux basés sur des algorithmes tels qu'un algorithme génétique[56] ou l'utilisation des approches d'apprentissage automatique. Au cours des dernières années, les entreprises et les chercheurs ont commencé à utiliser les bases des systèmes de recommandation dans l'apprentissage en ligne.

Les systèmes de recommandation(SR) aident les utilisateurs à trouver les produits et services correspondant à leurs préférences et à leurs besoins, et à réduire le temps nécessaire à la recherche des objets qu'ils recherchent. Ils jouent un rôle de plus en plus important dans diverses applications, telles que le commerce électronique, la musique, la recommandation de films et de livres, la recherche sur le Web, la santé et les plateformes d'e-learning (dans le processus d'enseignement).

SR peut être divisé en plusieurs types, tels que la recommandation basée sur le savoir ou le contenu(knowledge-based ou content-based recommendation), la recommandation

de filtrage social(Social filtering recommendation), la recommandation de filtrage collaboratif(collaborative filtering recommendation) et la recommandation hybride.

La recommandation de filtrage social (FS) utilise le contenu social des utilisateurs en fonction de leurs traces sur différents réseaux sociaux, afin de recommander un contenu pertinent correspondant à leurs profils sociaux. D'autre part, le filtrage collaboratif (FC) recommande une évaluation ou un produit pour un utilisateur en fonction des préférences d'évaluation d'utilisateurs similaires. FC est basé sur l'hypothèse qui dit que les utilisateurs avec le même goût ont une préférence similaire aux produits ou articles, il est divisé en une approche basée sur la mémoire(memory-based approach) et une approche basée sur un modèle(model-based approach). Dans le FC basé sur un modèle, des modèles sont développés à l'aide de différents algorithmes (réseau de neurones, algorithmes d'apprentissage automatique, etc.) afin de prédire et de recommander des produits ou des services pertinents. D'autre part, le FC basé sur la mémoire repose sur le calcul de la similarité entre les utilisateurs ou les articles. il est divisé en FC basé sur l'utilisateur(user-based) et en FC basé sur l'article(item-based).

Dans ce travail, chaque cours a un certain nombre d'objectifs pédagogiques(OP), notre objectif est de prédire et recommander l'objectif pédagogique adapté à l'apprenant. Par exemple, le cours de Java peut avoir différents objectifs pédagogiques (**l'installation de Java/les bases du java/les structures de contrôle/les tables** est un OP, **les objets/les classes/l'héritage** est un autre OP).

Le but ultime de l'approche proposée est de recommander des articles qui correspondent mieux au profil de l'apprenant. Les articles dans notre cas sont les concepts qui représentent les objectifs pédagogiques d'un cours, par exemple, les cours Java, Python ou Php.

L'approche proposée est née de l'idée que les performances de recommandation ont un impact considérable sur le succès de l'enseignement. Nous utilisons SR pour estimer les préférences potentielles des apprenants et recommander des cours pertinents (ou des objectifs pédagogiques), en fonction du contenu social (utilisation des réseaux sociaux) et du contenu du profil.

L'approche de recommandation proposée est divisée en deux parties :

- Approche de filtrage social (FS) : cette étape consiste à utiliser les profils des réseaux sociaux de chaque apprenant pour calculer deux facteurs : **productivité** et **motivation**. Dans cette étape, nous utilisons l'hypothèse qui dit que les personnes ayant une productivité et une motivation similaires auront un profil et des compétences d'apprentissage similaires. Cette étape nous aidera à trouver les apprenants les K plus proches voisins (apprenants similaires).

– Approche de filtrage collaboratif (FC) : lors de cette étape, nous construisons d'abord un profil pour chaque apprenant en fonction de ses connaissances, ce qui signifie qu'après la construction du profil, nous trouverons l'évaluation de l'apprenant pour chaque concept d'un OP. Ensuite, nous calculons la similarité entre l'apprenant cible(l'apprenant pour lequel nous voulons recommander un cours) et les autres apprenants pour recommander l'OP de l'apprenant le plus similaire (apprenant le plus proche). CF examine les profils des autres apprenants pour trouver l'OP le mieux adapté au profil de l'apprenant cible.

L'approche proposée peut traiter les problèmes traditionnels des systèmes de recommandation, à savoir : le problème du démarrage à froid(cold start), Sparsity (Rareté) et la scalabilité(scalability), en utilisant le contenu social, le profil de l'apprenant et la similarité, pour éviter tout problème pouvant affecter la qualité de la recommandation.

2.5.2 Revue de la littérature

Ces dernières années, de nombreuses approches ont été développées pour améliorer la qualité des recommandations et éviter les problèmes liés aux démarrages à froid, Sparsity et la scalabilité. Le champ d'application des systèmes de recommandation s'est également généralisé. Il est passé du domaine du commerce électronique (recommandation de produits) à d'autres domaines tels que la recommandation de films, de services ou également dans le domaine de l'éducation pour la recommandation des cours.

Ar et al. dans leur article[3] ont proposé une nouvelle approche du filtrage collaboratif(FC) basée sur un algorithme génétique(AG) pour améliorer le résultat de la prédiction en utilisant différentes mesures de similarité. La solution proposée vise à affiner et à améliorer les valeurs de similarité, c'est-à-dire les poids obtenus à l'aide de diverses métriques avec AG afin d'améliorer l'exactitude de la prédiction du FC. Les résultats montrent que l'approche évolutive a réduit d'une manière significative l'erreur de prédiction en utilisant les pondérations évoluées et que la similarité vectorielle de Cosine a montré les meilleures performances.

Les auteurs de[48] ont proposé une nouvelle méthode du FC pour rechercher les utilisateurs voisins sur la base d'une FC basée sur l'utilisateur(user-based FC). Ils ont utilisé le coefficient de corrélation de Pearson (PCC) comme mesure de similarité traditionnelle pour trouver des utilisateurs voisins, ainsi que la méthode de clustering k-means et le modèle de factorisation matricielle non négative (NNMF) en tant que méthodes de clustering traditionnelles. Pour les expériences, trois bases de données bien connues sont utilisées pour analyser l'approche proposée : **Movielens 100k (ML_100k), Movie-lens 1M (ML_1M) et Jester**. Les résultats montrent que la méthode proposée surpasse les autres

approches au niveau du rappel et de la précision, et que les algorithmes de classification fonctionnent mieux que les mesures de similarité pour trouver les utilisateurs voisins.

Liu et al. [54] ont présenté un nouveau modèle de similarité, pour améliorer les performances de la recommandation lorsque peu d'évaluations sont disponibles pour calculer les similarités de chaque utilisateur. Dans cet article, les auteurs ont montré les inconvénients de certaines mesures de similarité telles que cosine et peasrson. La mesure de similarité proposée est une fonction non linéaire, qui est la fonction **sigmoïde**, le principal objectif de cette approche est d'améliorer la similarité PIP (proximité, impact, popularité). Plusieurs expériences sont menées sur trois ensembles de données couramment utilisés. Comme résultats expérimentaux, les auteurs voient que la nouvelle mesure de similarité peut obtenir la meilleure performance que la plupart des autres méthodes. Ces résultats démontrent l'efficacité de la nouvelle mesure de similarité et qu'elle peut surmonter les inconvénients des mesures de similarité traditionnelles.

Les auteurs dans [13] ont proposé une nouvelle approche de similarité de filtrage collaboratif en améliorant l'algorithme traditionnel de similarité du cosine ajusté. Le processus de calcul de similarité d'utilisateur peut s'exécuter en mode hors connexion, ce qui permet de réduire le temps d'exécution recommandé et d'améliorer la vitesse de recommandation, et donc résoudre le problème du temps réel de la recommandation. L'approche FC proposée est basée sur la similarité optimisée d'utilisateur. Les résultats expérimentaux montrent que l'algorithme de filtrage collaboratif proposé permet d'optimiser d'une manière significative la précision de la similarité et d'obtenir de meilleurs résultats de recommandation.

Dans [69], les auteurs ont mis en œuvre deux approches : le filtrage collaboratif (FC) et le système de recommandation du réseau social (SNRS). Ils ont utilisé l'erreur absolue moyenne (EMA) et la précision pour comparer le résultat des deux approches mentionnées, et ils ont constaté que la méthode SNRS est considérée comme la version améliorée du FC et il est plus efficace. Les auteurs ont trouvé que le SNRS est plus efficace que le filtrage collaboratif traditionnel, car dans cette approche, les préférences de l'utilisateur, l'acceptation générale d'un article et l'influence des amis ont été prises en compte.

A titre de comparaison avec les méthodes présentées dans les paragraphes précédents, notre méthode se base sur une nouvelle idée pour grouper les apprenants(clustering) avec l'utilisation des données des réseaux sociaux ce qui va nous aider à éviter le problème de scalability. L'approche proposée utilise aussi les principes du FC afin d'augmenter la qualité de recommandation en se basant sur les connaissances des apprenants. Sans oublier que l'approche proposée peut traiter les problèmes classiques des SRs.

2.5.3 Approches de recommandation

En examinant de près le contenu du Web, nous remarquons que le nombre d'apprenants ayant commencé à apprendre en ligne augmente d'une manière explosive, en plus du grand nombre de cours publiés en ligne. Avec cela, il devient nécessaire d'utiliser les systèmes de recommandation pour trouver les cours optimaux et pertinents pour les apprenants.

Comme présenté précédemment, ce travail consiste à proposer une nouvelle approche de recommandation pour recommander aux apprenants des cours optimaux en fonction de leurs profils et de leur contenu social, afin d'améliorer le processus d'enseignement et de réduire le temps nécessaire à la recherche des cours pertinents.

Notre système de recommandation(SR) repose sur deux nouvelles approches :

- *Approche de filtrage social (FS)* qui utilise le contenu social des apprenants en fonction des profils de leurs réseaux sociaux pour définir des facteurs tels que **la productivité** et **la motivation**. Ces facteurs seront ensuite utilisés comme paramètres pour le regroupement (placer les apprenants ayant un contenu social similaire dans des clusters ou des grappes).
- *Approche de filtrage collaboratif (FC)* qui consiste à construire des profils d'apprenants basés sur leurs connaissances, puis à calculer la similarité entre l'apprenant actif(l'apprenant cible) et les autres apprenants du système, ou entre l'apprenant actif et les différents objectifs pédagogiques d'un cours.

Nous avons nommé l'approche proposée "**Filtrage collaboratif social (FCS)**", car elle utilise à la fois les avantages du FS et du FC pour trouver des cours pertinents et optimaux pour les apprenants (approche hybride entre le FS et FC). **FCS** consiste à recommander des cours pertinents à l'apprenant cible en fonction des profils et du contenu social des apprenants similaires.

L'approche de filtrage collaboratif social recherche les apprenants partageant des préférences similaires avec l'apprenant cible en matière de contenu social et pédagogique, et recommande l'OP de l'apprenant le plus similaire (apprenant le plus proche).

Dans notre SR, chaque cours est présenté comme un ensemble d'objectifs pédagogiques (OP). L'idée est que l'apprenant doit apprendre l'OP qui correspond à son profil et à son contenu social.

Approche de filtrage social (FS)

Le FS est la première étape de l'approche proposée. Il utilise les profils Facebook et Twitter des apprenants pour collecter des informations telles que des publications/tweets publiés, des likes(*j'aimes*) et des commentaires. Ces informations seront ensuite utilisées

pour calculer/définir deux mesures que nous avons nommées **la productivité** et **la motivation**.

Le principe du calcul de la productivité est comme celui présenté dans le chapitre précédent.

La motivation consiste à définir la motivation de chaque apprenant, qu'il soit motivé, démotivé ou neutre. Pour cela et après avoir défini la productivité, nous collectons les publications(de Facebook) et les tweets (de Twitter) publiés au cours de la période d'activité (la période au cours de laquelle l'apprenant publie beaucoup de contenu).

L'étape suivante consiste à classer les publications et les tweets récupérés en trois classes : **positive**, **négative** ou **neutre**. c'est-à-dire que chaque tweet ou message récupéré peut exprimer un sentiment positif, un sentiment négatif ou un sentiment neutre. Ensuite, si la classe majoritaire (MC) est la classe **positive**, nous considérons l'apprenant comme étant motivé, sinon, si MC est la classe négative, nous considérons l'apprenant comme démotivé et si MC est la classe neutre, nous considérons l'apprenant comme neutre.

Pour effectuer la classification, nous nous sommes basés sur la première approche proposée présentée précédemment (calcul de la similarité sémantique entre la publication ou le tweet à classer et trois documents d'opinion), voir la formule 21.

La formule 23 montre comment définir la motivation des apprenants.

$$Motivation = \max[\sum_{i=1}^{p} CV(T_p), \sum_{i=1}^{n} CV(T_n), \sum_{i=1}^{ne} CV(T_{ne})] \tag{23}$$

Avec :

- $\sum_{i=1}^{p} CV(T_p), \sum_{i=1}^{n} CV(T_n), \sum_{i=1}^{ne} CV(T_{ne})$ sont respectivement la somme des publications/tweets positifs, négatifs et neutres.

Ces deux mesures seront ensuite utilisées pour regrouper les apprenants en groupes(clusters). Puisque la productivité et la motivation ont les deux trois valeurs possibles (1, 2 et 3 pour la productivité ; positive, négative et neutre pour la motivation), nous avons donc 9 différents groupes ou classes (1-positive, 1-négative, 1-neutre, 2-positive, 2-négative, 2-neutre, 3-positive, 3-négative et 3-neutre).

À la fin de l'approche FS, nous classons les apprenants en groupes basés sur le contenu social. Pour chaque nouvel apprenant (apprenant cible ou actif) pour qui nous voulons recommander un cours (objectif pédagogique), nous appliquons les différentes étapes du FS pour définir la productivité et la motivation qui nous aideront après pour le regroupement, afin de trouver les K-plus proches voisins de l'apprenant cible (apprenants partageant la même productivité et la même motivation avec l'apprenant cible (appartenant au même groupe)).

Approche de filtrage collaboratif (FC)

FC est la deuxième étape de notre méthode. Dans notre plateforme, chaque apprenant est présenté par son profil de connaissances. Pour le construire, chaque apprenant doit répondre à un questionnaire qui se présente sous la forme de questions à choix multiples. Chaque cours de notre système a son questionnaire(Quiz) équivalent.

Dans notre système, chaque cours a un nombre des OPs, et chaque OP a un nombre des concepts qu'ils traitent. Par exemple, pour le cours du JAVA, la déclaration des tables, le remplissage d'une table, l'affichage d'une table sont des concepts.

Les questions du questionnaire représentent les différents concepts des différents objectifs pédagogiques du cours, c'est-à-dire que chaque OP contient un nombre de concepts, et chaque concept représente un article(item) de notre plateforme, et faire le quiz revient à donner une évaluation(rating) à chaque concept.

Après que l'apprenant complète le quiz et sur la base du résultat obtenu, nous construisons son profil qui se présente sous la forme d'un vecteur. Chaque élément du vecteur représente l'évaluation d'un concept. La note est égale à 0 si l'apprenant a répondu correctement à la question équivalente au concept et égale à 1 sinon.

Notre hypothèse est que si l'apprenant n'a pas répondu correctement à une question, cela signifie qu'il doit apprendre le concept lié à cette question, et nous considérons que son évaluation(rating) est égale à 1. Sinon, s'il répond correctement à une question, donc il maîtrise déjà le concept, c'est-à-dire il n'est pas nécessaire de le réapprendre, pour cela nous considérons son évaluation comme 0.

Après avoir trouvé les K-plus proches voisins(KPP) de l'apprenant cible à l'aide de l'approche de filtrage social (basée sur la productivité et la motivation), et après avoir construit son profil (sous forme de vecteur) sur la base du questionnaire, nous calculons la distance entre le profil de l'apprenant cible et les profils des KPP apprenants pour trouver l'apprenant le plus similaire. Nous nous sommes basés ici sur l'hypothèse que l'apprenant cible et son plus proche apprenant ont des préférences similaires et peuvent apprendre de la même manière. À la fin de ce processus, nous recommandons à l'apprenant cible les concepts de l'objectif pédagogique que l'apprenant le plus proche a déjà appris dans notre plate-forme.

Dans notre système, les profils des apprenants sont représentés par la représentation suivante :

$Profile = [R(C_1), R(C_2), R(C_3), ..., R(C_n)]$

Où :

– C : est un concept.

– R(C) : est l'évaluation du concept C après avoir passé le quiz (1 ou 0).

– n : est le nombre des concepts dans tous les objectifs pédagogiques d'un cours.

2.5.4 scalabilité et Sparsity

Les systèmes de recommandation(SR) ont des difficultés liées aux problèmes de **scalabilité** et de **sparsity** ou **cold start**(démarrage à froid). Chaque approche de SR doit prendre en compte ces deux problèmes (leur traitement) pour améliorer la qualité de recommandation.

L'approche proposée peut traiter ces deux problèmes, ce qui nous aide à éviter tout problème pouvant affecter notre système.

Le premier défi majeur rencontré par l'apprenant selon l'approche de filtrage collaboratif social est le problème de **scalabilité** tel qu'il peut être démontré lors de la recherche de l'apprenant le plus proche(KPP) si notre plateforme contient un très grand nombre d'apprenants et de cours. Pour éviter ce problème, nous adoptons deux niveaux :

– Le premier niveau consiste à regrouper les apprenants en groupes basés sur le contenu social (productivité et motivation) pour réduire le nombre d'apprenants (en ne conservant que les apprenants les plus similaires à l'apprenant cible). Après cela, nous n'avons plus besoin de rechercher l'apprenant le plus proche parmi tous les apprenants de notre plateforme, mais uniquement dans le groupe adéquat à l'apprenant cible (le groupe qui contient des apprenants similaires ayant un contenu social similaire à l'apprenant cible).
– Le deuxième niveau consiste à paralléliser notre travail à l'aide des technologies Big Data (Hadoop MapReduce et Hadoop Distributed File System HDFS) afin de partager le travail de recommandation entre un groupe de machines(cluster Hadoop).

Le deuxième défi majeur de notre système est lié aux problèmes de **sparsity et cold start**. Ils se produisent lorsqu'il est impossible de faire une recommandation fiable en raison d'un manque initial d'informations (apprenants similaires à l'apprenant cible). Par exemple, pour le premier utilisateur de notre plateforme, il est impossible de recommander des cours basés sur le contenu social car il n'y a pas encore des apprenants dans le système. Ou si nous ne trouvons pas d'apprenants similaires pour l'apprenant cible après l'application de l'approche du filtrage social.

Les problèmes de démarrage à froid(cold start) et sparsity se produisent dans l'approche proposée à trois niveaux. Le premier est lorsque le premier utilisateur souhaite utiliser notre plateforme pour avoir une recommandation. Le deuxième niveau que nous pouvons appeler démarrage à froid complet (complete cold start CCS) se produit lorsqu'aucun apprenant KPP (de l'apprenant cible) n'est disponible dans notre plateforme (aucun apprenant n'a un contenu social similaire à l'apprenant cible). Le troisième niveau

est lorsque la distance entre le profil de l'apprenant cible et les profils des apprenants de son groupe est inférieure à un seuil, ce qui signifie que parmi les apprenants similaires(en terme du contenu social), personne n'a un profil similaire à l'apprenant cible(en terme de contenu cognitif).

Si nous rencontrons l'un de ces problèmes, nous nous baserons uniquement sur le profil des apprenants et sur les objectifs pédagogiques pour recommander l'OP le mieux adapté. L'idée est qu'après la construction du profil de l'apprenant cible en complétant le questionnaire, nous construisons le vecteur du profil avec une évaluation pour chaque concept. Nous représentons également chaque OP sous la forme d'un vecteur en donnant 1 comme évaluation au concept traité par cet OP et 0 si un concept n'est pas traité par l'OP.

Ensuite, nous calculons la distance entre le profil de l'apprenant et les différents OP pour définir l'OP le plus proche à l'apprenant (l'OP qu'il doit apprendre).

2.5.5 Etapes de l'approche proposée et algorithme

Comme présenté précédemment, l'approche proposée repose sur plusieurs étapes importantes. La première consiste à utiliser les profils des réseaux sociaux des apprenants pour extraire certains facteurs sociaux tels que les tweets, les publications Facebook, les likes(j'aimes), les commentaires. Ces facteurs seront utilisés dans le calcul de la productivité et de la motivation, ce qui nous aidera à définir les K-plus proches voisins(KPP) de l'apprenant cible. La deuxième étape est la construction des profils des apprenants sur la base d'un questionnaire. Enfin, c'est le calcul de la distance entre le profil de l'apprenant cible et les profils des KPP apprenants pour définir l'objectif pédagogique recommandé (ou les concepts recommandés).

La figure II.22 montre les différentes étapes pour faire une recommandation :

L'*algorithme 5* présente les fonctions principales de l'approche proposée.

Où :

- $Social\_Login()$: est une fonction qui permet à l'apprenant de se connecter à notre plateforme à l'aide de ses comptes Facebook et Twitter.
- Social_Factors(8,12) : récupère le nombre de tweets, de publications Facebook, de likes et de commentaires publiés entre 8h et 12h(au cours de la première période).
- $Max(B_1, B_2, B_3)$: une fonction qui renvoie le maximum des trois mesures (B1, B2, B3).
- ClassOf($Posts\_Tweets$(Productivity)) : cette fonction définit la classe majoritaire des publications/tweets publiés par l'apprenant au cours de la période d'activité, la classe est soit motivé, démotivé ou neutre.

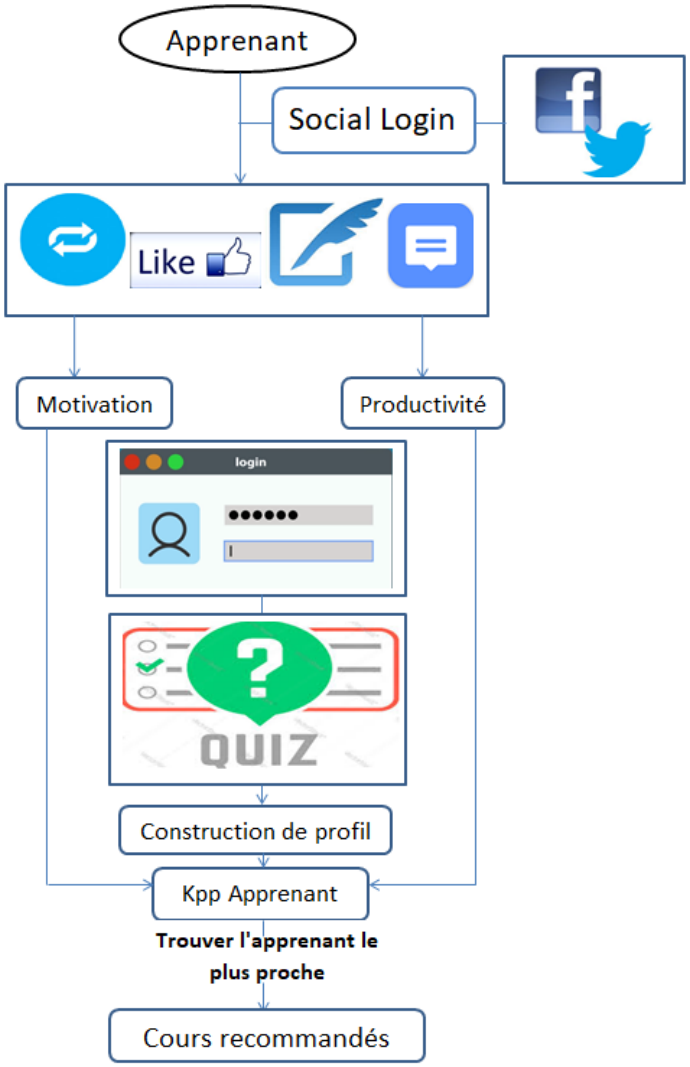


FIGURE II.22 – Etape de notre travail

- Clustering(Motivation, Productivity) : consiste à regrouper les apprenants en groupes pour définir les K-plus proches voisins de l'apprenant cible en fonction de la motivation et de la productivité.
- Distance($Target\_Learner, KNN\_Learners$) : cette fonction calcule la distance entre le profil de l'apprenant cible et les profils des KPP apprenants, pour extraire l'apprenant le plus proche.

Dans ce travail, nous avons décrit comment nous pouvons utiliser notre approche d'analyse des sentiments présentée précédemment(la première approche proposée) dans un problème réel(une plateforme d'apprentissage en ligne). Pour cela, nous avons présenté une nouvelle méthode pour recommander des cours aux apprenants basée sur leur contenu social et leur profil(connaissances). Notre méthode est une approche de filtrage collaboratif social qui utilise le contenu social des apprenants, tels que les tweets, les publications Facebook, les likes(j'aimes) et les commentaires, pour les regrouper en grappes. Il repose

Algorithm 5 Algorithme de filtrage collaboratif social

INPUT : Social Content
OUTPUT : Recommended Courses
$Social_Login()$
$B_1 \leftarrow Social\_Factors(8, 12)$
$B_2 \leftarrow Social\_Factors(2, 6)$
$B_3 \leftarrow Social\_Factors(7, 11)$
P $\leftarrow Max(B_1, B_2, B_3)$
if $P = B_1$ **then**
  Productivity $\leftarrow 1$
else if $P = B_2$ **then**
  Productivity $\leftarrow 2$
else if $P = B_3$ **then**
  Productivity $\leftarrow 3$
end if
Motivation $\leftarrow$ ClassOf($Posts\_Tweets$(Productivity))
$KNN\_Learners \leftarrow$ Clustering(Motivation, Productivity)
$Closest\_Learner \leftarrow$ Distance($Target\_Learner, KNN\_Learner`s$)
$Recommended\_Courses(Target\_Learner)$

également sur les connaissances des apprenants pour trouver l'apprenant similaire(le plus proche) à l'apprenant cible. L'approche proposée traite les problèmes de la scalabilité, de la sparsity et du démarrage à froid pour améliorer la qualité de la recommandation.

Les approches proposées dans les contributions 1 et 2 traitent les données des réseaux sociaux d'une manière claire sans prendre en considération l'ambiguïté et l'imprécision des sentiments. Dans le prochain chapitre, nous présenterons deux approches hybrides qui combinent la similarité sémantique, un dictionnaire et la logique floue afin d'analyser les données de manière floue.

Chapitre 3

Une approche multilingue pour la classification des données Twitter basée sur la logique floue.

Avec l'augmentation du nombre des utilisateurs qui utilisent les réseaux sociaux pour exprimer leurs opinions, les méthodes d'analyse des opinions doivent prendre en considération le flou et l'imprécision des sentiments. Dans ce travail, nous proposons deux nouvelles méthodes pour classer les tweets en trois classes : positive, négative ou neutre. Les méthodes proposées sont deux nouvelles approches hybrides basées sur la logique floue avec ses trois étapes importantes (fuzzification, inférences floue(rule inference)/agrégation et défuzzification), les concepts des systèmes de recherche d'informations (IRS) ; en calculant la similarité sémantique entre un tweet à classer et deux documents d'opinion(un pour les mots d'opinion positive et un autre pour les mots d'opinion négative) à l'aide du dictionnaire WordNet, et une approche basée sur le lexique en utilisant SentiWordnet. L'ensemble des données de tweets à classer et le résultat de la classification sont stockés dans le système de fichiers Hadoop(HDFS), pour l'application de notre proposition nous utilisons Hadoop MapReduce.

3.1 Logique Floue et l'analyse des sentiments

De nombreux travaux de la littérature utilisent des algorithmes d'apprentissage automatique ainsi que des techniques statistiques, toutes ces méthodes traitent la classification du sentiment d'une manière "noir et blanc", alors qu'en réalité les sentiments sont rarement clairs et contiennent des données imprécises, ambiguïtés et pouvant appartenir à plusieurs classes. A partir de là, nous utilisons dans notre travail la logique floue avec ses différentes étapes (fuzzification, inférence de règles floues, défuzzification). La logique floue peut bien gérer le flou et l'ambiguïté et elle ressemble au langage naturel et à la pensée naturelle, qui est proche du cerveau humain.

Dans ce chapitre, nous proposons deux nouvelles approches pour classer les tweets en

trois classes (positive, négative et neutre) en utilisant une approche hybride basée sur la logique floue, les concepts des systèmes de recherche d'informations (IRS) avec l'utilisation de la similarité sémantique, et la ressource lexicale(dictionnaire) SentiWordNet [1]. Pour cela, nous proposons deux mesures que nous avons appelées la **positivité** et la **négativité** des tweets. Pour calculer chaque mesure, nous basons sur les notions d'IRS et une approche de similarité sémantique, ou en utilisant le dictionnaire SentiWordNet.

Ces deux mesures joueront le rôle des entrées de notre système de la logique floue(Fuzzy Logic System FLS). En appliquant les différentes étapes d'un FLS(fuzzification, règles floues et défuzzification), on retrouve la classe du tweet (positive, négative ou neutre). Nous appliquons la logique floue sur la positivité et la négativité pour prendre en compte le flou et l'imprécision des sentiments ce qui affectent positivement la classification. chaque étape de notre système de la logique floue contient un ensemble de traitements. Par exemple, dans l'étape de fuzzification, nous allons utiliser une fonction pour calculer le degré d'appartenance des entrées à chaque ensemble flou, cette étape consiste à transformer les valeurs nettes(crisp values) de la positivité et de la négativité en valeurs floues.

3.2 Système à logique floue

Dans cette section, nous présenterons les principaux concepts de la logique floue : ensembles flous, fuzzification, inférence des règles floues et défuzzification.

L'idée de la logique floue est similaire au processus de sentiment et d'inférence de l'être humain, c'est une approche de calcul basée sur des "degrés de vérité" plutôt que la logique booléenne habituelle ("vrai ou faux" (1 ou 0)) sur laquelle l'ordinateur moderne est basé. La théorie de la logique floue vise principalement à transformer un problème noir et blanc en un problème gris [106] ; dans le contexte de la théorie des ensembles, la logique déterministe correspond à des ensembles nets(crisp sets)[55], l'idée de la logique floue a été inventé par le professeur L.A. Zadeh de l'université de Californie à Berkeley en 1965 [105].

Les idées floues et la logique floue sont utilisées dans notre vie quotidienne que personne ne fait même attention à elles. Par exemple, pour répondre à certaines questions de certains sondages, la plupart du temps, on pourrait répondre par "pas très satisfait" ou "assez satisfait", qui sont aussi des réponses floues ou ambiguës. Dans quelle mesure est-on satisfait ou insatisfait de certains services ou produits pour ces enquêtes ? Ces réponses vagues ne peuvent être créées et mises en œuvre que par des êtres humains, mais pas par des machines. Est-il possible pour un ordinateur de répondre directement à ces questions

1. http ://sentiwordnet.isti.cnr.it, est une ressource lexicale pour l'analyse des sentiments, il attribue à chaque synset de WordNet trois scores de sentiment : positivité, négativité et objectivité.

de sondage comme le faisait l'être humain ? c'est absolument impossible. Les ordinateurs ne peuvent comprendre que "0" ou "1" et "HAUT" ou "BAS". Ces données sont appelées données claires, nettes ou classiques(crisp value) et peuvent être traitées par toutes les machines[5].

3.2.1 Ensemble flou

Le concept de l'ensemble flou n'est qu'une extension du concept de l'ensemble classique ou net(crisp set). L'ensemble classique considère uniquement un nombre limité de degrés d'appartenance, tels que '0' ou '1', c'est-à-dire que la valeur de la fonction d'appartenance d'un objet dans un ensemble crisp est 0 ou 1. La valeur de **0** signifie que l'objet n'appartient pas à l'ensemble net tandis que la valeur **1** indique que l'objet appartient complètement à l'ensemble.

Un ensemble flou est une généralisation de l'ensemble classique ou crisp avec l'intervalle de [0,1]. Un objet ne peut appartenir que partiellement à un ensemble flou. Plus l'objet appartient à l'ensemble flou, plus le degré d'appartenance est élevé.

L'ensemble flou est un outil puissant qui nous permet de représenter des objets ou des membres d'une manière vague ou ambiguë. Par exemple, dans notre cas, nous voulons représenter le sentiment d'un tweet d'une manière ambiguë afin d'améliorer les résultats d'analyse et de classification.

3.2.2 Fuzzification

Pour appliquer la logique floue à un problème réel ; comme dans notre cas la classification des tweets, trois étapes consécutives sont nécessaires : *fuzzification*, *inférence floue* et *défuzzification*.

La fuzzification est la première étape pour appliquer un système à logique floue qui consiste à rendre floue une quantité précise. La plupart des variables existant dans le monde réel sont des variables claires ou classiques. Il faut convertir ces variables nettes (entrée et sortie) en variables floues. La fuzzification implique deux processus : dériver les fonctions d'appartenance(Membership Functions) pour les variables d'entrée et de sortie et les représenter avec des variables linguistiques.

Les fonctions d'appartenance transforment chaque valeur nette en un ensemble flou. En pratique, les fonctions d'appartenance peuvent avoir plusieurs différents types, tels que la forme d'onde triangulaire, la forme d'onde trapézoïdale, la forme d'onde gaussienne, la forme d'onde en cloche, la forme d'onde sigmoïdale et la forme d'onde S-curve.

Nous avons donc au début une variable nette ou claire(crisp variable) à fuzzifier, la

prochaine étape consiste à la fuzzifier à l'aide d'une fonction d'appartenance (triangulaire, trapézoïdale ...), puis pour cette variable, nous trouvons un degré d'appartenance compris dans l'intervalle [0,1].

3.2.3 Règles floues

La deuxième étape d'un système à logique floue (Fuzzy Logic System FLS) consiste à appliquer les règles aux variables d'entrée et de sortie. Une base de règles est construite pour contrôler la variable de sortie. Une règle floue est une simple règle IF-THEN avec une condition et une conclusion. Par exemple, **si(IF)** la positivité est élevée **et(AND)** la négativité **est** faible **alors(THEN)** le sentiment est positif. Les entrées et les sorties des règles ont des valeurs floues.

3.2.4 Défuzzification

La troisième étape dans un FLS est la défuzzification. Malgré le fait que la majeure partie des informations que nous assimilons au quotidien est floue, la plupart des actions ou décisions mises en œuvre par l'humain ou par les machines sont précises ou binaires. Les décisions que nous prenons sont binaires, le matériel que nous utilisons est binaire. Nous devons donc défuzzifier les sorties pour n'avoir qu'une valeur précise(crisp value) qui nous donne le résultat final de notre problème (le problème de la classification par exemple).

Ce processus de "défuzzification" a pour résultat de réduire un ensemble flou à une quantité précise d'une valeur unique, ou à un ensemble précis(crisp set), la sortie d'un processus flou peut être l'union logique de deux ou plusieurs fonctions d'appartenance floue définies sur l'univers de discours de la variable de sortie.

Il existe de nombreuses méthodes de défuzzification dans la littérature, telles que le principe d'appartenance maximale, la méthode Centroid (également appelée centre de gravité), la méthode de la moyenne pondérée et l'appartenance Moyenne-max (également appelée milieu de maxima).

3.3 Motivation et travaux connexes

Les classifieurs basés sur les techniques de ML traitent la classification des sentiments d'une manière "noir et blanc", alors qu'en réalité le sentiment est rarement clair, la plupart des travaux précédents se concentrent sur les méthodes des algorithmes déterministes sans considérer le flou des sentiments. La réalité est toujours loin d'être optimiste. Pre-

mièrement, les termes relatifs aux sentiments sont flous, le même mot peut expliquer des différentes orientations de sentiment, même dans le même domaine. D'autre part, les sentiments d'humain sont plusieurs fois flous. Par exemple, on peut utiliser un mot pour exprimer plus d'un sentiment en même temps.

Au cours des dernières années, les approches floues ont commencé à apparaître pour le traitement du texte et l'analyse des sentiments, quoique le nombre des articles repose sur la logique floue reste jusqu'à aujourd'hui faible par rapport au nombre des articles qui utilisent des techniques de ML ou des approches statistiques. Dans la littérature, nous trouvons quelques travaux qui utilisent la logique floue dans l'analyse des sentiments.

Les auteurs de[96] ont proposé un modèle informatique flou pour identifier la polarité des mots de sentiment chinois. Cet article présente principalement trois aspects, le premier consiste à calculer l'intensité de sentiment des mots à l'aide de trois lexiques de sentiment chinois existants. Deuxièmement, les auteurs de cet article ont construit un classifieur des sentiments flous et une fonction de classification correspondante du classifieur flou en vertu de la théorie des ensembles flous et du principe du degré d'appartenance maximale. Troisièmement, ils ont construit quatre bases de données des mots de sentiment pour démontrer les performances de leur modèle.

Un autre travail qui utilise les ensembles flous dans l'analyse des sentiments est celui présenté dans[98]. Dans cet article, les auteurs introduisent une nouvelle approche basée sur la logique floue pour la classification du texte, en particulier la classification du message de Twitter. Les entrées utilisées dans ce modèle sont de multiples fonctionnalités utiles extraites de chaque tweet. La sortie est le degré de pertinence de chaque tweet à un événement appelé "Sandy". Pour cela, ils ont utilisé un certain nombre de règles floues et les différentes méthodes de défuzzification existantes dans la littérature. Pour la fuzzification, ils ont sélectionné la fonction d'appartenance de forme trapézoïdale, parce qu'elle est simple et couramment utilisée. Le système flou proposé dans ce travail a comme entrées 7 variables linguistiques et comme sortie une variable linguistique qui est la pertinence du tweet avec "sandy". Comme résultats expérimentaux, ils ont comparé cinq méthodes de défuzzification couramment utilisées, et ils ont conclu que la méthode du centroïde était plus efficace et efficiente que les autres méthodes. En outre, ils ont effectué une comparaison avec la méthode de recherche par mot-clé bien connue, et les résultats révèlent que l'approche basée sur la logique floue est plus appropriée pour classifier les tweets pertinents et non pertinents.

Dragoni et Petrucci[19] ont proposé une méthode intégrant la logique floue pour la représentation de la polarité associée à des caractéristiques linguistiques appartenant à un domaine particulier. Cet article explique comment exploiter les chevauchements linguis-

tiques entre les domaines pour calculer la polarité des documents dans un environnement multi-domaine à l'aide des modèles flous.

Les auteurs de[2] présentent une approche hybride du problème de l'analyse des sentiments au niveau de la phrase. Cette nouvelle méthode utilise le traitement automatique du langage naturel(NLP) comme technique essentielle, un lexique des sentiments amélioré avec SentiWordNet, et les ensembles flous pour estimer la polarité de l'orientation sémantique et son intensité pour les phrases. Pour démontrer l'utilisation d'une approche hybride, les auteurs ont comparé leurs travaux à deux algorithmes d'apprentissage automatique supervisés : Naive Bayes(NB) et Maximum Entropy (ME). Les résultats expérimentaux montrent que la méthode hybride proposée utilisant des lexiques de sentiment, NLP et les ensembles flous, améliore significativement les résultats obtenus avec Naive Bayes (NB) et Maximum Entropy (ME), avec un haut niveau d'exactitude (88.02%) et de précision (84.24%).

Les chercheurs de l'article présenté dans [84] ont utilisé un réseau de neurones et les ensembles flous pour améliorer la qualité de la classification des sentiments. Cette méthode de classification utilise les avantages de la logique floue et du réseau de neurones pour créer un classifieur. Les auteurs ont proposé de fuzzifier les commentaires d'entrée à classer à l'aide de la fonction d'appartenance gaussienne. Ils ont utilisé le principe MAX pour la défuzzification. Pour la classification, ils ont utilisé un réseau de propagation par retour du perceptron multicouche (A Multi-layer perceptron back propagation network MLPBPN).

Dans [37], les auteurs ont proposé un système fondé sur des règles floues pour l'analyse des sentiments. Ils ont utilisé un classifieur basé sur des règles floues pour obtenir des degrés de sentiment. Ils utilisent également une fonction d'appartenance trapézoïdale pour la fuzzification et le maximum de sorties pour la défuzzification. Comme résultats expérimentaux de ce travail, les auteurs ont comparé la précision de l'approche proposée avec celle de deux autres algorithmes d'apprentissage automatique (Naive Bayes et arbre de décision), et les résultats montrent que la méthode floue proposée, atteint le même niveau de performance que les deux autres algorithmes.

Liu et al. [55] ont publié un article utilisant un système basé sur des règles floues comme modèles de calcul permettant une analyse précise et interprétable des sentiments. Les auteurs ont utilisé quatre ensembles de données des avis des films pour les classer à l'aide d'un système basé sur des règles floues. Pour cela, ils ont utilisé le système flou de **Tsukamoto** en raison du fait que ce type de systèmes flous s'applique aux problèmes de classification. Pour l'étape de fuzzification, ils ont utilisé la fonction d'appartenance floue trapézoïdale, les méthodes min/max pour l'application des règles et pour l'agréga-

tion, et le maximum des sorties pour l'étape de défuzzification. Comme résultats expérimentaux, ils ont comparé l'approche des règles floues avec des approches basées sur des algorithmes d'apprentissage automatique. En matière de précision de la classification en utilisant quatre bases de données, les résultats montrent que l'approche proposée (règles floues) est légèrement supérieure aux algorithmes bien connus (Naive Bayes et C4.5), en indiquant ainsi la pertinence des approches de règles floues pour les tâches d'analyse des sentiments.

Dans l'article de Wang et al. [97], une nouvelle méthode de calcul des sentiments définie comme un discriminateur des sentiments public (public sentiments discriminator PSD) est proposée, elle est basée sur la logique floue et la complexité des sentiments. Les expériences montrent que l'approche proposée(PSD) peut atteindre une précision et une F1-score similaires, mais davantage des résultats cognitifs par rapport aux méthodes traditionnelles d'apprentissage automatique bien connues.

Les auteurs de [23] utilisent les données provenant de Twitter pour analyser les performances des films indiens. Ils ont utilisé l'API Twitter4j pour extraire les tweets et les stocker dans la base de données MySQL. Les données extraites sont utilisées ensuite pour l'analyse des sentiments. Pour cela, ils ont utilisé le système d'inférence floue pour classer les avis des films en trois classes.

le travail de Karyotis et al. [45] a mis en œuvre une méthode de modélisation adaptative floue utilisant des paramètres optimisés à l'aide d'un algorithme génétique pour intégrer l'émotion humaine dans des systèmes informatiques intelligents. Il a également utilisé une infrastructure hybride de cloud computing pour mener des expériences à grande échelle et analyser les sentiments des utilisateurs et les émotions associées, en utilisant les données d'un million d'utilisateurs de Facebook.

Les approches proposées dans ce chapitre tirent parti des avantages des méthodes basées sur le lexique et sur la similarité sémantique(chapitre 1 et 2 de la deuxième partie) et aussi de la logique floue afin d'améliorer les techniques d'analyse des sentiments et pour prendre en considération l'imprécision des opinions. Comment nous avons utilisé un dictionnaire et la similarité sémantique en combinaison avec la logique floue, c'est ce que nous allons voir dans la prochaine section.

3.4 Fuzzification des approches de l'analyse des sentiments

Dans cette section, nous allons présenter les différentes étapes de notre travail et la méthodologie des approches proposées. Comme nous avons présenté précédemment, notre

travail a pour objectif de classer les tweets (analyse des sentiments) liés à un domaine, à un produit ou à des avis de films, etc. Nous classifions chaque tweet selon trois classes : positive, négative ou neutre (classification de subjectivité et d'objectivité) en utilisant deux approches, la première est basée sur un système à logique floue(Fuzzy Logic System **FLS**) et la similarité sémantique (les notions de systèmes de recherche d'informations), et la deuxième est basée sur FLS et la ressource lexicale SentiWordNet. Afin de réaliser nos approches, nous devons mettre en œuvre différentes étapes soit au niveau de la collecte de tweets, la préparation des tweets pour la classification (méthodes du pré-traitement du texte) ou au niveau de l'étape de la classification avec nos approches hybrides proposées.

Notre contribution apparaît dans la proposition de deux nouvelles approches pour classer les tweets en combinant la similarité sémantique et le système à logique floue dans la première approche ; SentiWordNet et la logique floue dans la seconde approche. En d'autres termes, nous exploitons les notions des systèmes de recherche d'informations, de la similarité sémantique, du dictionnaire WordNet, de la logique floue, et du dictionnaire SentiWordNet.

La nouveauté de notre travail apparaît en deux étapes, **la première** consiste à proposer deux nouvelles mesures que nous avons appelées **Positivité** et **Negativité**. Pour le calcul de ces deux mesures, nous avons utilisé les notions des systèmes de recherche d'informations, la similarité sémantique et les dictionnaires WordNet et SentiWordNet. L'idée de calculer la positivité et la négativité en utilisant la première approche est de construire deux documents d'opinion, l'un contenant des mots positifs comme "happy", "exiting" et l'autre contenant des mots négatifs comme "sad", "demotivated". Ensuite, nous calculons la similarité sémantique entre chaque tweet à classer et le document d'opinion positive pour trouver la valeur de la positivité, et entre le tweet et le document d'opinion négative pour calculer la valeur de la négativité. Le calcul de ces deux mesures en utilisant la deuxième approche est basé sur le dictionnaire SentiWordNet ainsi que sur l'étiquetage morpho-syntaxique(POS tags) en utilisant ApacheNLP. les prochaines sous-sections donneront plus de détails sur la manière de calculer ces deux mesures.

La deuxième étape de l'approche proposée consiste à appliquer les différentes étapes d'un FLS aux deux mesures calculées dans la première étape, afin de prendre en compte le flou et l'imprécision des sentiments exprimés dans un tweet. Pour cela, nous appliquons les différentes étapes d'un système flou (fuzzification, règles floues et défuzzification) sur la positivité et la négativité pour trouver à la fin de ce processus la classe du tweet.

Cette étape décrit comment utiliser les mesures calculées précédemment dans un système à logique floue. La positivité et la négativité des tweets joueront le rôle des entrées du système à logique floue, et en appliquant les trois étapes (fuzzification, Règles d'inférence

et défuzzification) sur ces deux mesures, on retrouve en sortie la classe du tweet (positif, négatif ou neutre). En utilisant les valeurs de positivité et de négativité, nous pouvons décider à l'aide du système à logique floue si le tweet est négatif, positif ou neutre.

Pour appliquer le système à logique floue (FLS) à un problème d'analyse des sentiments, nous devons disposer des variables qui joueront le rôle des entrées. C'est pour cette raison que nous proposons de calculer pour chaque tweet les deux mesures introduites précédemment.

Pour extraire les opinions et les sentiments à partir des données des réseaux sociaux, la plupart des articles publiés sont basés soit sur des algorithmes d'apprentissage automatique et les systèmes de la recherche d'informations(SRI) [11] sans tenir compte du flou des sentiments, soit uniquement sur des systèmes d'inférence floue [19]. L'approche proposée tire parti des deux méthodes pour améliorer les résultats de la classification. Nous utilisons un nouveau raisonnement pour proposer de nouvelles mesures (positivité et négativité des tweets), ainsi qu'un nouvel algorithme basé sur un système d'inférence floue pour classer un tweet en classes (extraire les sentiments des tweets).

Nous développons notre travail de manière parallèle en utilisant le framework Hadoop avec le système de fichiers distribué Hadoop (HDFS) et Hadoop MapReduce. Pour cela, nous proposons deux nouveaux algorithmes MapReduce basés sur les deux approches proposées. Nos approches sont multilingues, il prend en compte toutes les langues(voir la dernière section du chapitre 3 de la partie I).

3.4.1 Positivité et Négativité avec la première approche

Dans cette sous-section, nous présenterons comment nous calculons les deux mesures proposées : **la positivité** et **la négativité** des tweets, à l'aide des notions des systèmes de recherche d'informations (IRS), du dictionnaire WordNet et de quelques mesures de similarité sémantique(première approche).

Comme nous l'avons dit précédemment, la méthode proposée consiste à définir pour chaque tweet à classer deux mesures : **positivité** et **négativité** en calculant la similarité sémantique entre ce tweet et deux documents d'opinion Dp (document positif) et Dn (document négatif). Le document Dp contient des mots d'opinion positive tels que : "happy", "good", "exciting", "fabulous"...etc, et le document Dn contient des mots d'opinion négative tels que : anger, sadness, fear, etc.

Dans notre travail et parce que nous voulons faire une approche hybride entre la similarité sémantique et les concepts de la logique floue, nous devons calculer pour chaque tweet les valeurs initiales des variables d'entrée que nous avons appelées la positivité et la négativité.

Pour calculer la mesure de **positivité** d'un tweet, nous calculons la moyenne de la somme de la similarité sémantique entre chaque mot d'opinion du tweet et chaque mot du document positif Dp. Comme la similarité sémantique a une valeur comprise entre 0 et 1, la positivité du tweet sera également comprise entre 0 et 1.

De la même manière, pour la mesure de **négativité**, nous calculons la moyenne de la somme de la similarité sémantique entre chaque mot d'opinion du tweet et chaque mot du document négatif Dn, et parce que la valeur de la similarité sémantique est comprise entre 0 et 1 la négativité du tweet sera également comprise entre 0 et 1.

Pour le calcul de ces deux mesures, nous devons extraire les mots d'opinion du tweet, ces mots seront utilisés dans l'étape du calcul de la similarité sémantique avec les documents Dp et Dn.

Sur la base des résultats présentés dans le tableau II.4, nous décidons de calculer la similarité sémantique entre le tweet et les documents Dp et Dn en utilisant l'approche de Leacock et Chodorow. Pour mettre en œuvre cette approche, nous utilisons le dictionnaire sémantique **WordNet**.

La figure II.23 montre comment nous calculons la positivité et la négativité d'un tweet à l'aide de notre proposition.

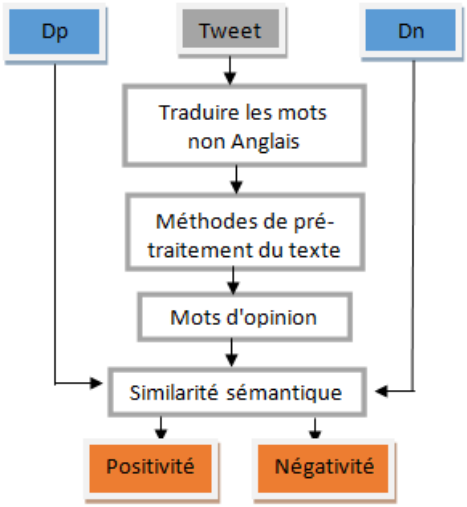


FIGURE II.23 – Calcul de la positivité et de la négativité avec la première approche

À partir de la figure II.23, la première chose à faire est de créer les deux documents d'opinion Dp et Dn (Dp contient les mots d'opinion positive et Dn contient les mots d'opinion négative), puis après l'application des différentes méthodes de pré-traitement du texte sur le tweet à classer (et le traduire s'il n'est pas écrit en anglais) et l'extraction des mots d'opinion, nous calculons la similarité sémantique en utilisant l'approche de Leacock et Chodorow entre chaque mot d'opinion du tweet et chaque mot de Dp pour calculer la mesure de positivité ; et entre chaque mot d'opinion du tweet et chaque mot

de Dn pour calculer la mesure de négativité.

Les formules suivantes (24) et (25) montrent comment nous calculons la positivité et la négativité en utilisant la première approche.

$$Positivité = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} Sim_{LC}(W_i, Wdp_j)}{N \times M} \tag{24}$$

$$Négativité = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} Sim_{LC}(W_i, Wdn_j)}{N \times M} \tag{25}$$

Où :

- N est le nombre de mots d'opinion dans le tweet.
- M est le nombre de mots d'opinion dans le document positif Dp, ou dans le document négatif Dn.
- W_i est le ième mot d'opinion du tweet.
- Wdp_j est le j-ème mot du document d'opinion positive Dp.
- Wdn_j est le j-ème mot du document d'opinion négative Dn.
- $Sim_{LC}(W_i, Wdp_j$ est la similarité sémantique entre un mot d'opinion du tweet et un mot du document d'opinion positive Dp en utilisant l'approche de Leacock et Chodorow.
- $Sim_{LC}(W_i, Wdn_j$ est la similarité sémantique entre un mot d'opinion du tweet et un mot du document d'opinion négative Dn en utilisant l'approche de Leacock et Chodorow.

À la fin de ce processus, nous trouvons la valeur nette(crisp value) de la positivité et de la négativité, par exemple si nous voulons classifier un tweet T, la première chose à faire est d'appliquer les méthodes de pré-traitement de texte (TP) pour extraire les mots d'opinion. Supposons qu'après l'étape d'application des méthodes TP nous trouvions 4 mots d'opinion qui sont : "happy, bad, exciting, sad", et après le calcul de la similarité sémantique entre T, Dp et Dn, nous trouvons que la positivité = 0.82 et la négativité = 0,625. Ces valeurs finales des variables positivité et négativité joueront le rôle des valeurs nettes initiales (entrées) de notre système à logique floue.

3.4.2 Positivité et Négativité avec la deuxième approche

Dans cette sous-section, nous décrirons comment nous utilisons SentiWordNet dans notre travail pour créer une approche hybride avec les concepts de la logique floue et comment nous l'utilisons pour calculer les valeurs des deux mesures proposées : la positivité et la négativité. SentiWordNet nous donne la possibilité de connaître pour chaque mot

son degré de positivité si le mot est positif (exprime un sentiment positif) et son degré de négativité si le mot est négatif. Ce degré est compris entre 0 et 1 si le mot est positif et entre 0 et -1 s'il est négatif.

Avant de calculer ces deux mesures, il est nécessaire de détecter la langue du tweet et de la traduire ; s'il n'est pas écrit en anglais, et appliquez également les différentes méthodes de pré-traitement du texte pour extraire uniquement les mots d'opinion, qui seront ensuite utilisés par SentiWordNet pour calculer nos deux mesures proposées. Pour le calcul de la positivité et de la négativité, nous devons savoir pour chaque mot d'opinion, s'il est positif ou négatif, pour cela nous calculons son degré à partir de SentiWordNet, si la valeur calculée est supérieure à 0, le mot est positif, sinon (s'il est inférieur à 0), le mot est négatif.

Pour calculer la mesure de positivité, nous calculons la moyenne de la somme du degré de chaque mot d'opinion positive ; et comme le degré de chaque mot positif dans sentiwordNet est compris entre 0 et 1, la positivité du tweet sera également comprise entre 0 et 1.

De la même manière, pour la mesure de négativité, nous calculons la moyenne de la somme du degré de chaque mot d'opinion négative ; et comme le degré de chaque mot négatif dans sentiwordNet est compris entre 0 et -1, la négativité du tweet sera également comprise entre 0 et -1, et pour rendre la négativité du tweet entre 0 et 1, nous la multiplions par -1.

Les formules suivantes (26) et (27) montrent comment nous calculons la positivité et la négativité en utilisant SentiWordNet.

$$Positivité = \frac{\sum_{i=1}^{P} SentiWordNet(Wp_i)}{P} \tag{26}$$

$$Négativité = -1 \times \frac{\sum_{i=1}^{N} SentiWordNet(Wn_i)}{N} \tag{27}$$

Où :

- P est le nombre de mots d'opinion positive dans le tweet.
- N est le nombre de mots d'opinion négative dans le tweet.
- Wp_i est le ième mot d'opinion positive du tweet.
- Wn_i est le ième mot d'opinion négative du tweet.
- $SentiWordNet(Wp_i)$ calcule le degré sentimental du mot d'opinion positive Wp_i en utilisant le dictionnaire SentiWordNet.
- $SentiWordNet(Wn_i)$ calcule le degré de sentiment du mot d'opinion négative Wn_i

en utilisant le dictionnaire SentiWordNet.

L'**algorithme 6** montre les différentes étapes suivies pour calculer la positivité et la négativité des tweets avec la seconde approche.

Algorithm 6 Positivité et négativité des tweets

```
Require: Positivity Value & Negativity Value
Require: tweet's class
  P ← 0
  N ← 0
  c ← 0
  d ← 0
  if Tweet is not in English then
    Tweet ← Translate(Tweet)
  end if
  Tweet ← TextPreProcesssing(Tweet)
  SE[] ← Split(Tweet)
  OW[] ← OpinionWord(SE)
  for all word ∈ OW do
    if SentiWordNet(word)>0 then
      P ← P+SentiWordNet(word)
      c ← c+1
    else if SentiWordNet(word)<0 then
      N ← N+(SentiWordNet(word) × -1)
      d ← d+1
    end if
  end for
  Positivity ← P/c
  Negativity ← N/d
```

Où :

- **Translate(Tweet)** : est une fonction qui traduit chaque tweet non anglais en anglais.
- **OpinionWord(Tweet)** : est une fonction qui permet de vérifier si un mot dans un tweet est un mot d'opinion ou non, c'est-à-dire si ce mot est un verbe, un adjectif ou un adverbe.
- **SentiWordNet(Word)** : calcule la polarité de chaque mot d'opinion du tweet.

Donc, à la fin de ce processus, nous trouvons la valeur nette de la positivité et de la négativité. Par exemple, si nous voulons classer un tweet T, la première chose à faire est d'utiliser les méthodes de pré-traitement du texte (TP) pour extraire les mots d'opinion. Supposons qu'après l'étape d'application des méthodes de TP, nous trouvions 4 mots d'opinion : "happy, bad, exciting, sad", qui sont 2 mots positifs :"happy" avec le degré

0,75 et "exciting" avec le degré 0,89 de SentiWordNet, et 2 mots négatifs : "bad" avec le degré -0,68 et "sad" avec le degré -0,57. En appliquant l'approche proposée *la positivité* $= (0.75 + 0.89)/2 = 0.82$ et *la négativité* $= ((-0.68 - 0.57)/2) \times -1 = 0.625$. Donc, pour le tweet T, le degré de positivité est égal à 0,82 et la négativité est égal à 0,625.

3.4.3 Système à logique floue proposé

Dans cette sous-section, nous décrivons nos deux approches hybrides d'analyse des sentiments, la première approche est basée sur la similarité sémantique/système à logique floue(FLS), et la deuxième est basée sur SentiWordNet/système à logique floue(FLS). Comme présenté précédemment, le système à logique floue commence par une valeur nette et après l'avoir été fuzzifié en utilisant différentes étapes (fuzzification, règles d'inférence) il renvoie une valeur nette dans la sortie en utilisant les méthodes de défuzzification (centroïde, mean/max...). La figure II.24 présente la structure générale de notre système à logique floue.

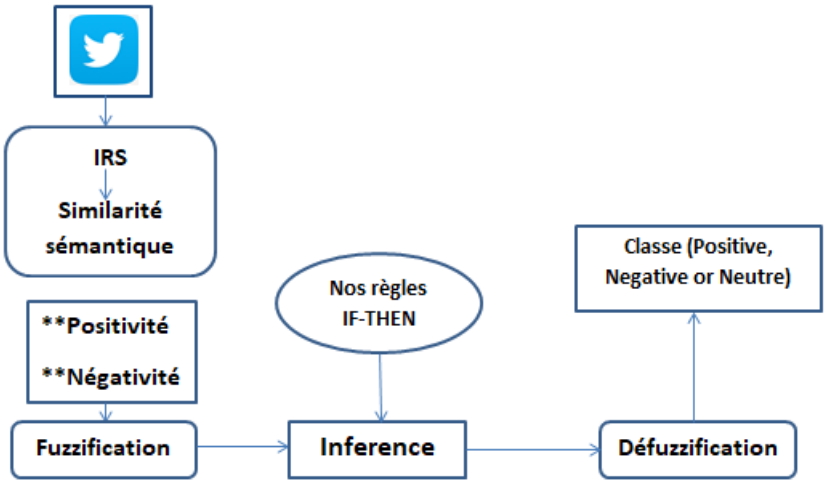


FIGURE II.24 – Système à Logique Floue Proposé

D'après la figure II.24, l'entrée (valeur nette) du système à logique floue correspond aux deux mesures (positivité et négativité) calculées à l'aide de la similarité sémantique entre le tweet et les deux documents d'opinion (figure II.23) ou avec l'utilisation du SentiWordNet. La sortie (valeur nette) correspond à la classe du tweet (positive, négative ou neutre).

Comme présenté précédemment, la première étape d'un FLS consiste à définir les entrées et les sorties. C'est-à-dire, la définition des variables linguistiques en entrée et en sortie. Dans notre cas et parce que nous voulons classer les tweets selon trois classes (positive, négative et neutre), nous définissons deux variables d'entrée qui sont la **positivité** et la **négativité** des tweets, et une variable de sortie qui est la **classe**(sentiment) du tweet.

Dans un FLS, chaque variable en entrée ou en sortie est appelée variable linguistique, chaque variable linguistique possède un certain nombre de valeurs pouvant être prises, ces valeurs sont appelées termes linguistiques ou ensembles flous. Dans notre cas, nous avons deux variables linguistiques en entrée, qui sont la positivité et la négativité, chacune d'elles a trois termes linguistiques qui sont **low, moderate et high**. Cela signifie que les variables de positivité et de négativité peuvent prendre trois valeurs possibles (low, moderate et high), ou en d'autres termes, peuvent appartenir à trois ensembles flous différents. De même, dans la sortie, nous avons une variable linguistique qui est la classe du tweet, et elle peut également prendre trois termes linguistiques différents qui sont **positive, négative et neutre**.

Après avoir défini les variables linguistiques et leurs termes linguistiques dans l'entrée et la sortie, la prochaine étape de notre FLS consiste à définir les valeurs nettes des entrées avec lesquelles nous allons commencer l'approche proposée. Pour cela et comme expliqué précédemment, nous utilisons la similarité sémantique, les notions des systèmes de recherche d'informations ou SentiWordNet, pour calculer la positivité et la négativité du tweet qui joueront le rôle des valeurs nettes de l'entrée.

Étape de fuzzification

$$f(x) = \begin{cases} 0 & \text{if } x \leqslant a \\ \dfrac{x-a}{b-a} & \text{if } a \leqslant x \leqslant b \\ 1 & \text{if } b \leqslant x \leqslant c \\ \dfrac{d-x}{d-c} & \text{if } c \leqslant x \leqslant d \\ 0 & \text{if } d \leqslant x \end{cases} \tag{28}$$

La figure II.25 montre une FA de forme trapézoïdale pour un terme linguistique avec a=0.2, b=0.4, c=0.6 et d=0.8.

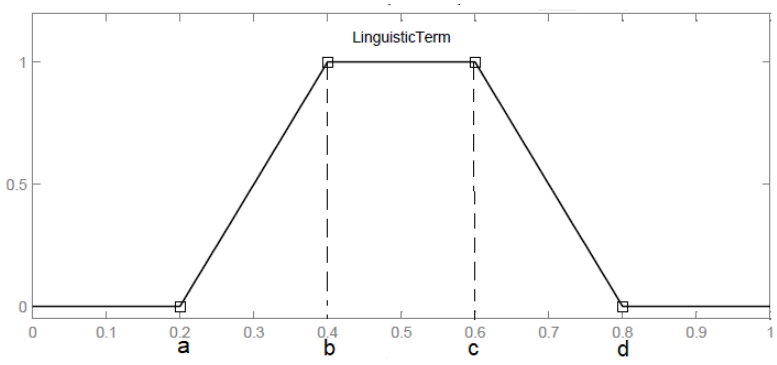


FIGURE II.25 – Fonction d'appartenance de forme trapézoïdale

D'autre part, triangulaire MF est une fonction qui dépend de trois paramètres scalaires a, b et c, tels que donnés par la formule 29 :

$$f(x) = \begin{cases} 0 & \text{if } x \leqslant a \\ \dfrac{x-a}{b-a} & \text{if } a \leqslant x \leqslant b \\ \dfrac{c-x}{c-b} & \text{if } b \leqslant x \leqslant c \\ 0 & \text{if } c \leqslant x \end{cases} \tag{29}$$

La figure II.26 montre une FA triangulaire pour un terme linguistique avec a=0.3, b=0.5 et c=0.7.

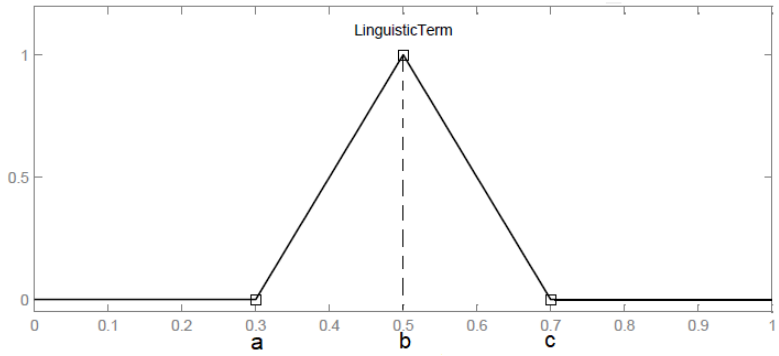


FIGURE II.26 – Fonction d'appartenance triangulaire

Dans notre cas, nous devons fuzzifier les variables d'entrée en utilisant l'une des FA présentées précédemment, pour cela nous devons définir la FA de chaque terme linguistique des entrées. Les variables linguistiques "positivité" et "négativité" ont trois termes linguistiques (trois ensembles flous), nous devons donc définir trois FA, une pour l'ensemble flou "low", une pour "moderate" et une autre pour "high". La prochaine étape pour calculer les fonctions d'appartenance est la définition des paramètres a, b, c et d pour chaque terme linguistique. Le choix de ces paramètres dépend du domaine d'application du système à logique floue et nécessite également la présence d'un expert dans ce domaine qui dispose de l'expérience et du matériel nécessaires pour définir ces paramètres. Par exemple, dans notre cas et comme expliqué précédemment, les entrées "positivité" et "négativité" ont des valeurs comprises entre 0 et 1, donc les valeurs a, b, c et d seront également comprises dans l'intervalle [0 ;1].

Comme nous avons présenté précédemment, si par exemple nous voulons travailler sans les concepts du FLS, nous définissons pour chaque terme linguistique (low, moderate et high) un intervalle compris entre 0 et 1, par exemple la positivité ou la négativité est "low" s'il est compris entre 0 et 0.4, "moderate" s'il est compris entre 0.4 et 0.6 et "high" s'il est compris entre 0.6 et 1. Donc, en utilisant l'ensemble classique, chaque valeur de

la positivité ou de la négativité appartient à un ensemble (low, moderate ou high) avec un degré d'appartenance égal à 1, ou elle n'appartient pas à un ensemble avec un degré d'appartenance égal à 0.

La réalité est toujours loin d'être optimiste. Premièrement, les termes relatifs aux sentiments sont flous. Le même mot peut expliquer différentes orientations de sentiment, même dans le même domaine. Deuxièmement, les sentiments de l'humain sont souvent flous. Par exemple, on peut utiliser un mot pour exprimer plus d'un sentiment à la fois. Dans cet article, et pour tenir compte du flou et de l'imprécision des sentiments, nous utilisons la logique floue pour classifier chaque tweet, c'est-à-dire que chaque valeur de positivité et de négativité appartiendra à chaque ensemble flou avec un degré d'appartenance entre 0 et 1, nous utilisons pour cela les fonctions d'appartenance(FA) pour chaque terme linguistique, et pour définir chaque FA, il faut choisir les paramètres (a, b, c ...).

Les figures II.27 et II.28 montrent respectivement la FA trapézoïdale et la FA triangulaire pour nos deux entrées : Positivité et Négativité :

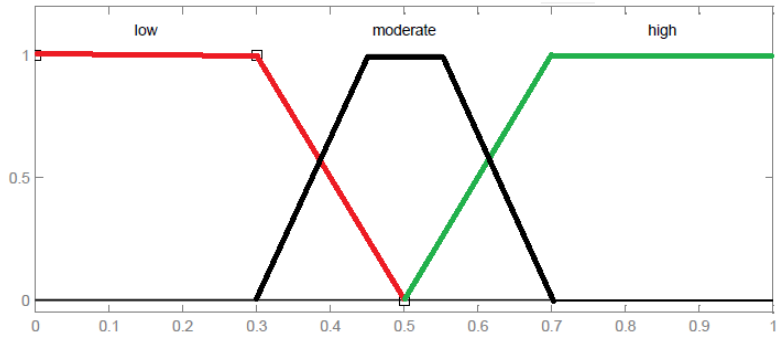


FIGURE II.27 – Trapézoïdale MF pour les variables d'entrée

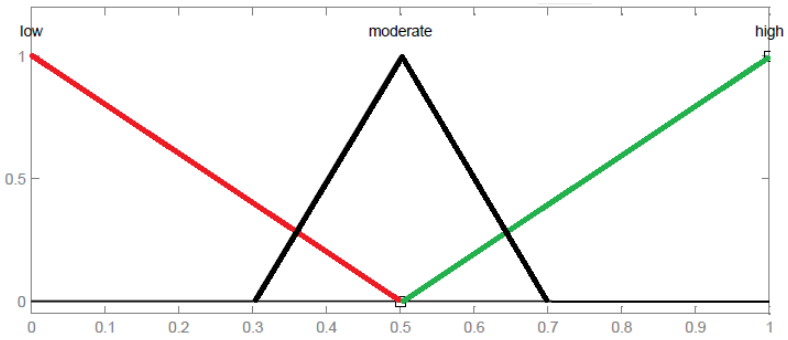


FIGURE II.28 – Triangulaire MF pour les variables d'entrée

A partir de ces deux figures (II.27 et II.28), par exemple pour le terme linguistique "moderate", les valeurs des paramètres sont a=0.3, b=0.4, c=0.6 et d=0.7 en utilisant la FA trapézoïdale ; a=0.3, b=0.5 et c=0.7 en utilisant la FA triangulaire. Les valeurs de ces

paramètres ont été définies empiriquement, et nous avons choisi les valeurs optimales qui donnent de bons résultats de classification.

Une fois les fonctions d'appartenance sont définies pour l'entrée et la sortie, l'étape suivante consiste à définir les règles de contrôle flou.

Règles d'inférence

L'étape suivante après la fuzzification des entrées est l'étape de la définition et de l'application des différentes règles de notre problème, c'est-à-dire la combinaison des fonctions d'appartenance avec les règles de contrôle pour obtenir la sortie floue. Dans notre travail et parce que nous avons deux entrées et une sortie avec trois termes linguistiques, nous avons défini neuf règles floues à l'aide du modèle IF-THEN avec l'opération logique AND entre la valeur des entrées. Les neuf règles de notre système à logique floue sont les suivantes :

- **IF** Positivité est low **AND** Negativité est low **THEN** Class est Neutre.
- **IF** Positivité est moderate **AND** Negativité est moderate **THEN** Class est Neutre.
- **IF** Positivité est high **AND** Negativité est high **THEN** Class est Neutre.
- **IF** Positivité est low **AND** Negativité est moderate **THEN** Class est Negative.
- **IF** Positivité est low **AND** Negativité est high **THEN** Class est Negative.
- **IF** Positivité est moderate **AND** Negativité est high **THEN** Class est Negative.
- **IF** Positivité est moderate **AND** Negativité est low **THEN** Class est Positive.
- **IF** Positivité est high **AND** Negativité est moderate **THEN** Class est Positive.
- **IF** Positivité est high **AND** Negativité est low **THEN** Class est Positive.

Après l'application des différentes règles de notre système, l'étape suivante consiste à les impliquer pour générer la valeur de la sortie de chacune d'entre elles. Dans notre cas et parce que nous utilisons l'opération AND entre les entrées, les sorties prennent la valeur minimale entre les entrées (formule 30). La dernière étape de l'inférence des règles est l'agrégation des résultats obtenus pour chaque sortie afin de trouver une valeur pour chacune. Par exemple, pour la sortie **neutre**, nous trouverons trois valeurs différentes en appliquant trois règles, et pour trouver la valeur finale de la sortie "neutre" nous calculons le maximum de ces trois valeurs.

$$Rule\_Output\_Value = min[Positivity\_Input, Negativity\_Input] \tag{30}$$

Défuzzification

Après la fuzzification des entrées et l'application des neuf règles de notre FLS, nous retrouvons le degré d'appartenance de notre sortie (classe du tweet) à chaque ensemble flou de sortie (positif, négatif et neutre), et pour trouver le résultat final de notre FLS

qui doit se présenter sous la forme d'une valeur nette, nous devons appliquer l'étape de défuzzification.

Le processus de défuzzification consiste à convertir la sortie floue en sortie nette ou classique. La conclusion ou sortie floue est toujours une variable linguistique et cette variable linguistique doit être convertie en variable crisp via le processus de défuzzification.

Dans ce travail, nous utilisons 4 techniques de défuzzification couramment utilisées, à savoir :

- **Principe d'appartenance maximale :** cette méthode calcule le maximum entre la valeur d'appartenance de la sortie à chaque ensemble flou.
- **Méthode centroïde :** parfois appelée centre de gravité, la méthode du centre de gravité (Center of Gravity COG) est la technique de défuzzification la plus répandue. Elle est largement utilisée dans les applications réelles. Cette méthode est similaire à la formule de calcul du centre de gravité en physique.
- **Méthode de la moyenne pondérée :** cette méthode n'est valide que pour les fonctions d'appartenance en sortie symétrique.
- **Méthode Mean-Max :** cette méthode (également appelée milieu-de-maxima) est étroitement liée à la première, à la différence que les emplacements de l'appartenance maximale peuvent être non uniques (c'est-à-dire que l'appartenance maximale peut être un plateau plutôt qu'un seul point).

Après avoir trouvé la valeur finale nette de la sortie (crisp value of the output CVO), la dernière étape consiste à comparer le résultat obtenu avec trois plages : tweet est négatif si CVO est compris entre 0 et 0.4, il est neutre si CVO est strictement supérieur à 0.4 et strictement inférieur à 0.6, et positif si CVO est compris entre 0.6 et 1.

3.4.4 Schématisation de notre travail

La figure II.29 donne les différentes étapes de notre méthode en utilisant la première approche proposée (similarité sémantique+FLS) :

La figure II.30 donne les différentes étapes de notre méthode en utilisant la deuxième approche proposée (SentiWordNet+FLS) :

3.4.5 Exemple d'application

Dans cette sous-section, nous décrivons à l'aide d'un exemple comment nous classifions un tweet selon trois classes (positive, négative ou neutre) en utilisant la méthode proposée. Pour cela, nous supposons que nous voulons classifier un tweet T, donc la première étape après la traduction de T s'il a été écrit dans une langue autre que l'anglais, l'application

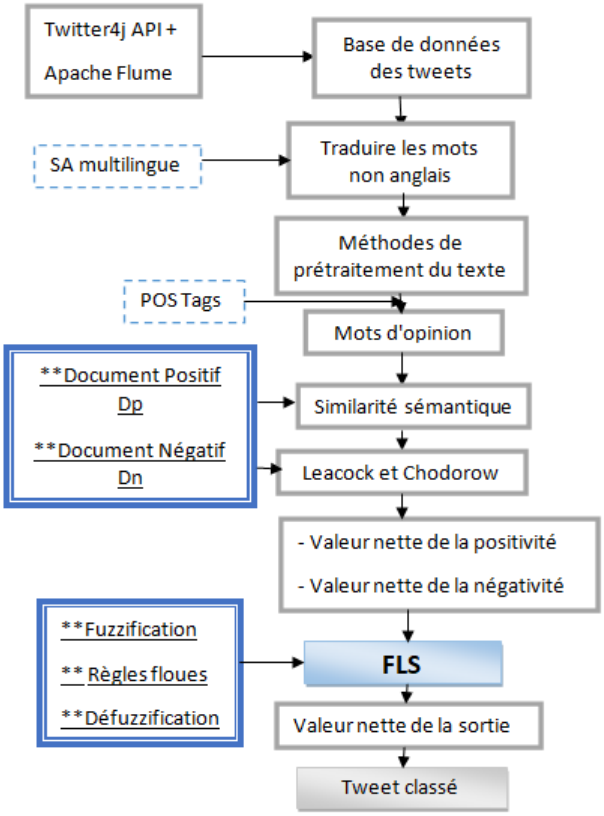


FIGURE II.29 – Différentes étapes basées sur la première approche

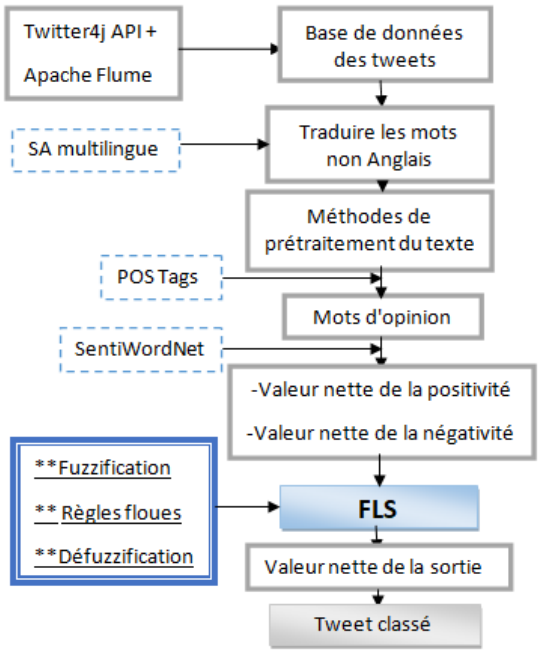


FIGURE II.30 – Différentes étapes basées sur la deuxième approche

des méthodes de pré-traitement du texte et après en avoir extrait les mots d'opinion, il s'agit de l'étape de calcul des variables d'entrée "positivité" et "négativité" (valeurs nettes)

à l'aide des approches décrites précédemment en utilisant soit la similarité sémantique avec l'approche de Leacock et Chodorow et le dictionnaire WordNet, ou en utilisant le dictionnaire SentiWordNet. Supposons qu'après toutes ces étapes, nous trouvions que les valeurs nettes des entrées sont : **positivité P=0.68** et **négativité N=0.35**.

Après avoir calculé les valeurs nettes(crisp values) des entrées, l'étape suivante est la fuzzification de ces valeurs nettes. Pour ce faire, nous utilisons la fonction d'appartenance de forme trapézoïdale présentée dans la formule 28 et les résultats illustrés dans la figure II.27. Avec tout cela, nous calculons le degré d'appartenance de la positivité P et de la négativité N pour chaque ensemble flou (Low, High et Moderate) comme suit :

- f(P,low)= **0**, car $P = 0.68 \geq d = 0.5$.
- f(P,moderate)=$\frac{d-P}{d-c} = \frac{0.7-0.68}{0.7-0.55} =$ **0.13**, car $c = 0.55 \leqslant P = 0.68 \leqslant d = 0.7$.
- f(P,high)=$\frac{P-a}{b-a} = \frac{0.68-0.5}{0.7-0.5} =$ **0.9**, car $a = 0.5 \leqslant P = 0.68 \leqslant b = 0.7$.
- f(N,low)=$\frac{d-N}{d-c} = \frac{0.5-0.35}{0.5-0.3} =$ **0.75**, car $c = 0.3 \leqslant N = 0.35 \leqslant d = 0.5$.
- f(N,moderate)=$\frac{N-a}{b-a} = \frac{0.35-0.3}{0.45-0.3} =$ **0.33**, car $a = 0.3 \leqslant N = 0.35 \leqslant b = 0.45$.
- f(N,high)= **0**, car $N = 0.35 \leqslant a = 0.5$

Après l'étape de fuzzification, c'est l'étape de l'application et de l'implication de nos neuf règles comme présentées ci-dessous :

- IF (P is low)=0 AND (N is low)=0.75 THEN (Tweet is Neutral)=min$(0, 0.75) =$ **0**.
- IF (P is moderate)=0.13 AND (N is moderate)=0.33 THEN (Tweet is Neutral)=min$(0.13, 0.33) =$ **0.13**.
- IF (P is high)=0.9 AND (N is high)=0 THEN (Tweet is Neutral)=min$(0.9, 0) =$ **0**.
- IF (P is low)=0 AND (N is moderate)=0.33 THEN (Tweet is Negative)=min$(0, 0.33) =$ **0**.
- IF (P is low)=0 AND (N is high)=0 THEN (Tweet is Negative)=min$(0, 0) =$ **0**.
- IF (P is moderate)=0.13 AND (N is high)=0 THEN (Tweet is Negative)=min$(0.13, 0) =$ **0**.
- IF (P is moderate)=0.13 AND (N is low)=0.75 THEN (Tweet is Positive)=min$(0.13, 0.75) =$ **0.13**.
- IF (P is high)=0.9 AND (N is moderate)=0.33 THEN (Tweet is Positive)=min$(0.9, 0.33) =$ **0.33**.
- IF (P is high)=0.9 AND (N is low)=0.75 THEN (Tweet is Positive)=min$(0.9, 0.75) =$

0.75.

L'étape suivante est l'agrégation de ces règles pour chaque ensemble flou de sortie.

- Tweet is neutral→ $\max(0, 0.13, 0) = 0.13$
- Tweet is Negative→ $\max(0, 0, 0) = 0$
- Tweet is Positive→ $\max(0.13, 0.33, 0.75) = 0.75$

Donc, après toutes ces étapes, nous trouvons le degré d'appartenance de la variable de sortie à chaque ensemble flou de sortie, et en utilisant les MFs de la sortie présentée dans la figure II.31 et la méthode du centroïde, nous défuzzifions la sortie pour trouver la valeur nette finale de la sortie (classe du tweet) comme présenté à la figure II.32.

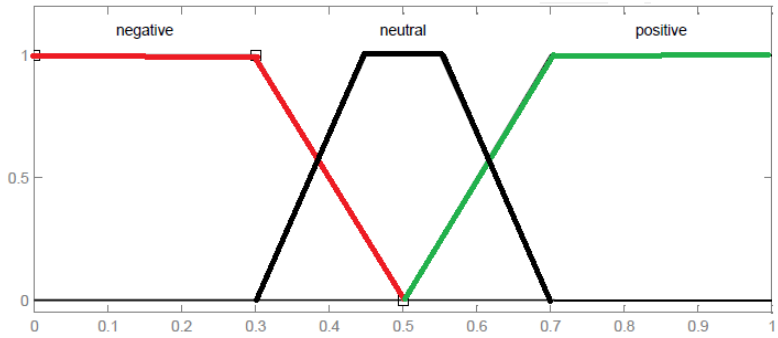


FIGURE II.31 – Trapézoïdale MF pour la sortie

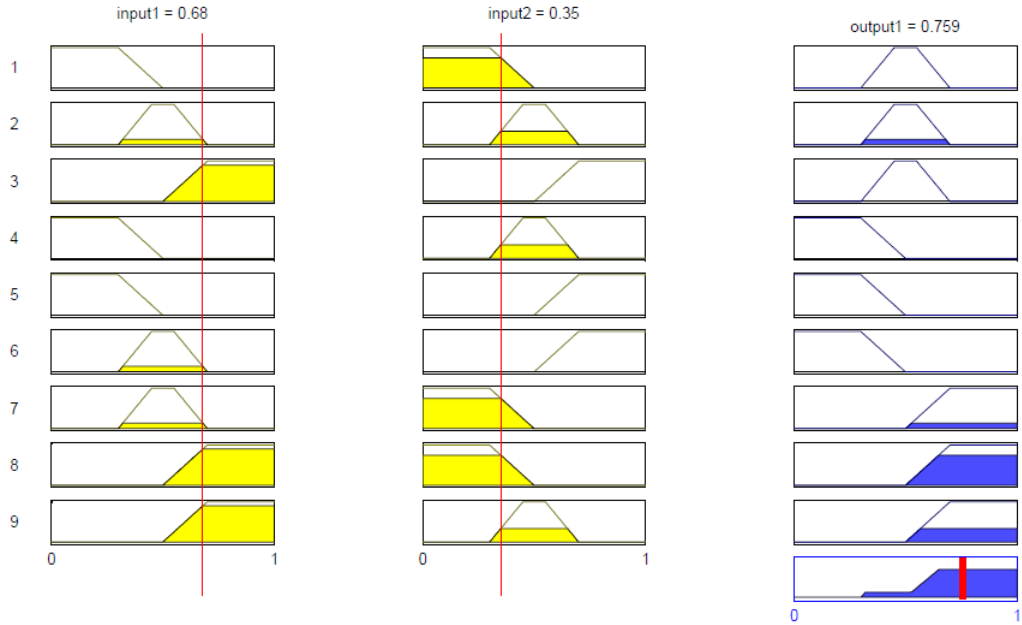


FIGURE II.32 – Résultat Final de notre FLS

Après l'étape de défuzzification en utilisant la méthode du centroïde, et comme présenté à la figure II.32, la valeur finale nette obtenue en appliquant la méthode proposée est égale à 0,759, et comme cette valeur est comprise entre 0,6 et 1, le tweet T est **positif**.

3.4.6 Etape de parallélisation

La figure II.33 montre les différentes étapes de la parallélisation de notre travail en utilisant la première approche.

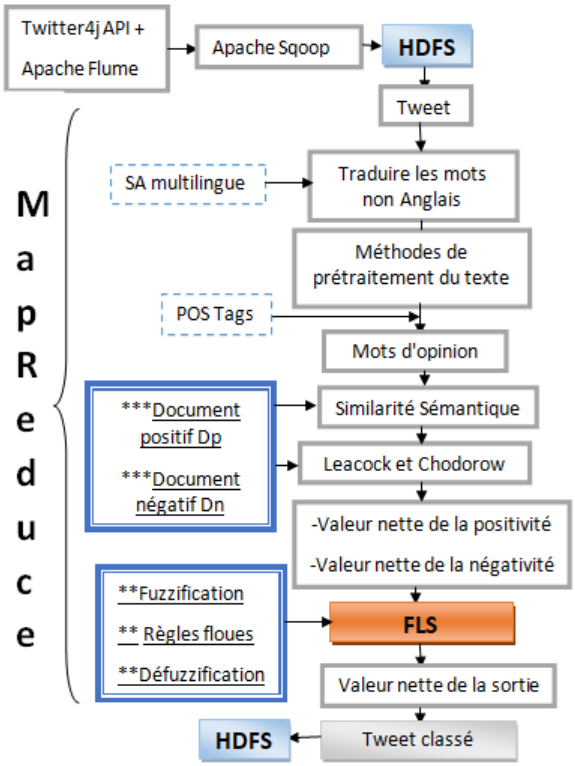


FIGURE II.33 – Différentes étapes pour la parallélisation en utilisant la première approche

La figure II.34 montre les différentes étapes de la parallélisation en utilisant la deuxième approche.

Comme présenté aux figures II.33 et II.34, nous avons stocké les entrées de notre travail (ensemble de données de tweets à classer) et la sortie (tweet classifié) dans HDFS. Le processus de la classification est effectué à l'aide du modèle de programmation MapReduce. L'entrée de MapReduce est un tweet, et après l'application de toutes nos étapes, nous obtenons dans la sortie le résultat de la classification (positive, négative ou neutre).

Nos algorithmes MapReduce suivis pour la classification des tweets avec la première et la deuxième approche proposée sont présentés respectivement dans l'**algorithme 7** et l'**algorithme 8** :

Où :

- Sim_{LC}**(word, term)** calcule la similarité sémantique en utilisant l'approche de Leacock et Chodorow entre chaque mot du tweet et un mot du document positif ou du document négatif.

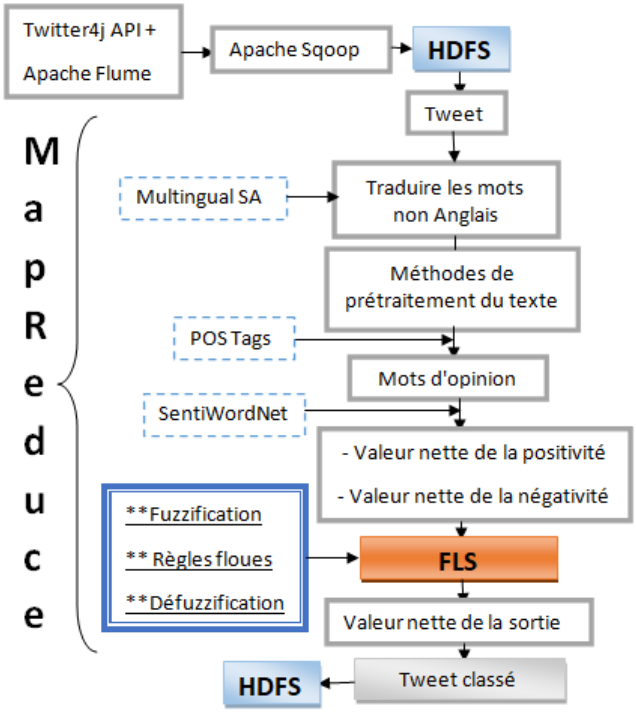


FIGURE II.34 – Différentes étapes pour la parallélisation en utilisant la deuxième approche

- **Fuzzification(Positivity AND Negativity)** permet de fuzzifier les entrées en utilisant leurs valeurs nettes et la fonction d'appartenance trapézoïdale.
- **Implication(Fuzzy Rules) & Aggregation(Fuzzy Rules)** applique les différentes règles pour trouver la valeur floue de chaque terme de sortie (positive, négative ou neutre).
- **Defuzzification(FuzzyOutput)** renvoie la valeur finale nette de la sortie (classe du tweet) en utilisant une méthode de défuzzification.

3.5 Résultats expérimentaux

Dans cette section, nous allons présenter quelques résultats expérimentaux de nos travaux. Comme présenté précédemment, la première étape consiste à construire la base de données des tweets (la collection de tweets) à l'aide de l'API Twitter4j et d'Apache Flume. Pour cela, notre ensemble de données contient des tweets publiés entre janvier 2016 et janvier 2017 et contiennent des mots liés à des produits tels que : "iPhone", "Samsung", ou à des événements tels que "World cup 2018", etc. Les tweets collectés seront stockés dans HDFS avec Apache sqoop.

Comme présenté précédemment, notre FLS contient deux variables d'entrée (positivité et négativité) et une variable de sortie (classe du tweet). La plage de chaque variable est comprise entre 0 et 1 et chacun comporte trois termes linguistiques (valeurs linguistiques

Algorithm 7 Modèle de programmation MapReduce en utilisant la première approche

Require: tweet's class
P ← 0
N ← 0
c ← 0
cc ← 0
if Tweet is not in English **then**
Tweet ← Translate(Tweet)
end if
Tweet ← TextPreProcesssing(Tweet)
SE[] ← Split(Tweet)
OW[] ← OpinionWord(SE)
Dp[] ← Split(Positive Document)
Dn[] ← Split(Negative Document)
for all word ∈ OW **do**
for all term ∈ Dp **do**
P ← P+Sim_{LC}(word,term)
c ← c+1
end for
for all term ∈ Dn **do**
N ← N+Sim_{LC}(word,term)
cc ← cc+1
end for
end for
Positivity ← P/c
Negativity ← N/cc
Fuzzification(Positivity AND Negativity)
Implication(Fuzzy Rules)
FuzzyOutput ← Aggregation(Fuzzy Rules)
CrispOutput ← Defuzzification(FuzzyOutput)
if CrispOutput≥ 0.6 **then**
sentiment ← positive
else if CrispOutput≤ 0.4 **then**
sentiment ← negative
else if 0.4<CrispOutput< 0.6 **then**
sentiment ← neutral
end if
write(tweet,sentiment)

ou ensembles flous) : "low", "moderate" et "high" pour les variables d'entrée ; négative, neutre et positive pour la variable de sortie. Sur la base de la valeur de la positivité et de la négativité, nous calculons le degré d'appartenance de chacun à chaque ensemble flou d'entrée ("low", "moderate" ou "high") à l'aide des fonctions d'appartenance. Ensuite, nous appliquons les différentes règles floues pour trouver le degré d'appartenance de la classe du tweet à chaque ensemble flou de sortie (positive, négative et neutre). La dernière

Algorithm 8 Modèle de programmation MapReduce en utilisant la deuxième approche

```
Require: tweet's class
  P ← 0
  N ← 0
  c ← 0
  cc ← 0
  if Tweet is not in English then
    Tweet ← Translate(Tweet)
  end if
  Tweet ← TextPreProcesssing(Tweet)
  SE[] ← Split(Tweet)
  OW[] ← OpinionWord(SE)
  for all word ∈ OW do
    if SentiWordNet(word)>0 then
      P ← P+SentiWordNet(word)
      c ← c+1
    else if SentiWordNet(word)<0 then
      N ← N+(SentiWordNet(word) × -1)
      cc ← cc+1
    end if
  end for
  Positivity ← P/c
  Negativity ← N/cc
  Fuzzification(Positivity AND Negativity)
  Implication(Fuzzy Rules)
  FuzzyOutput ← Aggregation(Fuzzy Rules)
  CrispOutput ← Defuzzification(FuzzyOutput)
  if CrispOutput≥ 0.6 then
    sentiment ← positive
  else if CrispOutput≤ 0.4 then
    sentiment ← negative
  else if 0.4<CrispOutput< 0.6 then
    sentiment ← neutral
  end if
  write(tweet,sentiment)
```

étape qui est la défuzzification (par l'application d'une méthode de défuzzification) nous donne la décision finale, c'est-à-dire si le tweet est négatif, positif ou neutre. Le tableau II.6 présente les différentes variables de notre système proposé avec leurs ensembles flous et leurs paramètres.

Pour calculer les valeurs initiales (valeurs nettes) pour les entrées (calcul de la positivité et de la négativité), nous utilisons soit la première approche basée sur la similarité sémantique et les notions des systèmes de recherche d'informations, soit la seconde approche basée sur le dictionnaire SentiWordNet, comme présentée précédemment. Dans

Variable	Variables linguistiques	intervalle	Valeurs linguistiques	Paramètre
Entrée	Positivité	0-1	Low	0-0.5
			Moderate	0.3-0.7
			High	0.5-1
	Négativité	0-1	Low	0-0.5
			Moderate	0.3-0.7
			High	0.5-1
Sortie	Sentiment	0-1	Négative	0-0.5
			Neutre	0.3-0.7
			Positive	0.5-1

TABLE II.6 – Paramètres d'entrées et de sortie

notre FLS, nous utilisons deux fonctions d'appartenance pour la fuzzification : **Trapezoidal MF** et **Triangular MF**, neuf règles IF-THEN et quatre méthodes de défuzzification (Max-Membership, Centroid, Moyenne pondérée et Mean-Max). À partir de là, nous avons huit combinaisons possibles pour choisir la méthode de fuzzification/défuzzification à utiliser dans le processus de classification.

Le tableau II.7 et II.8(respectivement avec la première approche proposée et la deuxième approche proposée) présentent les résultats obtenus pour le taux de la classification et d'erreur après la classification des tweets en utilisant huit combinaisons possibles : Trapézoïde MF/Principe d'appartenance maximale, Trapézoïde MF/Centroïde, Trapézoïdale MF/Moyenne pondérée, Trapézoïdal MF/Mean-Max, Triangulaire MF/Principe d'appartenance maximale, Triangulaire MF/Centroïde, Triangulaire MF/Moyenne pondérée, Triangulaire MF/Mean-Max.

Fuzzification	Défuzzification	CR	ER
Trapézoïde MF	Appartenance-Maximale	77%	23%
	Centroïde	86%	14%
	Moyenne pondérée	68%	32%
	Mean-Max	77%	23%
Triangulaire MF	Appartenance-Maximale	68%	32%
	Centroïde	75%	25%
	Moyenne pondérée	77%	23%
	Mean-Max	79%	21%

TABLE II.7 – Taux de classification et d'erreur en utilisant différentes combinaisons de Fuzzification/Défuzzification avec la première approche.

D'après ces deux tableaux, la meilleure combinaison de fuzzification/défuzzification

Fuzzification	Défuzzification	CR	ER
Trapézoïde MF	Appartenance-Maximale	62%	38%
	Centroïde	84%	16%
	Moyenne pondérée	72%	28%
	Mean-Max	75%	25%
Triangulaire MF	Appartenance-Maximale	62%	38%
	Centroïde	79%	21%
	Moyenne pondérée	72%	28%
	Mean-Max	72%	28%

TABLE II.8 – Taux de classification et d'erreur en utilisant différentes combinaisons de Fuzzification/Défuzzification avec la deuxième approche.

est celle dans laquelle nous avons utilisé la fonction d'appartenance de forme trapézoïdale pour la fuzzification et la méthode du centroïde pour la défuzzification avec un taux de classification élevé (86% en utilisant la première approche, et 84% avec la deuxième approche) et un taux d'erreur égal à 14% et 16% respectivement en utilisant la première et la deuxième approche. Sur la base de ces résultats, nous utilisons dans notre FLS proposé la FA de forme **trapézoïdale** pour fuzzifier les entrées et la méthode du **centroïde** pour trouver la valeur finale nette de notre système.

Une autre expérience de notre travail consiste à démontrer l'effet de l'utilisation de la logique floue sur la classification des tweets. pour cela, nous comparons les résultats obtenus avec nos approches proposées (similarité sémantique+la logique floue ou SentiWordNet+la logique floue) à des approches basées uniquement sur les concepts de similarité sémantique et les notions des systèmes de recherche d'informations ou uniquement sur la ressource lexicale SentiWordNet.

La figure II.35 présente les résultats obtenus pour le taux de classification (Classification Rate CR) et le taux d'erreur (Error Rate ER) avec nos approches et des approches basées uniquement sur la similarité sémantique ou SentiWordNet.

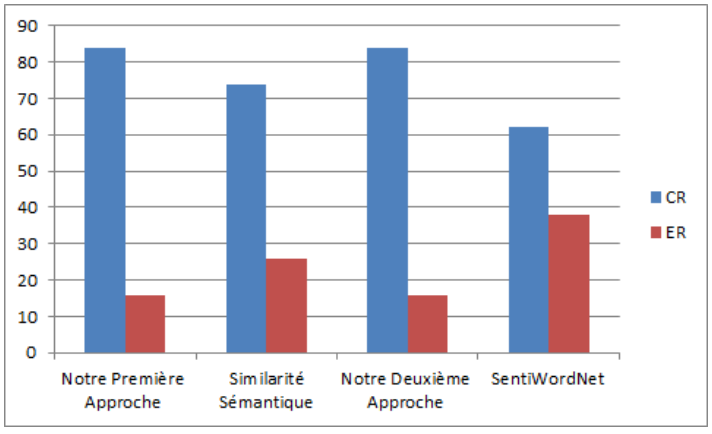


FIGURE II.35 – L'effet de la logique floue sur la classification

À partir de la figure II.35, nous remarquons qu'en utilisant notre première approche basée sur la logique floue et la similarité sémantique, le taux de classification augmente (de 74% à 86%) et celui du taux d'erreur diminue (de 26% à 14%) en comparaison avec l'approche qui utilise uniquement la similarité sémantique sans logique floue. Aussi, en utilisant la deuxième approche proposée (logique floue et SentiWordNet), le taux du CR augmente (de 62% à 84%) et celui du ER diminue (de 38% à 16%) par rapport à l'approche qui utilise uniquement SentiWordNet. C'est-à-dire qu'en ajoutant la logique floue, nous améliorons la qualité de notre système.

Pour démontrer les résultats obtenus en utilisant les approches proposées (première et deuxième approche), nous comparons la méthode proposée à d'autres techniques de la littérature, telles que : une méthode basée sur le lexique en utilisant le dictionnaire AFINN et WordNet[102], une approche basée sur la similarité sémantique qui calcule le degré de pertinence entre le tweet à classer et trois documents d'opinion : document positif, document négatif et document neutre (approche 1) , approche qui calcule la similarité sémantique entre chaque mot du tweet et les mots **négatif** et **positif** (approche 2)[59], et enfin une approche basée sur le dictionnaire SentiWordNet(senti.).

Cette expérience est basée sur les résultats de nos articles publiés [58][59], dans lesquels nous démontrons empiriquement que les approches basées sur le dictionnaire et celles basées sur la similarité sémantique surpassent les algorithmes bien connus d'apprentissage automatique tels que Naive Bayes, ID3 et KNN avec un taux de précision élevé. C'est pour cette raison que nous comparons nos approches proposées avec les approches basées sur les dictionnaires et celles basées sur la similarité sémantique.

La figure II.36 montre les résultats obtenus.

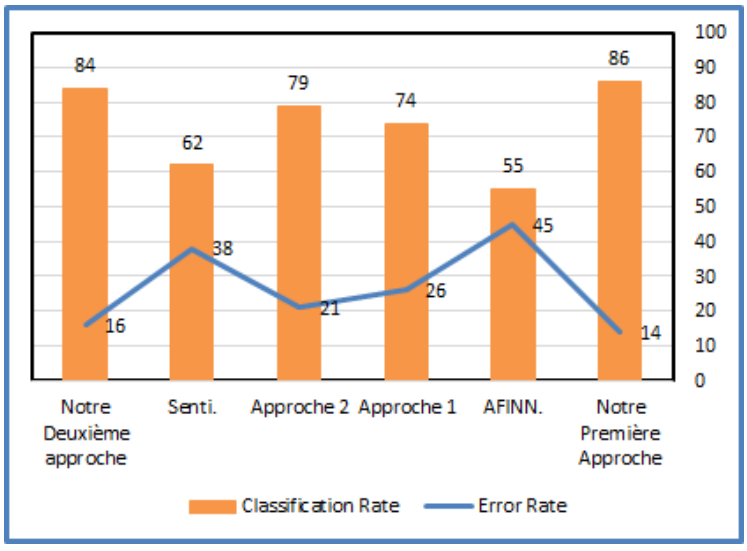


FIGURE II.36 – Résultats de comparaison

Selon la figure II.36, nos approches basées sur la logique floue surpassent les autres approches avec un taux de classification égal à 86% et 84%, et un taux d'erreur égal à 14% et 16%, en comparaison avec les autres approches qui n'utilisent pas la logique floue et qui consistent à classer les tweets d'une manière noir et blanc sans tenir compte du flou et d'imprécision des sentiments et des opinions, tout cela montre comment la logique floue peut améliorer la qualité de la classification avec une augmentation du taux de classification et une diminution du taux d'erreur.

La question la plus importante qui se pose est quelle est la qualité de notre modèle ? pour cela, nous décidons de calculer davantage de métriques en plus du taux de classification et d'erreur pour évaluer les approches proposées, ces métriques sont : exactitude(Accuracy), précision, rappel(Recall) et F1-score. Nous comparons nos approches avec 4 autres approches : une approche basée sur le dictionnaire AFINN, une approche basée sur le lexique qui utilise les dictionnaires AFINN et WordNet et les approches 1 et 2 qui reposent sur la similarité sémantique. La figure II.37 montre les résultats obtenus pour les 4 mesures (exactitude, précision, rappel et F1-score).

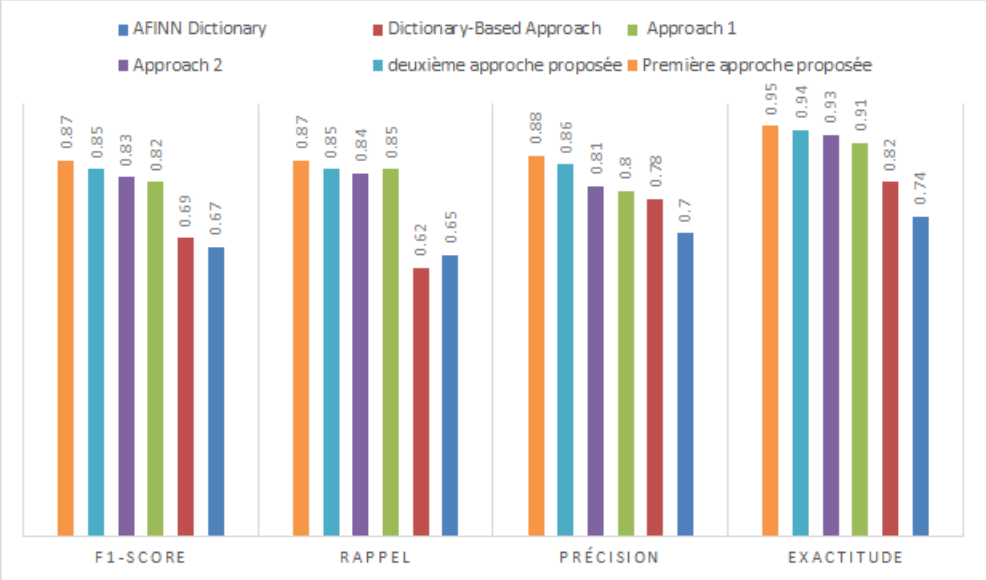


FIGURE II.37 – Évaluation de notre système d'analyse des sentiments

D'après la Figure II.37, nous remarquons que les approches que nous proposons donnent de bons résultats, que ce soit pour l'exactitude, la précision, le rappel ou le F1-score, par rapport aux autres approches. Ils ont une bonne exactitude de 95 % et 94%, la précision est améliorée de 88% et 86%, le rappel de 87% et 85%, et le F1-score de 87 % et 85%. Tout cela montre à quel point les approches que nous proposons dépassent les autres.

La dernière expérience que nous avons réalisée dans ce travail consiste à comparer nos approches proposées à deux travaux récemment publiés. Un certain nombre de chercheurs

ont exploré l'application des approches hybrides en combinant diverses techniques pour obtenir de meilleurs résultats qu'une approche standard avec un seul outil. Appel et al.[2] démontrent que l'utilisation d'une combinaison d'un lexique de sentiments, d'outils NLP et de techniques d'ensemble flou permet de surpasser les approches utilisant uniquement des algorithmes d'apprentissage automatique (ce qui démontre nos résultats obtenus). Les auteurs de cette approche utilisent Sentiwordnet pour extraire la polarité des mots, et après ils utilisent une inférence de logique floue avec cinq ensembles flous ("Poor", "Slight", "Moderate", "very", "most"). De plus, Dragoni et al.[19] utilisent également la logique floue pour calculer les informations de polarité sur la base de dictionnaires tels que senticnet. La figure II.38 présente une comparaison entre les approches proposées, l'approche **Appel**, l'approche **Dragoni**, l'approche Naive Bayes NB et l'approche d'entropie maximale(maximum entropy ME).

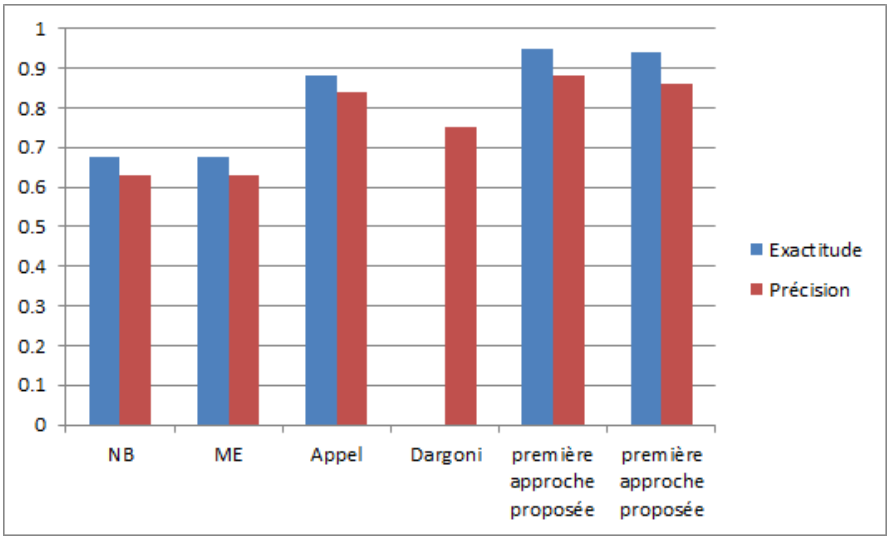


FIGURE II.38 – Résultat de la comparaison avec les approches hybrides

D'après la figure II.38, l'utilisation des approches utilisant plus d'une technique (l'approche proposée, Appel et Dargoni) donnent de bons résultats, par rapport aux méthodes basées uniquement sur des algorithmes d'apprentissage automatique (NB et ME). La figure montre également que nos approches surpassent les approches **Appel** et **Dragoni** au niveau de l'exactitude et de la précision.

Dans ce travail, nous avons présenté deux nouvelles approches basées sur la logique floue, la similarité sémantique à l'aide du dictionnaire WordNet et du dictionnaire SentiWordNet pour classer les tweets en trois classes (positive, négative ou neutre). Ce travail nous permet de familiariser avec les concepts de la logique floue et également comment calculer la positivité et la négativité des tweets sur la base de la similarité sémantique entre le tweet à classer et deux documents d'opinion, ou en utilisant SentiWordNet.

Notre système à logique floue utilise les résultats obtenus après le calcul de la positivité et de la négativité du tweet pour donner à la fin son sentiment (positif, négatif ou neutre). La méthode d'analyse des sentiments proposée est multilingue, et est développée parallèlement à l'aide du framework Hadoop avec le système de fichiers distribué Hadoop(HDFS) et le modèle de programmation MapReduce.

Les résultats expérimentaux montrent que l'utilisation de la logique floue dans la classification des tweets améliore la qualité de notre méthode et que les approches proposées surpassent d'autres méthodes de la littérature.

Conclusion générale et perspectives

L'analyse des données issues des réseaux sociaux a connu ces dernières années un grand développement. Ce domaine vise à classer les données en classes ou d'en prédire le degré de polarité d'un texte écrit. Dans cette thèse, nous avons présenté nos travaux de recherche qui consistent à utiliser les données issues de Twitter et Facebook pour les analyser et d'extraire des informations utiles(motivation, sentiments...), et également proposer des nouvelles approches et méthodes afin d'améliorer les résultats de la classification et d'analyse des données des réseaux sociaux.

Vu au grand nombre des utilisateurs des réseaux sociaux et le volume du contenu publié chaque jour, il devient nécessaire de chercher des méthodes optimales pour l'extraction et le stockage de cette énorme masse de données. Une des solutions existantes est de distribuer le travail d'extraction et du stockage entre plusieurs machines(Hadoop clutser), avec l'utilisation du terme Big Data avec ses technologies telles que le framework Hadoop, Apache Flume et Apache Sqoop.

Pour atteindre cet objectif, nous avons commencé notre thèse par une présentation des technologies Big Data, et comment nous pouvons extraire les données à partir du Twitter et Facebook sous forme des tweets et des publications Facebook. Nous avons décrit aussi comment stocker les données extraites dans HDFS pour distribuer le stockage, ce qui nous a permis de paralléliser les approches proposées.

Les données des réseaux sociaux sont non structurées, ils contiennent beaucoup des données inutiles(mots vides, URLs, ponctuation...etc), ce qui implique la nécessité d'une étape de pré-traitement pour rendre les données prêtes pour la phase de la classification. Pour cela, nous avons présenté les différentes méthodes du traitement du texte utilisées dans notre travail, et comment nous les avons adaptées dans nos approches afin d'augmenter la qualité de la classification. Nous avons décrit également comment rendre nos approches multilingues pour supporter plus qu'une seule langue.

Dans la première contribution présentée dans le chapitre 1 de la partie II, l'approche proposée consiste à classer les tweets en trois classes : positive, négative et neutre. Notre proposition consiste à enrichir le dictionnaire AFINN avec la sémantique en utilisant la base lexicale Wordnet. Nous avons également présenté la moyenne de paralléliser la classification en utilisant Hadoop (HDFS et MapReduce) pour optimiser le temps de la classification. Les résultats montrent que notre approche surpasse les approches basées

sur l'apprentissage automatique. Notre méthode donne de bons résultats au niveau du taux de classification et du taux d'erreur, en outre, le pré-traitement du texte améliore le résultat de la classification. Nous avons également montré l'effet de la parallélisation sur le temps de calcul. Enfin, Nous avons décrit notre système d'e-learning basé sur l'approche proposée et un algorithme génétique pour trouver un contenu pédagogique optimal pour un apprenant.

Pour améliorer les résultats de la classification, nous avons présenté deux approches basées sur la similarité sémantique pour classer les tweets en trois classes(positive, négative et neutre). La première approche calcule la similarité sémantique entre le tweet à classer et trois documents d'opinion(document positif, document négatif et document neutre), et la deuxième approche calcule la similarité sémantique entre le tweet et les mots **"positif"** et **"négatif"**. Les résultats montrent que les deux approches surpassent l'approche basée sur le lexique(AFINN+WordNet) et les approches d'apprentissage automatique(Machine Learning) soit au niveau de la précision, rappel ou F1-score. Une de ces approches est utilisée pour proposer une nouvelle méthode de recommandation pour recommander un contenu pédagogique optimal pour un apprenant. Notre système de recommandation est basé sur une approche de filtrage social et une approche de filtrage collaboratif.

Basé sur le fait que les sentiments de l'homme sont flous et vagues, nous avons proposé deux approches basées sur les concepts de la logique floue, les notions des systèmes de la recherche d'informations, similarité sémantique et le dictionnaire SentiWordNet. Notre travail consiste à proposer deux mesures que nous avons nommé la **positivité** et la **négativité**, ces deux mesures jouent le rôle des entrées de notre système à logique floue(fuzzy logic system **FLS**), et par l'application des étapes de fuzzification, règles floues et de la défuzzification, nous trouvons en sortie du FLS la classe du tweet(positive, négative ou neutre). Dans la partie des résultats expérimentaux, nous avons montré que l'utilisation de la logique floue augmente le taux de classification, et que nos approches surpassent les travaux existants.

Perspectives

Les différentes approches menées au cours de notre thèse ont démontré leurs importances dans beaucoup des domaines(e-learning, recommandation...), ce qui nous motive à continuer le travail dans cet axe de recherche scientifique. Nous croyons que nous devons travailler sur les points suivants afin d'enrichir le domaine de l'analyse des sentiments avec des nouvelles idées :

– Au niveau de l'approche de recommandation des objectifs pédagogiques pour un

apprenant, nous pensons que la proposition d'une nouvelle approche d'apprentissage par renforcement(reinforcement learning) peut augmenter la qualité de recommandation en trouvant le chemin optimal que l'apprenant(agent) doit suivre afin d'arriver à son objectif.

- L'utilisation de la capacité d'apprentissage en utilisant une approche d'apprentissage profond(**deep Learning**) et les capacités de gestion de l'incertitude et de l'imprécision de la logique floue pour fournir une prédiction de sentiment plus appropriée à l'utilisateur, avec la proposition d'une nouvelle méthode d'analyse des sentiments profonde(basée sur deep learning) et floue à la fois.
- Le rôle du Big Data dans l'analyse des sentiments est fondamental à cause du volume et type des données publiées chaque jour. En plus d'utiliser Hadoop MapReduce, nous voulons utiliser Apache **Spark** qui permet une analyse parallèle optimale et aussi en temps réel afin d'améliorer la vitesse des traitements.
- les paramètres des fonctions d'appartenances(membership functions) joue un rôle fondamental au niveau de la phase de fuzzification d'un FLS. Quoique la définition des valeurs de ces paramètres empiriquement donne des bons résultats, mais nous voulons proposer une nouvelle approche basée sur un algorithme d'optimisation afin de définir leurs valeurs de manière théorique, optimale et automatique.
- Le filtrage collaboratif, soit au niveau des articles(item-based approach) ou au niveau des utilisateurs est basé sur le calcul de similarité. La proposition d'une nouvelle approche optimale pour remplir ce travail est l'une de nos perspectives.
- La mise en place d'une application pour la recommandation des produits et films basée sur le Big Data et une nouvelle approche de recommandation basée sur l'analyse des sentiments et les notions des systèmes de recommandation.

Bibliographie

[1] A. M. Abirami and V. Gayathri. A survey on sentiment analysis methods and approach. In *2016 Eighth International Conference on Advanced Computing (ICoAC)*, pages 72–76, Jan 2017. 46

[2] Orestes Appel, Francisco Chiclana, Jenny Carter, and Hamido Fujita. A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Systems*, 108 :110–124, September 2016. 131, 156

[3] Yilmaz Ar and Erkan Bostanci. A genetic algorithm solution to the collaborative filtering problem. *Expert Systems with Applications*, 61 :122–128, November 2016. 117

[4] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0 : An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*, 2010. 47

[5] Ying Bai and Dali Wang. Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications. In *Advanced Fuzzy Logic Technologies in Industrial Applications*, pages 17–36. Springer, 2006. 128

[6] Anil Bandhakavi, Nirmalie Wiratunga, Stewart Massie, and P. Deepak. Emotion-Corpus Guided Lexicons for Sentiment Analysis on Twitter. In Max Bramer and Miltos Petridis, editors, *Research and Development in Intelligent Systems XXXIII*, pages 71–85. Springer International Publishing, 2016. 61

[7] Yanwei Bao, Changqin Quan, Lijuan Wang, and Fuji Ren. The Role of Preprocessing in Twitter Sentiment Analysis. In De-Shuang Huang, Kang-Hyun Jo, and Ling Wang, editors, *Intelligent Computing Methodologies*, Lecture Notes in Computer Science, pages 615–624. Springer International Publishing, 2014. 59

[8] Erik Boiy and Marie-Francine Moens. A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval*, 12(5) :526–558, October 2009. 48

[9] Erik Boiy and Marie-Francine Moens. A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval*, 12(5) :526–558, October 2009. 62

[10] Dilipkumar A. Borikar and Manoj B. Chandak. An Approach to Sentiment Analysis on Unstructured Data in Big Data Environment. In Aynur Unal, Malaya Nayak, Durgesh Kumar Mishra, Dharm Singh, and Amit Joshi, editors, *Smart Trends in Information Technology and Computer Communications*, Communications in Computer and Information Science, pages 169–176. Springer Singapore, 2016. 56

[11] Cagatay Catal and Mehmet Nangir. A sentiment classification model based on multiple classifiers. *Applied Soft Computing*, 50 :135–141, January 2017. 62, 134

[12] Ting-Yi Chang and Yan-Ru Ke. A personalized e-course composition based on a genetic algorithm with forcing legality in an adaptive learning system. *J. Netw. Comput. Appl.*, 36(1) :533–542, January 2013. 85

[13] Hao Chen, Zhongkun Li, and Wei Hu. An improved collaborative recommendation algorithm based on optimized user similarity. *The Journal of Supercomputing*, 72(7) :2565–2578, July 2016. 118

[14] Wei Yen Chong, Bhawani Selvaretnam, and Lay-Ki Soon. Natural Language Processing for Sentiment Analysis : An Exploratory Analysis on Tweets. In *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, pages 212–217, December 2014. 58

[15] Tapan Kumar Das and P Mohan Kumar. *BIG Data Analytics : A Framework for Unstructured Data Analysis*. 56

[16] Kerstin Denecke. Using SentiWordNet for multilingual sentiment analysis. In *2008 IEEE 24th International Conference on Data Engineering Workshop*, pages 507–512, April 2008. 61

[17] D. V. Nagarjuna Devi, Thatiparti Venkata Rajini Kanth, Kakollu Mounika, and Nambhatla Sowjanya Swathi. Assay : Hybrid Approach for Sentiment Analysis. In Suresh Chandra Satapathy and Amit Joshi, editors, *Information and Communication Technology for Intelligent Systems*, Smart Innovation, Systems and Technologies, pages 309–318. Springer Singapore, 2019. 66

[18] N. M. Dhanya and U. C. Harish. Sentiment Analysis of Twitter Data on Demonetization Using Machine Learning Techniques. In D. Jude Hemanth and S. Smys, editors, *Computational Vision and Bio Inspired Computing*, Lecture Notes in Computational Vision and Biomechanics, pages 227–237. Springer International Publishing, 2018. 63

[19] Mauro Dragoni and Giulio Petrucci. A fuzzy-based strategy for multi-domain sentiment analysis. *International Journal of Approximate Reasoning*, 93 :59–73, February 2018. 130, 134, 156

[20] Geli Fei, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. A Dictionary Based Approach to Identifying Aspects Implied by Adjectives for Opinion Mining. In *Proceedings of COLING 2012 : Posters*, pages 309–318, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. 60

[21] Xia Fu, Yajun Du, and Yongtao Ye. A Hybrid Method to Sentiment Analysis for Chinese Microblog. In Juanzi Li, Ming Zhou, Guilin Qi, Ni Lao, Tong Ruan, and Jianfeng Du, editors, *Knowledge Graph and Semantic Computing. Language, Knowledge, and Intelligence*, Communications in Computer and Information Science, pages 152–157. Springer Singapore, 2017. 65

[22] Elena Gabrielova. Implicit and explicit ways of expressing personal opinion on twitter : The tea party movement in the usa. *Higher School of Economics Research Paper No. WP BRP*, 90, 2015. 45

[23] Dipak Damodar Gaikar, Bijith Marakarkandy, and Chandan Dasgupta. Using Twitter data to predict the performance of Bollywood movies. *Industrial Management & Data Systems*, 115(9) :1604–1621, October 2015. 132

[24] Michael Gamon. Sentiment Classification on Customer Feedback Data : Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. event-place : Geneva, Switzerland. 59

[25] José Antonio García-Díaz, María Pilar Salas-Zárate, María Luisa Hernández-Alcaraz, Rafael Valencia-García, and Juan Miguel Gómez-Berbís. Machine Learning Based Sentiment Analysis on Spanish Financial Tweets. In Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, and Sandra Costanzo, editors, *Trends and Advances in Information Systems and Technologies*, Advances in Intelligent Systems and Computing, pages 305–311. Springer International Publishing, 2018. 63

[26] Geetika Gautam and Divakar Yadav. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In *2014 Seventh International Conference on Contemporary Computing (IC3)*, pages 437–442, August 2014. 63

[27] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009. 65

[28] Emma Haddi, Xiaohui Liu, and Yong Shi. The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, 17 :26–32, January 2013. 59

[29] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association*

for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, ACL '98/EACL '98, pages 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. 47

[30] Maha Heikal, Marwan Torki, and Nagwa El-Makky. Sentiment Analysis of Arabic Tweets using Deep Learning. *Procedia Computer Science*, 142 :114–122, January 2018. 63

[31] Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In *In Christiane FELLBAUM, editor, WordNet : An electronic lexical database, chapter 13, TheMIT Press*, pages 305–332, 1998. 99

[32] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM. 47

[33] Mohammad Rezwanul Huq, Ahmad Ali, and Anika Rahman. Sentiment Analysis on Twitter Data using KNN and SVM. *International Journal of Advanced Computer Science and Applications (ijacsa)*, 8(6), 2017. 64

[34] Amir Hussain and Erik Cambria. Semi-supervised learning for big social data analysis. *Neurocomputing*, 275 :1662–1673, January 2018. 56

[35] José Antonio Iglesias, Aaron García-Cuerva, Agapito Ledezma, and Araceli Sanchis. Social network analysis : Evolving twitter mining. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 001809–001814, Oct 2016. 70

[36] Rajkumar S. Jagdale, Vishal S. Shirsat, and Sachin N. Deshmukh. Sentiment Analysis on Product Reviews Using Machine Learning Techniques. In Pradeep Kumar Mallick, Valentina Emilia Balas, Akash Kumar Bhoi, and Ahmed F. Zobaa, editors, *Cognitive Informatics and Soft Computing*, Advances in Intelligent Systems and Computing, pages 639–647. Springer Singapore, 2019. 64

[37] Chris Jefferson, Han Liu, and Mihaela Cocea. Fuzzy approach for sentiment analysis. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, July 2017. 131

[38] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference (ROCLING 1997)*, pages 19–33, Taipei, Taiwan, August 1997. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP). 99

[39] Zhao Jianqiang. Pre-processing Boosting Twitter Sentiment Analysis ? In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 748–753, December 2015. 58

[40] Zhao Jianqiangn and Gui Xiaolin. Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis. *IEEE Access*, 5 :2870–2879, 2017. 58

[41] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1331–1336. AAAI Press, 2006. 44

[42] Ankur Patil Pratima Kakade Saumitra Vaidya Jyoti Nandimath, Ekata Banerjee and Divyansh Chaturvedi. Big data analysis using Apache Hadoop. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 700–703, August 2013. 56

[43] P. Kalaivani. Sentiment classification of movie reviews by supervised machine learning approaches. *Indian Journal of Computer Science and Engineering (IJCSE)*, 4(4), Aug-Sep 2013. 62

[44] Bharat Sri Harsha Karpurapu and Leon Jololian. A Framework for Social Network Sentiment Analysis Using Big Data Analytics. In Sang C. Suh and Thomas Anthony, editors, *Big Data and Visual Analytics*, pages 203–217. Springer International Publishing, 2017. 55

[45] Charalampos Karyotis, Faiyaz Doctor, Rahat Iqbal, Anne James, and Victor Chang. A fuzzy computational model of emotion for cloud based sentiment analysis. *Information Sciences*, 433-434 :448–463, April 2018. 132

[46] Mariam Khader, Arafat Awajan, and Ghazi Al-Naymat. The Effects of Natural Language Processing on Big Data Analysis : Sentiment Analysis Case Study. In *2018 International Arab Conference on Information Technology (ACIT)*, pages 1–7, November 2018. 59

[47] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. 47

[48] Hamidreza Koohi and Kourosh Kiani. A new method to find neighbor users that improves the performance of Collaborative Filtering. *Expert Systems with Applications*, 83 :30–39, October 2017. 117

[49] Sercan Kulcu, Erdogan Dogdu, and A. Murat Ozbayoglu. A survey on semantic web and big data technologies for social network analysis. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1768–1777, Dec 2016. 70

[50] Bac Le and Huy Nguyen. Twitter Sentiment Analysis Using Machine Learning Techniques. In Hoai An Le Thi, Ngoc Thanh Nguyen, and Tien Van Do, editors, *Advanced Computational Methods for Knowledge Engineering*, Advances in Intelligent Systems and Computing, pages 279–289. Springer International Publishing, 2015. 65

[51] Claudia Leacock and Martin Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. In Christiane Fellbaum, editor, *WordNet : An electronic lexical database.*, chapter 13, pages 265–283. MIT Press, 1998. 100

[52] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 99

[53] Bingwei Liu, Erik Blasch, Yu Chen, Dan Shen, and Genshe Chen. Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier. page 6. 56

[54] Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, and Xuzhen Zhu. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56 :156–166, January 2014. 118

[55] Han Liu and Mihaela Cocea. Fuzzy rule based systems for interpretable sentiment analysis. In *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, pages 129–136, Feb 2017. 127, 131

[56] Youness Madani, Jamaa Bengourram, Mohammed Erritali, Badr Hssina, and Marouane Birjali. Adaptive e-learning using Genetic Algorithm and Sentiments Analysis in a Big Data System. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 8(8) :394–403, 2017. 71, 101, 115

[57] Youness Madani, Jemaa Bengourram, and Mohammed Erritali. Social login and data storage in the big data file system hdfs. In *Proceedings of the International Conference on Compute and Data Analysis*, ICCDA '17, pages 91–97, 2017. 82, 85

[58] Youness Madani, Mohammed Erritali, and Jamaa Bengourram. Classification of Social Network Data Using a Dictionary-Based Approach. In *Cloud Computing and Big Data : Technologies, Applications and Security*, Lecture Notes in Networks and Systems, pages 202–219. Springer, Cham, October 2017. 109, 154

[59] Youness Madani, Mohammed Erritali, and Jamaa Bengourram. Sentiment analysis using semantic similarity and Hadoop MapReduce. *Knowledge and Information Systems*, 59(2) :413–436, May 2019. 154

[60] Sandip D Mali, Dr Sachin N Deshmukh, and Ashish A Bhalerao. SentiView : A Lexicon Based Approach for Twitter Sentiment Analysis. *International Journal of*

Innovative Research in Computer and Communication Engineering, 4(11) :8, 2007. 61

[61] Monica Malik, Sameena Naaz, and Iffat Rehman Ansari. Sentiment Analysis of Twitter Data Using Big Data Tools and Hadoop Ecosystem. In Durai Pandian, Xavier Fernando, Zubair Baig, and Fuqian Shi, editors, *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*, Lecture Notes in Computational Vision and Biomechanics, pages 857–863. Springer International Publishing, 2019. 57

[62] George A. Miller. Wordnet : A lexical database for english. *Commun. ACM*, 38(11) :39–41, November 1995. 47

[63] Ruslan Mitkov, Richard Evans, Constantin Orăsan, Iustin Dornescu, and Miguel Rios. Coreference Resolution : To What Extent Does It Help NLP Applications ? In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*, Lecture Notes in Computer Science, pages 16–27. Springer Berlin Heidelberg, 2012. 33

[64] Antonio Moreno-Ortiz and Javier Fernández-Cruz. Identifying Polarity in Financial Texts for Sentiment Analysis : A Corpus-based Approach. *Procedia - Social and Behavioral Sciences*, 198 :330–338, July 2015. 61

[65] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5 :1–5 :8, New York, NY, USA, 2012. ACM. 48

[66] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining Lexicon and Learning Based Approaches for Concept-level Sentiment Analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '12, pages 5 :1–5 :8, New York, NY, USA, 2012. ACM. event-place : Beijing, China. 66

[67] Tony Mullen and Nigel Collier. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *Proceedings of EMNLP 2004*, pages 412–418, Barcelona, Spain, July 2004. Association for Computational Linguistics. 64

[68] Deebha Mumtaz and Bindiya Ahuja. A Lexical and Machine Learning-Based Hybrid System for Sentiment Analysis. In Brajendra Panda, Sudeep Sharma, and Usha Batra, editors, *Innovations in Computational Intelligence : Best Selected Papers of the Third International Conference on REDSET 2016*, Studies in Computational Intelligence, pages 165–175. Springer Singapore, Singapore, 2018. 65

[69] Maryam Nayebzadeh, Akbar Moazzam, Amir Mohammad Saba, Hadi Abdolrahimpour, and Elham Shahab. An Investigation on Social Network Recommender Systems and Collaborative Filtering Techniques. page 6. 118

[70] M. S. Neethu and R. Rajasree. Sentiment analysis in twitter using machine learning techniques. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pages 1–5, July 2013. 48, 63

[71] Finn Årup Nielsen. A new ANEW : Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts' : Big things come in small packages 718 in CEUR Workshop Proceedings*, pages 93–98, May 2011. 67, 72

[72] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Languages Resources Association (ELRA). 60

[73] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3) :130–137, March 1980. 39

[74] Rudy Prabowo and Mike Thelwall. Sentiment analysis : A combined approach. *Journal of Informetrics*, 3(2) :143–157, April 2009. 48

[75] Rudy Prabowo and Mike Thelwall. Sentiment analysis : A combined approach. *Journal of Informetrics*, 3(2) :143–157, April 2009. 66

[76] Dibakar Ray. Lexicon Based Sentiment Analysis of Twitter Data. *International Journal for Research in Applied Science and Engineering Technology*, 5(X), 2017. 61

[77] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. 99

[78] Cristóbal Romero, Sebastián Ventura, and Paul De Bra. Knowledge Discovery with Genetic Programming for Providing Feedback to Courseware Authors. *User Modeling and User-Adapted Interaction*, 14(5) :425–464, January 2004. 85

[79] Nicolas Ruflin, Helmar Burkhart, and Sven Rizzotti. Social-data Storage-systems. In *Databases and Social Networks*, DBSocial '11, pages 7–12, New York, NY, USA, 2011. ACM. event-place : Athens, Greece. 57

[80] Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. page 8. 58

[81] Hassan Saif, Yulan He, and Harith Alani. Semantic Sentiment Analysis of Twitter. In *The Semantic Web – ISWC 2012*, pages 508–524. Springer, Berlin, Heidelberg, November 2012. 101

[82] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Semantic Patterns for Sentiment Analysis of Twitter. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *The Semantic Web – ISWC 2014*, pages 324–340, Cham, 2014. Springer International Publishing. 101

[83] Gerard Salton. *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. 83, 87

[84] Jaydeep Balkrishna Sathe and Manisha P. Mali. A hybrid sentiment classification method using neural network and fuzzy logic. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pages 93–96, Jan 2017. 131

[85] Wareesa Sharif, Noor Azah Samsudin, Mustafa Mat Deris, Rashid Naseem, and Muhammad Faheem Mushtaq. Effect of negation in sentiment analysis. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, pages 718–723, August 2016. 60

[86] Amira Shoukry and Ahmed Rafea. A Hybrid Approach for Sentiment Classification of Egyptian Dialect Tweets. In *Proceedings of the 2015 First International Conference on Arabic Computational Linguistics (ACLing)*, ACLING '15, pages 78–85, Washington, DC, USA, 2015. IEEE Computer Society. 65

[87] Philomina Simon and S. Siva Sathya. Genetic algorithm for information retrieval. *2009 International Conference on Intelligent Agent and Multi Agent Systems*, pages 1–6, 2009. 84

[88] Tajinder Singh and Madhu Kumari. Role of Text Pre-processing in Twitter Sentiment Analysis. *Procedia Computer Science*, 89 :549–554, January 2016. 59

[89] Shiliang Sun, Chen Luo, and Junyu Chen. A review of natural language processing techniques for opinion mining systems. *Information Fusion*, 36 :10–25, July 2017. 58

[90] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2) :267–307, June 2011. 47

[91] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2) :267–307, April 2011. 61

[92] Samir Tartir and Ibrahim Abdul-Nabi. Semantic Sentiment Analysis in Arabic Social Media. *Journal of King Saud University - Computer and Information Sciences*, 29(2) :229–233, April 2017. 101

[93] Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, August 2018. 51

[94] Abinash Tripathy, Ankit Agrawal, and Santanu Kumar Rath. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, 57 :117–126, September 2016. 62

[95] Dana Vrajitoru. Crossover improvement for the genetic algorithm in information retrieval. *Inf. Process. Manage.*, 34(4) :405–415, July 1998. 84

[96] Bingkun Wang, Yongfeng Huang, Xian Wu, and Xing Li. A Fuzzy Computing Model for Identifying Polarity of Chinese Sentiment Words. *Computational Intelligence and Neuroscience*, 2015. 130

[97] Xinzhi Wang, Hui Zhang, and Zheng Xu. Public Sentiments Analysis Based on Fuzzy Logic for Text. *International Journal of Software Engineering and Knowledge Engineering*, 26(09n10) :1341–1360, November 2016. 132

[98] KeYuan Wu, MengChu Zhou, Xiaoyu Sean Lu, and Li Huang. A fuzzy logic-based text classification method for social media data. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1942–1947, Oct 2017. 130

[99] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. 98

[100] Rui Xia and Chengqing Zong. A pos-based ensemble model for cross-domain sentiment classification. In *In Proceedings of the 5th International Joint Conference on Natural Language Processing-IJCNLP*, 2011. 49

[101] Qiang Ye, Ziqiong Zhang, and Rob Law. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, 36(3, Part 2) :6527–6535, April 2009. 64

[102] Madani Youness, Erritali Mohammed, and Bengourram Jamaa. A parallel semantic sentiment analysis. In *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, pages 1–6, October 2017. 154

[103] Madani Youness, Erritali Mohammed, and Bengourram Jamaa. Semantic Indexing of a Corpus. *International Journal of Grid and Distributed Computing*, 11(7) :63–80, July 2018. 38

[104] Yang Yu and Xiao Wang. World Cup 2014 in the Twitter World : A big data analysis of sentiments in U.S. sports fans' tweets. *Computers in Human Behavior*, 48 :392–400, July 2015. 57

[105] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, June 1965. 127

[106] Lotfi A. Zadeh. Fuzzy logic—a personal perspective. *Fuzzy Sets and Systems*, 281 :4–20, December 2015. 127

[107] Nurulhuda Zainuddin, Ali Selamat, and Roliana Ibrahim. Improving Twitter Aspect-Based Sentiment Analysis Using Hybrid Approach. In Ngoc Thanh Nguyen, Bogdan Trawiński, Hamido Fujita, and Tzung-Pei Hong, editors, *Intelligent Information and Database Systems*, Lecture Notes in Computer Science, pages 151–160. Springer Berlin Heidelberg, 2016. 66

[108] Nurulhuda Zainuddin, Ali Selamat, and Roliana Ibrahim. Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence*, 48(5) :1218–1232, May 2018. 66

[109] Lei Zhang and Bing Liu. Aspect and Entity Extraction for Opinion Mining. In Wesley W. Chu, editor, *Data Mining and Knowledge Discovery for Big Data : Methodologies, Challenge and Opportunities*, pages 1–40. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. 46